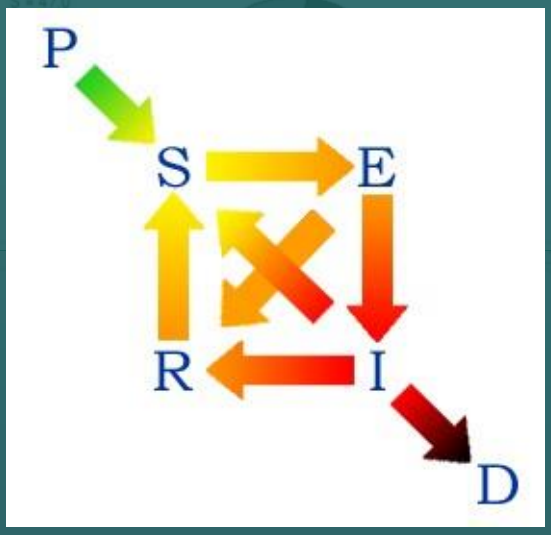



# *OUTBREAK*

## User's Manual
## Version 1.1

For *OUTBREAK* Version 2.10

Written by Carlo Pacioni, Sara Sullivan, Caroline M. Lees, Philip S. Miller, and Robert C. Lacy

15 June 2019

SCTI
Species Conservation Toolkit Initiative

Software for saving species;

Arthur Rylah Institute for environmental research

Chicago Zoological Society
Inspiring Conservation Leadership

IUCN SSC Species Survival Commission

CONSERVATION PLANNING SPECIALIST GROUP
Changing the Future for Wildlife

# *OUTBREAK*

*Manual written by:*
Carlo Pacioni, Sara Sullivan, Caroline M. Lees, Philip S. Miller, and Robert C. Lacy

*Software (version 2) written by:*
Robert C. Lacy, Chicago Zoological Society

Development and distribution of the *OUTBREAK* software and this accompanying manual was supported by the Species Conservation Toolkit Initiative and a Research Coordination Network grant (DEB-1146198) from the US National Science Foundation. Thanks are extended to the many colleagues within the IUCN SSC Conservation Planning Specialist Group, who helped in the conceptualization and development of the program. A special thanks to Dr. Scroggie for commenting on an early draft of this manual.

*OUTBREAK* is provided at no cost, in order to further conservation and science. It is distributed without warranty of its suitability for any particular use, and neither the program nor this manual are guaranteed to be free of errors, bugs, or potentially misleading information. It is the responsibility of the user to ensure that the software is appropriate for the uses to which it is put.

To submit suggestions, additions, or corrections (for the program or manual) or to offer assistance in translating this manual to other languages, please contact help@scti.tools.



Cover photo credit: Chicago Zoological Society.

*Citation of this manual:*
Pacioni, C., S. Sullivan, C.M. Lees, P.S. Miller, R.C. Lacy. 2019. *OUTBREAK User's Manual. Version 1.1*. Chicago Zoological Society, Brookfield, Illinois, USA.

*Citation of the software:*
Lacy, R.C., J.P. Pollak, P.S. Miller, L. Hungerford, P. Bright. 2020. *OUTBREAK version 2.10*. IUCN SSC Conservation Planning Specialist Group, Apple Valley, Minnesota, USA.

# Table of Contents

# Species Conservation Toolkit Initiative

SCTI is a partnership to ensure that innovations needed for species risk assessment, evaluating conservation actions and managing populations are developed quickly, available globally, and used effectively. More information about SCTI, downloads of software, and user support are available at www.scti.tools.

The past 30 years have seen major advances in the use of computer modeling to address the growing complexity of species conservation issues. The Species Conservation Toolkit currently includes the programs *VORTEX*, *VORTEX* Adaptive Manager, *OUTBREAK*, PMx, and *METAMODEL MANAGER*, which collectively have supported conservation decision-making for thousands of projects around the world.

Under the guidance of expert users, the toolkit allows conservation decision-makers and practitioners to assess risks to wildlife, evaluate conservation options, and guide active population management with greater efficiency and realism. It is essential that this toolkit be maintained and expanded to meet new needs and respond to new opportunities. To this end, a partnership of organizations has formed to foster lasting support for this suite of tools, ensuring its continued evolution, global distribution, and ongoing user-support by technical experts.

Thanks to Chicago Zoological Society, Species360, Smithsonian Conservation Biology Institute, European Association of Zoos and Aquaria, Association of Zoos and Aquariums, Auckland Zoo, Wildlife Reserves Singapore, Chester Zoo, Copenhagen Zoo, Zoological Society of London, SOS Rhino, San Diego Zoo Global, Oceans Initiative, Raincoast Conservation Foundation, San Francisco Zoo, Living Desert, Saint Louis Zoo, Woodland Park Zoo, Texas State Aquarium, Cincinnati Zoo, and Seattle Aquarium for helping to keep these conservation tools (and species!) alive and evolving.

Robert Lacy (Chicago Zoological Society)
Jonathan Ballou (Smithsonian Conservation Biology Institute)
Onnie Byers (IUCN SSC Conservation Planning Specialist Group)
Taylor Callicrate (Species Conservation Toolkit Initiative)
Sara Sullivan (Species Conservation Toolkit Initiative)

# About the manual

The purpose of this manual is to introduce the principles of epidemiological modeling of infectious disease in wildlife populations as modelled in *OUTBREAK* and to provide sufficient guidance to allow first-time users to build, run, and interpret results from *OUTBREAK* models. This manual does not go into the details of epidemiological or ecological concepts, knowledge of which may be necessary to model disease and demographic dynamics in a population adequately. However, Chapter 1 provides some background on epidemiological models along with an introduction to *OUTBREAK*. Chapter 2 provides guidance on installation, project creation, and file management. Each input screen is described in detail in Chapter 3. Chapters 4 and Chapter 5 describe how to run the simulations and access the results, respectively. Chapter 6 describes how to develop more complex models, and Chapter 7 provides examples with more detailed descriptions of how to model specific aspects. Chapter 8 describes where to access additional information and help.

Please send bug reports, suggestions, and any requests for additional information to help@scti.tools. We cannot guarantee that all comments and requests will be answered, but responses will be provided when possible. We do, however, value all feedback.

## Conventions

- Throughout the manual, input to the program is shown in **Bold Microsoft San Serif** font. Text is formatted in *italics* when it refers to the text present in the program's windows and in **bold** when it is the title of a section in the program's windows. Output file names are ***underlined***, ***bold*** and in ***italics*** for fixed parts of the file name (e.g., a suffix) and only **UNDERLINED** and **BOLD** for variable parts of the file name. For example, the output file **SCENARIO_NAME**_***day_stats.txt***, indicates that the suffix "day_stats.txt" is appended to the first half of the file name, which changes depending on the scenario name.
- Internal hyperlinks are formatted green; external hyperlinks are formatted blue. All terms featured in the glossary at the end of this manual are *italicized and hyperlinked*.
- *OUTBREAK* will use the decimal digit format (e.g. 0.5 or 0,5) based on the regional settings on the computer. However, as a convention throughout this manual, the dot "." will be used to indicate decimal digits.
- All input rates are intended as a function of the time step. The default time step is one day, and so the manual, as a convention, refers to a time step as one day. If the user modifies the time step definition, it is the user's responsibility to enter all rates on subsequent screens as a function of the chosen time step.

## Appendices

The appendices include additional information relevant to the topics covered in this manual.

- Appendix I provides an overview on using *METAMODEL MANAGER* as a platform to link *OUTBREAK* simulations into more comprehensive models of the diverse processes driving species dynamics, such as disease, species interactions, habitat change, dispersal, climate change, and genetic processes.
- Appendix II provides a glossary of the technical terms used in this document. It is advisable to review this glossary as the meaning of some terms can vary between disciplines. All technical terms are defined in the context of *OUTBREAK* and are italicized and hyperlinked to the glossary.

## References

A reference section is provided on page 97. For examples of the use of *OUTBREAK* in the scientific literature, see Chapter 8.

# Chapter 1. Introduction

## 1.1 Epidemiological models

Emerging *infectious diseases* pose serious threats to the persistence of wildlife species globally. Modelling population dynamics in response to infectious disease can help to increase our understanding of disease *transmission* and make quantitative predictions for the future course of an outbreak. Epidemiological models can also be used as a valuable tool to evaluate the efficacy of different management interventions that target the introduction of a disease *pathogen* into a population or the transmission of that pathogen among individuals within that population.

Classically, in mathematical epidemiology, compartmental models are used to describe the underlying dynamics of disease transmission. These models divide individuals in the study population into distinct groups, or compartments, based on their disease state and then track the number of individuals in each compartment over time. The compartments included in the model will depend on the pathogen's biology and its effect on the *host* but often include the following:

- **Susceptible (S)** - individuals can become infected with a *disease* following exposure to an etiological agent (i.e. disease pathogen)
- **Exposed (E)** - individuals have been exposed to and contracted the disease pathogen but are not infectious to other individuals
- **Infectious (I)** - individuals have been exposed to the pathogen (with or without causing disease) and can transmit the pathogen to other susceptible individuals
- **Recovered and resistant (R)** - individuals are no longer infectious as a result of acquiring permanent or temporary immunity

In the simplest epidemiological model, called an SI model, individuals are classified either as susceptible (S) or infectious (I). Individuals transition between the S and I states with a certain frequency, which is commonly indicated by the symbol $\beta$. This is called the transmission rate, transmission coefficient, or transmission parameter. An SI model can be used when describing the behavior of diseases with lifelong infectiousness (e.g., herpes). In cases where temporary or permanent immunity occurs when individuals recover from an infection (e.g., influenza), an SIR model would be appropriate. This model requires an additional parameter, the recovery rate ($\gamma$), to describe the rate of transition from states I to R.

Often, epidemiological models also include an exposed (E) disease state. Using a SEIR model is especially important when the time elapsed from being exposed to the disease and becoming infectious is long. In fact, in these situations, the number of S individuals will be overestimated if the *incubation period* is not taken into account, which can substantially influence the

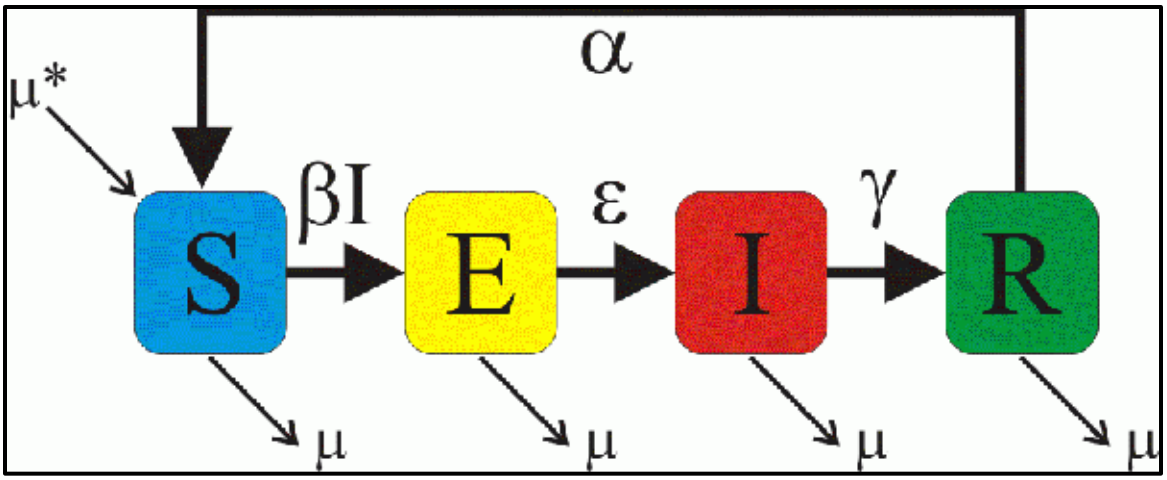



**Figure 1. A schematic representation of a SEIR epidemiological model. β: transmission rate, ε: incubation rate, γ: recovery rate, α: immunity loss rate. The term βI in the diagram illustrates that the probability that a susceptible individual becomes infected will be a product of the number of I individuals and the transmission rate.**

dynamics of the host-disease system being modelled (where the host is the organism that carries the disease).

A detailed description of all possible epidemiological models is beyond the scope of this manual but could take any of the following forms: SI, SIS, SIR, SIRS, SEIR, SEIRS. All of these models are typically implemented by resolving a set of differential equations (that is, equations that describe the change in abundance in each of the epidemiological compartments). These models are usually focused solely on disease dynamics, assuming a constant population size. That is, the models will predict changes in the distribution of individuals in the different epidemiological compartments but will not describe the demographic dynamics of the population.

However, from the foundations described above, it is possible to expand basic epidemiological models to include reproduction and natural mortality rates (i.e. those not due to the disease). For example, Figure 1 describes a SEIR model where (natural) demographic changes in the host population are included by modelling the production of (new) susceptible individuals (newborn: μ* = birth rate) and loss of individuals from each compartment due to natural mortality (death: μ = death rate). It is possible to envisage models with further sub-structuring of the host population, such as age classes, sexes, and multiple host species, or to take into account spatial dynamics. Such elaborations of more basic models are straightforward, as long as it is possible to build a matrix (or several matrices) that describe transition rates of individuals as functions of age, sex, spatial location, etc.

## Deterministic, stochastic, and individual-based models of infectious disease

The epidemiological models just described are classified as deterministic models. This is because solving the differential equations will describe the mean (expected) trajectories of the disease dynamics, which are determined by the parameter values inputted to the model. The solution to these equations will not change when the same equations are solved multiple times (as long as the same parameters are used). While such models are informative and have been demonstrated to describe epidemics in several circumstances adequately, they do not take into account the stochastic (i.e. random) variation in parameter values through time that is inherent in natural systems. This is particularly true when considering small populations. By chance alone, the overall result may change dramatically when only a few 'events' occur, and overall outcomes may be sensitive to the number of realized transitions between states.

Stochastic models were developed to deal with the random variability around whether particular events will occur and, when they do occur, with what level of impact. The strategy used by stochastic models is to determine the timing and magnitude of events by drawing a numerical value from a probability distribution each time an event has the potential to occur, rather than by using a single (fixed) parameter value to compute an event's outcome. This means that the results may change each time the model is run. By running the model multiple times and collating the results, it is possible to obtain a distribution of the expected outcomes. From this information, the likelihood of occurrence of any particular outcome can be computed.

Individual-based models are, in a sense, an extreme implementation of stochastic models. Each individual is treated as a separate entity in this type of model structure, with the probability of any given event impacting that individual determined at each point in time by specific random variables. For example, for each susceptible (S) individual that comes in contact with an infectious (I) individual, a stochastic model may draw a random value between 0 and 1. That value is then compared with the probability of disease transmission that has been defined in the model structure, in order to decide whether each contacted S individual will become infected (E). Clearly, the higher the probability of disease transmission, the greater the number of individuals infected. However, when the model is run multiple times, the number of individuals that become infected will vary stochastically.

Individual-based models, which keep track of each individual's fate over time, offer several advantages. Firstly, relatively complex processes can be broken down into smaller components, and each of these components can be simulated explicitly. This reduces the need to make (sometimes unrealistic) assumptions when biological processes are summarized by a single parameter (or a few parameters) in a model. For example, a simple deterministic model might explore population size changes using only one parameter (e.g., the growth rate), whereas a more complex, stochastic model might use the simulated contribution of each breeder to the population's reproductive output combined with the losses resulting from simulated variation in individual survival. Furthermore, individual-based models make it easier to tailor the

probability of certain demographic events, such as survival, reproduction, or disease status, to each individual's traits (e.g., sex, age, body condition, etc.). This detailed control of the mechanisms that are modelled allows for more in-depth hypothesis testing. For example, more complex models support evaluation of the effects of specific management interventions applied to only certain aspects of the overall system being considered. Further, because the models are a step-by-step reconstruction of the known biological processes, understanding and describing them is often more intuitive than is the case for more abstract models, which consider entire populations in aggregate, without reference to the status or traits of specific individuals.

Individual-based models also have disadvantages, the most obvious of which is that they are computationally intensive, especially when the simulations need to follow a large number of individuals. Modern technology provides some assistance in this regard, as computer speed continues to improve, and the large number of processor cores available on modern personal computers allow for parallel computation. However, computation time remains an issue that should not be underestimated. Most importantly, and not so frequently recognized, individual-based models tend to have a large number of parameters, and it is often difficult to obtain reliable estimates for all parameters from field or experimental data. Unreliable parameter estimates will lead to unreliable model predictions.

## 1.2 What is *OUTBREAK*?

*OUTBREAK* is an individual-based simulation of infectious disease in wildlife populations, developed to provide the capability to include disease in species risk assessments and conservation planning. *OUTBREAK* employs an S-E-I-R epidemiological model, tracking transitions of animals among *Susceptible (S)*, *Exposed (E)*, *Infectious (I)*, and *Recovered (R)* states. Transitions among the states are simulated as probabilistic events, with user-specified rates and durations -- including probabilities of encounter, transmission, and recovery, with durations of incubation (E), infection (I), and resistance (R). The model is highly flexible, allowing modeling of direct transmission via contact, distance-based transmission probabilities, maternal transmission, and infection from environmental sources. It also allows modeling of vaccination and removal programs for disease management. Input rates can be specified as fixed probabilities, entered as functions of population or individual characteristics, or sampled from distributions. *OUTBREAK* includes a demographic model, allowing disease to impact survival or fecundity. It can also be coupled with the *VORTEX* PVA software as a more powerful demographic model and for including genetic factors influencing disease transmission and recovery.

*OUTBREAK* was developed to address the important need to evaluate the impact of interacting epidemiological and demographic processes on population viability. While substantial effort has been directed toward constructing demographic models of wildlife population viability with greater realism and mathematical sophistication (e.g., Morris et. al 2002), considerably less attention is directed at the larger ecological factors that influence population persistence. One

such factor is infectious disease and its transmission dynamics among co-existing species. Typically, models of wildlife population viability do not adequately reflect the demographic effects of disease on a population, which can vary considerably depending upon the structure of the host population, the characteristics of the infectious agent, and environmental variables like habitat condition and availability. Similarly, the majority of available epidemiological models of infectious disease focus primarily on the disease state of individuals in the population and either assume a static population size or use only very simple exponential models of population change. As a result, important potential influences of demographic variability, population structure, and other ecological factors are lost. Several modelling platforms, including *OUTBREAK*, have been developed in recent years to overcome these limitations (e.g., Wilensky and Evanston 1999; Schumaker and Brookes 2018).

*OUTBREAK* has been used for modeling the spread of sylvatic plague in prairie dogs, tuberculosis in cape buffalo and lions, mange in bush dogs, yellow fever in howler monkeys, myxomatosis in rabbits, facial tumor disease in Tasmanian devils, chytridiomycosis in Panamanian golden frogs, and other wildlife diseases. Although the original impetus was to provide a tool to help with wildlife conservation, it does have important applications in both agriculture and human diseases. *OUTBREAK* is distributed for free by the Species Conservation Toolkit Initiative (www.scti.tools).

## 1.3 When to use *OUTBREAK*

When there is a special interest in specifically exploring the interaction between an infectious disease and demographic dynamics that would be difficult to investigate with conventional modelling approaches, *OUTBREAK* is a good candidate to tackle these problems. *OUTBREAK* aims to be a user-friendly program but, at the same time, to be a highly flexible and powerful tool for modelling a wide variety of situations. However, as an individual-based model, *OUTBREAK* can be computationally demanding, so its use for modelling very large populations may be somewhat impractical. It also often requires some creativity and work to find a way to use the available options in *OUTBREAK* to represent adequately the processes that determine the dynamics of the disease under consideration. Sometimes, it may not be possible to construct a satisfactory model within *OUTBREAK*. In such cases, a different epidemiological model – most likely one developed for the specific case – might be required.

Individual-based models require a large number of parameters, and *OUTBREAK* is no exception. There will be situations in which field data to estimate the necessary parameters are available or can be collected. In other cases, the ecology and biology of the host as well as of the disease component is well understood, so that it may be possible to define adequately a realistic range of values from existing information sources. However, there will also be situations in which it will be difficult to determine realistic parameter values. Whether the individual-based model developed in *OUTBREAK* is the right approach for a specific research project or for evaluating the efficacy of management programs will depend on the questions being investigated. It should be kept in mind that model predictions will be accurate only when the model accurately reflects

relevant dynamics in the system being modelled. If good data are not available to estimate the model's parameters and validate the model itself, then the user should be careful and avoid uncritically concluding that model will yield accurate predictions.

Nevertheless, it is important to recognize that *OUTBREAK* can be an invaluable tool for exploring disease dynamics, even in the absence of detailed data. For example, the aim of a given modeling project might be to identify parameters that strongly affect a given system under study, so that investment of available research or management resources can be more effectively prioritized. Conducting a sensitivity analysis (evaluating how a different parameterization of the model may – or may not – affect the results) can also help in assessing how robust a particular model may be due to a lack of knowledge of the ecology of the species or disease being simulated.

## Alone or with a meta-modelling approach?

This manual mostly focuses on the use of *OUTBREAK* as a stand-alone modelling platform. However, and as already mentioned above, *OUTBREAK* is designed to be linked to other simulation platforms or to use custom made programs or scripts within a metamodel framework, using the program *METAMODEL MANAGER* (Lacy, et al. 2013). There are several situations where such an approach could be preferred, including:

- The modelling of complex demographic systems that will be difficult to implement in *OUTBREAK* but are feasible in other PVA software (e.g., *VORTEX*).
- The modelling of different strains of a single pathogen where there is a need to keep track of each variant explicitly. Each strain could be represented by an *OUTBREAK* model linked through *METAMODEL MANAGER*.
- Similarly, the modelling of multiple pathogens, where again, each pathogen could be represented by an *OUTBREAK* model linked through *METAMODEL MANAGER*.
- Analysis of infectious disease dynamics within metapopulations. *OUTBREAK* is structured to simulate disease in a single population.
- Genetic determinants of disease dynamics. While it is possible to build a model that could approximate a genetic disease or genetic resistance (e.g., by forcing vertical transmission of the disease, or by simulating the offspring's genetic resistance by making them acquire some traits from their parent using individual State Variables), it will be difficult to implement more complex systems and may be easier to take advantage of other software packages that have in-built options for modelling inheritance (e.g., *VORTEX*).
- The modelling of indirect transmission (e.g., *parasites* with indirect life cycles) where it is relevant to model both the intermediate and definitive hosts.

# 1.4 *OUTBREAK*: A quick overview

Below is a short description of the program and output files. This quick overview is aimed at first-time users who want to decide whether *OUTBREAK* is relevant for their needs before proceeding further, or as a "refresher" for users who are experienced in using *OUTBREAK* or similar software. However, careful reading of the entire manual, as well as taking advantage of other training materials, are recommended if the user is to obtain a good understanding of *OUTBREAK*. Links to additional online training materials will be added to this user guide as they are developed.

## Purpose

*OUTBREAK* is a user-friendly program for developing individual-based models aimed at simulating the dynamics of diseases. *OUTBREAK* can be used for projects whose purpose is to: predict future trends or quantify risks in a population, given current or hypothesized scenarios; evaluate the effect of management options; or assess the relative importance of variables in the epidemiology of diseases. See Chapter 1.2 for more details.

## Installation requirements

*OUTBREAK* is compiled to run on computers running MS Windows operating system (Windows 7 or higher). It should be possible to run *OUTBREAK* using a Windows virtual machine running on Mac or Linux systems, although computation speed may be compromised. See Chapter 2 for more details.

## Navigation

*OUTBREAK* has two basic levels of navigation. At the top of the input window is a toolbar with high level menus for handling files, asking for help, running simulations, and viewing results. Below this toolbar, data input is carried out by stepping through input tabs that describe specify parameters relevant for the transition of individuals among epidemiological compartments. The most logical sequence in which to work will generally by to follow the order of the tabs.

## Primary features

### Data input (see Chapter 3)

The data input window contains several tabs that allow the user to:

- Set the overarching model parameters and labels, add new scenarios, and select optional output files on the **General Settings tab.**
- Define transitions between disease states on the **P (pre-susceptible) tab**, **S (susceptible)**, **E (exposed and infected)**, **I (infectious)**, and **R (recovered and resistant)** tabs.
- Include disease intervention options in the model by specifying vaccination or removal protocols on the **Vaccination tab** and **Removals tab**.
- Define the initial population size and distribution of individuals across age classes and disease states in the beginning of the simulation on the **Initial Population tab**.
- Use the **Demography tab** to define the life history traits of the population, including timing of breeding, annual or stage-based mortality, annual fecundity, proportion of breeding females, brood size and frequency, and carrying capacity.
- Include a (relatively simple) spatial component in the model by parameterizing the initial distribution of individuals and specifying rules for their movement over time on the **Spatial Settings tab**.

### Run simulations (see Chapter 4)

Click the *Run* button in the top toolbar of the program window to open a new window with options for adjusting model parameters, choosing scenarios to simulate, adding optional output files, and starting the simulations. The movement of individuals can also be displayed graphically on the right side of the opened **Run window** if the model has a spatial component.

### View results (see Chapter 5)

Once the simulations are complete, click the *Results* button in the top toolbar of the program window to explore results graphically and to save plots (or the underlying data used to generate plots) to disk. The opened **Results window** displays plots for the proportion of individuals in each disease state at the end of the simulation as well as the disease prevalence, number of individuals in each disease state, and change in population demographics over the simulated time period.

### Access output files (see Chapter 5.2)

The output data that are displayed in text, tables, and graphs are all saved in files that are placed in a _Results subfolder of your project folder. Default output files report summary statistics (mean and standard deviation) of demographic and disease variables, as well as more detailed data, on a per iteration basis. Additional output files can be requested to extract further information (e.g., individual lists with trait values). All output files that contain data are semicolon (";") delimited text files that can be loaded into other programs (e.g., R, Excel) for further processing.

## Advanced Features

### Functions (see Chapter 6)

Most data input fields allow values to be specified as functions of population or individual parameters (e.g., disease susceptibility as a function of age) in order to build more complex models. This allows certain events to be influenced by others, or to support the use of specific probability distributions to generate parameter values.

### State Variables (see Chapter 3.5)

From version 2.4 onwards, *OUTBREAK* also includes a **State Variables tab**. This provides the capacity to define Individual and Population State Variables, which can be used to define individual or population-level traits or to keep tallies of such traits in the population.

### Function Evaluator (see Chapter 6.2)

The combination of these two features (i.e. functions and State Variables) provides an extremely flexible modelling framework. However, note that the improper use of functions and State Variables can result in the creation of unintended dynamics in the model. The user is strongly advised to validate the addition of these features in the development phase of their model. To facilitate this process, *OUTBREAK* is equipped with a **Function Evaluator**, a tool for testing function output and editing function syntax.

## Helpful tips and warnings

- Click on input fields and hover the mouse over them to display tool tips that explain what type of data is expected (e.g., proportion or integer) and whether functions can be used.
- If a value outside the possible range for a parameter is entered, an error window will open. It will not be possible to save the project until the error is fixed.
- All rates are intended as a function of the time step. The default time step is one day, but the user may define an alternative period (e.g., a week, a month, a season). Throughout this manual, as a convention, we refer to a time step as one day.

# Chapter 2. Getting started

This chapter provides guidance on system requirements, software installation, file management, and project creation.

**Please register as an *OUTBREAK* user!**

*Registration can be completely anonymous and will provide us with information about how the SCTI software is being used and how we can better serve species conservation. Register at* scti.tools/register/.

## 2.1 System requirements

*OUTBREAK* is compiled to run on the MS Windows (Microsoft Corp. Redmond, Washington, USA) operating system, from Windows 7 onwards. Compatibility with all future versions of Windows is not guaranteed. It should be possible to run *OUTBREAK* using a Windows virtual machine running on Mac or Linux systems, although computation speed may be compromised.

*OUTBREAK* was written in the C# language, compiled with MS Visual Studio (Microsoft Corp. Redmond, Washington, USA), and uses some user interface controls from ComponentOne Studio WinForms (Grape City Inc., Pittsburgh, Pennsylvania, USA).

