User Manual for PatchImportance 1.0: Quantifying Relative Habitat Patch Importance Based on Metapopulation Persistence and Minimum Abundance

M.H. Grinnell and J.M.R. Curtis

Fisheries and Oceans Canada
Science Branch, Pacific Region
Pacific Biological Station
3190 Hammond Bay Road
Nanaimo, BC
V9T 6N7

2012

Canadian Technical Report of Fisheries and Aquatic Sciences 2977

Fisheries and Oceans Canada   Pêches et Océans Canada

Canada

Canadian Technical Report of Fisheries and Aquatic Sciences

Technical reports contain scientific and technical information that contributes to existing knowledge but which is not normally appropriate for primary literature. Technical reports are directed primarily toward a worldwide audience and have an international distribution. No restriction is placed on subject matter and the series reflects the broad interests and policies of Fisheries and Oceans Canada, namely, fisheries and aquatic sciences.

Technical reports may be cited as full publications. The correct citation appears above the abstract of each report. Each report is abstracted in the data base *Aquatic Sciences and Fisheries Abstracts*.

Technical reports are produced regionally but are numbered nationally. Requests for individual reports will be filled by the issuing establishment listed on the front cover and title page. Out-of-stock reports will be supplied for a fee by commercial agents.

Numbers 1–456 in this series were issued as Technical Reports of the Fisheries Research Board of Canada. Numbers 457–714 were issued as Department of the Environment, Fisheries and Marine Service, Research and Development Directorate Technical Reports. Numbers 715–924 were issued as Department of Fisheries and Environment, Fisheries and Marine Service Technical Reports. The current series name was changed with report number 925.

Rapport technique canadien des sciences halieutiques et aquatiques

Les rapports techniques contiennent des renseignements scientifiques et techniques qui constituent une contribution aux connaissances actuelles, mais qui ne sont pas normalement appropriés pour la publication dans un journal scientifique. Les rapports techniques sont destinés essentiellement à un public international et ils sont distribués à cet échelon. II n'y a aucune restriction quant au sujet; de fait, la série reflète la vaste gamme des intérêts et des politiques de Pêches et Océans Canada, c'est-à-dire les sciences halieutiques et aquatiques.

Les rapports techniques peuvent être cités comme des publications à part entière. Le titre exact figure au-dessus du résumé de chaque rapport. Les rapports techniques sont résumés dans la base de données *Résumés des sciences aquatiques et halieutiques*.

Les rapports techniques sont produits à l'échelon régional, mais numérotés à l'échelon national. Les demandes de rapports seront satisfaites par l'établissement auteur dont le nom figure sur la couverture et la page du titre. Les rapports épuisés seront fournis contre rétribution par les agents commerciaux.

Les numéros 1 à 456 de cette série ont été publiés à titre de Rapports techniques de l'Office des recherches sur les pêcheries du Canada. Les numéros 457 à 714 sont parus à titre de Rapports techniques de la Direction générale de la recherche et du développement, Service des pêches et de la mer, ministère de l'Environnement. Les numéros 715 à 924 ont été publiés à titre de Rapports techniques du Service des pêches et de la mer, ministère des Pêches et de l'Environnement. Le nom actuel de la série a été établi lors de la parution du numéro 925.

Canadian Technical Report of
Fisheries and Aquatic Sciences 2977

2012

USER MANUAL FOR PatchImportance 1.0: QUANTIFYING RELATIVE HABITAT PATCH IMPORTANCE BASED ON METAPOPULATION PERSISTENCE AND MINIMUM ABUNDANCE

by

M.H. Grinnell[*] and J.M.R. Curtis[†]

Fisheries and Oceans Canada
Science Branch, Pacific Region
Pacific Biological Station
3190 Hammond Bay Road
Nanaimo, BC
V9T 6N7

---

[*] E-mail: `matt.grinnell@dfo-mpo.gc.ca` | tel: (250) 756.7326
[†] E-mail: `janelle.curtis@dfo-mpo.gc.ca` | tel: (250) 756.7157

Cat. No. Fs 97-6/2977 E ISSN 0706-6457
Cat. No. Fs 97-6/2977 E-PDF ISSN 1488-5379

Correct citation for this publication:

Grinnell, M.H. and Curtis, J.M.R. 2012. User manual for PatchImportance 1.0: Quantifying relative habitat patch importance based on metapopulation persistence and minimum abundance. Can. Tech. Rep. Fish. Aquat. Sci. 2977: vi + 41 p.

Contents

List of Figures

List of Tables

List of Listings

Abstract

Grinnell, M.H. and Curtis, J.M.R. 2012. User manual for PatchImportance 1.0: Quantifying relative habitat patch importance based on metapopulation persistence and minimum abundance. Can. Tech. Rep. Fish. Aquat. Sci. 2977: vi + 41 p.

Those developing conservation strategies that include protected areas to ensure species persistence are often faced with the difficult choice of selecting a subset of the total area of suitable habitat for protection. In these instances, protecting the most important habitat patches will facilitate efficient resource use, and maximize the probability of metapopulation persistence. However, identifying important patches presents challenges, and important patches may be currently unoccupied which makes their identification even more difficult. To help identify candidate patches for protection, we developed **PatchImportance**, a tool that quantifies the relative importance of each patch based on its influence on the probability of metapopulation persistence, and metapopulation expected minimum abundance. We demonstrate our analysis using the British Columbia sea otter (*Enhydra lutris*) metapopulation as a case study.

Résumé

Grinnell, M.H. and Curtis, J.M.R. 2012. User manual for PatchImportance 1.0: Quantifying relative habitat patch importance based on metapopulation persistence and minimum abundance. Can. Tech. Rep. Fish. Aquat. Sci. 2977: vi + 41 p.

Dans le cadre des stratégies de conservation, en plein développement, comprenant des zones protégées pour garantir la survie des espèces, il faut souvent faire le choix difficile de sélectionner une sous-partie de la zone totale constituant un habitat approprié pour la protection. Dans ces cas, la protection des parcelles d'habitat les plus importantes permettra une utilisation plus efficace des ressources et maximisera les chances de survie de la métapopulation. Toutefois, la détermination des parcelles importantes pose des difficultés et il est possible que certaines parcelles importantes soient actuellement inoccupées, ce qui rend leur repérage encore plus compliqué. Pour aider à repérer les parcelles potentielles pour servir à la protection, nous avons conçu **PatchImportance**, un outil qui quantifie l'importance relative de chaque parcelle en fonction de son influence sur les chances de survie de la métapopulation et des attentes concernant l'abondance minimale de la métapopulation. Nous faisons une démonstration de notre analyse en utilisant la métapopulation de la loutre de mer (*Enhydra lutris*) de la Colombie-Britannique comme étude de cas.

1 Motivation

One common approach to increasing the probability of persistence for species at risk of extinction is to protect areas of suitable habitat (e.g., parks, reserves, no-take zones; Dudley 2008). In Canada, species listed under the Species at Risk Act benefit from the protection of critical habitat, defined as the habitat required for species survival or recovery (SARA 2011). Critical habitat does not typically encompass the entire area of occupancy (AO; i.e., every habitat patch); exceptions include cases when habitat availability limits species persistence and recovery, or when data limitations preclude otherwise (Hatfield 2009). When available habitat does not limit species persistence and recovery, a subset of suitable patches may be sufficient to meet conservation goals (Rosenfeld and Hatfield 2006).

In such cases, recovery teams are faced with the difficult problem of selecting which patches to protect (i.e., deciding the number, size, shape, and spatial configuration; Diamond 1975; Pascual-Hortal and Saura 2006; McLeod et al. 2009). For example, a given patch could be crucial to metapopulation persistence due to its location, despite supporting a low abundance of individuals (Jordán et al. 2003). Also, currently unoccupied patches could be important in the future to facilitate an expanding distribution. The issue of selecting patches for protection is further complicated by the complex nature of metapopulations, which are influenced by factors that include patch-specific population dynamics parameters that may be correlated among patches, dispersal, environmental stochasticity, and catastrophes. Selecting patches for protection can involve tradeoffs, such as: many small patches *versus* few large patches; or closely-spaced patches *versus* widely-spaced patches (Williams et al. 2005). Metapopulation models are valuable conservation tools which can incorporate the aforementioned (and other) complexities to reveal underlying patterns, compare alternative management actions, and ultimately guide the management of species at risk (Akçakaya et al. 2007).