## 2.2 Installation

*OUTBREAK* is available for free download online at https://scti.tools/outbreak/. The installation (OutbreakInstallation.msi) automatically creates a program short-cut on the desktop, configures necessary folders, and includes this manual as well as a few sample project files (with an .xml extension). If the user downloads the compressed (.zip) file rather than the installer, they will have to extract the two files and then run either OutbreakInstallation.msi or setup.exe if the former did not work.

### Updated Versions

Updated versions of *OUTBREAK* will be released periodically, either as minor fixes to bugs or as major upgrades with new features. Once a month, *OUTBREAK* will attempt to check for updates when opened if the computer on which it is installed is connected to the internet. The user can also check for updates by visiting the SCTI website (www.scti.tools) or by using the *Help* menu in the top toolbar of the program window to check if the current release on the website is more recent than the user's current version. Attempts will be made to make all upgrades backward compatible to accept *OUTBREAK* project files created with prior versions, but no guarantee can be provided. The updated installation of *OUTBREAK* will place into the *OUTBREAK* program folder a document file (OutbreakChangeLog.doc) that lists the primary changes that have been made recently to the *OUTBREAK* program. This change log is also available on the website at https://scti.tools/outbreak/.

## Folder and file management

During the installation process, the user is asked to select a location for the sample files (the default location is C:\OutbreakProjects) and for the installation files (the default location is C:\Program Files (x86) \Species Conservation Toolkit Initiative\Outbreak). If *OUTBREAK* project files, such as the sample files, are placed in a folder for which the user does not have reading and writing rights, *OUTBREAK* will fail to run the projects. Thus, some users might prefer or need to have the OutbreakProjects folder created in a location other than the default, where they have writing and reading rights (typically a sub-folder within MyDocuments).

# 2.3 Creating and saving projects

When *OUTBREAK* is loaded, the opening screen is displayed as shown below (Figure 2), giving information about the software version and the sponsors of *OUTBREAK*. Most users will create and run *OUTBREAK* population simulations from within the *OUTBREAK* user interface, although the functionality of the *OUTBREAK* simulation is available within program libraries (dll files) that can be accessed by other interfaces (such as *METAMODEL MANAGER*, available at https://scti.tools/metamodelmanager/).

The opening window provides options to:

- create a new project (which will initially have a default scenario);
- open an existing previously saved project;
- open a list of recently opened projects for quicker access;
- or exit the program.


**Figure 2. *OUTBREAK* opening window.**

## Creating a new project

Click the *New Project* button for *OUTBREAK* to open the input window, which contains several tabs. Chapter 3 provides a detailed description of each *OUTBREAK* input screen, including information about the features and choices available and their implications for modeling infectious diseases.

A default scenario ("Scenario1") will be available automatically, and it is important for the user to change the initial default input values on every input tab to values appropriate for the situation being modelled. The default values allow the user to run a test scenario to check that *OUTBREAK* is working as expected. The defaults are also often useful indicators of the magnitude and format of a typical input value (e.g., a percent, or an integer, or a number from 0 to 1). To clear all input values quickly, use the File menu in the top window toolbar to select *Clear Input*.

## Opening an existing project

To open an existing project, click the *Open Project* button. The Open dialog will initially search for an .xml file in the most recently used OutbreakProjects folder, but the user can also navigate to other folders. For quicker access to a list of recently opened projects click the *Recent Projects* button. To switch from an opened project to a different existing project, use the File menu in the top toolbar of the program window to select *Load input file* to search for a .xml file in the most recently used OutbreakProjects folder or *Open Recent* to select from a list of recently opened projects.

## Saving a project

To save a PMx project, use the File menu in the top window toolbar to select one of the following options:

- *Save Project to* automatically saves the project with the name chosen when the project was created (*.xml).
- *Save Project As* to specify the name of the project file and/or location to which it will be saved.

# 2.4 Helpful tips and warnings

- Generally, parameter values in *OUTBREAK* are numbers. However, in most cases, functions can be entered to allow, for example, a value to be drawn from a probability distribution or to link the value of a parameter to particular variables. See Chapter 6 for a detailed explanation of the use and syntax of functions in *OUTBREAK*.

- After clicking in a field where parameter values can be changed, hovering the mouse over that field displays a tool tip. These help messages explain what type of data

*OUTBREAK* is expecting (e.g., a proportion or an integer) and whether functions can be used.

- Any value that is modified will be highlighted in red to remind the user of the modifications that have been made since the project was loaded. If a value outside the possible range for the parameter being modified is entered, an error window will pop up. When this occurs, it will not be possible to save the project because *OUTBREAK* detects that there is a critical error. The user is encouraged to address these problems immediately so that the project can be saved at regular intervals.

- If a function with incorrect syntax or an illegal character is entered in a field, *OUTBREAK* will detect this problem only when it evaluates the function. That is, *OUTBREAK* will report the error only when the simulations are run. For this reason, users are strongly encouraged to check functions using the **Function Evaluator** at the time they are entered (see Chapter 6). To access the Function Evaluator, click *Function Evaluator* in the top toolbar of the program window.

- When a field requires a value between zero and one, and the function entered returns a value outside this range, *OUTBREAK* will truncate the value to 0 or 1.

- **Throughout this manual, as a convention, we refer to time step as one day.**

  *This is because the default time step in OUTBREAK is one a day (with a year defined as 365 days). While the user can specify the time steps differently on the **General Settings tab**, it is the user's responsibility to enter rates in subsequent input tabs as a function of the time step chosen.*

# Chapter 3. Data input

This chapter takes the user through each *OUTBREAK* screen, explaining the features and choices available and their implications for modeling infectious diseases.

## 3.1 General Settings tab

Upon opening a project, *OUTBREAK* displays the input window, which contains several tabs. The first of these is the **General Settings tab** (Figure 3), where the user can set the overarching model parameters and labels. The data fields and options available on this tab relate to:

- project name and description;
- scenario management;
- projection settings; and
- optional data output files.

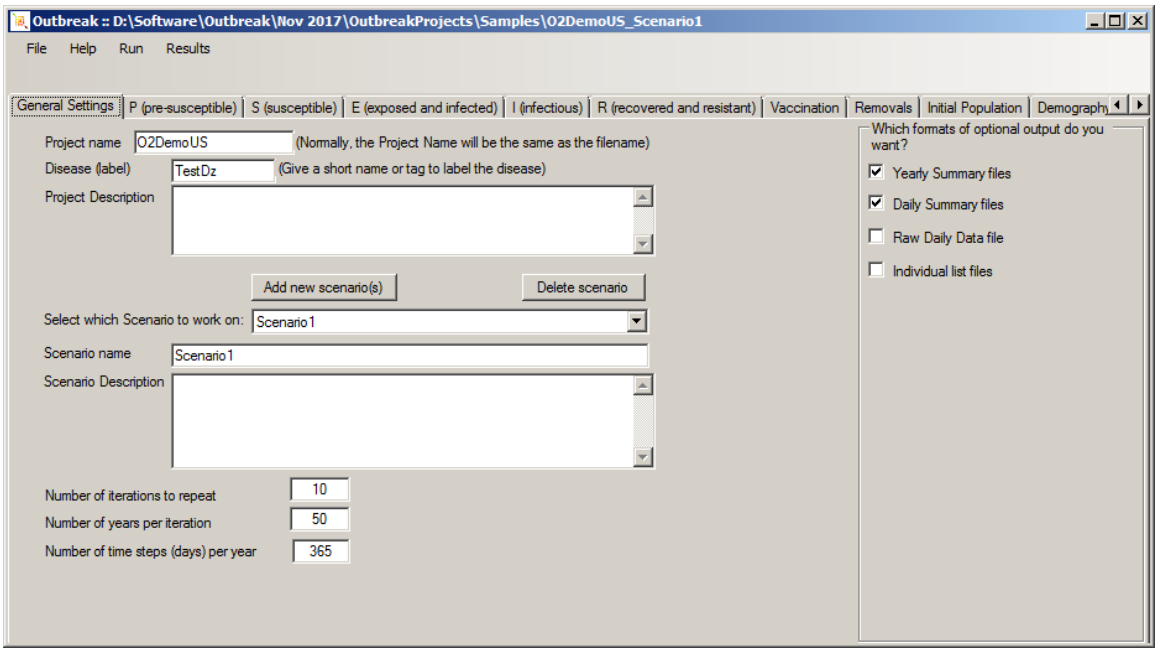



**Figure 3. *OUTBREAK* General Settings tab**

### Project name and description

*Project Name* The *Project name* will be the same as the file name. If desired, modify this name either here in the *Project name* field or via the *Save as* option in the File menu at the top of the input window. Consider using relatively short project names to avoid very long file names.

*Disease (label)* Record a short label in this field to identify the disease.

*Project Description; Scenario Description* These text fields are provided for the user's convenience to add notes, reminders, or justifications of decisions made while developing the project. The user is strongly encouraged to make use of these spaces to document ideas and the reasons behind decisions, as these can easily be lost unless without proper record-keeping.

### Naming your project

*Almost any label can be used. However, because the project name is used as the file name, it is important not to use characters, such as '/', '\', '?', or '*'. These characters are invalid in Windows file names and will prevent the project from being saved successfully. That is, once the input window is closed and the user tries to save the file at the prompt, the project will close and the changes made will be lost!*

## Scenario management

*Add new scenario(s)* Click this button to add a scenario. If multiple scenarios already exist, a dialog box appears prompting the user to select which existing scenario should be used as a template for the new scenario(s). Use the second dialog box to specify how many copies should be made. The new scenario will be named with the suffix "_copy" and a number, which will depend on the number of copies requested.

*Delete scenario* Click this button to delete the currently selected scenario.

*Select which Scenario to work on* Use this drop-down menu to select a scenario to become active. Any further changes in the model will affect only that scenario.

*Scenario name* View and edit the name of the currently selected scenario here. Any scenario name changes are automatically reflected in the *Select which Scenario to work on* drop-down menu. Each scenario name should be brief yet descriptive enough that the user can quickly identify its essential characteristics relative to other scenarios in the project.

## Projections settings

*Number of iterations to repeat* Enter the desired number of (independent) repetitions of the simulation that should be run. It is wise to start with a small number here as the simulation may be slow. *OUTBREAK* simulates and keeps track of each individual in the population, and the computation time can be long

depending on population size, number of simulated years, and model complexity. Once the user has tested the developed model and is satisfied with the model structure and settings, it is generally wise to run a larger number of iterations for the final analysis (at least 100 - 500 to ensure that the variability in results is properly explored). Spaces, commas, or dots should not be used to separate digits in large numbers.

**Defining a time step**

- *Double check that rates entered in subsequent input tabs are expressed as a function of the time step chosen.*
- *If running OUTBREAK in conjunction with other platforms using METAMODEL MANAGER, review Appendix I for more information on synchronizing sub-model time steps. The specification of time steps in OUTBREAK influences the number of cycles of the program that need to run before METAMODEL MANAGER hands data to the next application.*

*Number of years per iteration*

Enter the number of years that should be simulated in each iteration. Note that *OUTBREAK* distinguishes between time steps (days) and years. A year is a collection of time steps. Again, it is recommended to start with simulations with a short duration (perhaps 1 year) to test that the model is properly set up, and then to increase the simulation to the desired length once the model structure and settings are decided upon. The duration of any model scenario is determined by many factors: species life-history, biological characteristics of the disease under consideration, the disease management decision-making context, etc. Think carefully about this parameter in the model development process.

*Number of time steps (days) per year*

Enter the number of time steps that make up a year in the simulation. The default time step is a day, and a year is defined as 365 days. While the user can specify the time steps differently, it is imperative that rates entered in subsequent input tabs are expressed as a function of the time step chosen.

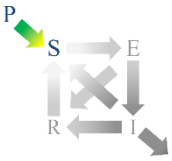
## Optional data output files

*Which formats of optional output do you want?*

On the right side of the **General Settings tab**, the user can select any of the following optional output files by clicking the relevant boxes. These options are also available when running the simulation in the **Run window**. For a detailed description of these formats, see Chapter 5.2.

## 3.2 Defining transitions between disease states

The following five tabs are used to enter parameters that define the transitions between disease states. The transition dynamics among individuals in a population will be the most complex part of the model setup.




### P (pre-susceptible) tab

The pre-susceptible state includes all individuals from birth to the earliest age of susceptibility. Examples might include newborns who have some duration of immunity through acquisition of maternal antibodies, or individuals for whom exposure first occurs after departure from a nest or den. Some or all individuals may be susceptible from birth, and some may never become susceptible, depending on the disease and situation. The **P (pre-susceptible) tab** is shown below in Figure 4. The data fields and options available on this tab relate to:

- initial acquisition of susceptibility;
- maternal-offspring transmission; and
- maternally derived immunity.

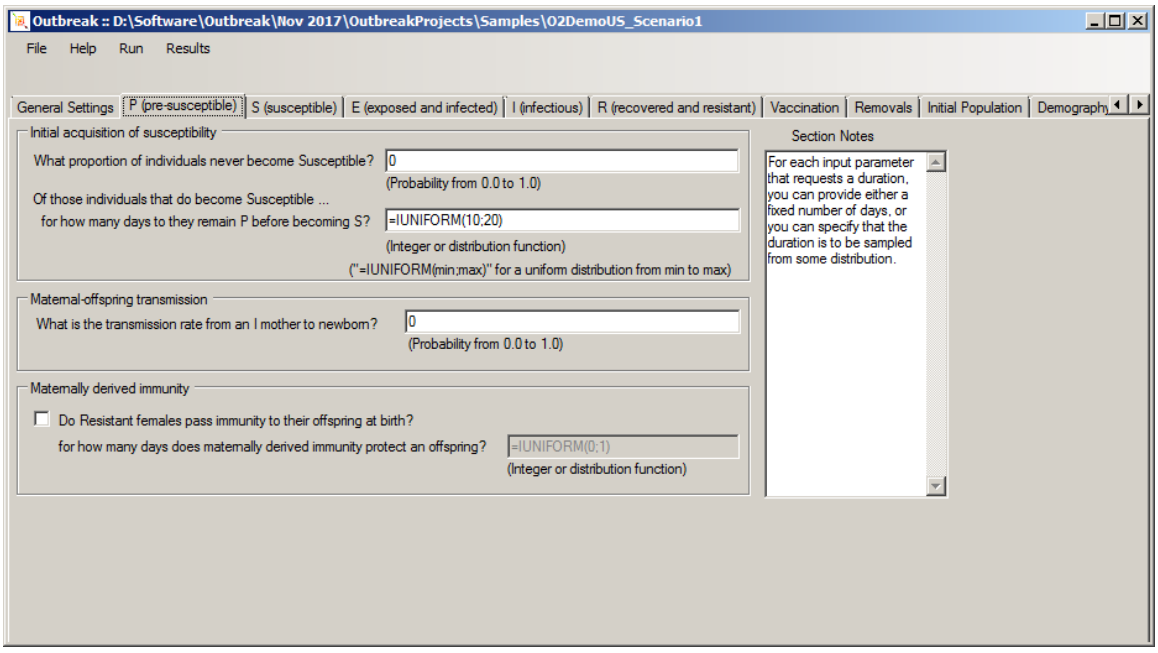



**Figure 4. *OUTBREAK* P (pre-susceptible) tab**

## Initial acquisition of susceptibility

*What proportion of individuals never become susceptible (S)?*

Enter the number of individuals that will never become susceptible, expressed as a proportion (i.e. a number between 0 and 1). If 0, all individuals will become susceptible. If 1, no individuals will ever become susceptible, and all will remain in state P (pre-susceptible) for their lifespan.

For example, if the value 0.2 is entered, OUTBREAK will randomly select 80% of newborns and change their status from P to S after the time specified in the next field has passed. The remaining 20% will never become susceptible.

*Of those individuals that do become susceptible …*

*For how many days do they remain pre-susceptible (P) before becoming susceptible (S)?*

This setting defines the age (in days) at which a young individual may become susceptible to the disease agent. This parameter is relevant only for those individuals that do become susceptible. For example, if there is maternally conferred immunity, the user could enter the number of days following birth after which maternal antibodies are no longer protective. For sexually transmitted diseases this would be the age of sexual maturity. For disease exposure that occurs on dispersal, this will be the age of dispersal.

The default value for this parameter is a uniform distribution of integers (days) for which the user sets a minimum and a maximum value. For example, an entry of "=IUNFORM(0;5)" means that each individual that can become susceptible is assigned an integer value randomly selected from 0 to 5. Those individuals that are allocated the value 0 will become immediately susceptible.

To capture variability other than a simple uniform distribution, it is possible to enter a function to specify that distribution. For example, an entry of "=5+2*NRAND" will sample the duration of the pre-susceptible state from a normal distribution with mean = 5 and standard deviation (SD) = 2 days. For more information on using functions, see Chapter 6.

## Maternal-offspring transmission

*What is the transmission probability from an infectious (I) mother to a newborn?*

This parameter is used in cases where the disease agent can be transmitted directly to offspring from their mother (i.e. vertical transmission). It specifies the proportion of newborns from an infectious (I) mother that become exposed to the agent (E) and is expressed as a value between 0 and 1.

A value of 0 indicates that the infection cannot be transmitted directly to offspring from I mothers, and 1 indicates that all offspring of I mothers become infected (i.e. contract the disease but are not yet infectious to others).

**The *Maternal-offspring transmission route* is independent of the parameter entered in the *Initial acquisition of susceptibility* section discussed above.**

*For example, if the intent is to model a disease that is only transmitted vertically, that can be achieved by setting the proportion of individuals that never become susceptible to 1 and then modifying the transmission probability from an I mother to a newborn to a sensible value for the disease being modelled.*

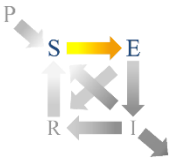
## Maternally derived immunity

*Do resistant (R) females pass immunity to their offspring at birth?*

Check this box when there is passive maternal immunity. That is, when newborns from an R mother will have immunity to the disease. When this box is selected, the following field becomes active.

*For how many days does maternally derived immunity protect an offspring?*

Specify how many days the passive maternal immunity will last in the offspring.




## S (susceptible) tab

A susceptible individual is capable of becoming infected (E) given exposure to a disease agent. This tab, shown below in Figure 5, determines if and how a susceptible individual becomes exposed (i.e. transitions from S to E). The transition from S to E can occur either:

- by encountering an infectious (I) individual; or
- through contact with an environmental disease source.

**Note the distinction between exposed or infected (E) and infectious (I).**

*Both E and I individuals have been exposed to the pathogen and may or may not show clinical signs of the disease; however, ONLY I individuals are shedding the pathogen. That is, an I individual is a source of infection for S individuals.*

*OUTBREAK* will determine the fate of each susceptible individual based on the encounter rates and probabilities of disease transmission entered on this tab. The number of individuals that become exposed (E) depends on the number of infectious (I) individuals, the number of susceptible (S) individuals, the encounter rate, and the probability of disease transmission. If an environmental disease source exists, the rate of transition from S to E is given by the encounter rate with the pathogen in the environment and the probability of disease transmission from an environmental disease source. Note that a number of variables are available within the program to identify characteristics of susceptible and infectious individuals (e.g., ***SAGE***, ***IAGE***, ***SDEX***, ***ISEX***, etc.). Specific information on the use of these variables and more general functions in *OUTBREAK* can be found in Chapter 6.

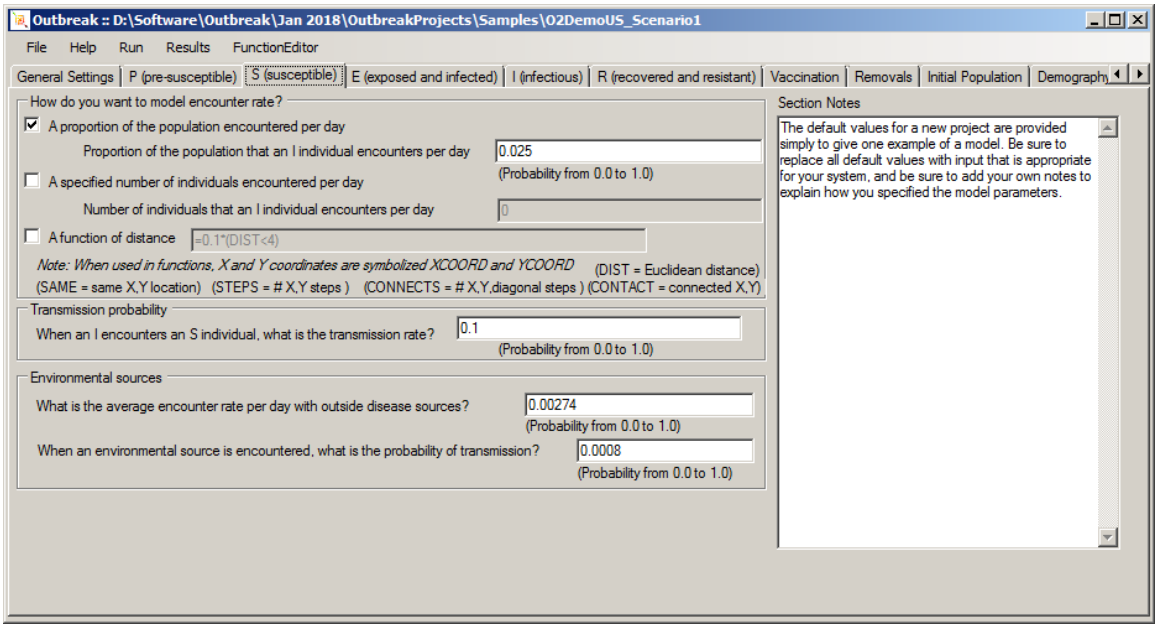



**Figure 5. *OUTBREAK* S (susceptible) tab**

## Box 1. How to quantify transmission rate, encounter rate, and transmission probability.

Quantifying the transmission rate (β), which is determined by the interplay between individuals in a population coming into contact and the probability of transmitting the disease when an I individual is involved in these contacts, is notoriously difficult (McCallum, *et al*. 2001; Vynnycky and White 2010). Most often, researchers have resorted to quantifying the force of infection (λ=βI) by monitoring the number of new infections over time (e.g., following individuals from birth and monitoring changes in their immunological status, Grenfell and Anderson 1985). In other cases, β has been quantified using multi-state mark-recapture analysis coupled with serological data of captured individuals (Tompkins, *et al.* 2009). An additional commonly used approach is to fit a statistical model to field data to estimate the parameter of interest. For example, by fitting a generalized linear model to prevalence data, Caley and Ramsey (2001) estimated leptospirosis and bovine tuberculosis transmission rates in possums. More recently, mathematical models have been developed to estimate epidemiological parameters from pathogens' molecular data (Kühnert, *et al*. 2011; Stadler, *et al*. 2012; Kühnert, *et al*. 2013; Stadler, *et al.* 2013; Gavryushkina, *et al*. 2014). These molecular methods, despite being extremely promising, might not be accurate in complex systems, such as wildlife populations, where many underlying assumptions may not be met. Such methods may need further validation in the field before they can be generalized.

Things are somewhat more complicated in *OUTBREAK,* because it separates the encounter rate from the probability of disease transmission. Therefore, while the examples above may provide an indication of what the mean value of the product of these two parameters would be, they will not estimate either of them individually. Therefore, the user could enter an encounter rate and a probability of disease transmission whose product matches that estimated in field studies. However, some of the approaches mentioned above can be combined with new technologies or additional field techniques to estimate separately the encounter rate and the probability of disease transmission.

Camera traps are a possible option and have been used to estimate encounter rates between wild and farmed white-tailed deer (Vercauteren, *et al.* 2007) and between white-tailed deer and livestock (Kukielka, *et al*. 2013). The latter study also evaluated the effect of environmental conditions (seasons) on encounter rates by fitting a regression model to the data.

Many attempts have been made to quantify encounter rates from the characteristics of population social networks, and a variety of techniques have been used. For example, using GPS collars, Schauber, *et al.* (2007) have reconstructed direct and indirect contact

## Box 1. Cont'd. How to quantify transmission rate, encounter rate, and transmission probability.

matrices in white-tailed deer by keeping track of encounters between individuals and landscape usage. VanderWaal, *et al*. (2014) quantified individuals' home range overlap proportions to estimate possible interaction between giraffes across the landscape and used an observational study to determine the social network in the study population. Proximity data loggers have been used to evaluate potential differences in the transmission rate of Tasmanian devil facial tumor disease between seasons (Hamede, *et al.* 2009). Once the encounter rates are estimated, it is possible to extrapolate the probability of infection, potentially for each demographic class (e.g., males, females, or age classes) if incident data (e.g., serological data over time) are available (e.g., Wallinga, *et al.* 2006). Indeed, it has been suggested that social network analysis is of the most promising tools for obtaining insights into disease transmission in wildlife (Craft 2015). However, some challenges still exist. Encounter rates need to be estimated over a relevant time frame with respect to the infectious period (Craft 2015). Also, some difficulties exist in defining what kind of contact is relevant for the disease being investigated (Vynnycky and White 2010). Commonly, field data are collected for a relatively small sample, especially in observational studies or where expensive technology or field work is involved (e.g., GPS collars, proximity data loggers). Therefore, some uncertainty exists about how confidently the resulting information can be applied at the population level (Schauber, *et al.* 2007). A few studies have considered the correlation between indirect measurements of encounter rates (e.g., measuring home range overlap between individuals) and actual contact between individuals, and it is reassuring that a general relationship between the two was confirmed (Schauber, *et al.* 2007; Robert, *et al*. 2012). However, the strength of these correlations was demonstrated to be quite variable (Schauber, *et al*. 2007; Robert, *et al.* 2012). Use of genetic data from pathogens has been recommended as a means of confirming that adequate and relevant encounter rate measurements were obtained, with some approaches being possible alternatives to intensive field approaches (VanderWaal, *et al*. 2014; Craft 2015).

This area of research is in quick and active development. No attempt was made here to make the information provided exhaustive. Instead, the aim was merely to give a sense of the possible options. The final decision on how to estimate model parameter values rests with the modeler and needs to be made on a case-by-case basis depending on the purpose of the model and the resources and time available.

## How would you like to model the encounter rate?

In this section, the user decides how to model the encounter rate in *OUTBREAK*. To facilitate the modelling of diseases that can be transmitted in multiple ways, the user can select multiple options in this section. In these cases, *OUTBREAK* will sequentially step through the different transmission modes.

- **A proportion of the population encountered per day**

  Use this option to model the number of individuals encountered as a function of the population size. When this box is selected, the following field becomes active:

  | | |
  |---|---|
  | *Proportion of the population that an I individual encounters per day* | Enter the proportion of the population, per day, that an infectious individual encounters. For example, if the value 0.1 is entered, it means that an infectious individual encounters 10% of the population every day, through its movement. |

- **A specified number of individuals encountered per day**

  Use this option to express the encounter rate as a fixed number of individuals per day. If this box is selected, the following field becomes active:

  | | |
  |---|---|
  | *Number of individuals that an I individual encounters per day* | Enter the number of individuals encountered per day. |

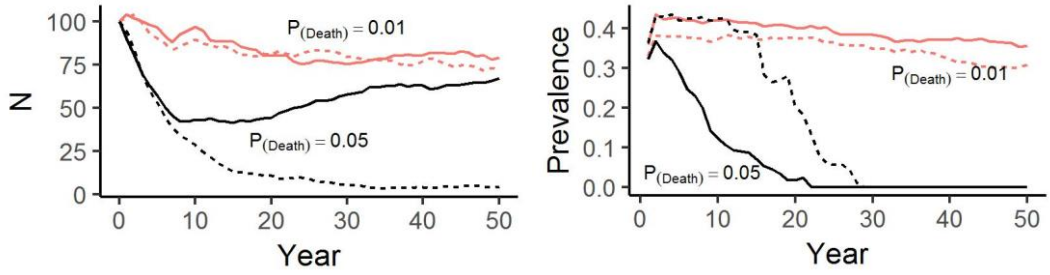
### Note that *OUTBREAK* does not choose specific individuals to be encountered.