Although most patches have the potential to support species survival and recovery, patches may differ in their relative importance. In these cases, quantifying the relative importance of each patch may be useful to the decision making process (Urban and Keitt 2001). For example, identifying the most important patches can help prioritize areas for protection, and inform the development of conservation strategies to maximize the probability of species persistence given limited conservation resources (Jordán et al. 2003). In this paper, we describe our simulation approach using **PatchImportance**, which quantifies relative patch importance according to each patch's influence on the probability of metapopulation persistence, and metapopulation expected minimum abundance (EMA).

2 Background

This user manual follows the same outline as the **NetworkDistances** user manual (Grinnell and Curtis 2011) because both tools may be useful to similar audiences, and because both tools require similar data and programmes. Users familiar with the **GRIP** (Curtis and Naujokaitis-Lewis 2008*a*,*b*) version 2.0 script, or the **NetworkDistances** version 1.0 code may find it easier to implement the **PatchImportance** code, but prior knowledge of either tool is not necessary. The **PatchImportance** code can be implemented on a personal computer using easily accessible software. Like **GRIP**, **PatchImportance** is written in the programming language **R** (RDCT 2011), and interacts with **RAMAS Metapop** (Akçakaya 2005). Unlike **GRIP**, **PatchImportance** quantifies the relative importance of each patch, measured by its influence on the probability of metapopulation persistence, and metapopulation EMA.

We demonstrate **PatchImportance** by analysing the British Columbia (BC) sea otter (*Enhydra lutris*) metapopulation, a marine mammal of Special Concern in Canada (COSEWIC 2007). We provide some background information on BC sea otter habitat, metapopulation dynamics, and oil spill catastrophes in order to illustrate concepts, limitations, and opportunities where appropriate. The **PatchImportance** code is generic, and could be applied to other species given sufficient metapopulation dynamics data. We attempt to highlight sections of the code that may be customized to investigate other species, or provide advice on related science-based questions.

Please contact the authors if you have questions, comments, suggestions, or concerns regarding this manual, or the code. We are attempting to keep track of this code's use; please cite this manual and contact the authors if you use **PatchImportance** for research. Note that **PatchImportance** comes with absolutely no warranty.

3 British Columbia sea otters

BC sea otters were hunted for their dense fur and extirpated in the early 20th century. Since their reintroduction to Checleset Bay between 1969 and 1972, sea otters have increased in abundance and distribution (Figure 1; Nichol et al. 2009). BC sea otters are currently listed as Special Concern because the small population and limited range is susceptible to becoming Threatened or Endangered due to identified threats such as oil spills (COSEWIC 2007). Despite the lack of quantitative recovery targets, range expansion is crucial to reduce threats from oil spill catastrophes (Nichol 2007).

Quantifying patch importance is a critical step in identifying areas of high conservation value for sea otters. In addition to supporting species-specific conservation actions, such areas could be used to inform science-based processes to identify high-priority areas for protection in Canadian marine spatial planning initiatives (Clarke and Jamieson 2006*a*,*b*). To identify important sea otter patches using **PatchImportance**, we modeled the BC sea otter metapopulation using a realistic habitat map and population dynamics data, as well as a possible future oil spill catastrophe scenario (Table 1).

3.1 Habitat suitability and patches

Gregr et al. (2008) quantified BC sea otter habitat suitability on a 0.5×0.5 kilometre

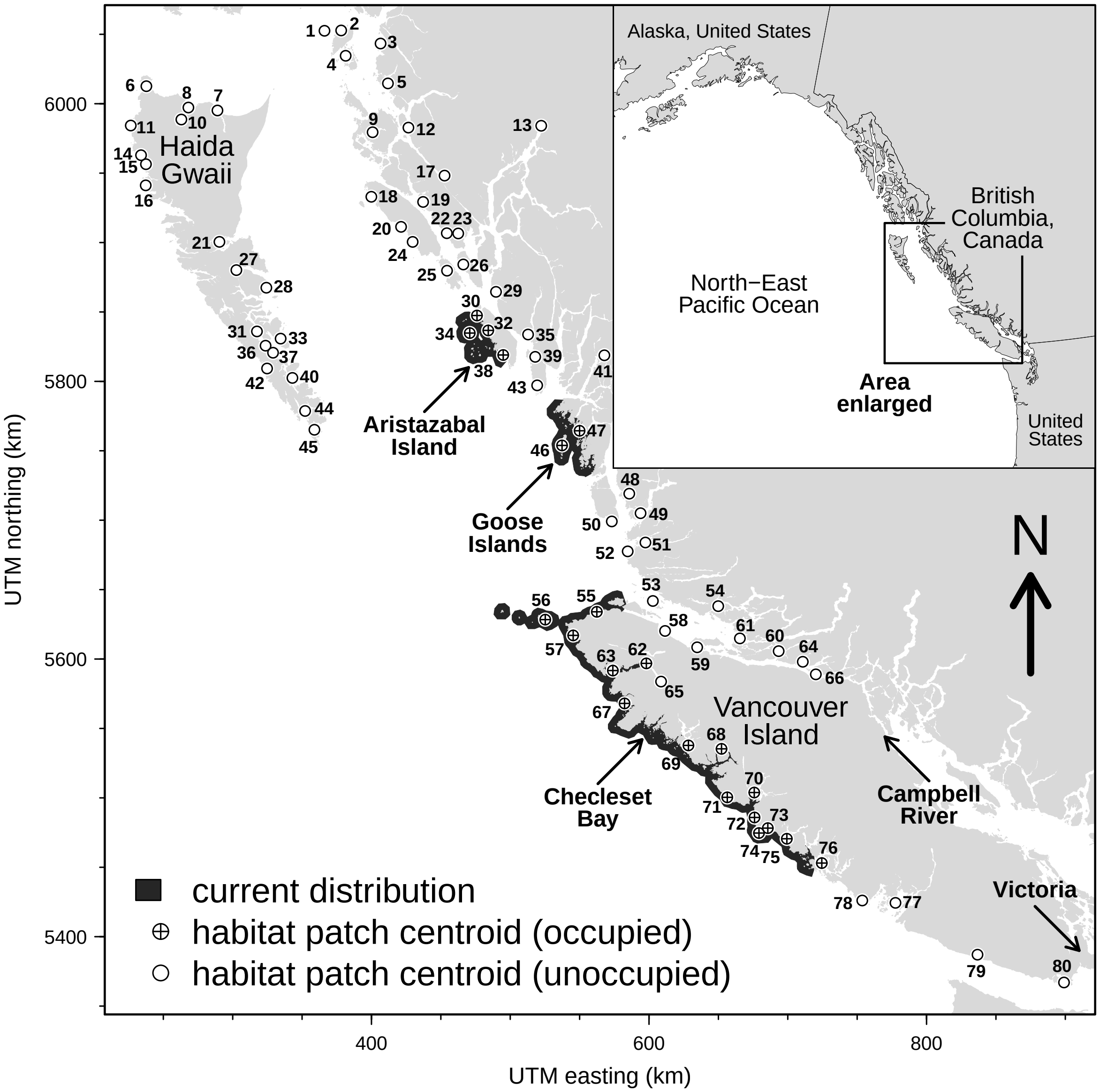


Figure 1. Current British Columbia sea otter distribution (Nichol et al. 2009), habitat patches, and place names mentioned in the text. Geographic coordinates are projected in Universal Transverse Mercator (UTM; zone 9), in kilometres (km).

(km) grid as a function of coastline and bathymetric complexity. We used this habitat suitability map to identify discrete patches based on neighbourhood distance, and a habitat suitability threshold. As in Gregr et al. (2008), we excluded the area East of Vancouver Island between Victoria and Campbell River, and areas > 5 km from land. We ran the **RAMAS Patch** programme (Akçakaya 2005) on this habitat suitability map, and identified 80 discrete patches of suitable sea otter habitat in BC. Although the current distribution only covers a portion of the BC coast (Nichol et al. 2009), we

Table 1. Parameters used to model the British Columbia (BC) sea otter metapopulation. Units: distance is in kilometres (km).

Parameter description	Value	Type or units	Reference
Neighbourhood distance*	2.500	km	Loughlin (1980)
Habitat suitability threshold†	0.200	proportion	Gregr et al. (2008)
Initial abundance‡	0.510	females · km^{-2}	Gregr et al. (2008)
Carrying capacity	1.270	females · km^{-2}	Gregr et al. (2008)
Maximum growth rate	1.186	proportion	Watson et al. (1997)
Survival of pups to juveniles	0.600	proportion	Krkosek et al. (2007)
Juvenile survival	0.631	proportion	Krkosek et al. (2007)
Survival of juveniles to adults	0.269	proportion	Krkosek et al. (2007)
Adult survival	0.900	proportion	Krkosek et al. (2007)
Adult fecundity	0.450	proportion	Krkosek et al. (2007)
Local oil spill probability$^{\S\blacktriangle}$	0.175	probability	See footnotes$^{\S\blacktriangle}$
Regional oil spill probability§	0.029	probability	See footnote§
Local oil spill multiplier$^\diamond$	0.770	proportion	Gerber et al. (2004)
Regional oil spill multiplier$^\diamond$	0.580	proportion	Gerber et al. (2004)

* Maximum distance between suitable cells to consider them part of the same discrete patch.

† Minimum habitat suitability for breeding.

‡ Patches within the current distribution; patches outside the current distribution had initial abundance equal to zero.

§ Yearly probabilities based on frequencies of 97 and 16 spills per 2.20×10^{13} litres (l) of crude oil transported for local and regional spills, respectively (Anderson and LaBelle 2000), and 3.98×10^{10} l of hydrocarbons transported annually in BC (Dickins 1995).

$^\blacktriangle$ Coast-wide probability of a local oil spill; patch-specific probability is $\frac{0.175}{80} = 0.002$.

$^\diamond$ Proportion of abundance remaining after the oil spill occurs.

included all 80 patches in our analysis to account for potential range expansion.

3.2 Metapopulation dynamics

We restricted our metapopulation model to females because of their importance in regulating population growth and driving population trends (Tinker et al. 2006). We modeled metapopulation dynamics using a stage-structured model with a yearly time step. Beverton-Holt density dependence affected all vital rates based on the abundance of all stages (Akçakaya 2005). We excluded Allee effects from growth functions because it is unlikely that small otter populations are impacted by such effects (Tinker et al. 2008). However, we set the local extinction threshold at 1 female, which is the minimum patch abundance to consider the patch "occupied." We modeled log normal environmental stochasticity and demographic stochasticity with a coefficient of variation of 0.1

(Akçakaya 2005).

We initialized the metapopulation model by setting abundance to 40% of carrying capacity, k for the 21 patches with centroids within the current distribution in accordance with predicted and observed population densities (Gregr et al. 2008); we set abundance to zero for patches outside the current distribution. These 21 initially occupied patches supported an initial metapopulation abundance, $N = 1\,785$ females, had an area of occupancy, $\mathrm{AO} = 3\,514\,\mathrm{km}^2$ as the sum of patch areas, and an extent of occurrence, $\mathrm{EO} = 24\,732\,\mathrm{km}^2$ using the minimum convex polygon method on patch centroids (IUCN 2010; COSEWIC 2010).

In **RAMAS Metapop**, patches of suitable habitat are treated synonymously as spatially-structured populations linked by dispersal and modeled as a metapopulation (Akçakaya 2005). As in **RAMAS Metapop**, we define a *metapopulation* as a set of spatially-structured interacting populations, and a *population* as the individuals in a habitat patch. Sea otters migrate between patches, but movement and dispersal patterns among patches remain poorly quantified in BC. We used data from a California sea otter radio-tracking study (Ralls et al. 1996 in Krkosek et al. 2007) to fit a dispersal distance function

$$m_{ab} \;=\; 0.0524 + 1.2901e^{-0.6D_{ab}} \tag{1}$$

where m_{ab} is the yearly migration rate for juveniles and adults (i.e., proportion of population a) from patch a to patch b, which are separated by centre-to-centre distance D_{ab}. We modeled migration according to Equation 1 when $0 < D_{ab} \leq 100\,\mathrm{km}$; we set m_{ab} to zero for pups, and when $D_{ab} > 100\,\mathrm{km}$ (Garshelis et al. 1984). Using this maximum dispersal distance allowed females to occupy offshore islands such as Haida Gwaii, but prevented females from migrating along the entire coast in one year.

We modeled spatial environmental stochasticity by assuming that spatially proximate patches are subject to more similar environmental conditions (i.e., weather events) than distant patches. For example, two spatially proximate patches may have coincident variability in vital rates (Akçakaya 2005). For our analysis, we used Equation 1 to model the correlation of fecundity, survival, and carrying capacity among patches.

3.3 Oil spill catastrophes

We calculated probabilities for local (e.g., between 0.16 and 16 million litres; l) and regional (e.g., greater than 16 million l) oil spills based on global tanker crude oil spill frequencies between 1985 and 1999 (Anderson and LaBelle 2000), the volume of hydrocarbons transported in BC waters annually (Dickins 1995), and 80 patches. We assume that our calculated oil spill probabilities represent a baseline; conditions may have changed since these data were collected, and may be different for BC waters. For example, spill probabilities may be lower due to short transit times in BC waters, or higher due to more navigational hazards, and an increased volume of transported oil since the data were collected. Also, spill frequencies based on tankers carrying crude oil do not account for spills caused by other vessels such as barges, and other hydrocarbons. We assumed that the occurrence of local (i.e., patch specific) and regional (i.e., metapopulation wide) oil spills in BC is uncorrelated.

Due to lack of data on long-term effects of oil spills on sea otter fitness, we modeled the optimistic situation in which catastrophes only affected abundance the year in which they occurred (i.e., no residual effects). However, sea otters exposed to oil may have reduced reproductive success for more than one generation (Mazet et al. 2001). Our model could easily be updated to incorporate new information on residual effects.

4 Implementing PatchImportance

We assume that users have at least a working ability with the **R** statistical and graphing programme (RDCT 2011), and are familiar with **RAMAS Metapop** version 5.0 software (Akçakaya 2005), both of which must be installed. Users can run **PatchImportance** on non-Windows operating systems provided they install the **WineHQ** programme (WPD 2010), which is required by **RAMAS Metapop**.

4.1 Set-up and user-defined variables

A minimum of three files are required in the working directory: `batch.txt` (Listing 1); the **RAMAS Metapop** input file (e.g., `seaotter.mp`); and the **PatchImportance** code, `PatchImportance.R`. Two additional files must be present to run **RAMAS Metapop** on non-Windows operating systems: `beforeMP`; and `afterMP` (Listing 2). Because **PatchImportance** removes various temporary files and directories from the working directory, additional files or directories in the working directory may be removed inadvertently.

Specify appropriate values for the required user-defined variables before sourcing the **PatchImportance** code, `PatchImportance.R` (Listing 3, Appendix):

`mpFile` **RAMAS Metapop** input file name, with appropriate values and settings for the metapopulation (Akçakaya 2005). Note that patch names must follow the **GRIP** naming convention (e.g., `Pop 1`, `Pop 2`, `Pop 3`, ..., `Pop num`, where `num` is the number of patches satisfying the constraint `num` ≥ 2) to locate specific lines in the input file. Value: character (e.g., `‘‘seaotter.mp’’`).

Listing 1. The `batch.txt` file is used by **PatchImportance** to create **RAMAS Metapop** (Akçakaya 2005) batch files. Note that the third line references the **RAMAS Metapop** executable.

```
START /WAIT
"R_SAM"
"C:\Program Files\RAMASGIS\Metapop.exe"
"
" /RUN=YES /TEX
```

Listing 2. Two one-line files are required to convert end-of-line characters from unix to dos (`beforeMP`; a) and dos to unix (`afterMP`; b) when running **PatchImportance** on non-Windows operating systems using **WineHQ** (WPD 2010).

```
(a)
unix2dos *.mp

(b)
dos2unix *.mp
```

`nYr` Number of years to project the metapopulation, and quantify patch importance. Note that `nYr` over-rides the number of years specified in `mpFile`. Value: integer satisfying the constraint $1 \leq$ `nYr` ≤ 500 (Akçakaya 2005).

`nRep` Number of replications per iteration. Note that `nRep` over-rides the number of replications specified in `mpFile`. Value: integer satisfying the constraint $4 \leq$ `nRep` $\leq 10\,000$ (Akçakaya 2005).

`nIter` Number of iterations. Users must determine the number of iterations required: for example, the minimum number required to achieve consistent patch importance and variability estimates. Value: integer satisfying the constraint `nIter` ≥ 1.