*In most models, OUTBREAK does not choose specific individuals to be encountered but will instead randomly select the desired number of individuals from the total population. This is unlikely to be realistic in species that have a strict social structure. For example, in OUTBREAK an individual lion might encounter a random set of 10 individuals each day, but the actual individuals encountered will change daily. For social animals like lions, it would be more realistic for individual lions to be modelled as mostly encountering only the 10 individuals in their social group. If more specific social structures are desired for the model, you can use State Variables to create that social structure (See Chapter 3.5), or consider linking OUTBREAK to more sophisticated population structure models designed specifically for the population in question (e.g., Keet, et al. 2009). These more complicated encounter probabilities can then be used as input data for each daily time step in OUTBREAK to model the transmission of disease. For more information on metamodels, see Appendix I.*

## Box 2. Encounter rate: as a proportion of the population or a fixed number of individuals.

The choice of how to express encounter rates has important implications for disease and population dynamics. Imagine that a decision is made to model the encounter rate as a fixed number of individuals per day. If the population size decreases over time, either due to natural causes, the disease itself, or a combination of reasons, the 'per capita' risk of encountering an I individual increases (Keeling and Rohani 2011). Conversely, if the population size increases over time, then the 'per capita' risk of encountering an I individual decreases. When the disease has an effect on the demography of the population being modelled, this can lead to substantially different results than those that would be obtained if the encounter rate was modelled as a proportion of the population size. In the latter case, the 'per capita' risk of encountering an I individuals is constant, because the encounter rate would scale linearly with the population size (Keeling and Rohani 2011).

The results presented below, which were obtained by modifying the sample project distributed with *OUTBREAK*, are intended to exemplify these effects.




In the two plots above, the mean host population trajectory (left plot) and disease prevalence (right plot) are shown, for simulations where the probability of death as result of infection was set to 0.01 (red lines) or to 0.05 (black lines). Starting population size was set to 100 individuals. Encounter rates were set to 2.5 individuals/day (dashed lines) and 2.5% of the population (solid lines). No environmental disease sources were included. When the effect of the disease on the demography of the population is minor (P(death from infection) = 0.01), the trajectories created by the two different encounter rates are similar. However, when the impact of the disease is more severe (P(death due to infection) = 0.05), the resulting dynamics are quite different. As theory forecasts (Keeling and Rohani 2011), population size predictions are far less optimistic when the encounter rate is frequency-dependent rather than density-dependent.

- **A function of distance**

  Use this option to model encounters between infectious individuals and other individuals in the population as a function of the distance between them. When this option is used, the **Spatial Settings tab** will become active and the user will have to specify how movement patterns are regulated throughout the landscape.

  Chapter 6 describes in detail the use of functions in *OUTBREAK*. The following are useful operators in this context:

  - ***XCOORD*** and ***YCOORD*** indicate X and Y coordinates, respectively.
  - ***SAME*** can be used to force encounter only when the individuals have the same location (it returns true or false).
  - ***STEPS*** indicates the vertical or horizontal distance between individuals.
  - ***CONNECTS*** calculates the minimum distance between individuals, including diagonal movements. The difference between ***STEPS*** and ***CONNECTS*** is that ***STEPS*** counts only vertical or horizontal steps, while ***CONNECTS*** allows cells to be traversed diagonally.
  - ***DIST*** indicates Euclidean distance ($\sqrt{[(XCOORD_A - XCOORD_B)^2 + (YCOORD_A - YCOORD_B)^2]}$) and
  - ***CONTACT*** indicates that individuals have to be in contiguous cells (note that ***CONTACT*** returns true (1) or false (0)).

| | | |
|---|---|---|
| A | | |
| B | | C, D |
| | | E |

**Figure 6. Graphic representation of individuals on a 3X3 grid in *OUTBREAK*.**

  Figure 6 represents the distribution of five individuals on a 3x3 grid. The ***SAME*** function will return true only for the pair C and D. ***CONTACT*** is true only between the pairs A and B, and C or D with E, but not for the pair C and D. The distance between the pair A and E calculated with ***STEPS*** would be 4 (2 horizontal and 2 vertical steps), with ***CONNECTS*** would be 2, and with ***DIST*** would be 2.83.

  Any function can be entered for the function of "distance", so you are not restricted to specifying a distance-dependent function for the encounter rate. Indeed, a function specified as a "Proportion of the population encountered per day" (see above) will work the same if the function is entered instead as a function of distance. (In either case, the function specifies the probability that an I individual will encounter a given S individual.) The only difference between entering the function in these two places is that when it is entered as a function of distance, then *OUTBREAK* will make available the

above distance measures (***DIST***, ***CONNECTS***, ***STEPS***, etc.). However, the calculation of these measures for every possible I x S pair of individuals will slow the simulation, so you should avoid it if your function does not include distance metrics.

## Box 3. How to model parasitic diseases caused by parasites that require intermediate and definitive hosts.

Certain parasites have an indirect life cycle. That is, they require more than one host species, infected in sequence, to complete their life cycle. While in the definitive host (sometimes also called the final or primary host), the parasite is at the sexual stage of its life cycle; in the intermediate host (also sometimes called secondary host), it is at an asexual stage. It is often the case that each host species is not infectious to conspecifics. That is, the intermediate host cannot infect directly other intermediate hosts, and definitive hosts are not infectious to other definitive hosts. For example, the final host of several gastro-intestinal parasites must consume the intermediate host in order to become infected, allowing completion of the life cycle. In consuming prey whose status is "exposed", a final host will itself become exposed and possibly infected. Though it might not be infectious to other final hosts, it may become a disease source for intermediate hosts through, for example, contamination of the environment with oocysts, which can infect a susceptible intermediate host once ingested. These diseases are potentially complex to model, and there are several different means of modelling the dynamics of such systems using *OUTBREAK*.

When there is no intraspecific transmission, individuals that are exposed are not infectious to other conspecifics, hence they remain E as far as *OUTBREAK* modelling is concerned. The user can extend the duration of E to, or beyond, the maximum longevity of individuals. This will allow those who are infected (E) but not infectious (I) to remain in state E for their remaining lifespan.

If the interaction between intermediate and definitive host is not a limiting factor for the disease (which is often the case when one of the host species is an invertebrate and the interest is in the dynamics of the other host, which is a vertebrate species), it is easier (and probably faster!) to simplify the model and use the environmental disease source option in *OUTBREAK* as a proxy for the parasite stage in the invertebrate host. This approach will still allow for a level of control of the 'abundance' and 'distribution' of the invertebrate by adjusting the parameters in the model for the **Environmental sources** section on the **S (susceptible) tab** (e.g., linking them to changes over the year or season to simulate invertebrate responses to variable climatic conditions). If the changes of status between intermediate and final hosts are to be modelled explicitly, then the development of a multi-species model using *METAMODEL MANAGER* will be required.

## Transmission probability

*When an I encounters an S individual, what is the probability of transmission?*

Enter the probability that an encounter between an S and I individual results in the transmission of the pathogen to the susceptible individual.

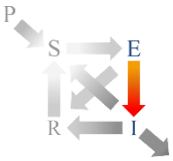
## Environmental sources

*What is the average encounter rate per day with an outside disease source?*

This parameter defines the probability of coming into contact with an environmental disease source. It is relevant when exposure to disease may also occur through contact with sources other than an infectious individual. Outside sources include environmental contamination (e.g., soil, pens, etc.), contaminated food, air or water, and vectors (e.g., mosquitoes, ticks, etc.). Enter the probability per day that an individual will encounter one of these environmental disease sources.

*When an environmental source is encountered, what is the probability of transmission?*

For the disease of interest, specify the average probability of disease transmission from the environmental sources. This may vary depending on the type of source, the extent of contamination, and environmental conditions.




# E (exposed and infected) tab

An exposed (E) individual is one that has encountered an infectious individual or an outside disease source and contracted the disease agent but has not yet become infectious. Such individuals can remain in this state for an extended period of time that is determined by the incubation, or latent period, of the disease. In the model, the duration of this period will be defined by the range of days entered on the **E (exposed and infected) tab** (Figure 7). In a population, exposed animals do not contribute to the infection of S individuals and so do not influence the rate of disease spread. However, by explicitly modelling the latency period, the model will more accurately reflect the change in the pool of S individuals. From a demographic point of view, if being in an E state results in altered reproductive or survival rates, then these mechanisms can be modelled on the **Demography tab**.

## Incubation period

*What is the duration of the incubation period (latency) in days?*

Enter the number of integer days required for an individual to become infectious following exposure. This is the incubation or latent period for the disease. When 0 is used, *OUTBREAK* will change an individual's disease state the following day in the simulation (as it does when 1 is used).

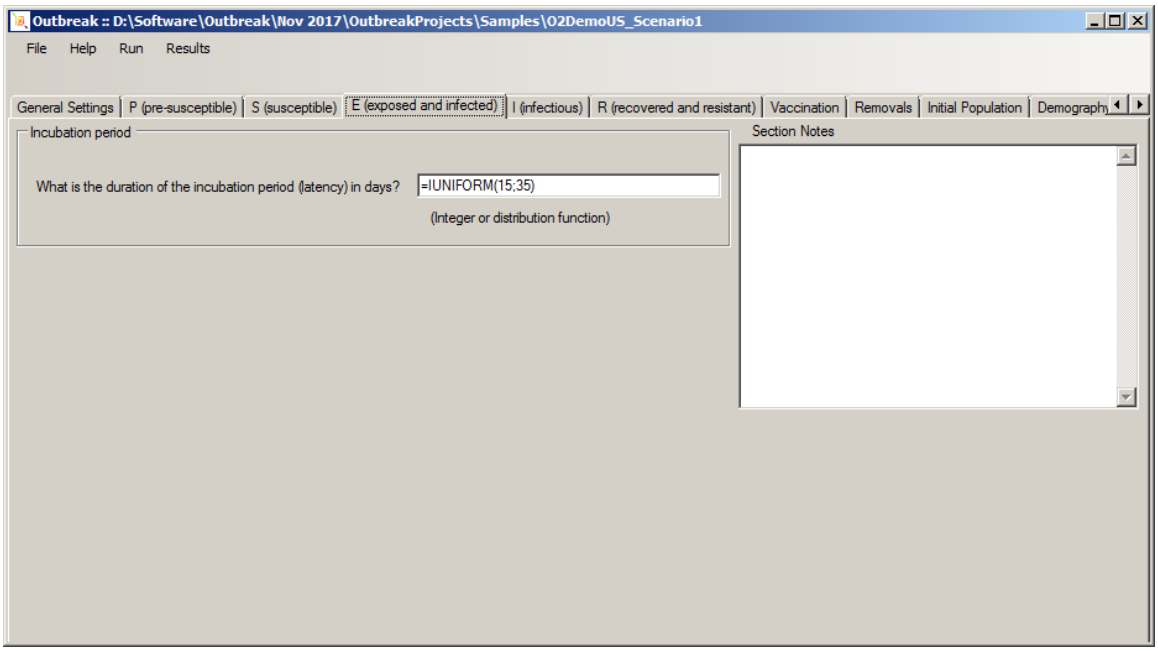
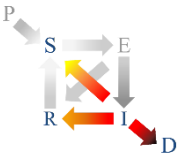



**Figure 7. *OUTBREAK* E (exposed and infected tab)**




## I (infectious) tab

An infectious individual is one that is actively shedding the disease agent and is, therefore, capable of transmitting the disease to another individual. The **I (infectious) tab**, displayed below in Figure 8, determines the average number of days for which an individual remains infectious and the fate of the infectious individual based on disease outcome probabilities.

An infectious individual may:

- remain permanently infectious (remain in I);
- recover without immunity (move to S);
- recover with immunity (move to R); or
- die.

Each of these options has a specific probability of occurring. The program will use these probabilities to simulate disease progression for each individual. Note that there is no immediate way to model a transition from I to E, except by the individual returning to S and then transitioning to E. If there is a need to model this transition explicitly, this may be achieved by creating an Individual

State Variables that tracks whether an 'I' individual can still infect others (see Chapter 3.5 for more information on using the **State Variables tab**).

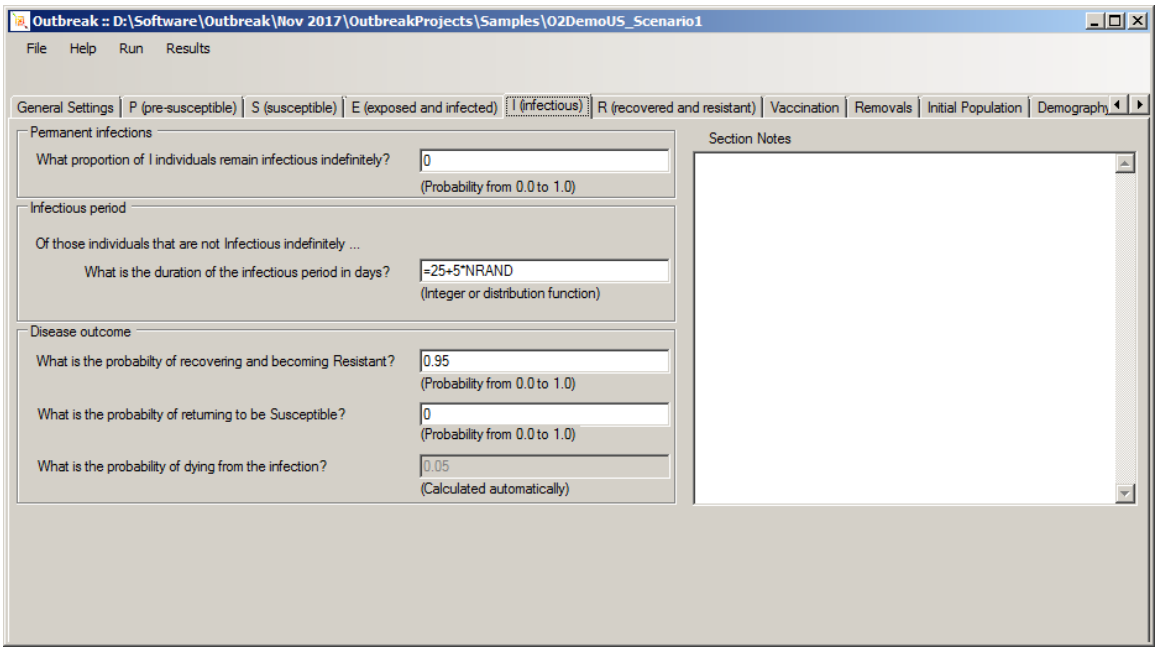



**Figure 8. *OUTBREAK* I (infectious) tab**

## Permanent infections

*What proportion of I individuals remain infectious indefinitely?*

Enter the proportion of infectious individuals that remain in this state for the rest of their life.

### Modelling permanent infectiousness by altering other parameters

*Permanent infectiousness can also be modelled by altering the parameters in the **Infectious period** and **Disease outcome** sections on this tab. It may be beneficial to do this, for example, if infectious individuals have higher mortality rates. The user should carefully evaluate which of the possible approaches is more appropriate. Similarly, if the duration of the infectious period is linked to other variables, such as age, the user can enter a function in the Infectious period section to handle these cases, as exemplified below.*

## Infectious period

*Of those individuals that are not Infectious indefinitely …*

*What is the duration of the infectious period in days?*

This parameter defines the number of days an individual may be shedding the disease organism and is capable of transmitting it to other individuals. It is relevant only for the individuals that were not defined as permanently infectious in the **Permanent infections** section on this tab. When a value of 0 is used here, the individual will immediately change disease status.

### Remember that "infectious" really refers to individuals that shed the pathogen.

*Being infected (E) does not necessarily imply that the individual is infectious (I). A good example is toxoplasmosis in cats. Infected cats normally shed oocysts for only a few weeks despite remaining infected for a much longer period. In this case, the user has to ensure that the number entered here is the actual number of days that the pathogen is shed. For certain diseases, whether an infected individual remains infectious or not will depend on other specific factors. For example, individuals infected with the Bovine Viral Diarrhea virus (BVDV) remain infectious for their lifetime if they have contracted the disease in utero. When they contract the disease as adults, they are usually infectious for only a limited number of days. This could be modelled by entering* ***0*** *in the* ***Permanent infections*** *section, and a function, for example* ***=IF(AGE=1;X;Y)****, where* ***X*** *is the maximum age of the species being modelled and* ***Y*** *is the infectious period for individuals infected later on in life. This will result in the duration of the infectious period being evaluated in all individuals. For those that were infected in their first day of life (i.e. born infected), the infectious period is equal to their lifetime; for all others the infectious period is set to* ***Y****. For more information on functions, see Chapter 6.*

## Disease outcome

In this section, enter the probabilities for each of the possible disease outcomes when an individual leaves the I state. The three possible outcomes for I individuals include:

- recovering and becoming resistant;
- returning to being susceptible; or
- dying from the disease.

The probabilities associated with these three events (labeled $P_{(Resistance)}$, $P_{(Susceptibility)}$, $P_{(Death)}$, respectively) are linked by the following relationship: $P_{(Resistance)} + P_{(Susceptibility)} + P_{(Death)} = 1$. As no other outcomes are possible, the three probabilities must sum to 1, so the user needs to know only two of the probabilities in order to calculate the third.

**Be careful!**

*On this tab only the probability of death strictly related to the disease should be considered. Any other natural (or additional) cause of mortality should be dealt with on the **Demography tab**. Also note that permanently infectious individuals are not included in this equation to the left.*

- **Recovering and becoming resistant**

  Depending on the disease, an individual's immune response may allow it to eliminate the infection and may, in addition, confer some level of immunity or resistance to future infection.

  *What is the probability of recovering and becoming Resistant?* — Enter the probability that an infected individual will become resistant (i.e. that it will move from I to R).

- **Returning to being susceptible**

  *What is the probability of returning to be Susceptible?* — Enter the probability that an individual returns to the S state following infection. Note that this transition is functionally equivalent to a transition from I to R, followed by an instantaneous transition from R to S.

- **Dying from the disease**

*What is the probability of dying from the infection?*

The probability of mortality due to disease for infectious individuals that do not remain permanently infectious is automatically calculated by *OUTBREAK* using the probabilities of becoming resistant and of returning to susceptibility. That is,

$$P_{(Death)} = 1 - [P_{(Resistance)} + P_{(Susceptibility)}]$$

Individuals that remain permanently infectious are not included in this calculation. This option should be the primary way to model mortality that is due to the disease.

**Note that, classically in epidemiology, the disease state R has been used to describe several conditions.**

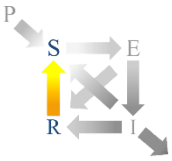
*The disease state R has been used to describe several conditions, all resulting in the temporary or permanent removal of individuals from the susceptible (S) and infectious (I) pools. For example, R has been used to mean "Resistant" (in the sense that OUTBREAK uses it) or to mean physically "Removed" from the population, either due to death or to some other reason. There are also cases in which behavioral changes prevent transmission of the disease.*

*OUTBREAK handles most of these cases separately. For example, removal of individuals can be modelled explicitly on the **Removals tab**. Similarly, the effect of a vaccination program can be explicitly modelled on the **Vaccination tab**. When death occurs as result of infection, this is modelled by modifying the probability of dying from the infection or the mortality rates associated with the disease state. However, there are situations when the R status in OUTBREAK can be interpreted differently from the natural immune resistance to diseases. For example, because the removal plan applied within the management options is permanent, if the user is interested in evaluating the efficacy of a temporary removal (e.g., moving infectious (I) individuals into a quarantine facility), this could be achieved by considering R individuals as I individuals that are temporarily removed to quarantine and then returned to the S pool at a later time.*

### Box 4. More about the infectious period and disease outcome.

It is important to keep in mind that the *Infectious period* parameter on the **I (infectious) tab** describes the time it takes for individuals to transition from state I to any of the possible disease outcomes. This assumes that transitions to these outcomes take roughly equal amounts of time. Where this is not the case, the user has a few options. The easiest option is simply to ignore any differences, which is appropriate when they can be considered irrelevant in terms of the overall dynamics being modelled. Alternatively, the timing can be adjusted using functions. For example, if individuals can be infectious for 10 days, but the disease takes a longer clinical course that may result in death, say, 30 days after becoming infectious, it is possible to set the *Infectious period* to 30 days on the **I (infectious) tab** and then limit the *Transmission probability* on the **S (susceptible) tab** to the first 10 days of an individual's infectious state (e.g., using the function *=(IDSTATE<11) * X*, where *X* is the transmission probability).

Another example includes a disease that has a hyper-acute form that generally results in death but also a sub-acute form that results in individuals either becoming resistant or returning to being susceptible. Individuals that die due to the disease will normally do so in a short period of time, while those that will eventually recover or return to the susceptible state will persist in the infectious state for a longer period. In an example where there are only two possible outcomes: death and becoming resistant, this could be modelled in *OUTBREAK* by specifying the infectious period as a bimodal distribution. Then, the probabilities of transitioning from I to R could be made to be a function of the time spent in the I state. If the probability of becoming resistant is, say, 1, when the I individual did not die in the first 10 days from being I, then the function for the probability of recovering and becoming resistant could be *=(DSTATE>10)*. The probability of returning to being susceptible would be set to 0, and *OUTBREAK* would automatically generate the probability of dying, effectively, *DSTATE<11.* For more information on using functions, see Chapter 6.




# R (recovered and resistant) tab

A recovered individual has survived an infection and has cleared the disease agent, while a resistant individual is one that has recovered and cannot be infected again. Most frequently, this will be due to immunity acquired from a previous infection. Becoming R has the effect of preventing or delaying a return to the S state. The **R (recovered and resistant) tab**, shown below in Figure 9**,** defines the duration of resistance for individuals that recover from the disease.

## Permanent resistance

*What proportion of R individuals acquire permanent immunity?*

Enter the proportion of R individuals that become permanently resistant to the disease.

## Duration of resistance for those with permanent immunity

*What is the duration of immunity in days?*

If immunity is not permanent, enter the number of days R individuals will remain resistant to infection. After this time, R individuals will return to the S state. When 0 is used, *OUTBREAK* will change the individual's disease status the next day in the simulation (as it does when 1 is used).

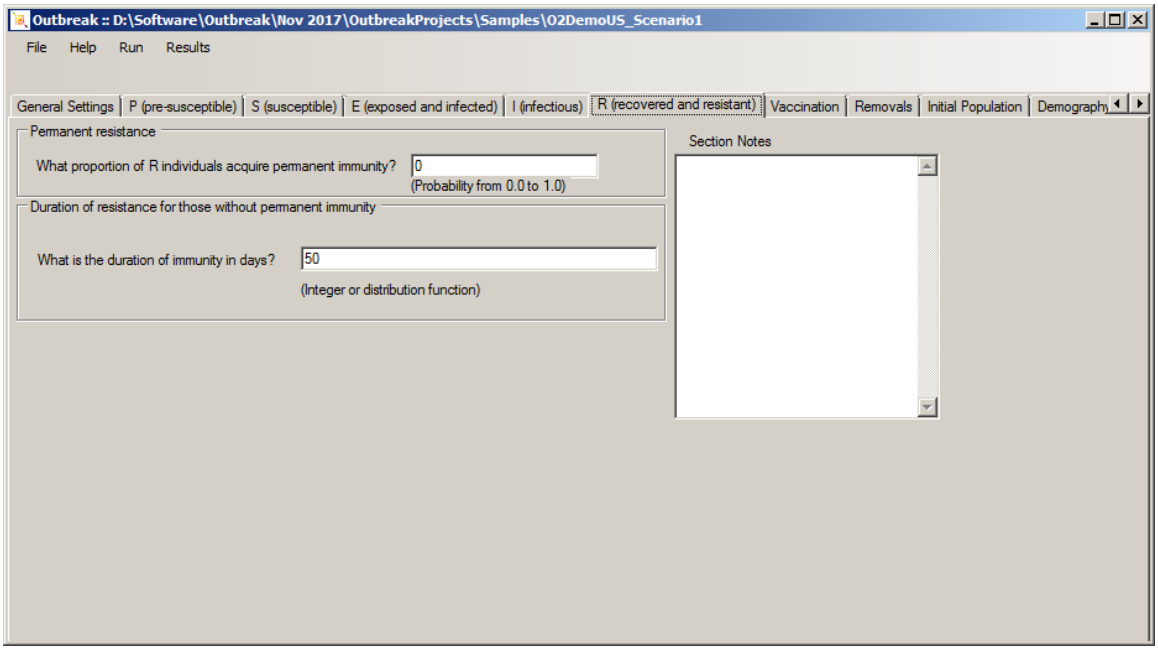



**Figure 9. *OUTBREAK* R (recovered and resistant) tab**

# 3.3 Management Actions

The next two tabs allow the user to enter management options that could be used to stop or prevent the spread of the disease through a population.

## Vaccination tab

Vaccination programs will protect the population (with a certain level of efficacy) from the disease of interest. In an epidemiological sense, vaccinated animals move to an R category without stepping through the sequence E → I → R. Effectively, applying this option aims to reduce the pool of S individuals. The input screen for this management option is shown below in Figure 10. The data fields and options available on this tab specify:

- when to vaccinate;
- what proportion of each age class to vaccinate; and
- the efficacy and duration of the vaccination program.

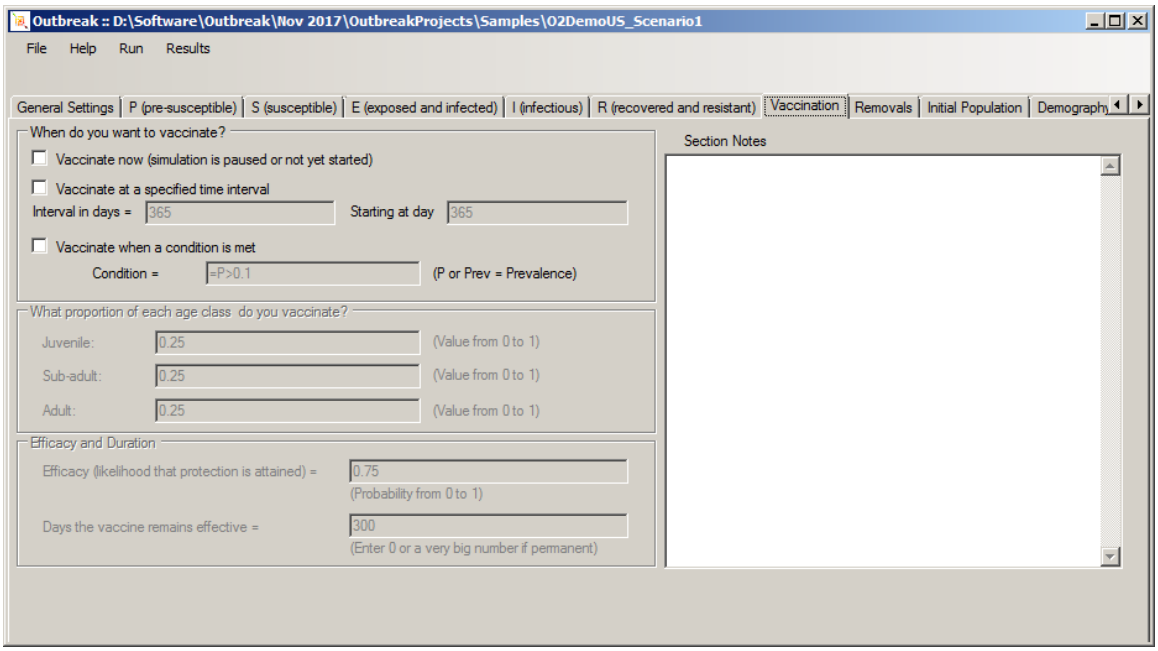



**Figure 10. *OUTBREAK* Vaccination tab**

## When do you want to vaccinate?

This section of the tab controls when the vaccination program should be applied. *OUTBREAK* provides three different vaccination options:

- **Vaccinate now (simulation is paused or not yet started)**

  Use this option to apply vaccination at the beginning of the simulation.

- **Vaccinate at a specified time interval**

  Use this option to apply vaccination at a specific moment in the simulation. When this box is selected, the following two fields become active:

  *Interval in days* Enter the number of days between vaccinations.

  *Starting at day* Specify (in days) when the vaccination program starts. These can be at the beginning of the simulation or sometime later.

- **Vaccinate when a condition is met:**

  Use this option when specific conditions need to be met to implement a vaccination program. When this box is selected, the following field will become active, and the user can enter a function to code what conditions need to be met to trigger the vaccination program.