`doSave` Whether output simulation data should be saved to the hard disk. Set to `TRUE` to save output data; set to `FALSE` to skip this step. If `doSave`, users must ensure that sufficient hard disk space is available to save the large number of **RAMAS Metapop** input and output files (Subsection 4.3). Value: logical.

`wineDrive` Drive letter indicating the location of the **RAMAS Metapop** executable. Note that this is only required on non-Windows operating systems using **WineHQ**. Value: character (e.g., `‘‘C:’’`).

The analysis will issue errors and warnings if the required system-dependent files are absent, or if the user-defined variables have values that are outside the aforementioned constraints. Note that the **PatchImportance** code has extensive comments to enhance useability.

4.2 Algorithm outline

The goal of the **PatchImportance** code is to quantify the relative importance of each patch according to the probability of metapopulation persistence, and metapopulation EMA. Generally, the **PatchImportance** algorithm is as follows: (1) identify the most important patch; (2) include the identified patch(es) in the metapopulation, and identify the next most important patch; (3) repeat Step 2 until the least important patch is identified; and (4) repeat Steps 1 to 4 to quantify patch importance variability.

More specifically, the algorithm has an outer loop `q` over 1 : `nIter` iterations, and an inner loop `m` over 1 : `num` patches within each iteration (Figure 2). In the code and in this manual, we use patch numbers 1, 2, 3, ..., `num` to refer to patch names `Pop 1`, `Pop 2`, `Pop 3`, ..., `Pop num`, respectively. The procedure for the inner loop is as follows:

1. Run **RAMAS Metapop** once for each patch separately for `nYr` years and `nRep` replications.
2. Identify the patch that maximizes the probability of metapopulation persistence

   $$\mathrm{P}_{\mathrm{Pers}} = 1 - \mathrm{P}_{\mathrm{Ext}} \qquad (2)$$

   where $\mathrm{P}_{\mathrm{Ext}}$ is the cumulative probability of metapopulation extinction (i.e., zero individuals) over `nYr` years (Akçakaya 2005). However, sometimes multiple patches maximize $\mathrm{P}_{\mathrm{Pers}}$; we resolved these patch importance ties by selecting the patch that maximizes metapopulation EMA over `nYr` years, $\mathrm{N}_{\mathrm{Min}}$ (Akçakaya 2005) from the subset of patches that also maximize $\mathrm{P}_{\mathrm{Pers}}$. Note that **RAMAS Metapop** calculates $\mathrm{N}_{\mathrm{Min}}$ as the mean (over the `nRep` population trajectories) of the minimum metapopulation abundance. We used EMA as a secondary statistic because EMA is a strong predictor of persistence (McCarthy and Thompson 2001). In cases where multiple patches were equally important in terms of both $\mathrm{P}_{\mathrm{Pers}}$ and $\mathrm{N}_{\mathrm{Min}}$, we selected a patch at random from the subset of patches that maximized both $\mathrm{P}_{\mathrm{Pers}}$ and $\mathrm{N}_{\mathrm{Min}}$.[1] Ultimately, this step identifies the most influential patch (i.e., rank 1), and includes this patch in successive **RAMAS Metapop** runs.
3. For each remaining patch, run **RAMAS Metapop** to simulate population dynamics in scenarios that include the new patch, and all patches previously identified as influential.
4. Identify the next most influential patch in the metapopulation using the procedure outlined in Step 2, and include this patch in successive **RAMAS Metapop** runs.
5. Repeat Steps 3 & 4 until the least influential patch is identified (i.e., rank `num`).

Iterate this inner loop `nIter` times to account for the variability in patch ranks due to stochasticity (e.g., environmental, demographic, catastrophic) modeled by **RAMAS Metapop** (`nRep` replications; Akçakaya 2005). Results are written to the $\mathtt{q}^{\mathrm{th}}$ row of three text files at the end of each iteration: patch ranks in `ranks.txt`; $\mathrm{P}_{\mathrm{Pers}}$ in

---

[1] Patch importance ties usually occur when $\mathrm{P}_{\mathrm{Pers}} = 0.00$ or $\mathrm{P}_{\mathrm{Pers}} = 1.00$, and their prevalence may be reduced by using a different `nYr`, quasi-extinction threshold, or both. The analysis counts the number of ties broken using the secondary statistic, and by selecting a patch at random, and prints a message to the **R** console if the number is ≥ 1. Note that the analysis evaluates $(\mathtt{num} - 1) \times \mathtt{nIter}$ ranks.

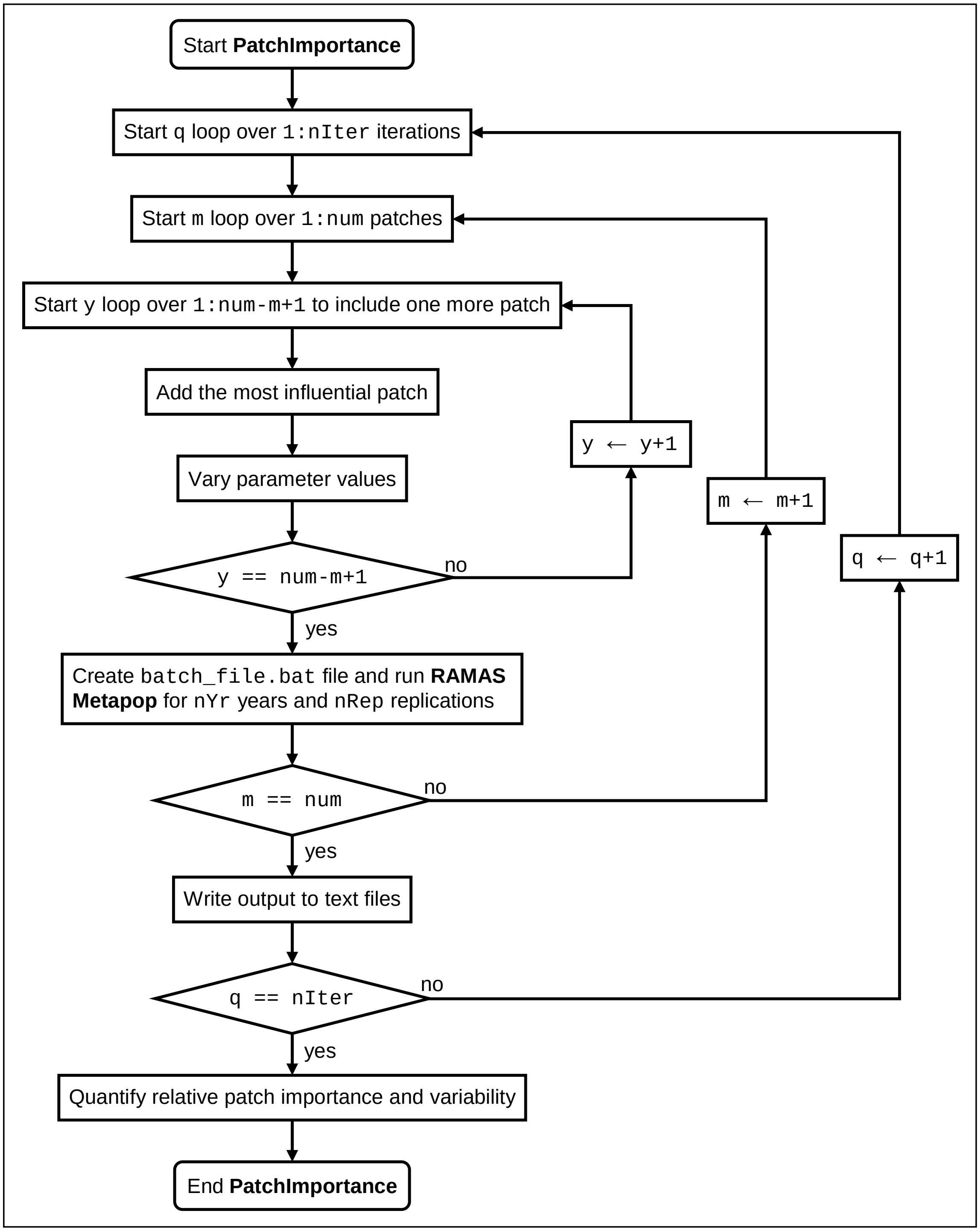


Figure 2. Simplified flow diagram of the **PatchImportance** algorithm which quantifies the relative importance of habitat patches based on their influence on the probability of metapopulation persistence and the metapopulation expected minimum abundance using **RAMAS Metapop** (Akçakaya 2005).

`pProbs.txt`; and $\mathrm{N}_{\mathrm{Min}}$ in `nAbunds.txt`. Because of the two nested loops, the number of times that **RAMAS Metapop** is run, $N_{\mathbf{mp}}$ is a function of `nIter` and `num`

$$N_{\mathbf{mp}} = \texttt{nIter}\left(\frac{\texttt{num}\,(\texttt{num}+1)}{2}\right) \tag{3}$$

which can result in a large number of **RAMAS Metapop** runs. To reduce computation time, users can divide the `nIter` iterations among several processors, and then append the aforementioned text files by row, `q`. The analysis prints a progress message to the **R** console after each iteration to allow users to estimate the required computation time.

4.3 Output

For each iteration, we ranked patches according to $\mathrm{P}_{\mathrm{Pers}}$ and $\mathrm{N}_{\mathrm{Min}}$, and output patch ranks to the file `ranks.txt` as an `nIter` × `num` matrix:

$$\begin{array}{ccccc} r_{1,1} & r_{1,2} & r_{1,3} & \cdots & r_{1,\texttt{num}} \\ r_{2,1} & r_{2,2} & r_{2,3} & \cdots & r_{2,\texttt{num}} \\ r_{3,1} & r_{3,2} & r_{3,3} & \cdots & r_{3,\texttt{num}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_{\texttt{nIter},1} & r_{\texttt{nIter},2} & r_{\texttt{nIter},3} & \cdots & r_{\texttt{nIter},\texttt{num}} \end{array} \tag{4}$$

where $r_{\texttt{q},\texttt{m}}$ identifies the $\texttt{m}^{th}$ most influential patch in the $\texttt{q}^{\mathrm{th}}$ iteration. For example, $r_{3,1} = 2$ indicates that patch `Pop 2` was the most influential patch ($\texttt{m} = 1$) in the third iteration ($\texttt{q} = 3$). Similarly, $\mathrm{P}_{\mathrm{Pers}}$ and $\mathrm{N}_{\mathrm{Min}}$ are output to the files `pProbs.txt` and `nAbunds.txt`, respectively. We used patch ranks to calculate a more intuitive measure of patch influence, which we call relative patch importance

$$i_{\texttt{q},\texttt{m}} = \max(r) - r_{\texttt{q},\texttt{m}} + 1 \tag{5}$$

where $\max(r)$ is the maximum rank (e.g., `num`). Unlike patch ranks, high relative importance values, $i_{\texttt{q},\texttt{m}}$s correspond to influential patches. We rescaled relative patch importance values to range between 0.0 and 1.0 to facilitate interpretation (Li and Wu 2004)

$$z_{\texttt{q},\texttt{m}} = \frac{i_{\texttt{q},\texttt{m}} - \min(i)}{\max(i) - \min(i)} \tag{6}$$

where $\min(i)$ is the minimum relative patch importance value (e.g., 1.0). Henceforth we refer to these rescaled relative patch importance values, z as "relative importance." Compared to the average patch with $z = 0.5$, more influential patches have higher relative importance $0.5 < z \leq 1.0$, while less influential patches have lower relative importance $0.0 \leq z < 0.5$.

We quantified patch importance, z among iterations by the median, and variability by the 50^{th} and 95^{th} percentile ranges. These summary statistics could be used to identify patches that differ significantly in importance, such as patches for which the 95th percentile range does not overlap 0.5. Because we define patch importance by the 95th percentile range, we expect that approximately 5% of patches will be significantly different from 0.5 due to chance alone.

The function `CalcPatchImp(dat1, dat2, dat3)` calculates the aforementioned summary statistics for z, $\mathrm{P}_{\mathrm{Pers}}$, and $\mathrm{N}_{\mathrm{Min}}$, and writes these statistics to the file `RelativeImportance.csv`. The function also displays patch importance statistics in a figure, `PatchImportance.pdf` (Figure 3). Two additional figures are created: the relationship between the number of patches and $\mathrm{P}_{\mathrm{Pers}}$, `Persistence.pdf`; and the relationship between the number of patches and $\mathrm{N}_{\mathrm{Min}}$, `Abundance.pdf` (Figure 4a & b, respectively). Finally, the function returns a list, `patchImp` with four objects: the aforementioned summary statistics in the data frame `patchImp$stats`; z in the matrix `patchImp$imps`; $\mathrm{P}_{\mathrm{Pers}}$ in the matrix `patchImp$probs`; and $\mathrm{N}_{\mathrm{Min}}$ in the matrix `patchImp$abunds`. If desired, users could modify this function to calculate additional statistics and plot additional figures.

Although we have tested the **PatchImportance** code with several **RAMAS Metapop** input files, users must ensure that **PatchImportance** results are meaningful. For example, investigate the **RAMAS Metapop** input and output files (e.g., `rep_y.mp`, and `IntExtRisk_y.txt`, respectively), where `y` indexes the **RAMAS Metapop** run. If specified (e.g., if `doSave`), these files are saved in the directory `DataOutput/output.q.m/`, where `q` indexes the outer loop, and `m` indexes the inner loop.

5 Sea otter habitat patch importance

For our BC sea otter metapopulation case study with 80 patches, we calculated $\mathrm{P}_{\mathrm{Pers}}$ and $\mathrm{N}_{\mathrm{Min}}$ at `nYr` = 100 years, which corresponds to approximately 13 generations (COSEWIC 2007). Using a 100 year timeline was sufficient for metapopulation abundance to stabilize (results not shown). This timeline is also suggested for evaluating the probability of extinction for Canadian species at risk (Criterion E; COSEWIC 2010), which is adapted from the International Union for Conservation of Nature's Red List categories and criteria (IUCN 2010). We also specified `nRep` = 50 replications, and `nIter` = 200 iterations, which was sufficient to stabilize patch ranks and variability.

5.1 Relative patch importance

The 95th percentile range of patch importance for the majority of patches overlaps 0.5, indicating that these patches are not significantly different than the average patch (Figure 3). However, the 95th percentile range for 5 patches lies completely above 0.5, suggesting that these patches may be more important than the average patch. For example, the 95th percentile range for the fourth most important patch, `Pop 70` on the West Coast of Vancouver Island, is completely above 0.5, indicating that `Pop 70` has a