  *Condition* Enter a function that determines the conditions that need to be met to apply the vaccination program. A common condition might be that the disease prevalence is higher than a certain threshold. In the sample default value for this parameter (=***PREV>0.1***), for example, vaccination is applied when the prevalence is larger than 10%.

## What proportion of each age class do you vaccinate?

Use this section to specify what proportion of each age class will receive the vaccine. These parameter values can be entered as proportions, or as functions that evaluate to a number bound between 0 and 1. For example, modelling different vaccination proportions for males and females can be achieved with the function =***IF(S = 'F'; X; Y)***, where ***X*** proportion of females and ***Y*** proportion of males are vaccinated.

| | |
|---|---|
| *Juvenile* | Specify the proportion of juveniles that will be vaccinated. |
| *Sub-adult* | Specify the proportion of sub-adults that will be vaccinated. |
| *Adult* | Specify the proportion of adults that will be vaccinated. |

## Efficacy and Duration

*Efficacy (likelihood that protection is attained)*

Enter the probability that the vaccination confers full protection against the disease in a given individual. This model definition is consistent with the traditional definition of vaccine efficacy: The percentage reduction of disease in a vaccinated group of individuals compared to an unvaccinated group.

*Days the vaccine remains effective*

Enter the number of days for which vaccinated individuals are protected against the disease. If protection is permanent, the user should enter a number equal to or larger than the maximum age (in days) of the species being modelled. Maximum age is entered on the **Demography tab**.

# Removals tab

Removal of individuals can be effective in preventing or limiting the spread of a disease by decreasing contact rates, reducing the number of S individuals, or both. The input screen for this management option is shown below in Figure 11. The data fields and options available on this tab determine:

- when to remove individuals; and
- what proportion of each age class to remove.

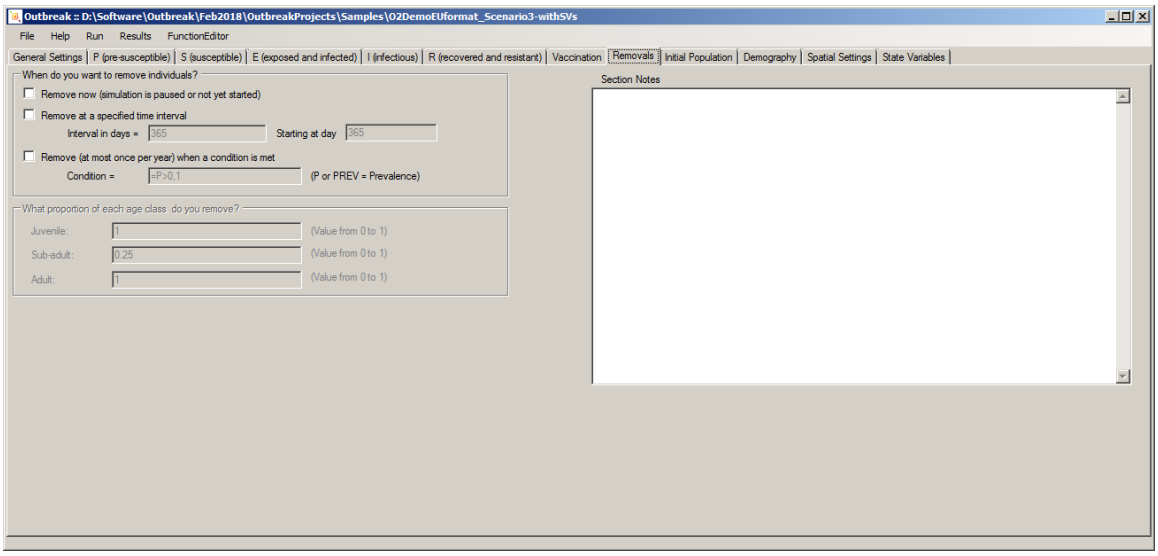



**Figure 11. *OUTBREAK* Removals tab**

## When do you want to remove individuals?

This section controls when individuals are removed from the population. *OUTBREAK* provides three different removal options:

- **Remove now (simulation is paused or not yet started)**

  Use this option to remove individuals at the beginning of the simulation.

- **Remove at a specified time interval**

  Use this option to apply removals at a specific moment in the simulation. When this box is selected, the following two fields become active:

  *Interval in days* Enter the number of days between removals.

  *Starting at day* Specify (in days) when the first individuals should be removed. This can be at the beginning of the simulation or sometime later.

- **Remove (at most once per year) when a condition is met**

  Use this option when specific conditions trigger the removal of individuals. When this box is selected, the following field will become active and the user can enter a function to describe what conditions need to be met.

  *Condition* Enter a function that determines the conditions that need to be met to trigger the removal of individuals. A common condition might be that the disease prevalence is higher than a certain threshold. In the sample default value for this parameter (=***PREV>0.1***), for example, individuals are removed when the prevalence is larger than 10%.

## What proportion of each age class do you remove?

Use this section to specify what proportion of each age class will be removed. These parameter values can be entered as proportions, or, for more complex models, as functions (that evaluate to a number bound by 0 and 1). For example, modelling different removal proportions for males and females can be achieved using the function =***IF(S = 'F'; x; y)***, where ***x*** proportion of females and ***y*** proportion of males are removed.

*Juvenile* Specify the proportion of juveniles that will be removed.

*Sub-adult* Specify the proportion of sub-adults that will be removed.

*Adult* Specify the proportion of adults that will be removed.

# 3.4 Demographic characteristics of the population

The next three tabs are used to define the characteristics of the population to be modeled in terms of initial population size and life history parameters and to model the location of individuals in the landscape (if desired).

## Initial Population tab

An image of the screen to input the Initial population parameters is shown below in Figure 12. The data fields and options available on this tab specify the:

- initial population size;
- initial distribution of disease states; and
- length of juvenile, subadult, and adult age classes in days.

***Remember that OUTBREAK* can only model one population per simulation.**

*METAMODEL MANAGER can be used to model multiple populations or species and can incorporate interactions between them. Through METAMODEL MANAGER, OUTBREAK can be linked to other more sophisticated demographic models, such as VORTEX.*

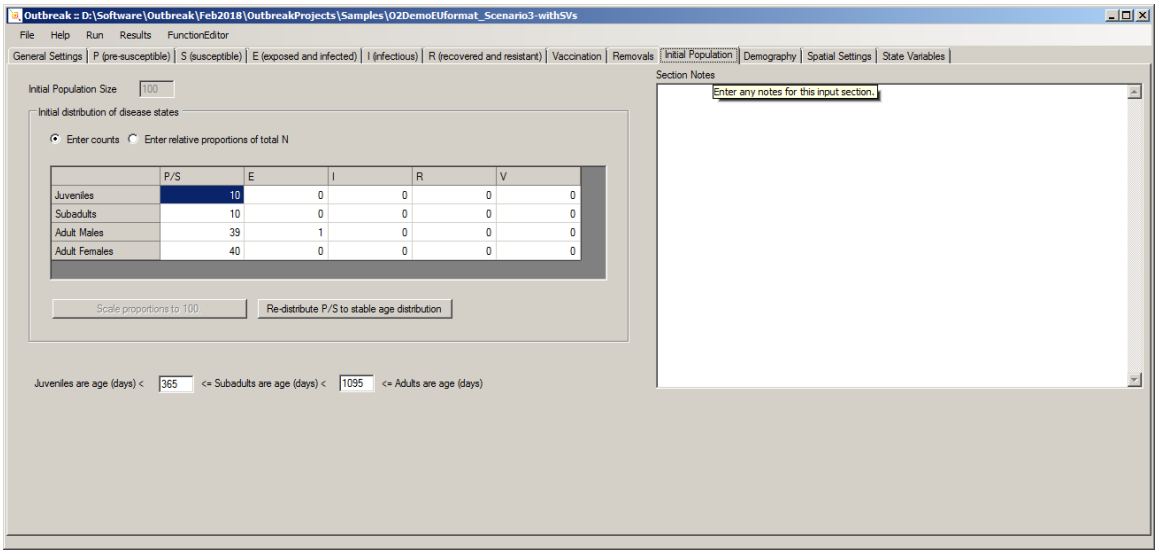



**Figure 12. *OUTBREAK* Initial Population tab**

## Initial Population Size

*Initial Population size*

Enter the number of individuals in the population at the beginning of the simulation. This *Initial Population Size* field is available only when the *Enter relative proportions of total N* option is selected (see the **Initial distribution of disease states** section below).

***As population size increases, the simulation will take longer to run and require more memory.***

*It may be sensible to start with a small total population to test and develop the model, and then increase the total number of individuals to that required in the final simulations.*

## Initial distribution of disease states

Use this section to define how individuals in the population are distributed across age classes and disease states. P/S is Pre-susceptible/Susceptible, E is Exposed, I is Infectious, R is Resistant, and V is Vaccinated. The meaning of the values entered will be different depending on which of the following options is selected.

- **Enter Counts**

  If this option is selected, the numbers in the table are interpreted as actual numbers of individuals.

- **Enter relative proportions of total N**

  If this option is selected, *OUTBREAK* will calculate the number of individuals in each age class based on its allocated proportion, with the total population size obtained by summing all of the entries in the table. The values entered, then, do not need to total to 1, to 100, or to N.

  *Scale proportions to 100* — This button rescales the proportions of individuals across age classes and disease states to add to 100.

  *Re-distribute P/S to stable distribution* — This button redistributes the values in the P/S column according to the expected age distribution calculated from the birth and death rates reported in the S column on the **Demography tab**.

***Testing the impact of different population sizes.***

*If the user has count data that they wish to use to initialize the population, but then would like to test whether there are differences in the results if N is different, it is possible to do this by first selecting the* Enter counts *option and entering specific counts in each age class. Then the option should be changed to* Enter relative proportions of total N. *By doing this, in any additional scenarios created for which the initial N is modified, the same proportional distribution will be retained.*

## Juvenile, subadult, and adult age classes (in days)

*Juveniles are age (days)*; *Subadults are age (days)*:

These last two fields at the bottom of this tab define when juveniles become sub-adults and when sub-adults become adults. These are expressed in days. *OUTBREAK* allows the creation of (only) three age classes with the main purpose being to allow the user to apply different demographic rates and implement different management actions to different age classes.

It should be noted that it is possible to specify an age of first reproduction on the **Demography tab** that is less than the age at which individuals enter into the adult age class. Therefore, some consideration should be given to those species that have multiple age classes. When multiple age classes exist, they may need to be grouped in a way that reflects biological differences in mortality rates among ages, or to reflect the applicability of management options.

For example, consider a carnivore whose offspring are in a den until weaning occurs, then later venture outside but do not follow their parents hunting, then spend a period associating with their parents, finally becoming independent adults that disperse. If, as might be expected, the largest shift in mortality rate occurs when animals leave the den, rather than when they begin to go off hunting with their parents, it might make most sense in the model to define the pups in the den as one life-stage (i.e. juveniles) and the young animals either hanging around outside the den or hunting with their parents as another (i.e. sub-adults).

# Demography tab

This **Demography tab**, shown below in Figure 13, defines the life history traits of the population. The data fields and options available on this tab relate to:

- timing of breeding;
- life table rates; and
- carrying capacity (K).

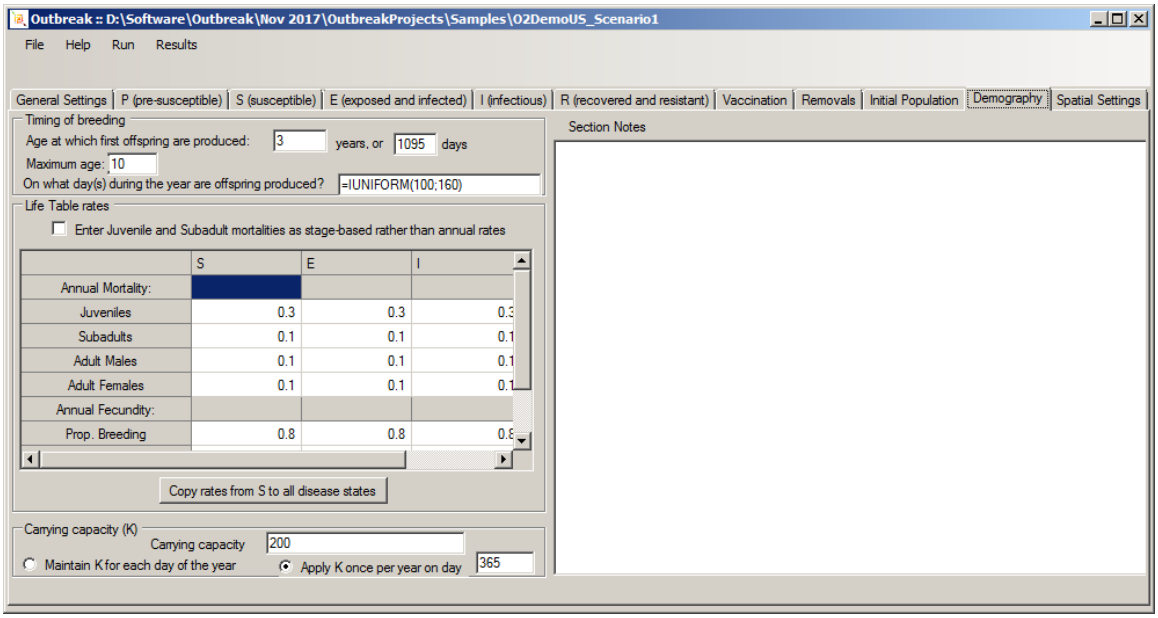



**Figure 13. *OUTBREAK* Demography tab**

## Timing of breeding

*Age at which first offspring are produced*

Enter the age at which individuals are capable of producing offspring. For most cases, this will be the age of sexual maturity. This can be entered in years or days. When one value is modified, *OUTBREAK* automatically calculates the other. Note that the *years* field does not accept fractions of a year (e.g., 0.5, 2.5, etc.), but these can be entered as days (e.g., 180, 912, etc.). When an age is entered in days, the year field is greyed out, and further modification can only be made in days.

*Maximum age*

Enter the maximum lifespan in years.

### *OUTBREAK* assumes maximum longevity and oldest age of reproduction are the same.

*OUTBREAK assumes that the value entered for maximum longevity will also be the oldest age at which individuals are capable of producing offspring. If there is a discrepancy between the oldest age of reproduction and maximum lifespan (that is, individuals can live longer than they can reproduce for), this can be taken into account by using functions in the reproduction rates. For example, if the baseline reproduction rate is* ***m*** *and no reproduction occurs after age* ***x****, the following formula could be entered into the* Annual Fecundity *field:* ***=m*[A<=(365*x)]*** *so that when* ***A*** *is smaller than* ***x****, individuals will breed, but when it is greater than* ***x*** *but smaller than the Maximum age, they will continue to live but without reproducing (note that the age,* ***A****, in OUTBREAK is expressed in days, hence* ***365*x****, under the assumption that a year is made up of 365 days). For more information on using functions, see Chapter 6.*

*On what days during the year are offspring produced?*

This time period defines the range of days in the year when offspring are born. Most commonly, a function with a distribution will be needed to generate a suitable range, but it is also possible to provide a fixed value. If the year is made up of 365 days, and the user is following a calendar year, January 1st is generally considered Day 1; however, this can be modified based on the needs of the model.

For example, it may be convenient to have seasons grouped together rather than split across different years in order to better reflect biological phases in the modelled species. In most demographic models, the knowing the exact days of the breeding season is not important. However, in a disease model, such as *OUTBREAK*, the breeding season results in a pulse of newly susceptible individuals entering the population. Consequently, the timing of reproduction can influence the dynamics of infectious disease.

## Life Table rates

This section determines the dynamics of the population based on annual mortalities and birth rates. The last row in this table reports the (approximate) deterministic calculation of growth rate ($\lambda = N_{t+1}/N_t$) based on the rates entered. These calculations may be only approximate because, when functions are used to define mortality or birth rates, it may not be possible to know in advance the value(s) returned by the functions.

For example, there is no guarantee that *OUTBREAK* will use the mean values returned by functions as it may use any possible value that that function may take. Similarly, when density-dependent functions are used, *OUTBREAK* will calculate the results based on the initial values or fixed values for the parameters used, and these values may be different to those actually applied when the simulations are run.

### Annual Mortality

This section specifies age-specific mortality rates for each epidemiological compartment of the population. These are S for Susceptible, E for Exposed, I for Infectious, R/V for Resistant or Vaccinated. The mortality rates provided for S individuals are also applied to Pre-Susceptible individuals, although this can be changed if the mortality is set to be a function of age.

| | |
|---|---|
| *Enter Juvenile and Subadult mortalities as stage-based* | By ticking this box the mortality rates are interpreted as the probability of an individual dying before reaching the next age class. Otherwise mortality rates are expressed as annual rates (i.e. the probability of dying before reaching the next year). Adult age class mortality rates are always expressed as annual rates. |

**Generally, these mortality rates should be the natural mortality rates in the absence of the disease.**

*If changing the status from S to E, I, R or V also alters the individuals' mortality (or reproduction rates), this can be modelled by modifying the relevant values in the successive columns of this table. Most commonly, mortality directly due to the disease is entered on the **I (infectious) tab**, and only secondary effects of the disease on the demography should be entered in the life table.*

*An example of such a secondary effect would be when recovered individuals still suffer from residual health effects that can make them less likely to breed or more susceptible to other causes of mortality. Different rates can be applied to R and V in the last column of this table by using a function to select the status, for example: =**IF(Z=4; x; y)**, where **Z** is the variable that codes for disease status (and Z = 4 identifies individuals in the V state), **x** is the rate to be applied if the individual's disease status is R, and **y** if the individual's disease status is V. For more information on using functions in OUTBREAK, see Chapter 6.*

### Annual Fecundity

These reproductive parameters determine the number of offspring produced each year. The final mean reproductive output (i.e. the number of offspring in a year) can be calculated with the following formula: *Prop. Breeding X adult population X # Broods/year X Brood size*.

*Prop. Breeding* — Enter the proportion of adult females in the population that produce offspring each year.

*# Broods/year* — Enter the mean number of broods produced per each female per year.

*Brood size* — Enter the mean number of offspring per brood. Different broods within the same year can be identified with the variable **BROOD** and different means can be applied. For example, using the function **=IF(BROOD=1;5;4)** would set a mean of five offspring in the first brood in that year and four in subsequent broods.

For convenience, click the *Copy rates from S to all disease states* button to copy all values in the S column into all the other columns.

## Carrying capacity (K)

Enter the maximum population size that can be sustained by the habitat. A number of factors, such as habitat size and resource availability (food, water, shelter) can limit K. In cases where K may vary over time, the in-built variable ***YEAR*** or ***DAY*** can be used to define functions that determine K in relation to the year since the beginning of the simulation or the day within the simulated year. When the number of individuals exceeds K, additional individuals are eliminated from the population. These individuals are randomly selected.

The manner in which *OUTBREAK* enforces K is specified by selecting one of the two following options.

- *Maintain K for each day of the year* — Surplus individuals are removed (killed) at the end of each day.
- *Apply K once per year on day* — Surplus individuals are removed (killed) once per year. When this option is selected, a box becomes active to enter the day of the year (specified using an integer) when this should occur. This may reflect a time of the year that is generally stressful for the population, such as when there are water or food shortages, or challenging weather conditions (e.g., extreme heat, floods etc.).

# Spatial Settings tab

If transmission rates are influenced by distances between individuals, several parameters need to be entered to inform the model of a) how individuals are initially distributed and b) how they then move through the landscape. This can be achieved using the **Spatial Settings tab**, shown below in Figure 14. This tab may already be active if the user has selected the option *A function of distance* as the way to model the encounter rate on the S (susceptible) tab.

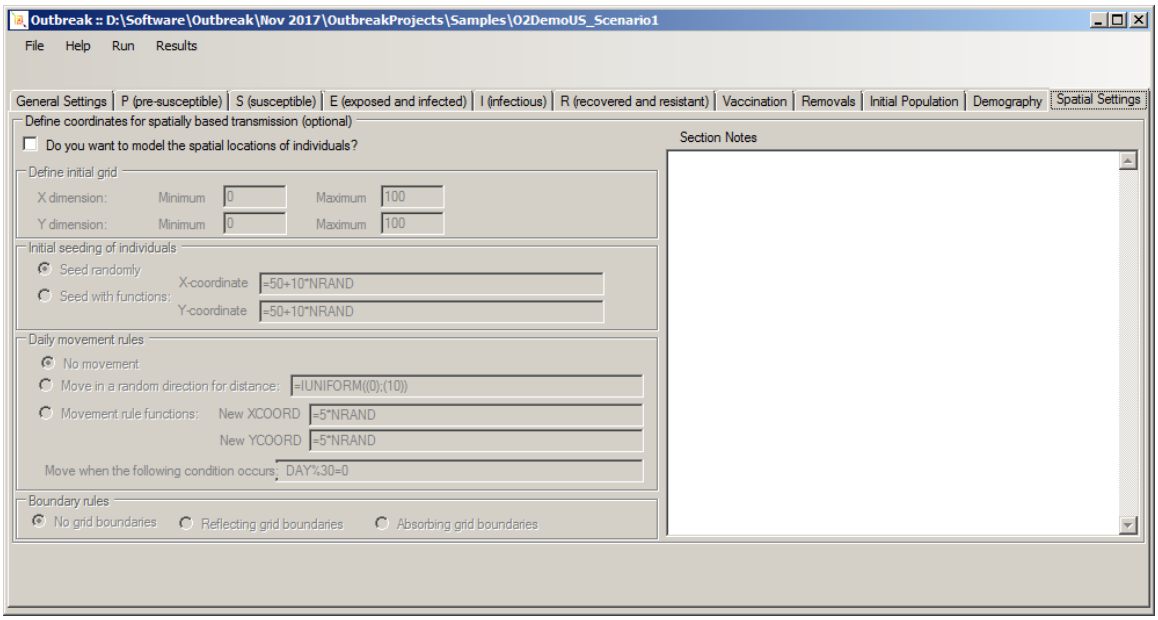



**Figure 14. *OUTBREAK* Spatial Settings tab**

## Define coordinates for spatially based transmission (optional)

*Do you want to model the spatial locations of individuals?* — Check this box to make this tab active (if not already).

## Define initial grid

This section defines the size of the grid (landscape) available to individuals. This can only be a square or rectangular shape. The *Minimum* and *Maximum* fields set the horizontal (X) and vertical (Y) size of the grid and are unitless. These serve only to provide a coordinate system that can be used to control individual distribution and movements. The *Minimum* values of X and Y define the lower, left corner of the grid.

## Initial seeding of individuals

This section provides two options for how individuals are to be distributed across the landscape at the beginning of the simulation. Note that newborn individuals inherit their locations from their mother.

- *Seed randomly*

  Select this option for individuals to be distributed randomly throughout the grid at the beginning of the simulation.

- *Seed with functions*

  Select this option for individuals to be distributed at the beginning of the simulation according to the values entered in the *X-coordinate* and *Y-coordinate* fields. It is possible, for example, to locate all the individuals in a cell by providing one number or to restrict the initial distribution of individuals by providing a range of values. For example, if **=IUNIFORM(1;25)** is used for both fields (*X-coordinate* and *Y-coordinate*), then the locations where the individuals are initially distributed would be limited to the lower, left corner of a 100x100 grid.

## Daily movement rules

This section provides several options to define if and how individuals move.

- *No movement.*

  Select this option if no movements occurs. This can be used when the distance between individuals is relevant for the epidemiology of the disease, but individuals have fixed locations.

- *Move in a random direction for distance*

  Enter the distance, as a number of cells, moved by each individual. When this option is used, the direction of movements is random.

- *Movement rule functions*

  Select this option to establish rules that govern how individuals move. The *New XCOORD* and *New YCOORD* fields are used to provide the new coordinate locations on the X and Y axes of the grid, respectively. For example, it is possible to link movements to an individual's disease status. If, for example, the following function is used: **=IF[(Z#2) OR (Z#3); XCOORD+5*NRAND; XCOORD]**, individuals will move to a new X-coordinate, which will be at a normally distributed distance from the original X-coordinate with mean 0 and standard deviation 5 but only if the individuals are neither E or I. If they are either E or I, they will remain in their original locations.

Another example might be when there is the intention to simulate movements within the home range.

If the users chooses *Move in a random direction for distance* or *Movement rule functions*, the following box becomes active:

| | |
|---|---|
| *Move when the following condition occurs* | Use this option to regulate when movements should occur. In the sample default value (***DAY%30=0***), for example, movements occur only every 30 days. |

## Boundary rules

The last section of this tab provides three options for specifying what should happen if individuals encounter the boundary of the grid.

- *No grid boundaries*

  When an individual encounters the original grid boundary, it continues on its trajectory, and the grid is increased in size to allow tracking its new location.

- *Reflecting grid boundaries*

  When individuals encounter a grid boundary, they will be reflected, and the movement will be completed according to the remaining number of steps.

- *Absorbing grid boundaries*

  For that day of the simulation, movements do not progress further than the grid boundary, and the individuals that encounter it will stop there. These individuals are still retained in the simulation and may move the next time that movements are allowed, depending on the direction of the new movement.

## Box 5. Limiting movements within home ranges

Often individuals movements are (mostly) limited to within their home range, and it may be important to include this ecological feature in the model.

This can be achieved by assigning each individual a defined home range center, and then allowing movements within a certain distance from the center. Such an approach requires creating an Individual State Variable (see Chapter 3.5 for information on using the **State Variable tab**) that will be the center of the home range (e.g. in a 100x100 grid: *HRXcenter=iuniform(0;100)* and *HRYcenter=iuniform(0;100)* for the X and Y axes of the grid respectively). The movement rule can then be set as:

*New XCOORD= HRXcenter + 5*NRAND*

*New YCOORD= HRYcenter + 5*NRAND*

In this example, the X and Y coordinates of the home rage will be randomly set for each individual, but subsequent movements will have a mean distance 0, with a standard deviation of 5 from the center of the individual's home range. In other words, 95% of the movements will be within 1.96x5=9.8 from the center of the home range along the X and Y axes of the grid. Further details and examples are provided in Chapter 7.

# 3.5 State Variables tab

## Introduction to State Variables

State variables can now be created within *OUTBREAK* on the **State Variables tab** (see Figure 15).

**Correct use of State Variables requires an advanced knowledge of *OUTBREAK* features and functions.**

*The use of State Variables can add considerable flexibility and power to the OUTBREAK model. However, their use may be difficult and sometimes unpredictable for the inexperienced user. Hence, only advanced users should use these in their models. One use of Individual and Population State Variables is to validate the use of functions. For example, if a function is entered to provide a parameter value, this can be set up as a State Variable. By doing this, the result of the function is stored in the output files, and the user can verify that the calculations return the expected values when the simulation is run.*

Any number of State Variables can be created, and they can be defined at a population or individual level. Population State Variables describe characteristics of the population, compute summary statistics (e.g., means), or keep a tally of individuals with specific traits. Some useful variables to use in Population State Variables are ***ITOTn***, ***IMEANn***, ***IMINn***, and ***IMAXn*** to compute the sum, mean, minimum, or maximum values across Individual State Variables (further details on these and additional functions are provided in Chapter 6). Individual State Variables most commonly describe individual attributes. For example, dominance status might be coded as Dominant = 1.0; Subdominant = 2.0; and Subordinate = 3.0. Or an Individual State Variable might be used to represent body condition.

Population and Individual State Variables can be referenced in functions by either the State Variable number (e.g., PS2, IS1) or their labels. It is important to ensure that the labels used are not reserved for any built-in functions or variables (such as ***DAMZ***, ***BROOD***). Examples of how to use State Variables are provided in scenario 3 ("Scenario3-withSVs") in the sample project ("O2Demo.xml") and in Chapter 7.

## Using the State Variables tab in *OUTBREAK*

State Variables must take numeric values or logic ones (i.e. TRUE and FALSE), which can be coded as a numeric values. The *Add* button allows creation of one or more State Variables. A variable can be removed by clicking on its row, and then clicking the *Delete* button. For each variable, a *Label* must be entered. The label can be any text but should not contain spaces or punctuation. Labels should be chosen to help the user remember what parameters the State Variables represent.

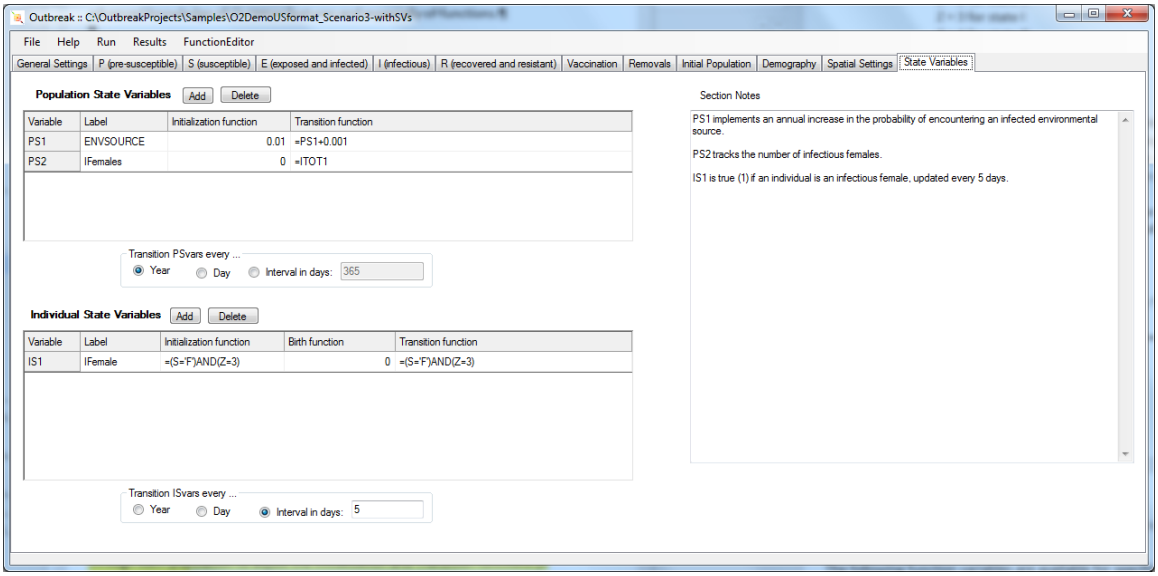



**Figure 15. *OUTBREAK* State Variable tab**

## Population State (PS) Variables

For each variable created, in addition to the *Label*, an *initialization* and a *transition function* must be entered. The first sets the variables' starting values at the beginning of the simulation, while the second determines the new values that the variables take when updated. Either functions or constants can be used in these fields.

*Transition PSvars every…* This section allows the user to control when Population State Variables are updated. This can be every year (by clicking on *Year*), every day (by clicking on *Day*) or with a set *Interval in days*.

## Individual State (IS) Variables

For each Individual State Variable, as for the Population State Variables, an initialization and transition function must be entered. The first sets the variable's starting value for each individual at the beginning of the simulation, while the second determines the new value that the variable takes when updated. Individual State Variables also have a *Birth function* that needs to be entered. This is used to set the variable's value for each newborn individual. Here too, functions or constants can be used.

*Transition ISvars every…* This section allows the user to control when Individual State Variables are updated. This can be every year (by clicking on *Year*), every day (by clicking on *Day*) or with a set *Interval in days*.

# Chapter 4. Running the simulation

Once the model is ready to be executed, the *Run* button at the top of the program window can be used to start the simulations. Clicking this button opens the **Run window** (Fig. 16), with options to:

- edit projection settings;
- select additional optional output files;
- graphically display the spatial locations of all individuals during the simulation; and
- run the simulation.

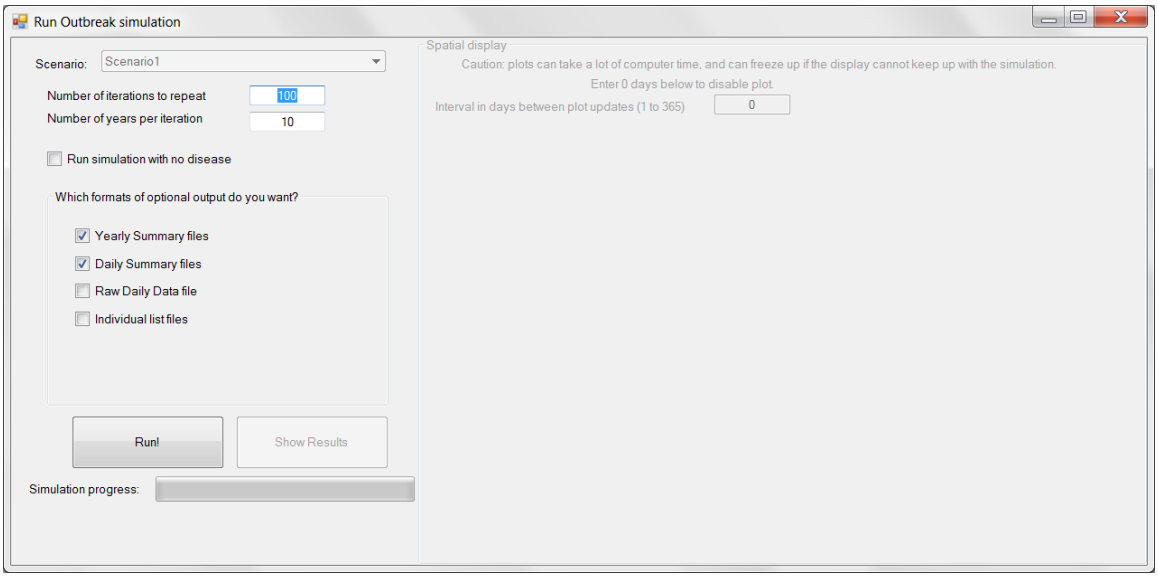



**Figure 16. *OUTBREAK* Run window, where the user can modify projection settings and start the simulation.**

## Projection settings

*Scenario* Use this drop-down menu to select the scenario to be simulated. If only one scenario exists, this drop-down menu is disabled.

The following two fields, *Number of iterations to repeat* and *Number of years per iterations,* are also available on the **General Settings tab** and are repeated here for convenience. Any modification made to these fields here on the **Run window** will automatically be reflected on the **General Settings tab** and vice versa. See Chapter 3.1 for a detailed explanation of these fields.

*Run simulation with no disease:* Select this box to disable all disease components of the model. This is often useful to test whether the life history parameters provided are realistic and result in a biologically plausible model. In this case, *OUTBREAK* will apply the parameter values provided in the S column on the **Demography tab**, under the assumption that these would model a population free from disease with no E or I individuals. This option is provided to carry out a check 'on the fly'.

**Note that the output files will not record that this simulation was run with the disease components disabled.**

*Also, as the output files are named using the scenario name, when the simulation is re-run (without the no disease option), the results will be overwritten. Therefore, if the intent is to generate a baseline model without disease, against which the effect of the disease can be compared, then it may be a better to create a dedicated scenario (named, for example, "NoDisease") where the disease component is explicitly removed (without E or I and with no environmental disease source).*

## Optional output files

The option to select additional output files is also available on the **General Settings tab** and is repeated here for convenience. If any additional output files are selected here on the **Run window**, this will be reflected automatically on the **General Settings tab** and vice versa**.**

*Which formats of optional output do you want?* Use this section to select additional outputs. Details of the data stored under each output option are provided in Chapter 5.2.

## Spatial display

The last section of this tab determines if and how often a graphical display of the spatial locations of all individuals is updated during the simulation. This section will only be active if the *Do you want to model the spatial locations of individuals?* box is checked on the **Spatial Settings tab**.

*Interval in days between plot updates* This field will only be active if the model has a spatial component. By changing this parameter to any number greater than zero, the user can instruct *OUTBREAK* to display the simulation graphically. The

number given here will determine the interval in days between updates of the graph. Some consideration should be given to the appropriate interval to use. Requesting updates to the graphs too frequently will slow down the simulations if the simulations can run faster than the computer's graphic card can update the screen. Even in those cases where the graphic card can cope with the speed of the simulation, it is often the case that the updates are so fast that it is not possible to appreciate changes. Conversely, too large an interval may compromise the visualization of spatial dynamics.

## Run the simulation

Once the relevant settings for the simulation are set, the simulation can be executed by clicking the *Run!* Button. Click the *Show Results* button once the simulation is complete to see a graphic display of the results. The next chapter explains in detail what graphs and output files are available to view simulation results.

# Chapter 5. Viewing results

## 5.1 Results window

Once the simulations are complete, the **Simulation Results window** (Fig. 17) can be accessed by clicking the *Show Results* button in the **Run window**. Alternatively, in a project that has results available, the same graphical display of results can be obtained by clicking on the *Results* button at the top of the main program window.

Four plots are displayed by default in **Simulation Results window.** These plots are explained in more detail on the following page.

- **End Counts** (top left); displays the proportion of individuals in each disease state at the end of the simulation.
- **Disease Prevalence** (top right); displays the disease prevalence in the population over the simulated time period.
- **Disease Dynamics** (bottom left); displays the mean number of individuals in each disease state over the simulated time period.
- **Demographic Dynamics** (bottom right); displays the mean number of individuals in each age class over the simulated time period.

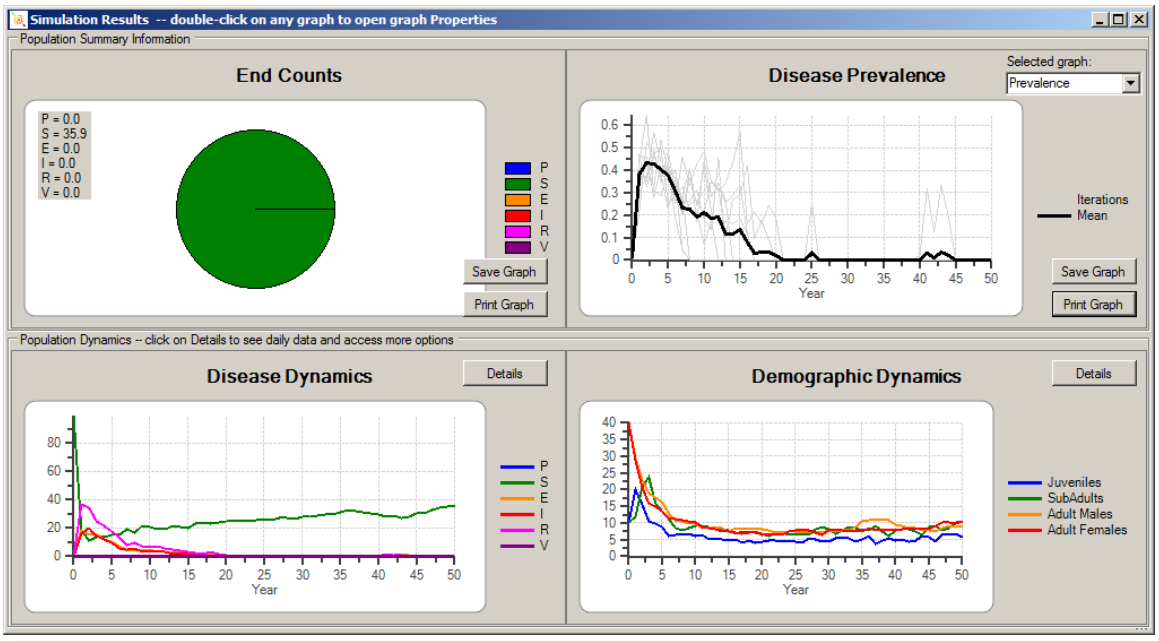
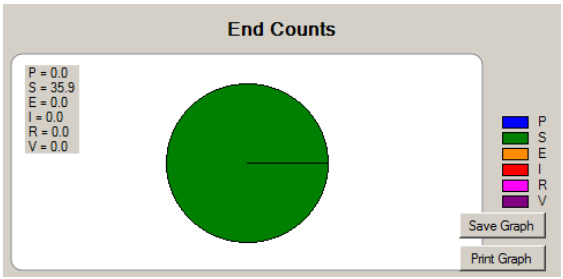



**Figure 17. *OUTBREAK* simulation Results window**

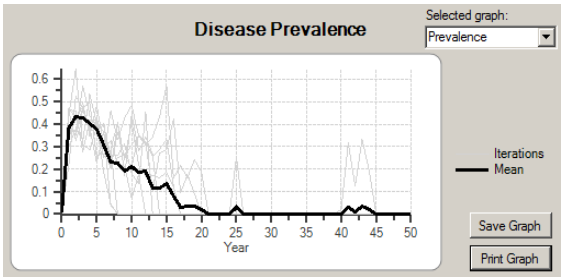
## End Counts

This pie chart depicts the proportion of individuals in each disease state at the end of the simulation. The disease states are color coded, and a legend is provided on the right side of the chart. The actual numbers of individuals in each disease state are shown to the left of the pie chart.




## Disease Prevalence

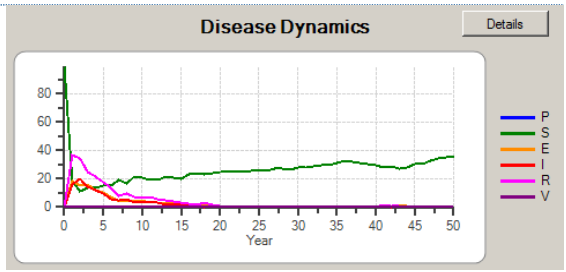
This line graph shows disease prevalence in the population over the simulated time period with time in years on the X-axis and (by default) prevalence as a proportion on the Y-axis. Individual iterations are the grey lines, and the mean prevalence across all iterations is represented by the black line. A drop-down menu is available to select variables other than prevalence. Possible options include: population size, probability of extinction, probability of terminal quasi-extinction, and probability of interval quasi-extinction. The difference between the last two terms is that terminal quasi-extinction means being below the threshold at the end of the simulation. Interval quasi-extinction means being below the threshold at some time during the simulation (not necessarily at the final time step). If Population State Variables were included in the model, these will also be available in the drop down menu.




**Did you know?** *In all graphs, it is possible to change the formatting of the plots by double-clicking on the graph area.*

## Disease Dynamics

This line graph shows (by default) the mean number of individuals in each disease state over the simulation period, across all iterations. Disease states are color coded, and a legend is provided on the right side of the plot. By clicking on the *Details* button, the **Disease Dynamics Details window** will open (Fig. 18).



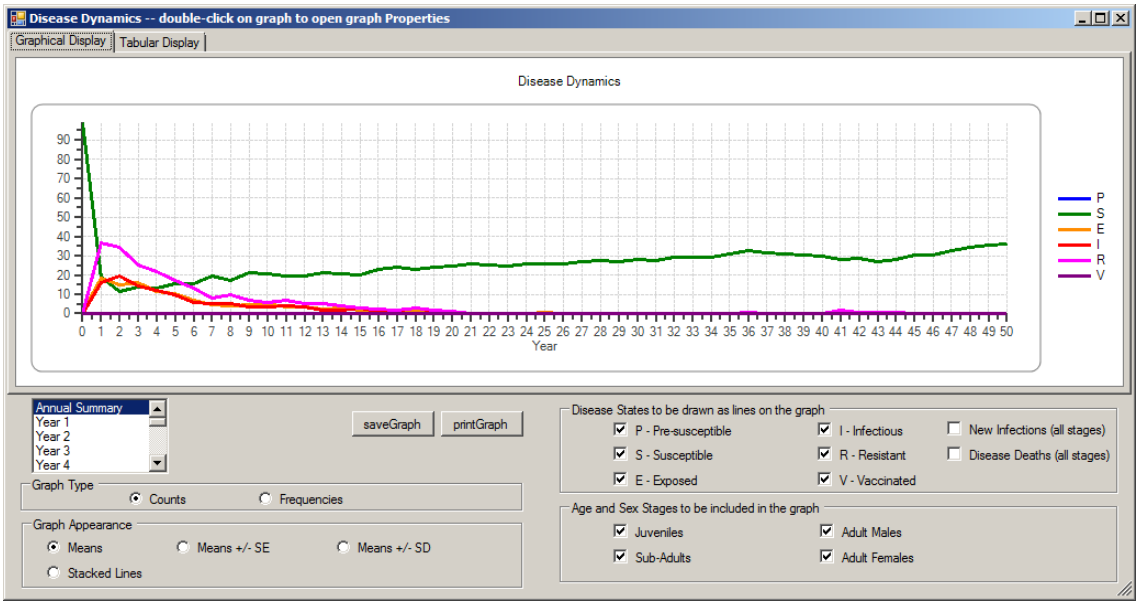


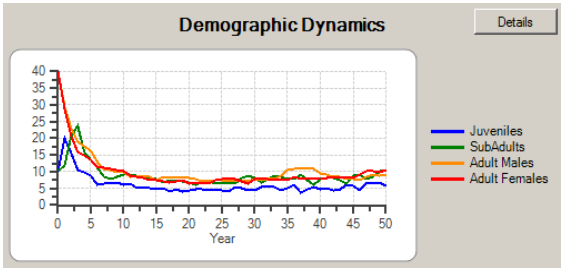

**Figure 18. *OUTBREAK* Disease Dynamics Details window – Graphical Display tab**

The **Disease Dynamics Details window** has two tabs: **Graphical Display** and **Tabular Display**. On the **Graphical Display tab** of this window, it is possible to modify the graph in the following ways:

- Use the drop-down menu below the left side of the graph to modify the temporal span of the data plotted (all years or one year at a time).
- In the **Graph Type** section, select whether the number of individuals in each disease state should be plotted as relative proportions instead of counts.
- Specify whether standard error or standard deviation should also be plotted or whether the plot should use stacked lines by selecting the relevant options in the **Graph Appearance section**.
- Select a different data series for inclusion in the graph in the **Disease States to be drawn as lines on the graph** section.
- Include data only for individuals of specified age class or sex by selecting the relevant options in the **Use the Age and Sex Stages to be included in the graph** section.

The second tab, **Tabular Display**, provides the data used to generate the plots. It is possible to export the data as a text file for use in other software, with the *Export Table* button.

## Demographic Dynamics




The change in demographics of the population over the simulated time period is displayed in this line graph. The mean number of individuals in each age class (Y-axis) is plotted over the time in years (X-axis). Age classes are color coded and a legend is provided on the right hand side. By clicking on the *Details* button, a **Demographic Dynamics Details window** will open (Fig. 19), where it is possible to select various options to modify the graph and to export the data as described for the **Disease Dynamics** plot on the previous page.

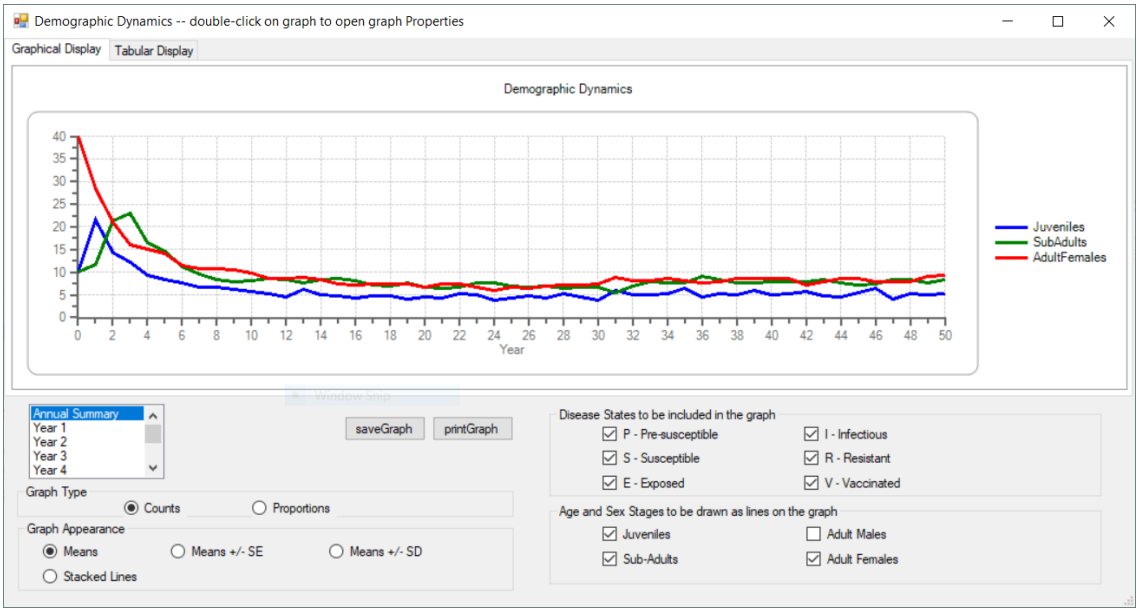



**Figure 19. *OUTBREAK* Disease Dynamics Details window – Graphical Display tab**

## 5.2 File output

*OUTBREAK* saves the results of simulations in a subfolder within the folder where the project file is located. The folder name has the suffix ***_Results*** appended to the project name. By default, *OUTBREAK* uses the semicolon (";") as a field separator for text files, for both data and header lines, which makes it easier to open these files in Excel or load them in other software (e.g., R) for further processing. For each scenario run, the following files are saved:

- ***Project input file (.xml)***

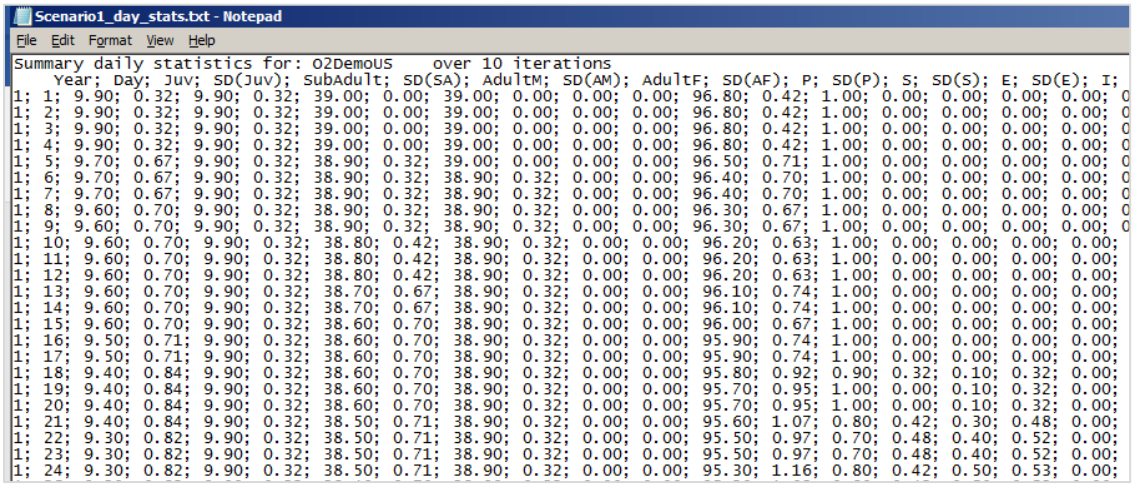
  This file can be loaded in *OUTBREAK* and is the file with the exact settings that were used to run the simulation. If the user has modified the project, but not saved the changes, these will not be recorded in the .xml file where the project is stored, while the .xml file in the Results folder will include all the settings that were present at the moment when the project was run, including any unsaved changes.

- ***Input parameter file (.inp)***

  This is a 'human-friendly' version of the input file. All parameters are saved in a legible format in a text file. The file includes all simulation settings, including details of the software version, project and scenario names, numbers of iterations and years, and all other disease and demographic parameters as well as Population and Individual State Variables present in the model.

- ***Daily statistic file (SCENARIO_NAME_day_stats.txt)***

  This file contains daily descriptive statistics (mean and standard deviation) across all iterations for the demographic (juvenile, sub-adult, adult, male, female, etc.) and disease (P, S, E, I, R, V) classes (Fig 20). Note that AM and AF are abbreviations for Adult Male and Adult Female; for example, AMS stands for Adult Male Susceptible. If Population State Variables were included in the model, their mean and standard deviation will also be reported in the last columns of this file.


Scenario1_day_stats.txt - Notepad

File Edit Format View Help

```
Summary daily statistics for: O2DemoUS    over 10 iterations
    Year; Day; Juv; SD(Juv); SubAdult; SD(SA); AdultM; SD(AM); AdultF; SD(AF); P; SD(P); S; SD(S); E; SD(E); I;
1; 1; 9.90; 0.32; 9.90; 0.32; 39.00; 0.00; 39.00; 0.00; 0.00; 0.00; 96.80; 0.42; 1.00; 0.00; 0.00; 0.00; 0.00; 0
1; 2; 9.90; 0.32; 9.90; 0.32; 39.00; 0.00; 39.00; 0.00; 0.00; 0.00; 96.80; 0.42; 1.00; 0.00; 0.00; 0.00; 0.00; 0
1; 3; 9.90; 0.32; 9.90; 0.32; 39.00; 0.00; 39.00; 0.00; 0.00; 0.00; 96.80; 0.42; 1.00; 0.00; 0.00; 0.00; 0.00; 0
1; 4; 9.90; 0.32; 9.90; 0.32; 39.00; 0.00; 39.00; 0.00; 0.00; 0.00; 96.80; 0.42; 1.00; 0.00; 0.00; 0.00; 0.00; 0
1; 5; 9.70; 0.67; 9.90; 0.32; 38.90; 0.32; 39.00; 0.00; 0.00; 0.00; 96.50; 0.71; 1.00; 0.00; 0.00; 0.00; 0.00; 0
1; 6; 9.70; 0.67; 9.90; 0.32; 38.90; 0.32; 38.90; 0.32; 0.00; 0.00; 96.40; 0.70; 1.00; 0.00; 0.00; 0.00; 0.00; 0
1; 7; 9.70; 0.67; 9.90; 0.32; 38.90; 0.32; 38.90; 0.32; 0.00; 0.00; 96.40; 0.70; 1.00; 0.00; 0.00; 0.00; 0.00; 0
1; 8; 9.60; 0.70; 9.90; 0.32; 38.90; 0.32; 38.90; 0.32; 0.00; 0.00; 96.30; 0.67; 1.00; 0.00; 0.00; 0.00; 0.00; 0
1; 9; 9.60; 0.70; 9.90; 0.32; 38.90; 0.32; 38.90; 0.32; 0.00; 0.00; 96.30; 0.67; 1.00; 0.00; 0.00; 0.00; 0.00; 0
1; 10; 9.60; 0.70; 9.90; 0.32; 38.80; 0.42; 38.90; 0.32; 0.00; 0.00; 96.20; 0.63; 1.00; 0.00; 0.00; 0.00; 0.00;
1; 11; 9.60; 0.70; 9.90; 0.32; 38.80; 0.42; 38.90; 0.32; 0.00; 0.00; 96.20; 0.63; 1.00; 0.00; 0.00; 0.00; 0.00;
1; 12; 9.60; 0.70; 9.90; 0.32; 38.80; 0.42; 38.90; 0.32; 0.00; 0.00; 96.20; 0.63; 1.00; 0.00; 0.00; 0.00; 0.00;
1; 13; 9.60; 0.70; 9.90; 0.32; 38.70; 0.67; 38.90; 0.32; 0.00; 0.00; 96.10; 0.74; 1.00; 0.00; 0.00; 0.00; 0.00;
1; 14; 9.60; 0.70; 9.90; 0.32; 38.70; 0.67; 38.90; 0.32; 0.00; 0.00; 96.10; 0.74; 1.00; 0.00; 0.00; 0.00; 0.00;
1; 15; 9.60; 0.70; 9.90; 0.32; 38.60; 0.70; 38.90; 0.32; 0.00; 0.00; 96.00; 0.67; 1.00; 0.00; 0.00; 0.00; 0.00;
1; 16; 9.50; 0.71; 9.90; 0.32; 38.60; 0.70; 38.90; 0.32; 0.00; 0.00; 95.90; 0.74; 1.00; 0.00; 0.00; 0.00; 0.00;
1; 17; 9.50; 0.71; 9.90; 0.32; 38.60; 0.70; 38.90; 0.32; 0.00; 0.00; 95.90; 0.74; 1.00; 0.00; 0.00; 0.00; 0.00;
1; 18; 9.40; 0.84; 9.90; 0.32; 38.60; 0.70; 38.90; 0.32; 0.00; 0.00; 95.80; 0.92; 0.90; 0.32; 0.10; 0.32; 0.00;
1; 19; 9.40; 0.84; 9.90; 0.32; 38.60; 0.70; 38.90; 0.32; 0.00; 0.00; 95.70; 0.95; 1.00; 0.00; 0.10; 0.32; 0.00;
1; 20; 9.40; 0.84; 9.90; 0.32; 38.60; 0.70; 38.90; 0.32; 0.00; 0.00; 95.70; 0.95; 1.00; 0.00; 0.10; 0.32; 0.00;
1; 21; 9.40; 0.84; 9.90; 0.32; 38.50; 0.71; 38.90; 0.32; 0.00; 0.00; 95.60; 1.07; 0.80; 0.42; 0.30; 0.48; 0.00;
1; 22; 9.30; 0.82; 9.90; 0.32; 38.50; 0.71; 38.90; 0.32; 0.00; 0.00; 95.50; 0.97; 0.70; 0.48; 0.40; 0.52; 0.00;
1; 23; 9.30; 0.82; 9.90; 0.32; 38.50; 0.71; 38.90; 0.32; 0.00; 0.00; 95.50; 0.97; 0.70; 0.48; 0.40; 0.52; 0.00;
1; 24; 9.30; 0.82; 9.90; 0.32; 38.50; 0.71; 38.90; 0.32; 0.00; 0.00; 95.30; 1.16; 0.80; 0.42; 0.50; 0.53; 0.00;
```

**Figure 20. Example of *OUTBREAK* daily statistics output file**

Also included in the daily statistics output file (as of version 2.9) is R0 – the basic reproduction rate – for the disease up through that day of the year. R0 is calculated from those individuals that have become I during the simulation and have exited that state (by way of becoming R, S, or dead). It is the number of mean number of infections caused by I individuals. Note that often R0 will decline as an epidemic proceeds, because the declining population size and declining number of S individuals means that there are fewer S that can be encountered by each I. Indeed, if an epidemic sweeps through the population and infects all individuals, then R0 will necessarily become approximately 1.0, because every animal became infected.