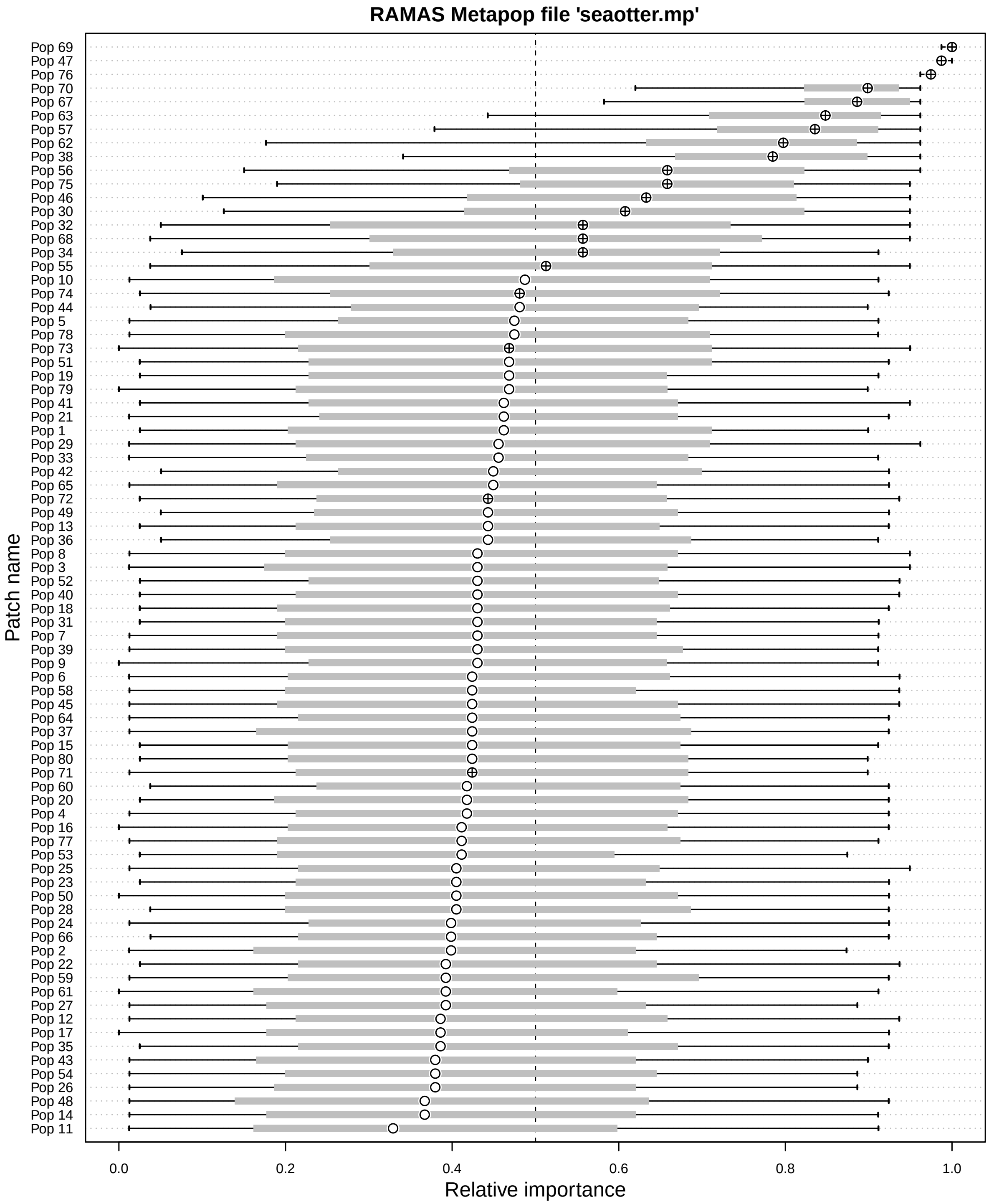
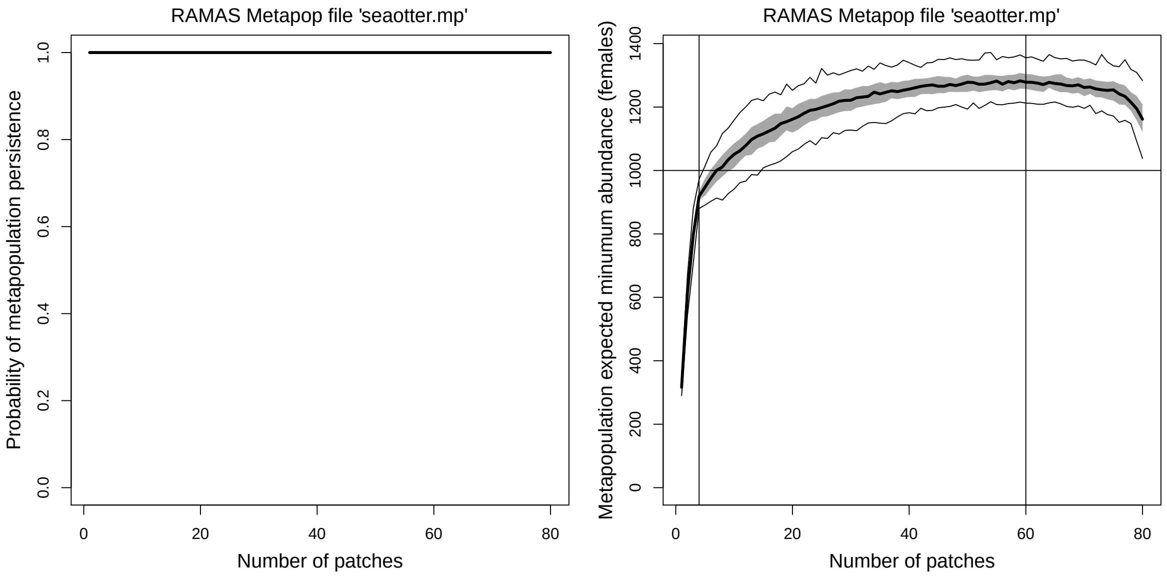


Figure 3. British Columbia sea otter habitat patches ordered by decreasing median relative importance (dots) and 95th percentile range (horizontal lines with ticks), from top to bottom (`nIter` = 200 iterations). Grey rectangles indicate 50th percentile ranges. Note that patch names correspond to patch numbers in Figure 1. Patch symbols: initially occupied, $\oplus$; and initially unoccupied, $\bigcirc$.



(a) Probability of metapopulation persistence, $\mathrm{P}_{\mathrm{Pers}}$. Note that variability (e.g., 50th percentile range) overlaps the median.

(b) Metapopulation expected minimum abundance, $\mathrm{N}_{\mathrm{Min}}$, and values mentioned in the text. Vertical lines: 4 and 60 patches; horizontal line: $\mathrm{N}_{\mathrm{Min}} = 1\,000$ females.

Figure 4. Relationship between the number of habitat patches and the two metapopulation statistics used to quantify relative patch importance for British Columbia sea otters. Thick lines indicate medians, grey polygons indicate 50th percentile ranges, and thin lines indicate 95th percentile ranges (`nIter` = 200 iterations).

stronger influence on the probability of metapopulation persistence (or metapopulation EMA) than the average patch. The high relative importance of `Pop 70` suggests that it might be more valuable to BC sea otters than other patches on the BC coast. These 5 patches, and the 12 other patches with median relative importance $z > 0.5$, were all initially occupied. None of the patches have significantly lower-than-average patch importance values, and patches with lower importance tend to have more variability (e.g., wider 95th percentile range).

5.2 Probability of metapopulation persistence

In our BC sea otter case study, our analysis suggests that the probability of metapopulation persistence is always equal to 1.0, even when only one patch (e.g., the most important patch) is included in the metapopulation (Figure 4a). The secondary statistic (metapopulation EMA) was used to break 99.8% of patch importance ties, and 0.2% of ties were broken by selecting a patch at random. That is to say, none of the sea otter patch ranks in our metapopulation model were determined by solely considering the probability of metapopulation persistence.

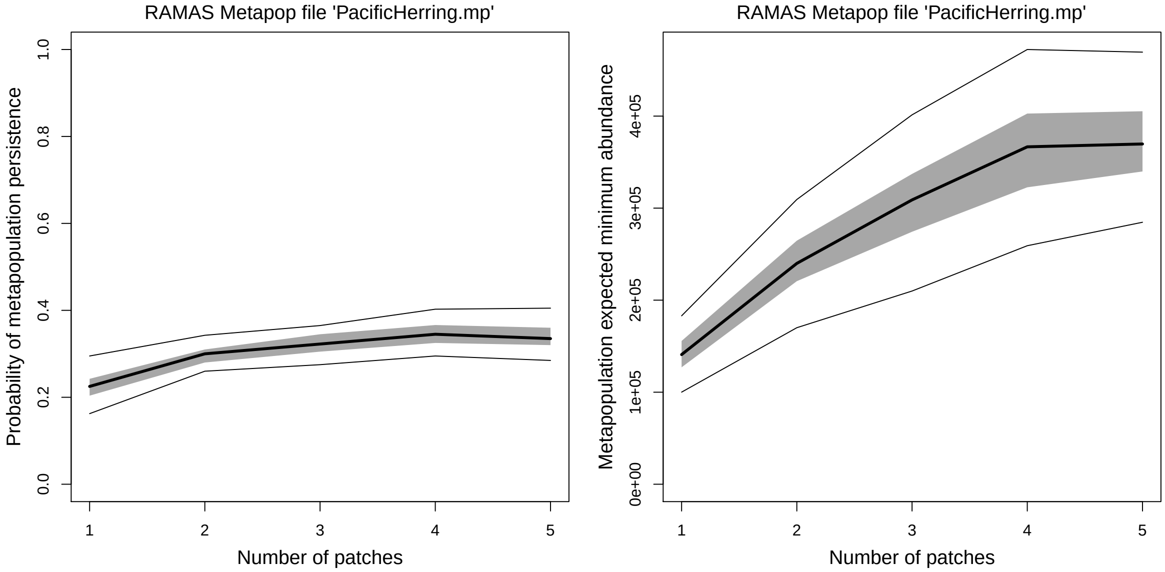
The high probability of persistence is not entirely unexpected because extinction risks for population models tend towards either zero or one over a wide range of parameter values (McCarthy and Thompson 2001). There are at least two explanations for the consistent high probability of BC sea otter metapopulation persistence: (1) model parameters inadvertently over-estimate persistence; or (2) persistence is assured even with very few patches. Of these two explanations, we believe that our BC sea otter metapopulation model may be inadvertently over-estimating persistence because of one or more overly-optimistic parameter values. For example, parameter values based on data collected during a phase of rapid population growth could be overly-optimistic, which could be relevant to this analysis. Also, our Beverton-Holt model of density dependence assumes compensatory dynamics which means that populations tend to increase towards carrying capacity.