Note that R0 as calculated by *OUTBREAK* is not exactly the same as the standard definition, because R0 is usually defined as the mean number of secondary infections in a population where all other individuals are S. In *OUTBREAK*, R0 is the number of secondary infections that occur in a population that might have many individuals that are not S, especially in the later stages of an epidemic. The peak value for R0 reported by *OUTBREAK*, usually occurring in early stages of the disease outbreak, would more closely approximate the standard concept of the basic reproduction rate. However, keep in mind that R0 will change with population density (even if all other individuals are S). While this is not a concern for large populations that are stable in size, it makes the concept of R0 less useful (or at least less well-defined) for small wildlife populations that are subject to large fluctuations in size due to the disease and other factors.

- ***Yearly statistic file (SCENARIO_NAME_year_stats.txt)***

  This file is similar to the *Daily statistic file* described on the previous page but contains yearly descriptive statistics (mean and standard deviation) across all iterations for the demographic (juvenile, sub-adult, adult, male, female, etc.) and disease (P, S, E, I, R, V) classes. Note that AM and AF are abbreviations for Adult Male and Adult Female; for example, AMS stands for Adult Male Susceptible. The R0 reported in the *Yearly statistic* file is the cumulative number of secondary infections (see above) from Day1 through the last day (usually, Day 365) of that year (i.e., it is the same as what is reported for Day 365 in the *Daily statistic* file). If Population State Variables were included in the model, their mean and standard deviation will also be reported in the last columns of this file.

- ***Prevalence statistic file (SCENARIO_NAME_prev_stats.txt)***

  This file contains two blocks of data. The first is a two-column block with the mean prevalence (second column) across all iterations for each year (first column; Fig. 21a). The second is a three-column block with the prevalence (third column) for each year (second column) for each iteration (first column; Fig. 21b).

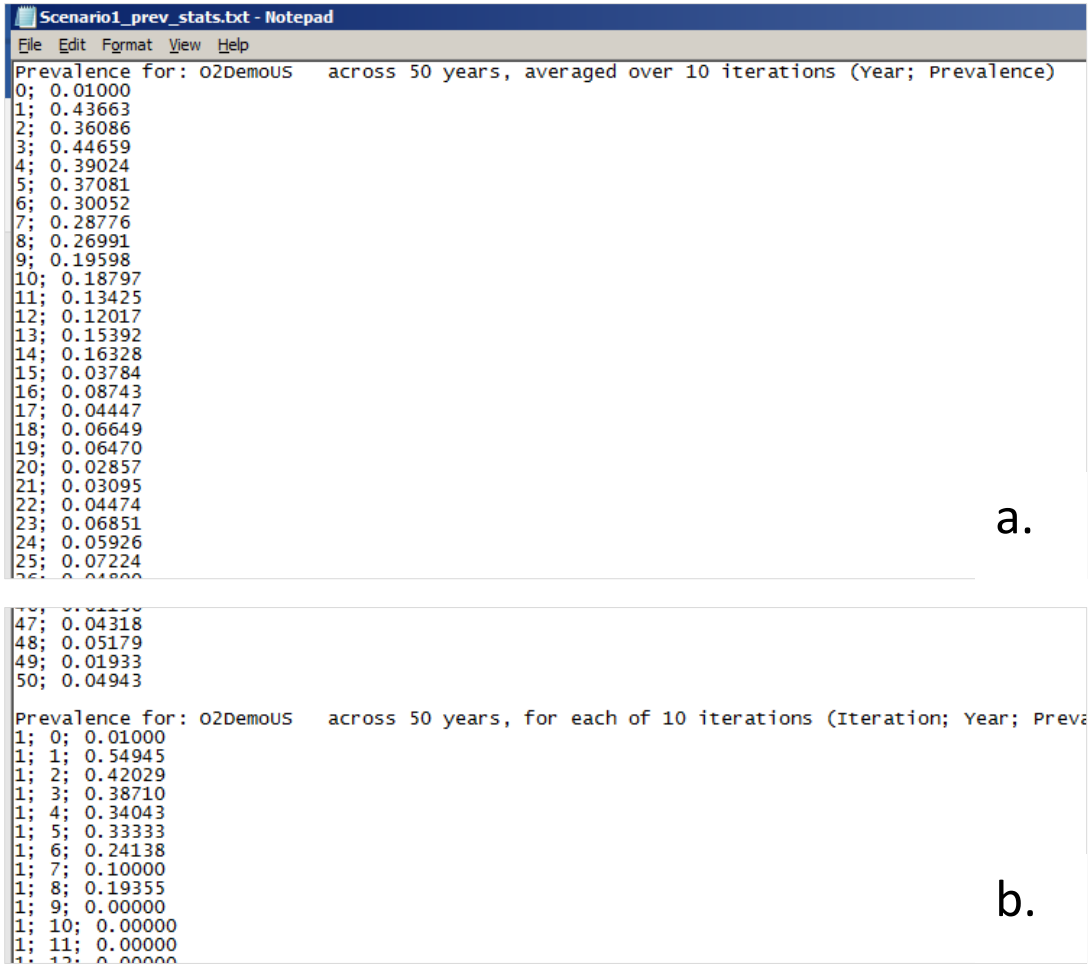



**Figure 21. Example of *OUTBREAK* prevalence statistic output file with (a) the mean prevalence across all iterations for each year shown in the first data block and (b) the prevalence for each year for each iteration displayed in the second data block.**

- ***Yearly population size file (SCENARIO_NAME_ryN.txt)***

  This file contains the population size (third column) in each year (second column), for each iteration (first column). If Population State Variables were included in the model, their values will also be reported in the last columns of this file. Example file contents are shown below in Figure 22.

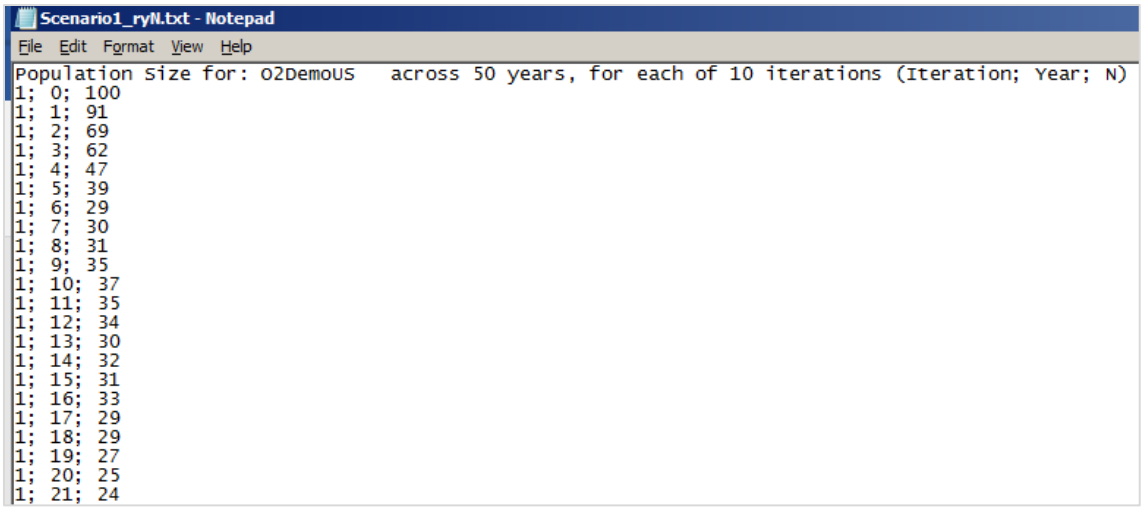



**Figure 22. Example of *OUTBREAK* yearly population size output file**

## Optional output

From the **Run window** or the **General Settings tab**, it is possible to specify that several other output files are generated. These include:

- ***Year summary file (SCENARIO_NAME_year_summary_y.txt)***

  For each year, a summary file is saved that contains a block of data for each iteration. The first line of each data block reports the disease prevalence for that year. The following lines are the counts for the demographic classes (juvenile, sub-adult, adult male and adult female), broken down into disease classes (P, S, E, I, R, V). See Figure 23.

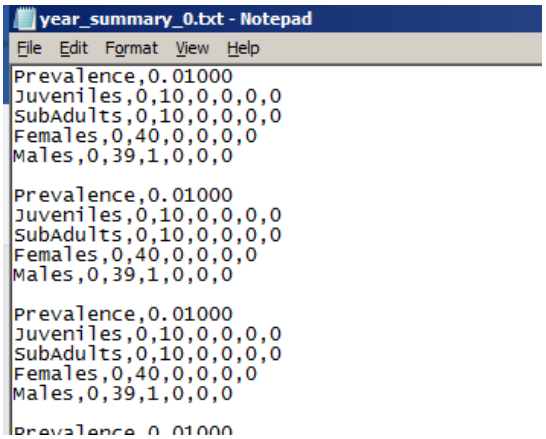



**Figure 23. Example of *OUTBREAK* year summary output file.**

- ***Year daily file (SCENARIO_NAME_year_daily_y.txt)***

  For each year, a summary file is generated that contains a block of data for each simulated day in each iteration. The first line of each data block reports the disease prevalence for that day in that year. The following lines are the counts for each demographic class (juvenile, sub-adult, adult male and adult female), broken down into disease classes (P, S, E, I, R, V).

- ***Raw daily data file (SCENARIO_NAME RawData.txt)***

  This file contains daily counts (for each year and each iteration) of the disease classes (P, S, E, I, R, V) as well as a tally of the new infections and the deaths due to disease (Fig. 24).

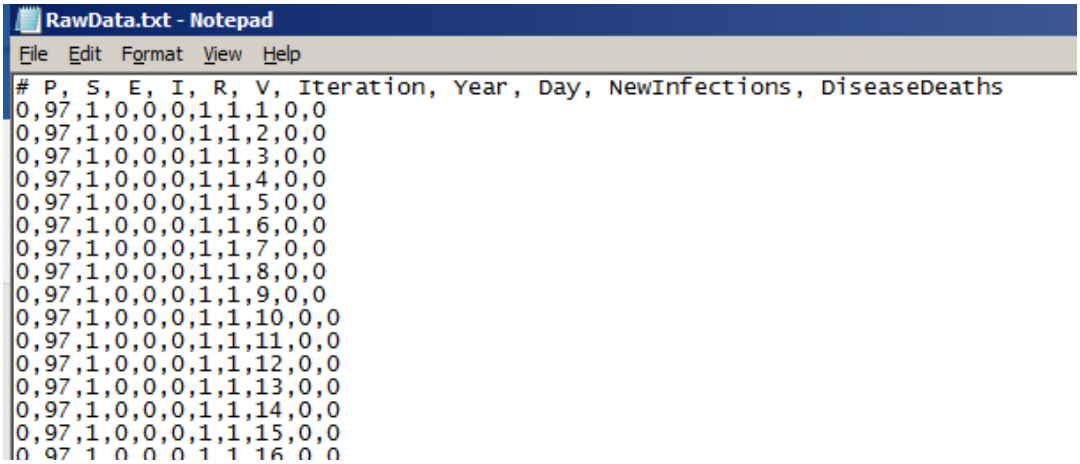



**Figure 24. Example of *OUTBREAK* raw daily data output file.**

- ***Individual list files (SCENARIO_NAME_indlist_iter_i_yr_Y.txt)***

  A file is created for each iteration and for each simulated year with a list of individuals and their trait values at the end of that year (Fig. 25). Note that a year 0 file is provided to show the starting population. The first line of each file reports the prevalence in that iteration and year. Subsequent lines contain an animal identifier, the age (in days), sex, disease state, whether the disease state was permanent, how many days the individual was in that disease state, whether it was vaccinated and whether it was alive. When a spatial component is present in the model, this file will also include two additional columns: XCOORD and YCOORD. If Individual State Variables were included in the model, their values will also be reported in the last columns of this file. When the simulation is run for many iterations and for many years, this option will create a large number of potentially large files. Therefore, usually these files are requested only for testing purposes (running just a few iterations for a few years) or when very detailed simulation results are needed for further analysis in other programs.

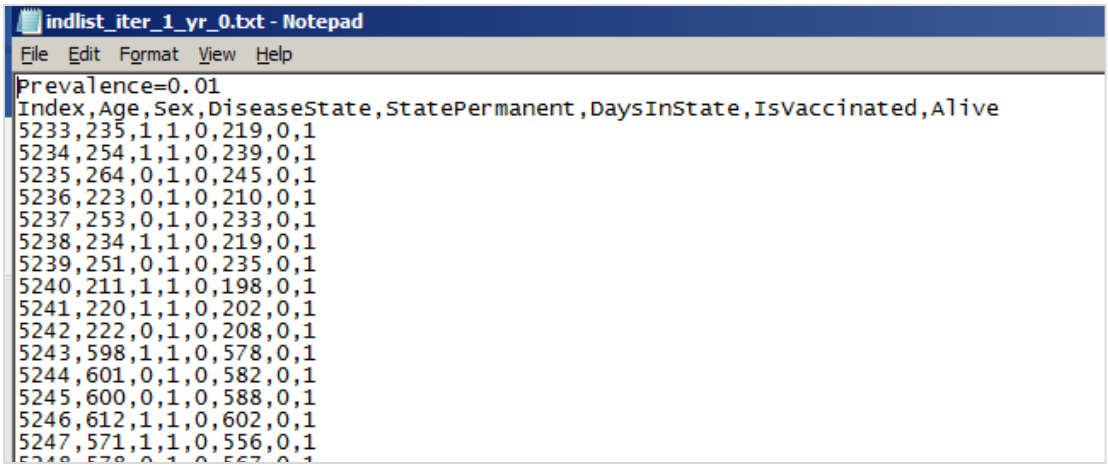



**Figure 25. Example of *OUTBREAK* individual list output file.**

# Chapter 6. Using functions in *OUTBREAK*

## 6.1 Specification of input parameters as functions

*OUTBREAK* provides the option to model most input values as functions of population or individual parameters. Generally, any input parameter that is specified as a rational number (such as a contact probability, transmission rate, duration of a state, or demographic rate) can be specified either as a number or as a function that evaluates to a number. The population descriptors that can be used as variables in the functions include year in the simulation, day in the year, iteration, population size, carrying capacity, numbers of juveniles (animals in the first age class), subadults, adult females, and adult males, and disease prevalence (the proportion in either E or I state). Individual characteristics that can be entered as variables in these functions include ID#, sex, age, disease state, days in state, vaccination status, and location on the landscape (for spatial models). Functions are evaluated each time that the input parameter is needed.

**Include the intended type of rounding within the function itself to control exactly how non-integer values will be treated.**

*Functions return rational numbers (positive or negative numbers that may include digits after a decimal delimiter), calculated to double-precision (8-byte numbers, with about 17 significant digits). If the required value must be an integer (e.g., number of days in a state, or a carrying capacity), Outbreak might truncate, round, or probabilistically round the value to obtain an integer (see descriptions of input rates, above), so it is best to include the intended type of rounding within the function itself so that the user controls exactly how non-integer values will be treated.*

### Why might I use functions?

1. The flexibility to specify input values as functions rather than as fixed constants allows users to model specific disease and population dynamics. With some creativity and perhaps considerable effort, *OUTBREAK* can now model many of the kinds of dynamics that can be envisioned. The following are just a few examples:

    - disease susceptibility might be a function of age;
    - disease incubation and duration might be sex-specific;
    - probability of environmental sources harboring disease might be a function of the prevalence of disease in the population;

- it might be known that carrying capacity will change in the future;
- movement patterns might be age and sex dependent.

2. Another important use of functions for input parameters is to specify that a parameter value is not precisely known and should instead be sampled each iteration from a distribution that represents the uncertainty regarding the parameter value.

## How do I enter a function?

To enter a function rather than a constant for an input variable, you can type the function directly into the input box. You should precede the function with an "=" sign (to distinguish the specification of a value as a function rather than as a constant).

**Did you know?** *The syntax of functions in OUTBREAK is similar to that used in Excel and is the same as used in VORTEX.* ***The easiest way to learn the syntax rules is to examine the examples provided in this chapter.***

*OUTBREAK* will assume that floating point numbers are formatted according to the Region settings on the computer. E.g., in the USA and Australia, 31/10 = 3.1, whereas in most of Europe, 31/10 = 3,1.

One important syntax rule is that the separator used between variables in a binary operator must be a semi-colon (;) – not a comma (,) – to avoid problems arising from the comma being used as a decimal separator with some regional data settings. Thus, the correct format is, e.g.,

MAX(A;B), *not* MAX(A,B)

POW(C;10), *not* POW(C,10)

Order of precedence of operators, within any parentheses, follows standard rules. Order of precedence of operators (evaluated first at the top), with left-to-right evaluation for sets of operators on the same line, is:

^ (exponentiation)
- (negation)
! (logical NOT)
*, /, % (multiplication, divide, and modulus [remainder from integer division])
+, -

<, >, <=, >=
=, != ("=", "==", and "EQUALS" are all equivalent; and "#" is the same as "!=")
AND, OR, NAND, NOR ("&" and "&&" and "AND" are all equivalent; "|" and "||" and "OR" are all equivalent; "NAND" can be coded as "$" and "NOR" can be coded as "~".)

(Parentheses), [brackets], and {braces} may be used interchangeably to indicate the order of operations. Although the choice of brackets or braces makes no difference to the program, it can be useful to alternate among them to help with the readability of functions. For example, the following are equivalent, but the second one might be easier to read.

SRAND((A+3)*C)+5^(SQRT((D+5)*LOG(ABS(F)))+M)

SRAND[(A+3)*C]+5^{SQRT[(D+5)*LOG(ABS[F])]+M}

The case of function names and variables is ignored. All letters that are entered in a function are converted to upper case by *OUTBREAK*, with the exception of filenames and R commands, which can be case-sensitive.

The logical values TRUE and FALSE, can be used interchangeably with 1 and 0 (i.e. TRUE = 1 and FALSE = 0).

'F' [in single quotes to signify a character, not the variable F] can be used for FEMALE and is coded as (and can be replaced by) 0.

'M' [in single quotes to signify a character, not the variable M] can be used for MALE and is coded as (and can be replaced by) 1.

'E' [in single quotes to signify a character, not the variable E] will be translated into the base of the natural logarithms.

### Hint for speed!

*Functions can be much slower to evaluate than are simple constant values. Therefore, do not enter* ***=1/4*** *(which must be evaluated as a function each time it is used), when you could have just entered* ***0.25*** *(which does not require a function evaluation).*

## 6.2 Function evaluation

The **Function Evaluator** in *OUTBREAK* provides a tool for editing, testing, and plotting functions (see Fig. 26). The **Function Evaluator** can be accessed by clicking *Function Evaluator* at the top to the program window.

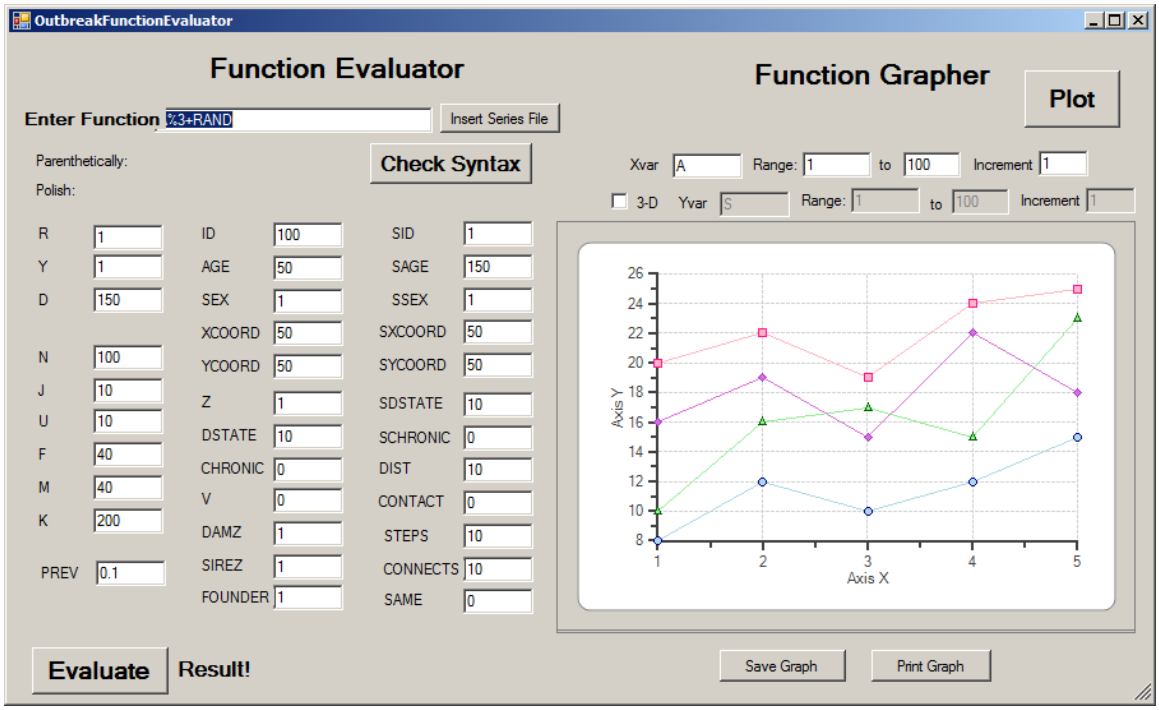



**Figure 26. *OUTBREAK* Function Evaluator**

If the cursor is on a data entry field that can accept a function when the **Function Evaluator** window is opened, the field content will be copied in the *Enter Function* field; otherwise, the **Function Evaluator** will start with a simple default function (A%3+RAND). In either case, a function can be typed or edited into this field. The *Insert Series File* button allows browsing to a file that it is needed when a list of values from a file is used (with ***FILE(filename)***). In the **Function Evaluator**, it is optional to precede the function with an '=' sign.

Click the *Check Syntax* button to test if the function is considered valid by *OUTBREAK* (i.e. all variables are recognizable, parentheses match, and required number of values are passed to built-in functions and operators). This check will also:

- display the function with extra parentheses added to make clear the order of operators; and
- give the function in Reverse Polish Notation to verify further that the function and its order of operators is being interpreted correctly (Note that Reverse Polish Notation is a notation for mathematical expressions in which operators

follow their operands and no parentheses are needed to control unambiguously the order of operations. For example, the expression "A*(B+C^2))" would be represented as "A B C 2 ^ + *" or, equivalently, as "C 2 ^ B + A*" in Reverse Polish Notation.)

Click the *Evaluate* button to evaluate the function with any specific set of variable values in order to see if it is returning the expected result. This button is disabled (greyed out) if the function syntax is invalid. A list of variables available for use in *OUTBREAK* is provided just below the *Check Syntax* button. The values in the fields next to the variables' labels are the values that are used to evaluate the function.

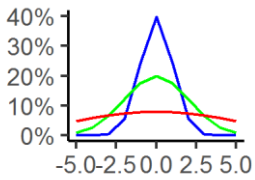
Click the *Plot* button to plot the values returned by a function in respect to variable values. This button is disabled if the function syntax is invalid. Plots can be printed or saved using the *Save Graph* and *Print Graph* buttons below the plot area. The name of the variable to be plotted on the x-axis, which will be copied automatically by *OUTBREAK* when a variable is used in the function, can be typed or edited in the *Xvar* field, along with its range and increment. If a second variable is used, it can be included in the plot by ticking the *3-D* box and providing the variable's label with its range and increment. If there are variables in the function for which a range of values is not provided (that is, they are not plotted on the horizontal or vertical axis), the **Function Evaluator** will use the values provided in the list just below the *Check Syntax* button.

To retain the function, click the *Accept* button. The function will be transferred into the current input data box, if it was a box that allows a function to be entered. Otherwise, click the *Cancel* button or the close window icon ('x') to discard the function and return to the main *OUTBREAK* input window.

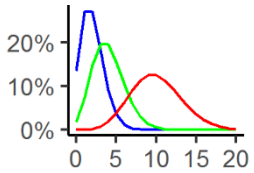
## 6.4 Density distributions

The section below shows the density distribution of common functions and how they change when their parameters are modified. These are provided as a reminder for the user. This is not an exhaustive representation of how to obtained distribution shapes nor on their parameterization.

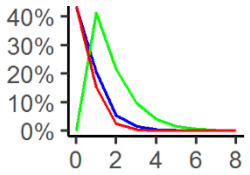
- Density of a normal distribution with mean 0 and standard deviation of 1 (blue), 2 (green) and 5 (red).



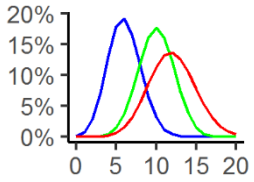

- Density of a Poisson distribution with lambda 2 (blue), 4 (green) and 10 (red). Increasing lambda shifts mode to the right.



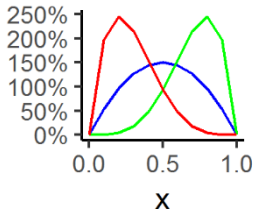

- Density of a gamma distribution with shape parameters 0.5; 1 (blue), 1.5; 1 (green) and 0.5; 1.5 (red). Increasing the first shape parameter concentrates the mode (if>1), increasing the second shape parameter flattens.




- Density of a binomial distribution with probability 0.3 and size 20 (blue), probability 0.5 and size 20 (green) and probability 0.3 and size 40 (red). Increasing probability or size shifts mode to the right.




- Density of a beta distribution with shape parameters 2; 2 (blue), 5; 2 (green) and 2; 5 (red). Increasing the first shape parameter shifts the mode to the right, increasing the second shape parameter shifts the mode left.



## 6.5 Variables and operators available for use in functions

The following tables list the many variables and operators that are available for use in *OUTBREAK* functions. Note that some variables have both 1-letter codes and synonymous short labels; either form can be used in the functions.