Although the probability of BC sea otter metapopulation persistence is always equal to 1.0 in our case study, some example metapopulations have a different relationship. For example, consider the BC herring sample file available with the **RAMAS Metapop** installation, `PacificHerring.mp` (based on Fu et al. 2004). We used **PatchImportance** to quantify the relative importance of the five BC herring patches with the following parameters: `nYr` = 100 years; `nRep` = 200 replications; and `nIter` = 100 iterations (results not shown). Our analysis suggests that the median probability of BC herring metapopulation persistence is low, but increases from 0.225 to 0.335 as more patches are included in the metapopulation (Figure 5a). We also show the relationship between the number of patches and median metapopulation EMA, which increases from 140 868 to 369 727 herring as more patches are included in the metapopulation (Figure 5b).

5.3 Metapopulation expected minimum abundance

The relationship between BC sea otter metapopulation EMA and the number of patches can be broken up into three zones based on the number of patches in the metapopulation: between 1 and 4 patches; between 5 and 60 patches; and between 61 and 80 patches (Figure 4b). Initially, median metapopulation EMA is low, but increases rapidly to approximately 920 females as important patches are added to the small metapopulation. This zone of rapid increase in median metapopulation EMA is followed by a zone of diminishing marginal gains and wide variability as the number of patches approaches 60. The addition of these less important patches causes median metapopulation EMA to increase to about 1 280 females. Finally, median metapopulation EMA declines slightly to about 1 160 females as the last few patches are added to the metapopulation.

The presence of patches which reduce metapopulation EMA may indicate that there are sink populations, which have negative population growth rates (e.g., deaths exceed births). Sink populations are in contrast to source populations, which have positive population growth rates. Although population sinks may reduce abundance, their existence is not necessarily detrimental to metapopulations because sinks can increase connectivity between source populations, or buffer against catastrophes (Akçakaya et al. 2007). Additionally, patches that currently act as sinks due to the prevalence of low quality



(a) Probability of metapopulation persistence, P_{Pers}.

(b) Metapopulation expected minimum abundance, N_{Min}.

Figure 5. Relationship between the number of habitat patches and the two metapopulation statistics used to quantify relative patch importance for British Columbia herring. Thick lines indicate medians, grey polygons indicate 50th percentile ranges, and thin lines indicate 95th percentile ranges (`nIter` = 100 iterations).

habitat could become sources in the future if environmental conditions change (e.g., climate change). Using metapopulation models to identify valuable and less valuable (e.g., sink) patches may be a step towards identifying ecological traps, which are preferentially selected low-quality patches (Dwernychuk and Boag 1972). In contrast to population sinks, identifying population traps is a conservation concern because their presence can lead to metapopulation extinction (Battin 2004).

5.4 Spatial autocorrelation of important patches

Our analysis suggests that median patch importance values exhibit significant positive spatial autocorrelation, indicating that sea otter patches of similar importance are somewhat clustered (Moran's $I = 0.233$, $p < 0.001$; Bivand 2011; Figure 6). Important patches are clustered in the current distribution on the West Coast of Vancouver Island, the Goose Islands, and Aristazabal Island. Important patches may be underrepresented on Haida Gwaii in part because the habitat suitability map did not capture habitat characteristics in this area, despite likely supporting a high otter abundance in the past (Gregr et al. 2008). Additionally, otters migrating from initially occupied patches to Haida Gwaii would have had to travel through several intermediate patches which would delay their occupation; the only route to Haida Gwaii from the mainland

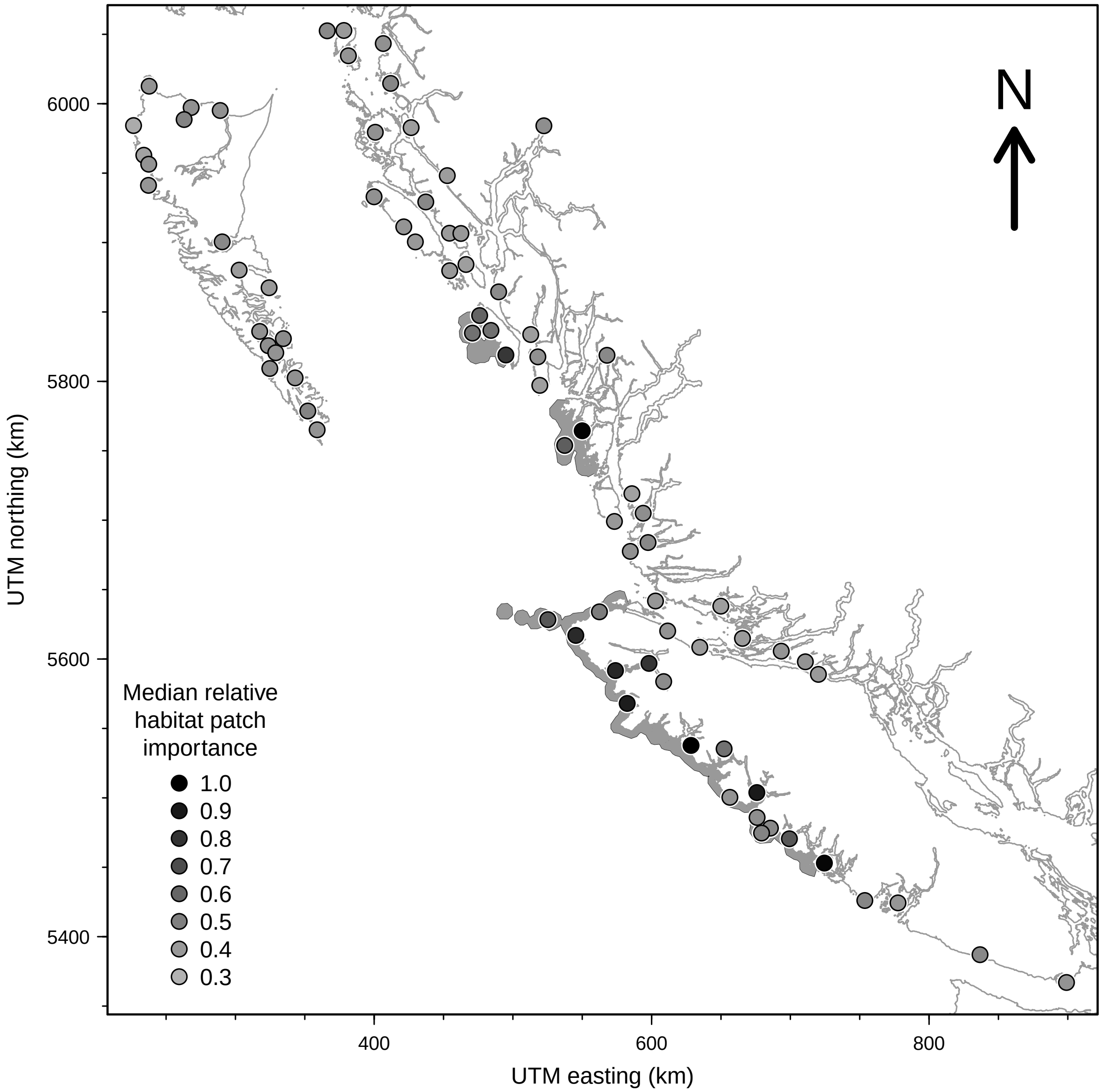


Figure 6. Median relative importance of British Columbia sea otter habitat patches (`nIter` = 200 iterations). Geographic coordinates are projected in Universal Transverse Mercator (UTM; zone 9), in kilometres (km). The grey polygon indicates the current distribution (e.g., area encompassing patches with non-zero initial abundance; Nichol et al. 2009).

is via `Pop 1` to `Pop 7`.