**It is important to be aware when the variables are updated as the simulation is running.**

*For many variables, it is obvious; for example, the variable **Y** (for year) is incremented at the beginning of each year of the simulation, and **DAY** is updated at the beginning of the simulation of each day. However, population tallies, such as **N**, **F**, **M**, **J**, and **PREV** will be updated dynamically within each day after any vaccinations, then again after any removals, then after disease state transitions, and finally after movements.*

# Scenario, population, and individual descriptors

### Valid Function Variables: Scenario descriptors

**R**, or **RUN** = Run (simulation iteration)
**Y**, or **YEAR** = Year
**D**, or **DAY** = Day of the year

### Valid Function Variables: Population descriptors

**F** = Number adult females
**J** = Number of juveniles (age 0-1)
**K** = Carrying capacity
**M** = Number of adult males in the population
**N** = Population size
**P**, or **PREV** = Prevalence
**U** = Number of subadults (age > 1, < breeding age)
**NP** = Number of P individuals
**NS** = Number of S individuals
**NE** = Number of E individuals
**NI** = Number of I individuals
**NR** = Number of R individuals
**NV** = Number of V individuals
**PS1, PS2**, etc. Value of 1st, 2nd, etc. Population State variable
**ITOTn** (e.g. **ITOT1**, **ITOT2**) = Total, across living individuals, of **n**th individual state variable.
**IMEANn** (e.g. **IMEAN1**, **IMEAN2**) = Mean across individuals of **n**th individual state variable.
**IMINn** (e.g. **IMIN1**, **IMIN2**) = Minimum, across individuals, of **n**th individual state variable.
**IMAXn** (e.g. **IMAX1**, **IMAX2**) = Maximum, across individuals, of **n**th individual state variable.

## Valid Function Variables: Individual descriptors

| | |
|---|---|
| **A**, or **AGE** | Age |
| **S**, or **SEX** | Sex (0 or 'F' for female, 1 or 'M' for male) |
| **ID** | numerical code for an individual, set sequentially as individuals are created. |
| **Z** | Disease State<br>Z = 0 for state P<br>Z = 1 for state S<br>Z = 2 for state E<br>Z = 3 for state I<br>Z = 4 for state R<br>Z = 5 for state V |
| **DSTATE** | Days in state |
| **C**, or **CHRONIC** | Chronic (permanent) state (1 or 0) |
| **V** | Vaccinated (1 or 0) |
| **IS1, IS2**, etc. | Value of 1st, 2nd, etc. Individual State variable |
| **DAMZ** | Dam disease state (at birth of offspring) |
| **SIREZ** | Sire disease state (at birth of offspring) |
| **BROOD** | During breeding, which brood of the year for that female is being produced (this can be useful if brood sizes or survival rates vary from 1st to 2nd to later broods) |
| **FOUNDER** | Individual is a founder present at the start of the simulation (1 or 0), in which case it won't have values for DAMZ or SIREZ (both will be set to -1) |

## Variables for use with spatial modeling

| | |
|---|---|
| **X**, or **XCOORD** | X-coordinate of the individual |
| **W**, or **YCOORD** | Y-coordinate of the individual (note: Y is used for Year, not for Y-coord) |

The following function variables are available for specifying the characteristics of an Infectious (I) individual and a Susceptible (S) individual when they encounter each other. This can be useful if, for example, the probability of encounter or disease transmission is influenced by the age, sex, or other characteristics.

## Individual descriptors of a Susceptible individual in an encounter

| | |
|---|---|
| **SAGE** | Age |
| **SSEX** | Sex (0 or 'F' for female, 1 or 'M' for male) |
| **SID** | numerical code for an individual, set sequentially as individuals are created. |
| **SDSTATE** | Days in state |
| **SCHRONIC** | Chronic (permanent) state (1 or 0) |
| **SXCOORD** | X-coordinate of the individual |
| **SYCOORD** | Y-coordinate of the individual (note: Y is used for Year, not for Y-coord) |
| **SIS1, SIS2**, etc. | Value of 1st, 2nd, etc. Individual State variable |

## Individual descriptors of an Infected individual in an encounter

| | |
|---|---|
| **IAGE** | Age |
| **ISEX** | Sex (0 or 'F' for female, 1 or 'M' for male) |
| **IID** | numerical code for an individual, set sequentially as individuals are created. |
| **IDSTATE** | Days in state |
| **ICHRONIC** | Chronic (permanent) state (1 or 0) |
| **IXCOORD** | X-coordinate of the individual |
| **IYCOORD** | Y-coordinate of the individual (note: Y is used for Year, not for Y-coord) |
| **IIS1, IIS2**, etc. | Value of 1st, 2nd, etc. Individual State variable |

Note that when considering an encounter between an I individual and an S individual, the standard variables for describing an individual (AGE, SEX, etc.) describe the I individual. Thus, IAGE = AGE, ISEX = SEX, IIS1 = IS1, etc.

## Variables for describing the spatial relationship between individuals

| | |
|---|---|
| **DIST** | Euclidean distance between two individuals |
| **SAME** | T or F: in same X,Y location (grid cell) |
| **CONTACT** | T or F: in contacting, but not the same, grid cell |
| **STEPS** | minimum steps in X or Y direction to traverse to reach other individual |
| **CONNECTS** | minimum number of connecting cells to traverse to reach other individual |

# Operators for use in functions

## Valid Arithmetic Operators for use in Functions

| Operator | Description | Example |
|---|---|---|
| ABS | Absolute value | ABS(-10) = 10 |
| NEG | Negative | NEG(-10) = 10 |
| CEIL | Ceiling | CEIL(3.12) = 4 |
| FLOOR | Truncate | FLOOR(3.12) = 3 |
| ROUND | Round | ROUND(3.12) = 3; ROUND(5.5) = 6 |
| PROUND | Probabilistic round | PROUND(3.12) = 3 (88% of the time) or 4 (12% of the time) |
| SQRT, SQR | Square root | SQR(1.44) = 1.2 |
| LN, LOG | Natural logarithm | LN(1.60) = 0.47 |
| LOG10 | Base 10 logarithm | LOG10(1.60) = 0.20412 |
| EXP | e raised to the power | EXP(0.47) = 1.60 |
| + | Addition | 1.0+2.0 = 3.0 |
| - | Subtraction | 2.0–1.0 = 1.0 |
| * | Multiplication | 2.0*3.0 = 6.0 |
| / | Division | 6.0/2.0 = 3.0 |
| POW, ^ | Exponentiation | POW(10;0.20412) = 10^0.20412 = 1.60 |
| MAX | Maximum | MAX(3.12;4.21) = 4.21 |
| MIN | Minimum | MIN(3.12;4.21) = 3.12 |
| MOD, % | Modulus | MOD(33;8) = 33%8 = 1 |

## Defined Constants

| | | |
|---|---|---|
| TRUE | 1.0 | (10>5) = TRUE |
| FALSE | 0.0 | !TRUE = FALSE |
| PI | 3.1415927 … | SIN(PI/4) = 0.7071067 |
| 'E' | 2.7182818 … | LN('E') = 1.0 |

| Operator | Description | Example |
|---|---|---|
| **Trigonometric Operators** | | |
| SIN | Sine | SIN(PI/2) = 1.0 |
| COS | Cosine | COS(PI/2) = 0.0 |
| TAN | Tangent | TAN(PI/4) = 1.0 |
| ASIN | Arcsine | ASIN(1.0) = 1.5707963 |
| ACOS | Arccosine | ACOS(0.0) = 1.5707963 |
| ATAN | Arctangent | ATAN(1.0) = 0.7853981 |
| DEGREES | Convert Radians to | DEGREES(PI/4) = 45 |
| RADIANS | Convert Degrees to Radians | RADIANS(45) = PI/4 = 0.7854 |
| **Logical (Boolean) Operators** | | |
| ==, = | Is equal to | (3=2) = 0 = FALSE |
| NOT, ! | Negation | !(3=4) = 1 = TRUE |
| !=, # | Not equal to | (3!=4) = 1 |
| AND, && | And | ((3=4)AND(3!=4)) = 0 |
| OR, \|\| | Or | ((3=4)OR(3!=4)) = 1 |
| > | Greater than | (3>4) = 0 |
| < | Less than | (3<4) = 1 |
| >= | Greater than or | (3>=3) = 1 |
| <= | Less than or equal to | (3<=3) = 1 |
| COMPARE | Compare two values | COMPARE(A;B) = 1 (if A>B), or -1 (if A<B), or 0 (if A=B) |
| IF | Conditional evaluation | IF(A>B;5;10) = 5 (if A>B) or 10 (if A<=B) |

| Operator | Description | Example |
|---|---|---|
| **Random Number Generators** | | |
| **RAND** | Uniform random (0 – 1) | **RAND = 0.2341** or **0.8714** or … |
| **NRAND** | Normal random deviate | **NRAND = 0.512** or **–0.716** or … |
| **UNIFORM** | Uniform random (A – B) | **UNIFORM(1;5) = 1.5** or **.3.6** or … |
| **IUNIFORM** | Random integer (A – B) | **IUNIFORM(1;5) = 1.0** or **2.0** or … |
| **POISSON** | Poisson, with mean m | **POISSON(3) = 0.0** or **1.0** or **2.0** or |
| **POISSON1** | 0-truncated Poisson; Poisson without 0 class | **POISSON1(3) = 1.0** or **2.0** or … |
| **SRAND** | A "seeded" random number; hence, **SRAND(x)** provides a random number between 0 and 1 with the seed value **x** | |
| **SNRAND** | A "seeded" random normal deviate; hence, **SNRAND(x)** returns a number from a (0,1) normal distribution with the seed value **x** | |
| **SUNIFORM** | A "seeded" random uniform; hence, **SUNIFORM(a;b;x)** provides a random number between a and b (inclusive) with the seed value **x** | |
| **SIUNIFORM or SIRAND** | A "seeded" uniform random integer; hence, **SIUNIFORM(a;b;x)** provides a random number between a and b (inclusive) with seed value **x** | |
| **SPOISSON** | A "seeded" random number generator; **SPOISSON(m;x)** provides a random number from a Poisson distribution with mean **m** with seed value **x** | |
| **SPOISSON1** | A "seeded" 0-truncated Poisson; **SPOISSON1(m;x)** provides a random number greater than 0 from a Poisson with mean **m** with seed value **x** | |
| **BINOMIALN SBINOMIALN SBETA** | The number of successes sampled from a binomial distribution with parameters mean **m** and sample size **n**.; S**BINOMIALN(m;n)** uses seed value **x**. **SBETA(a;b;x)** uses seed value **x** | |
| **BETA, SBETA** | A beta distributed random number; **BETA(a;b)** samples from a beta with shape parameters **a** and **b**. **SBETA(a;b;x)** uses seed value **x** | |
| **BETAM, SBETAM** | A beta distributed random number; **BETAM(m;s)** samples from a beta with mean **m** and SD **s**. **SBETAM(m;s;x)** uses seed value **x** | |
| **GAMMA, SGAMMA** | A gamma distributed random number; **GAMMA(a;b)** samples a random number from a gamma with shape parameters **a** and **b**. **SGAMMA(a;b;x)** uses seed value **x** | |
| **GAMMAM, SGAMMAM** | A gamma distributed random number; **GAMMAM(m;s)** samples a random number from a gamma with mean **m** and SD **s**. **SGAMMAM(m;s;x)** uses seed value **x** | |
| **DECAY, SDECAY** | An integer sampled from the distribution representing exponential decay at a constant rate; **DECAY(p)** samples a number of timesteps until transition in a process with probability **p** each timestep. **SDECAY(p;x)** uses seed value **x**. If **p**<= 0, then the function returns 0 (not infinity). | |

## Operators for creating and modifying lists

The operators below, used to create, modify, and operate on lists will not be used by most *OUTBREAK* users, but they can be powerful ways to create specific sequences of values. These can then be used to set different input values for each age class or at each year or in each population.

| Function | Description | Example |
|---|---|---|
| **Operations on Lists** | | |
| **LOW** | Lowest value | **LOW(3;5;7;1;8) = 1.0** |
| **HIGH** | Highest value | **HIGH(3;5;7;1;8) = 8.0** |
| **MEDIAN** | Median value | **MEDIAN(3;5;7;1;8) = 5.0** |
| **MEAN** | Mean value | **MEAN(3;5;7;1;8) = 4.8** |
| **SUM** | Sum | **SUM(3;5;7;1;8) = 24.0** |
| **PRODUCT** | Product | **PRODUCT(3;5;7;1;8) = 960.0** |
| **LOOKUP** | Value at the **x** location in the subsequent list | **LOOKUP(3;5;7;1;8) = 1.0** |
| **CHOOSE** | Randomly chosen value from the list | **CHOOSE(3;5;7;1;8) = 3** or **5** or **7** or **1** or **8** |
| **SCHOOSE** | Seeded randomly chosen value from the list | **SCHOOSE(x;3;5;7;1;8) = 3** or **5** or **7** or **1** or **8**, with seed **x** |
| **COUNT** | Number of values | **COUNT(3;5;7;1;8) = 5.0** |
| **FIND** | Location of value **x** in the subsequent list (0 if | **FIND(1;5;7;1;8) = 3.0** |
| **PERCENTILE** | Value in the subsequent list that is at the **x** | **PERCENTILE(40;5;7;1;8) = 7.0** |
| **PRANK** | Percentile of value **x** in the subsequent list | **PRANK(3;5;7;1;8) = 5.0** |
| **TIMESERIES** | List item at the location given by current year, **Y** | **TIMESERIES(3;5;7;1;8) = 5.0** in year 2 |

Note that the following operations to create or modify lists cannot stand alone in a function, because they do not return a single value. Instead, they must be used within one of the functions listed on the previous page that operates on a list. For example, a function of "=SORT(1;3;5)" is not valid, but "=FIND(3;SORT(1;3;5;4;2))" is valid and returns the answer 3.

| Function | Description | Example |
|---|---|---|
| FILE | Creates a list from the values in a text file | FILE("C:/Temp/MyFile.txt") |
| FILECOL | Creates a list from the values in a column or row of a file | FILECOL(C:/Temp/MyFile.txt;2) |
| FILEROW | | FILEROW(C:/Temp/MyFile.txt;3) |
| FILECELL | Gets the value at row **r** and column **c** in a text file | FILECELL(MyFile.txt;2;3) |
| SEQUENCE | A list with values from **a** to **b**, by increments of **c** | SEQUENCE(1;6;2) = (1;3;5) |
| SORT | Sorts from low to high | SORT(1;7;3;5;8) = (1;3;5;7;8) |
| SORTREV | Sorts from high to low | SORTREV(1;7;3;5;8) = (8;7;5;3;1) |
| REVERSE | Reverses the order | REVERSE(1;7;3;5;8) = (8;5;3;7;1) |
| SUBSET | Creates a list from **c**; **d**; …, starting with the **a** position and ending at **b** position | SUBSET(2;4;6;7;5;8;9) = (7;5;8) |
| SAMPLE | Creates a list of **x** items randomly chosen without replacement. | SAMPLE(3;1;7;3;5;8) = (1;5;7) or (8;1;3) or (3;1;5) or … |
| SSAMPLE | Creates a list of **x** items chosen with seed **s** without replacement | SSAMPLE(123;3;1;7;3;5;8) = (1;5;7) or … with the same list returned every time. |
| APPEND | Appends **x** to the list | APPEND(2;1;7;3;5) = (1;7;3;5;2) |
| PREPEND | Inserts **x** at the beginning | PREPEND(2;1;7;3;5) = (2;1;7;3;5) |
| INSERT | Inserts **x** at the **n**th position | INSERT(6;2;1;7;3;5) = (1;6;7;3;5) |
| REMOVE | Removes first item equal to **x** | REMOVE(3;7;3;5;3;8) = (7;5;3;8) |
| REMOVEALL | Removes all items equal to **x** | REMOVEALL(3;7;3;5;3;8) = (7;5;8) |
| REMOVEAT | Removes the **n**th item | REMOVEAT(3;7;3;5;8) = (7;3;8) |

**FILE(filename)** creates a list from the values in a text file. E.g., **LOOKUP(3;FILE(myfile.txt))** will look up the third value in the list provided in myfile.txt. This can be useful if you want to specify a series of values in a file that will be read into *OUTBREAK*. The filename should contain the complete path. The filename can be in quotes or not, but it is safer to use the quotes, because then any punctuation in the filename will not be mis-interpreted as a delimiter between parameters. The filename cannot contain parentheses.

For the next three operators, the rows and columns must be specified as constant numeric values, and cannot be variables (such as **A** or **N**), because the file is read once at the beginning of the scenario simulation to obtain the row, column, or cell values.

**FILECOL(filename;n)** creates a list from the values in the nth column of a text file (with columns of each row separated by semi-colons, tabs, or spaces, but not commas). E.g., **LOOKUP(3;FILECOL(myfile.txt;4))** will look up the third value in the 4th column of data in myfile.txt.

**FILEROW(filename;n)** creates a list from the values in the nth row of a text file (with values separated by semi-colons, tabs, or spaces, but not commas). E.g., **LOOKUP(3;FILEROW(myfile.txt;4))** will look up the third value in the 4th row of data.

**FILECELL(filename;r;c)** reads the value in column c and row r in a text file. E.g., **FILECELL(myfile.txt;4;3)** gives the same value as **LOOKUP(3;FILEROW(myfile.txt;4))** and **LOOKUP(4;FILECOL(myfile.txt;3))** It is also possible to read a cell value from a matrix by using the **LOOKUP** function. For example, **LOOKUP((r-1)*n+c;FILE(myfile.txt))** will fetch the value at row r, column c, from a matrix with n columns. Using this format allows the use of functions for the r, c, and n variables.

## Using random number generators in functions

Random number generators can be used to specify that rates or durations vary over a range of values, or are uncertain. However, the proper use of these functions requires careful consideration of how the "seed" values (implicit, as in **RAND** and **NRAND**, or explicit, as in **SRAND** and **SNRAND**) determine when new random numbers are selected. Repeated calls to the random number return the same value if the same seed is specified. Random numbers produced with different, even sequential, seeds will not be correlated. The "unseeded" forms (**RAND** and **NRAND**) set their own unique (or nearly so) seed each time they are called.

**Did you know?** *The very first use of a random number generator in OUTBREAK uses a seed based on the number of seconds elapsed since the turn of the century.*

Each call to an unseeded random number generator also sets a new seed for the next call for an unseeded random number. Thus, identically configured computers starting the same simulation at exactly the same second on their clocks would produce identical results for an analysis. This synchrony may require, however, that all memory storage locations (including hard disk caches) and even the hard disk contents are identical on the systems (because they will affect the time required for each read or write to the disk). Thus, it really is quite random.

The specification of random number seeds allows synchronization of sequences of random numbers. This can be used to create synchrony of events, such as environmental variation affecting multiple input parameters or autocorrelations among years (time lags or cycles). If several different demographic rates are specified by functions containing random number generators, care must be taken to create the desired synchrony or lack of synchrony. If two functions contain the same seed values, they will return the same random number. Seed values must be distinct to create independence of random numbers. Proper use of random number seeds can be difficult. Think carefully about the effect of any seed that you use in a function, to be certain that it will produce the same random numbers when you want them, and independent random numbers otherwise. Any variable (e.g., **A** for age, **Y** for year, **R** for run) included within the seed will cause the same "random" number to be chosen for each case with the same value for those variables (**A**, **Y**, **R**). For example, if you specify **SRAND(Y)** within a function, then each year will get an independent random number, and that set of random numbers will be the same over all calls to evaluate that function (such as for every day, every run, and every individual). If you specify **SRAND((R*100)+Y)**, then each run will get a new independent random number each year of the simulation. See the examples below for further information about random number seeds.

The seeds used by *OUTBREAK* will be converted to integers between 0 and 65536. Non--integer seeds will be truncated [hence, **SRAND (35.23) = SRAND(35.89)**] and values above 64K will be "wrapped" [the modulus taken, so that **SRAND(65636)** = **SRAND(100)**].

## Some further notes regarding function syntax and use

- Variables of trigonometric functions are assumed to be in radians, except for the function **RADIANS**, which takes radians and returns a value in degrees. The function DEGREE converts angles in radians, to their equivalent in degrees.

- The operator **NEG** is the same as using a minus (-) sign before a number. By the context, *OUTBREAK* will interpret whether a minus sign signifies subtraction (a binary operation) or the negative (a unary operation).

- **CEIL**, **FLOOR**, and **ROUND** convert rational numbers to integer values, but all expressions are evaluated as rational numbers. For example, `FLOOR(3.7)/FLOOR(4.1) = CEIL(2.1)/CEIL(3.7) = ROUND(3.1)/ROUND(3.6) = 0.75.`

- Numbers may be written with or without leading and trailing zeroes. Decimal points for integral values are optional. For example, all of the following are valid expressions (in the USA; see the next point): `3; 3.00; 0.03; .03; -0.30; -5`.

- Note, however, that the decimal delimiter (point or comma) used by *OUTBREAK* will be the appropriate one for the regional data format set in Windows on the computer. Thus, ¼ would be written as `0,25` in most of Europe, but as `0.25` in the USA and Australia.

- Functions containing invalid mathematical expressions are prohibited, such as:

    - `SQR(-10)` Square root of a negative number
    - `LN(-10)` Natural log of a negative number or zero
    - `5/0` Division by zero
    - `TAN(1.5707963)` Tangent of PI / 2
    - `ASIN(1.1)` Arcsine or arccosine of a value > 1, or < –1

- Some mathematically valid functions would be ambiguous or meaningless. For example, functions of **K** should not include **A** (age) or **S** (sex) as parameters, because the condition of exceeding the carrying capacity is a population-level phenomenon.

- There is no limit on the length of a function. Often there are multiple functions that will achieve the same purpose. A more concise function may be easier to type and may also run faster in the program. However, it may be easier to see the logic in a more explicit but longer function that evaluates to the same result.

## Some examples of the use of functions in *OUTBREAK*

| | |
|---|---|
| • Sample a duration (e.g., incubation period of the infection) from a uniform distribution of integers ranging from 10 to 20 days | =IUNIFORM(10;20) |
| • Sample a duration from a normal distribution with mean 10 and SD = 5 | =10+5*NRAND |
| • Set the probability of encounter to be 10% whenever the distance between two individuals is less than 4. | =0.1*(DIST<4) |
| • Set the probability of encounter to be 50% whenever the two individuals are in the same grid cell. | =0.5*SAME |
| • Set the probability of encounter to be a declining function of the distance between two individuals. | =0.5^DIST |
| • Set the probability of transmission to be higher for infected males than for infected females | =IF(SEX='M';0.02;0.01) |
| • Set the probability of transmission to be higher when the Infected individual and the Susceptible individual are of the same sex | =(ISEX=SSEX)*0.01+0.01<br>*Note that an equivalent function would be: =IF(ISEX=SSEX;0.02;0.01)* |
| • Set the movement with respect to the X-coordinate to be a random distance distributed as a normal with SD = 5. | =XCOORD+5*NRAND |
| • Let movement occur only every 30$^{th}$ day (i.e., once a month) | =(DAY%30)=0 |

# Chapter 7. Worked examples

This chapter provides a more detailed description of common elements that may need to be included in a model. This chapter is still under development, and more examples will be added in the future.

## 7.1 Setting up a home range

Conditioning animal moments to resemble the existence of a home range is a common model requirement. To this end, a simple example is provided in Box 5. Limiting *movements within home ranges*). Using this as a starting point, we provide a detailed explanation on how to set this up using the example project "O2Demo.xml," which is distributed with the *OUTBREAK* installation file.

### Example 1. Limiting movements within a certain distance from a central point

#### Step 1. Create a new scenario

Load the "O2Demo.xml" project in *OUTBREAK.* From the **General Settings tab**, create a copy of the Scenario2-withSpatial, and rename the new scenario as Scenario2-withSpatial_HR1.

#### Step 2. Create State Variables

In the **State Variables tab**, create a new individual variable, label it ***HRXcenter*** and type in the *Initialization function* and *Birth function* fields ***=iuniform(0;100)***. Type in the *Transition function* ***=IS1***. This variable can be referred to, within *OUTBREAK,* as ***IS1*** or ***HRXcenter***, and will be used to set the X-coordinate of the center of each individual's home range.

Create a second individual variable, label it ***HRYcenter*** and type in the *Initialization function* and *Birth function* fields ***=iuniform(0;100)***. Type in the *Transition function* ***=IS2***. This variable will be used to set the Y-coordinate of the center of each individual's home range.

With this set up (see Fig. 27), individuals created at the beginning of the simulation and newborns during the simulation will be assigned random X and Y coordinates between 0 and 100. Setting the transition functions to ***=IS1*** or ***=IS2*** means that the value of the variables will stay the same throughout the simulation.

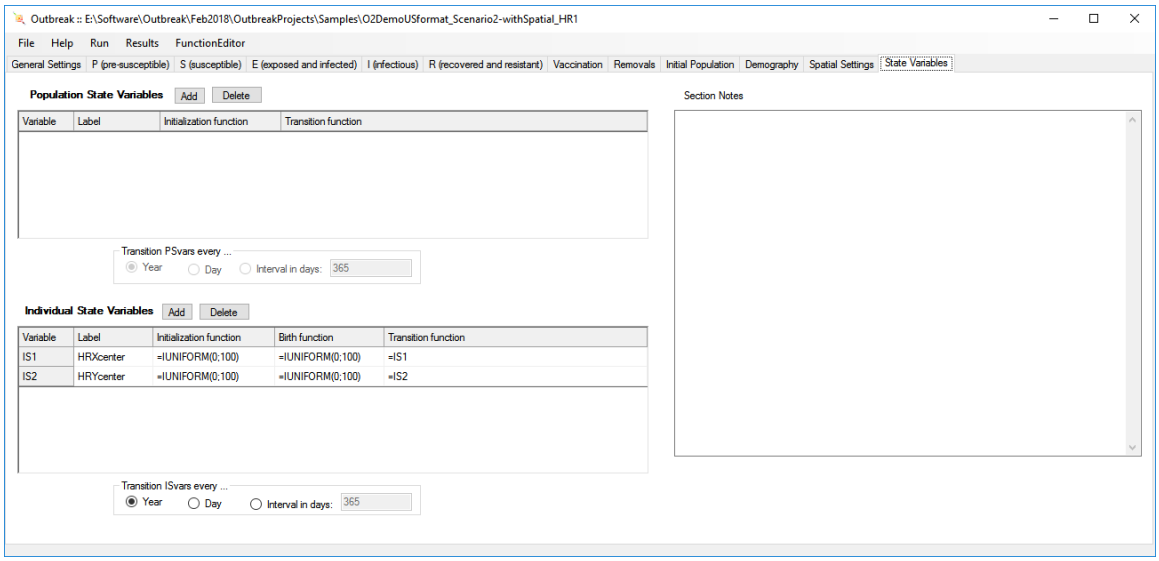



**Figure 27. *OUTBREAK* State Variables tab with two individual State Variables (IS1 and IS2) created to set the X and Y coordinates of the center of each individual's home range.**

## Step 3. Adjust the initial seeding of individuals

Select the **Spatial Settings tab**, change the **Initial seeding of individuals** to *Seed with functions.* Then type in ***=IS1*** and ***=IS2*** in the *X-coordinate* and *Y-coordinate* fields, respectively. In the **Daily movement rules** section, set the *Movement rule functions* to ***=IS1+5*NRAND*** and ***=IS2+5*NRAND*** for the XCOORD and YCOORD, respectively. Once this is done, the settings on this tab should look like Figure 28.

When the simulation is run, the initial distribution of individuals on the grid will be determined by the positions assigned in IS1 and IS2. Subsequent movements will have a bivariate normal distribution with mean distance 0, and standard deviation of 5 from the center of the individual's home range. Note that the settings in the original scenario that was used as template were to move individuals every 30 days. Unless this is changed, the distances moved in this simulation would represent the monthly distance covered by individuals.

## Step 4. Validate that individuals are moving as expected

A way to quickly visualize whether individuals in the model are moving approximately as expected is to change the initial population size to a very small number (e.g., 5) so that it is possible to visualize the movement while the simulation is running. A more formal approach is to request the optional individual list files, and check that the simulate movements are consistent with the allowed rules. Note that the individual list files are saved at the end of each year, so it will not be possible to actually check the positions of individuals at each movement.

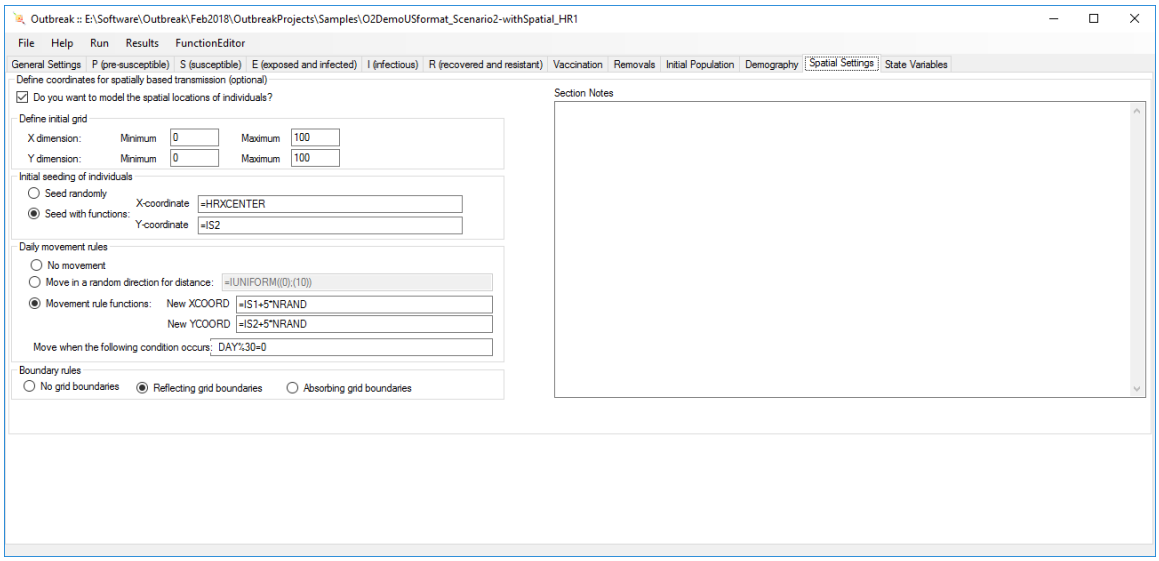



**Figure 28. Outbreak Spatial Settings tab set up using functions to determine the initial distribution of individuals and their subsequent movements on the grid.**

# Example 2. Random, short term, continuous movements constrained within a home range

The user might like to model short term movements, which are typically shorter than the overall length of the home range size, while ensuring that, over time, these movements are within the home range boundaries. For example, let's say that the user wants to model the monthly movements as having a normal distribution of mean zero and standard deviation 2, but within a home range of radius 10.

## Step 1. Create a new scenario

If not already open, load the "O2Demo.xml" project in *OUTBREAK.* From the **General Settings tab**, make a copy of the model Scenario2-withSpatial_HR1 and rename it Scenario2-withSpatial_HR2.

## Step 2. Create State Variables

On the **State Variables tab** create two Individual State Variables and label them ***NewXCOORD*** and ***NewYCOORD*** respectively. Set the *Initialization function* and *Birth function* fields to ***IS1*** for ***NewXCOORD*** and ***IS2*** for ***NewYCOORD***. Set the *Transition function* to ***=XCOORD+2*NRAND*** and ***=YCOORD+2*NRAND*** respectively. Lastly, change the transition to be updated every 30 days. Once done, this tab should look like the image below (Fig. 29).

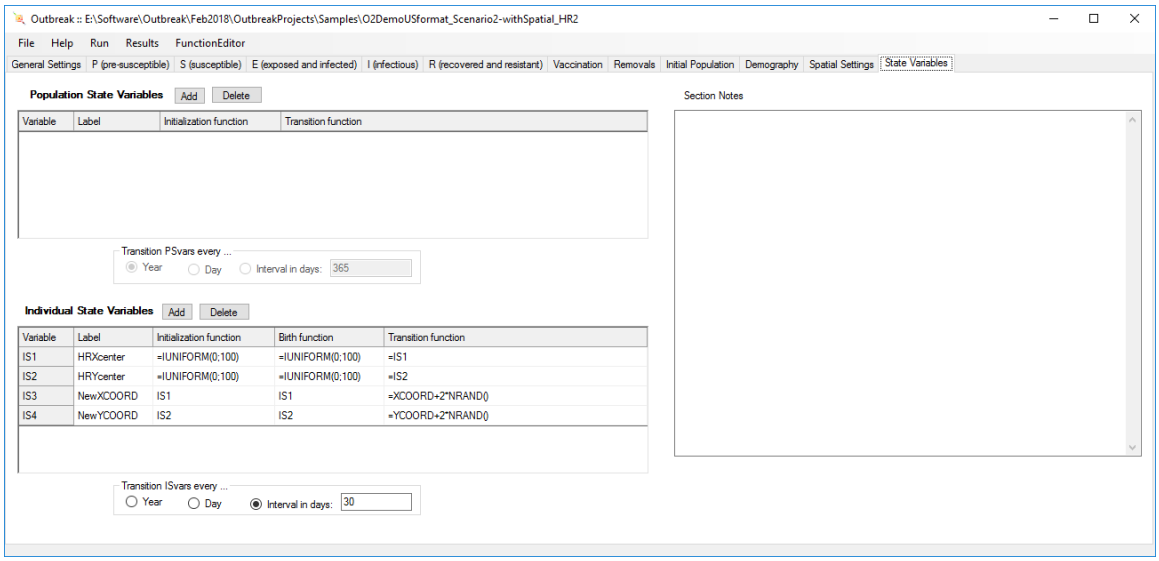



**Figure 29. *OUTBREAK* State Variables tab with two Individual State Variables (IS3 and IS4) created to set a new X and Y coordinate at each movement.**

## Step 3. Adjust the daily movement rules

On the **Spatial Settings tab**, change the movement rules to

*New XCOORD* *=IF(ABS(IS1 - IS3)<10; IS3; XCOORD)*

*New YCOORD* *=IF(ABS(IS2- IS4)<10; IS4; YCOORD)*

In this model, individuals will acquire, at each movement, a new X- and Y-coordinate that it has a mean distance zero and standard deviation 2 from their previous coordinate, but if the distance of the new coordinate from the center of the home range exceeds 10, they will retain their previous coordinates. Alternatively, the movement can be truncated at the home range boundaries by modifying the movement rules as follows:

*=IF(ABS(IS1 - IS3)<10; IS3; IS1 + 10)*

and

*=IF(ABS(IS2- IS4)<10; IS4; IS2 + 10)*

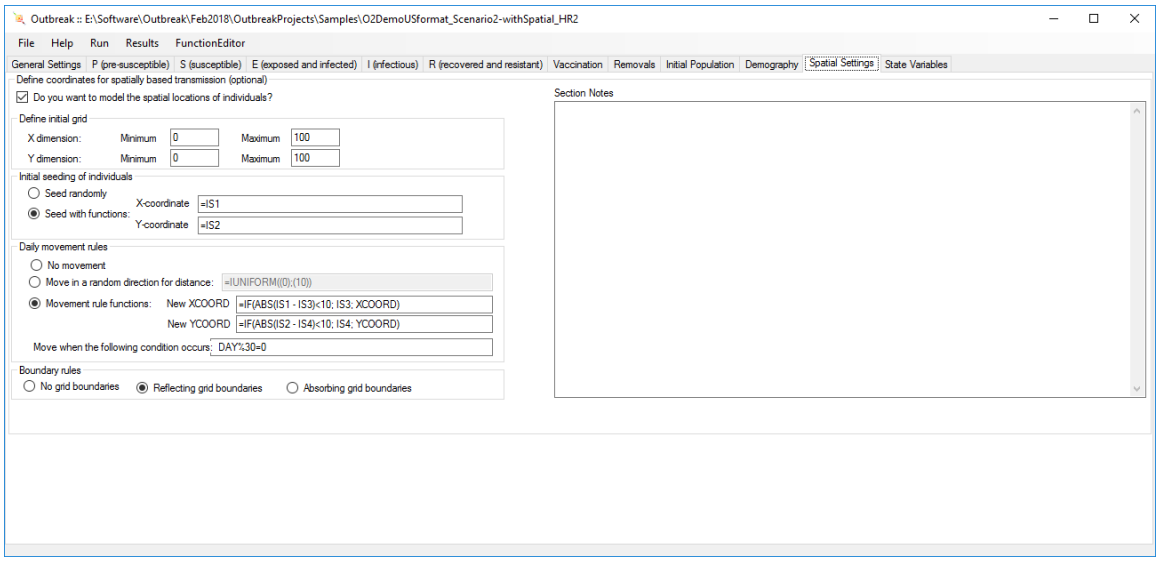



**Figure 30. *OUTBREAK* Spatial Settings tab set up using functions to determine the initial distribution of individuals and their subsequent short-term movements on the grid.**

# Example 3. Shifting home range position over time and controlling for home range 'shape'

Often, home ranges' positions are not fixed but may shift over time as a result of individuals' exploration in the surrounding areas. This additional complexity can be easily incorporated in the model as explained below.

## Step 1. Create a new scenario.

If not already open, load the "O2Demo.xml" project in *OUTBREAK.* From the **General Settings tab**, make a copy of the scenario created above and rename it Scenario2-withSpatial_HR3.

## Step 2. Create State Variables.

On the **State Variables tab**, change the *Transition function* of the IS1 and IS2 variables to ***=IS1 + 2 * NRAND * (DAY=365)*** and ***=IS2 + 2 * NRAND * (DAY=365)***, respectively. Each time these variables are updated, they will change based on a normal distribution with mean zero and standard deviation. Note that the term ***(DAY=365)*** can take only value 0 or 1 (FALSE or TRUE). It is used to force the updating of these variables only once a year, as it is commonly the case that shifts in home range locations are slow and progressive (i.e. over the years) rather than sudden. The ***IS3*** and ***IS4*** variables are to be updated monthly. However, the State Variables, in *OUTBREAK* algorithm, are updated after the movements. Hence, the term ***((DAY+1)%30=0)*** needs to be added to their transition functions. This will

cause the variables to update the day before the movements occur, so that a new value is considered at each movement. Lastly, the transition of the variables needs to be changed to 1 to ensure that the variables are updated when needed.

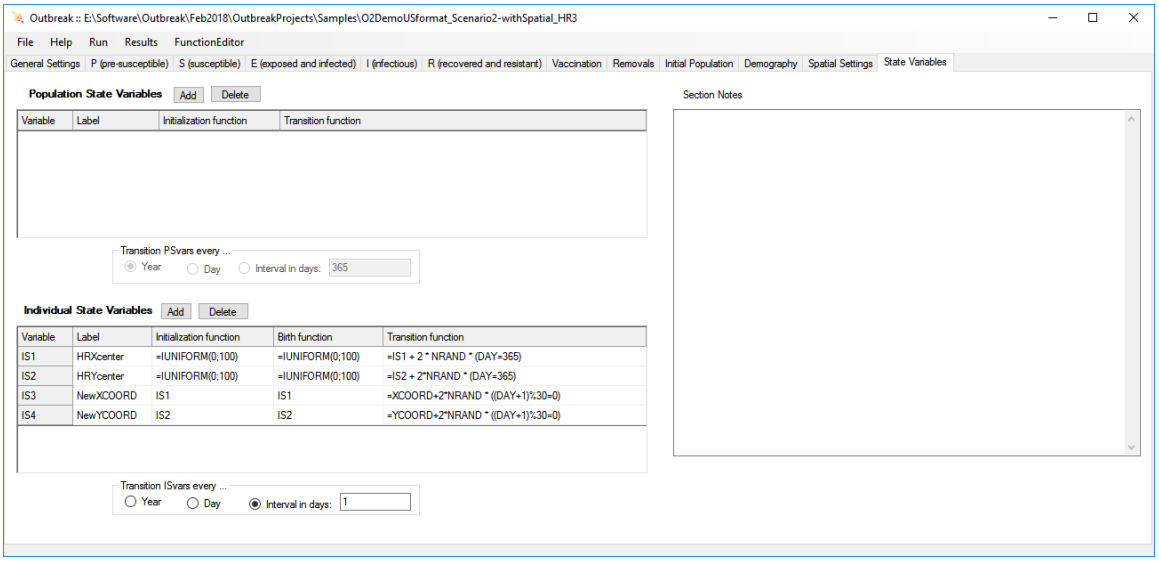



**Figure 31. Outbreak State Variables tab with IS1 and IS2 set to update once a year based on a normal distribution with mean zero and standard deviation.**

## Step 3. Control for home range "shape"

So far, it was assumed that a circular shape would adequately describe home ranges. However, there may be situations where other shapes may be more appropriate. For example, if an elongated shape is desired, this can be easily obtained by modifying the maximum distance allowed on the X- or Y-coordinate. For example, modifying the movement rule function in the **Spatial Settings tab** from:

*=IF(ABS(IS1 - IS3)<10; IS3; XCOORD)*

To

*=IF(ABS(IS1 - IS3)<5; IS3; XCOORD)*

Will cause the home range shape to be elongated, with a narrower radius along the X-axis, 5, and a vertical radius of 10.

# Chapter 8. Getting help

Within the *OUTBREAK* program, this manual is accessible via the *Help* menu at the top of the program window. In addition, hitting the F1 key at any time will open the manual to a page that is relevant to the user's current place in the program (e.g., an input section or results window).

The *OUTBREAK* webpage offers a few additional resources for technical training and user support. Additional training materials, including online modules and videos, will be developed in the future. Users are encouraged to check the website regularly for updates. A list of frequently asked questions is available at: scti.tools/help-support/

Several introductory videos can be downloaded from the following link:

- https://scti.tools/downloads/#EducationalMaterials

Lastly, a few publications where *OUTBREAK* has been used is provided on the next page.

# Examples of the use of *OUTBREAK* in the scientific literature

Bradshaw, C.J.A., McMahon, C.R., Miller, P.S., Lacy, R.C., Watts, M.J., Verant, M.L., Pollak, J.P., Fordham, D.A., Prowse, T.A.A. and B.W. Brook (2011). Novel coupling of individual-based epidemiological and demographic models predicts realistic dynamics of tuberculosis in alien buffalo. Journal of Applied Ecology 49: 268–277

Lacy, R.C., Miller, P.S., Nyhus, P.J., Pollak, J.P., Raboy, B.E., and Zeigler, S.L. (2013) Metamodels for Transdisciplinary Analysis of Wildlife Population Dynamics. PLoS ONE 8(12)

Prowse, T.A.A., Johnson, C.N., Lacy, R.C., Bradshaw, C.J.A., Pollak, J.P., Watts, M.J. and B.W. Brook (2013). No need for disease: testing extinction hypotheses for the thylacine using multi-species metamodels. Journal of Animal Ecology 2013 (82): 355–364

Shoemaker, K.T., Lacy, R.C., Verant, M.L., Brook, B.W., Livieri, T.M., Miller, P.S., D.A. Fordham, D.A. and H.R. Akcakaya (2014). Effects of prey metapopulation structure on the viability of black-footed ferrets in plague-impacted landscapes: a metamodelling approach. Journal of Applied Ecology 2014 (51): 735–745

Wells, K., Brook, B.W., Lacy, R.C., Mutze, G.J., Peacock, D.E., Sinclair, R.G., Schwensow, N., Cassey, P., O'Hara, R.B. and D.A. Fordham (2015). Timing and severity of immunizing diseases in rabbits is controlled by seasonal matching of host and pathogen dynamics. J. R. Soc. Interface 12:20141184

# References

Anderson, R.M., and May, R.M. 1991. Infectious diseases of humans. Oxford University Press, Oxford, UK.

Caley P, Ramsey D. 2001. Estimating disease transmission in wildlife, with emphasis on leptospirosis and bovine tuberculosis in possums, and effects of fertility control. Journal of Applied Ecology 38:1362-1370.

Craft ME. 2015. Infectious disease transmission and contact networks in wildlife and livestock. Philosophical Transactions of the Royal Society B: Biological Sciences 370.

Gavryushkina A, Welch D, Stadler T, Drummond AJ. 2014. Bayesian inference of sampled ancestor trees for epidemiology and fossil calibration. PLoS Comput Biol 10:e1003919.

Grenfell, B.T., and Dobson, A.P., eds., 1995. Ecology of infectious diseases in natural populations. Cambridge University Press, Cambridge, UK.

Grenfell B, Anderson R. 1985. The estimation of age-related rates of infection from case notifications and serological data. Epidemiology and Infection 95:419-436.

Hamede RK, Bashford J, McCallum H, Jones M. 2009. Contact networks in a wild Tasmanian devil (Sarcophilus harrisii) population: using social network analysis to reveal seasonal variability in social behaviour and its implications for transmission of devil facial tumour disease. Ecology Letters 12:1147-1157.

Jakob-Hoff RM, MacDiarmid SC, Lees C, Miller PS, Travis D, Kock R. 2014. Manual of Procedures for Wildlife Disease Risk Analysis. World Organisation for Animal Health, Paris, 160 pp. Published in association with the International Union for Conservation of Nature and the Species Survival Commission.

Keeling MJ, Rohani P. 2011. Modeling infectious diseases in humans and animals: Princeton University Press.

Keet D F , Davies-Mostert H, Bengis RG, Funston P, Buss P, Hofmeyr M, Ferreira S, Lane E, Miller P, B. Daly G. 2009 . Disease Risk Assessment Workshop Report: African Lion ( *Panthera leo* ) Bovine Tuberculosis . Conservation Breeding Specialist Group (CBSG SSC / IUCN) / CBSG Southern Africa. Endangered Wildlife Trust , Skukuza, South Africa .

Kühnert D, Stadler T, Vaughan TG, Drummond AJ. 2013. Simultaneous reconstruction of evolutionary history and epidemiological dynamics from viral sequences with the birth-death SIR model. Journal of the Royal Society of London Interface 11.

Kühnert D, Wu C-H, Drummond AJ. 2011. Phylogenetic and epidemic modeling of rapidly evolving infectious diseases. Infection, Genetics and Evolution 11:1825-1841.

Kukielka E, Barasona JA, Cowie CE, Drewe JA, Gortazar C, Cotarelo I, Vicente J. 2013. Spatial and temporal interactions between livestock and wildlife in South Central Spain assessed by camera traps. Preventive Veterinary Medicine 112:213-221.

Lacy RC, Miller PS, Nyhus PJ, Pollak J, Raboy BE, Zeigler SL. 2013. Metamodels for transdisciplinary analysis of wildlife population dynamics. PloS One 8:e84211.

Lacy RC, Pollak JP. 2014. VORTEX: a stochastic simulation of the extinction process. Version 10. Brookfield: Chicago Zoological Society.

McCallum H, Barlow N, Hone J. 2001. How should pathogen transmission be modelled? Trends in ecology & evolution 16:295-300.

Morris, W.F., and Doak, D.F. 2002. Quantitative conservation biology. Theory and practice of population viability analysis. Sinauer, Sunderland, Massachusetts, USA (replacing Sjogren-Golve and Hanski 2000 as well as Lacy and Pollak 2014 on page 11).

Robert K, Garant D, Pelletier F. 2012. Keep in touch: Does spatial overlap correlate with contact rate frequency? The Journal of Wildlife Management 76:1670-1675.

Schauber EM, Storm DJ, Nielsen CK. 2007. Effects of joint space use and group membership on contact rates among white-tailed deer. Journal of Wildlife Management 71:155-163.

Schumaker NH, Brookes A. 2018. HexSim: a modeling environment for ecology and conservation. Landscape Ecology.

Sjögren-Gulve P, Hanski I. 2000. Metapopulation Viability Analysis Using Occupancy Models. Ecological Bulletins:53-71.

Stadler T, Kouyos R, von Wyl V, Yerly S, Böni J, Bürgisser P, Klimkait T, Joos B, Rieder P, Xie D, et al. 2012. Estimating the Basic Reproductive Number from Viral Sequence Data. Molecular Biology and Evolution 29:347-357.

Stadler T, Kühnert D, Bonhoeffer S, Drummond AJ. 2013. Birth–death skyline plot reveals temporal changes of epidemic spread in HIV and hepatitis C virus (HCV). Proceedings of the National Academy of Sciences 110:228-233.

Tompkins D, Ramsey D, Cross M, Aldwell F, De Lisle G, Buddle B. 2009. Oral vaccination reduces the incidence of tuberculosis in free-living brushtail possums. Proceedings of the Royal Society of London B: Biological Sciences 276:2987-2995.

VanderWaal KL, Atwill ER, Isbell LA, McCowan B. 2014. Linking social and pathogen transmission networks using microbial genetics in giraffe (Giraffa camelopardalis). Journal of Animal Ecology 83:406-414.

Vercauteren KC, Lavelle MJ, Seward NW, Fischer JW, Phillips GE. 2007. Fence-Line Contact Between Wild and Farmed White-Tailed Deer in Michigan: Potential for Disease Transmission. Journal of Wildlife Management 71:1603-1606.

Vynnycky E, White R. 2010. An introduction to infectious disease modelling: Oxford University Press.

Wallinga J, Teunis P, Kretzschmar M. 2006. Using data on social contacts to estimate age-specific transmission parameters for respiratory-spread infectious agents. American Journal of Epidemiology 164:936-944.

Wilensky U, Evanston I. 1999. NetLogo: Center for connected learning and computer-based modeling. Northwestern University, Evanston, IL 4952.

# Appendix I. Using *OUTBREAK* in *METAMODEL MANAGER*

*METAMODEL MANAGER* is a modeling platform that allows simulations, such as *OUTBREAK,* to be linked together into more comprehensive models of the diverse processes driving species dynamics, such as disease, species interactions, habitat change, dispersal, climate change, and genetic processes. *METAMODEL MANAGER* and its manual can be downloaded from https://scti.tools/downloads/, and the metamodeling approach and the program are described in Lacy et al. (2013). For users of *METAMODEL MANAGER*, the following brief description provides some information about how to link *OUTBREAK* into metamodels.

Within *METAMODEL MANAGER*, *OUTBREAK* can be run as either a System-level model (creating the initial population and doing its own demography) or as a Modifier model (changing only the disease state of individuals whose demography is managed by *VORTEX* or some other System model). When *OUTBREAK* is used as a Modifier model in a metamodel, its demography is disabled – that is, the values on the Initial Population and Demography input tabs are ignored, because they are replaced by the demography provided by the System-level model.

The demographic simulation in *OUTBREAK* (when it is used as the System-level model) differs in several ways from the one in *VORTEX*. Primarily, *OUTBREAK* models events on a daily basis, so that demographic events are interlaced with disease processes, and *OUTBREAK* does not include many aspects of stochasticity (e.g., environmental variation, catastrophes, genetic drift) that are modeled in *VORTEX*.

*OUTBREAK* (from version 2.5 onwards) provides a utility for the user to create Population and Individual State Variables. Additionally, *OUTBREAK* always creates a Population State Variable called "Prevalence," and Individual State Variables called "DiseaseState," "DaysInState," "StatePermanent," and "IsVaccinated". Within functions defining rates in *OUTBREAK*, these variables should be referenced by the labels PREV, Z, DSTATE, CHRONIC, and V, respectively. When linked to *METAMODEL MANAGER*, *OUTBREAK* must declare the Population and Individual State Variables that are shared with *METAMODEL MANAGER*. Variables can be shared by using the same labels and, when so, their values can be changed by other models in a metamodel.

*OUTBREAK* can also use any other Global, Population, or Individual State Variables created by other programs and handed to it by METAMODEL MANAGER. Thus, for example, if a *VORTEX* scenario creates an Individual State Variable called "BodyCondition," the probability of disease transmission (or any other input variable) can be specified to be a function of BodyCondition within the linked *OUTBREAK* . Similarly, with an Individual State Variable set to be the genotype at one or more loci, disease susceptibility or recovery rates can be made to be dependent on the genotypes that are simulated in *VORTEX*.

*OUTBREAK* does provide a utility to simulate movements of individuals on the landscape, because the distance between individuals can be an important determinant of disease transmission. The

spatial model in *OUTBREAK* is relatively basic, with the user needing to specify the functions that define movement probabilities on an X-Y grid at each daily time step, but this is a capability that is not included in *VORTEX*. Therefore, in a linked *VORTEX*-*OUTBREAK* metamodel, *OUTBREAK* can be used to simulate individual movements, and *VORTEX* can use the X-Y location of each individual as a determinant of demographic rates or events. For example, the criteria for acceptable mates in *VORTEX* can be that a potential mate must be within a certain Euclidean distance (*VORTEX* variable DIST) to be accepted. Or, the probability of mortality can be specified to be higher in some regions of the X-Y space.

When using *OUTBREAK* or *VORTEX* in a metamodel with spatial movement, the locations of individuals should be obtained with variables XCOORD and YCOORD, rather than X and Y. This is because X and Y have alternative meanings as built-in variables in *VORTEX* (X = number of females, Y = year). Thus, the Individual State Variables in *VORTEX* that will be used for spatial locations and linked to the *OUTBREAK* simulation should be labeled XCOORD and YCOORD, and any functions of location in *VORTEX* or *OUTBREAK* should use these labels.

As with *VORTEX*, after the metamodel has run, you can view the tables and graphs of *OUTBREAK* results (the summaries of disease dynamics) by opening the *OUTBREAK* project in the full *OUTBREAK* interface.

# Appendix II. Glossary of terms

Many of the following glossary of terms are taken or adapted from the IUCN SSC/OIE *Manual of Procedures for Wildlife Disease Risk Analysis* (Jakob-Hoff *et al*., 2013). As the meaning of some terms can vary between disciplines, all terms are defined in the context of *OUTBREAK*. Note that italicized words within definitions refer to other words included in the glossary.

| | |
|---|---|
| **Disease** | Any impairment of the normal structural or physiological state of a human or animal (also see *Infectious Disease).* |
| **Disease risk analysis** | The application of risk analysis to identify diseases which may enter a specified animal population, to identify the likelihood of such introductions, assess their consequences and to identify measures which may be applied to mitigate either the likelihood of introduction or the magnitude of consequences. |
| **Epidemic** | A sudden, rapid spread or increase in the prevalence of a *parasite* or *disease*. An epidemic is often the result of a change in circumstances which favor parasite transmission, such as a rapid increase in host population density or the introduction of a new parasite. |
| **E (exposed and infected) state** | The disease state in *OUTBREAK* that includes individuals who have been exposed to and contracted the *pathogen* but are not infectious to other individuals. |
| **Host** | Any animal that is capable of harboring a *parasite*, regardless of whether they play a role in further *transmission* of the parasite. |
| **Incubation period** | The time that elapses between *infection* with a *parasite* and the onset of *disease*. Used synonymously with latency in *OUTBREAK* (see *Latent period*). |
| **Infection** | The entry and development or multiplication of a *parasite* in the body of a *host*, where it may or may not cause *disease*. |
| **Infectious disease** | The debilitating effects of *infection* or infestation by a *parasite*. It is possible for a *host* to be infected by a *parasite* but to show no symptoms of *disease* (see also *Disease*). |

| | |
|---|---|
| **Infectious period** | The period during which the infected individual is able to transmit the *infection*. |
| **I (infectious) state** | The disease state in *OUTBREAK* that includes individuals who have been exposed to the *pathogen* (with or without causing disease) and can transmit the pathogen to other *susceptible* individuals. |
| **Latent period** | The period when an individual is infected but not yet capable of transmitting the *infection*. |
| **Parasite** | An agent that lives on or within a *host* and that survives at the expense of the host regardless of whether a *disease* state follows. This definition includes both *microparasites* (e.g., bacteria, viruses) and *macroparasites* (e.g., helminths, arthropods). |
| **Pathogen** | Any *disease*-causing *parasite*. |
| **P (pre-susceptible) state** | The disease state in *OUTBREAK* that includes all individuals from birth to the earliest age of susceptibility. |
| **Prevalence** | The proportion of the host population with *infection* or *disease* (i.e. the proportion in the E or I states). |
| **R (recovered and resistant) state** | The disease state in *OUTBREAK* that includes individuals who are no longer infectious as a result of acquiring permanent or temporary immunity. |
| **S (susceptible) state** | The disease state in *OUTBREAK* that includes individuals who can become infected with a *disease* following exposure to an etiological agent (i.e. disease *pathogen*). |
| **Transmission** | The process by which a *parasite* passes from a source of *infection* to a new *host*. |
| **Uncertainty** | The lack of precise knowledge of the input values which is due to measurement error or to lack of knowledge of the steps required, and the pathways from hazard to risk, when building the scenario being assessed. |
| **Vaccination** | The immunization of *susceptible* animals through the administration of a vaccine comprising antigens appropriate to the disease to be controlled. |