Spatial planning initiatives attempting to increase the probability of BC sea otter metapopulation persistence and metapopulation EMA may maximize their impact by protecting the most important patches, or areas with clusters of important patches. Coast-wide spatial planning initiatives for BC sea otters may also benefit by using habitat suitability maps that capture habitat characteristics on Haida Gwaii. Other factors to guide the selection of protected area size and shape include: the ability to monitor and enforce regulations (McLeod et al. 2009); ecological interactions (e.g., prey dynamics); and social, political, and economic constraints (Akçakaya et al. 2007).

Generally, patches with non-zero initial abundance tend to be more important for sea otters than patches that are initially unoccupied (Table 2). For example, the 17 most important patches were initially occupied; for these patches, larger patches (i.e., larger AO, which supports a higher abundance) tend to be more important than smaller patches. Importance values for the remaining 63 patches do not appear to follow an obvious trend with respect to the calculated metapopulation statistics.

Table 2. British Columbia sea otter hatitat patches ordered by decreasing median relative importance, z, and then by the 95th percentile range of relative importance (not shown), from top to bottom (`nIter = 200` iterations). Also indicated is whether the patch was initially occupied, $N_0 > 0$. Metapopulation statistics: female metapopulation abundance, N; area of occupancy (AO) as patch area; number of patches (NP); and extent of occurrence (EO) as minimum convex polygon. Units: AO and EO are in square kilometres (km^2). Note that N assumes that every patch is at carrying capacity, $k = 1.27$ females $\cdot$ km^{-2} (Gregr et al. 2008).

					Cumulative			
Patch	$N_0 > 0$	z	N	AO	NP	N	AO	EO
Pop 69	Yes	1.000	1208	952	1	1 208	952	952
Pop 47	Yes	0.987	1073	845	2	2 282	1 796	1 796
Pop 76	Yes	0.975	722	569	3	3 004	2 365	7 569
Pop 70	Yes	0.899	227	179	4	3 231	2 544	7 569
Pop 67	Yes	0.886	224	176	5	3 455	2 720	12 094
Pop 63	Yes	0.848	167	131	6	3 622	2 852	12 540
Pop 57	Yes	0.835	162	127	7	3 783	2 979	14 936
Pop 62	Yes	0.797	113	89	8	3 897	3 068	14 936
Pop 38	Yes	0.785	151	119	9	4 048	3 188	19 118
Pop 56	Yes	0.658	68	54	10	4 116	3 241	21 147
Pop 75	Yes	0.658	78	61	11	4 194	3 302	21 351
Pop 46	Yes	0.633	73	57	12	4 267	3 360	21 351
Pop 30	Yes	0.608	57	45	13	4 324	3 404	22 989
Pop 32	Yes	0.557	36	28	14	4 359	3 432	22 989
Pop 68	Yes	0.557	33	26	15	4 393	3 459	22 989
Pop 34	Yes	0.557	28	22	16	4 420	3 480	23 871
Pop 55	Yes	0.513	6	5	17	4 427	3 486	23 871
Pop 10	No	0.487	10	8	18	4 437	3 494	42 849
Pop 74	Yes	0.481	8	6	19	4 445	3 500	43 711
Pop 44	No	0.481	179	141	20	4 624	3 641	56 644

Table 2 (*continued*)

					Cumulative			
Patch	$N_0 > 0$	z	N	AO	NP	N	AO	EO
Pop 5	No	0.475	398	313	21	5 022	3 954	73 779
Pop 78	No	0.475	113	89	22	5 135	4 043	78 047
Pop 73	Yes	0.468	14	11	23	5 149	4 054	78 047
Pop 51	No	0.468	264	208	24	5 413	4 262	78 047
Pop 19	No	0.468	173	136	25	5 586	4 399	78 047
Pop 79	No	0.468	10	8	26	5 597	4 407	95 817
Pop 41	No	0.462	54	43	27	5 651	4 450	103 154
Pop 21	No	0.462	151	119	28	5 802	4 568	104 203
Pop 1	No	0.462	17	14	29	5 819	4 582	107 619
Pop 29	No	0.456	43	34	30	5 862	4 616	107 619
Pop 33	No	0.456	97	76	31	5 959	4 692	107 619
Pop 42	No	0.449	21	16	32	5 980	4 709	108 344
Pop 65	No	0.449	10	8	33	5 990	4 716	108 344
Pop 72	Yes	0.443	4	4	34	5 994	4 720	108 344
Pop 49	No	0.443	193	152	35	6 187	4 872	108 344
Pop 13	No	0.443	49	38	36	6 236	4 910	130 582
Pop 36	No	0.443	7	5	37	6 243	4 916	130 582
Pop 8	No	0.430	9	7	38	6 252	4 922	130 866
Pop 3	No	0.430	117	92	39	6 368	5 014	131 532
Pop 52	No	0.430	6	4	40	6 374	5 019	131 532
Pop 40	No	0.430	150	118	41	6 524	5 137	131 532
Pop 18	No	0.430	271	213	42	6 794	5 350	131 532
Pop 31	No	0.430	3	2	43	6 797	5 352	131 532
Pop 7	No	0.430	105	83	44	6 903	5 435	131 532
Pop 39	No	0.430	135	106	45	7 038	5 542	131 532
Pop 9	No	0.430	1467	1155	46	8 505	6 697	131 532
Pop 6	No	0.424	103	81	47	8 607	6 778	134 550
Pop 45	No	0.424	46	36	48	8 653	6 814	135 828
Pop 58	No	0.424	76	60	49	8 729	6 873	135 828
Pop 37	No	0.424	29	23	50	8 758	6 896	135 828
Pop 64	No	0.424	2	1	51	8 760	6 897	135 828
Pop 15	No	0.424	43	34	52	8 803	6 932	138 302
Pop 80	No	0.424	88	69	53	8 891	7 000	153 759
Pop 71	Yes	0.424	10	8	54	8 900	7 008	153 759
Pop 60	No	0.418	101	80	55	9 001	7 088	153 759
Pop 20	No	0.418	50	39	56	9 051	7 127	153 759
Pop 4	No	0.418	375	295	57	9 426	7 422	153 759
Pop 16	No	0.411	26	20	58	9 452	7 442	154 438
Pop 77	No	0.411	682	537	59	10 134	7 980	154 438

Table 2 (*continued*)

					Cumulative			
Patch	$N_0 > 0$	z	N	AO	NP	N	AO	EO
Pop 53	No	0.411	936	737	60	11 070	8 716	154 438
Pop 25	No	0.405	350	276	61	11 420	8 992	154 438
Pop 23	No	0.405	1	1	62	11 421	8 993	154 438
Pop 50	No	0.405	26	21	63	11 448	9 014	154 438
Pop 28	No	0.405	27	22	64	11 475	9 036	154 438
Pop 24	No	0.399	17	14	65	11 492	9 049	154 438
Pop 66	No	0.399	107	84	66	11 600	9 134	154 438
Pop 2	No	0.399	30	23	67	11 629	9 157	154 498
Pop 22	No	0.392	33	26	68	11 662	9 183	154 498
Pop 59	No	0.392	14	11	69	11 676	9 194	154 498
Pop 61	No	0.392	781	615	70	12 457	9 809	154 498
Pop 27	No	0.392	4	4	71	12 462	9 812	154 498
Pop 12	No	0.386	7	6	72	12 469	9 818	154 498
Pop 17	No	0.386	14	11	73	12 483	9 829	154 498
Pop 35	No	0.386	47	37	74	12 530	9 866	154 498
Pop 43	No	0.380	80	63	75	12 610	9 929	154 498
Pop 26	No	0.380	95	75	76	12 704	10 004	154 498
Pop 54	No	0.380	217	171	77	12 921	10 174	154 498
Pop 48	No	0.367	72	56	78	12 993	10 231	154 498
Pop 14	No	0.367	1	1	79	12 994	10 232	154 624
Pop 11	No	0.329	20	16	80	13 014	10 247	154 893

5.5 Setting conservation targets

Patch importance values could be used to identify the number and location of patches required to achieve a target metapopulation statistic. For example, common metapopulation statistics mentioned in species at risk recovery documents include: metapopulation abundance, N of either all individuals, or mature individuals; AO; number of patches; and EO (IUCN 2010; COSEWIC 2010). Patch importance values could help set recovery targets for species at risk: for example, consider the hypothetical recovery target of protecting the habitat required to support a theoretical maximum of $N = 5\,000$ female otters.[2] According to our analysis, protecting the 21 most important patches (i.e., `Pop 69`, `Pop 47`, `Pop 76`, ..., `Pop 5`) meets this target with cumulative $N = 5\,022$ females when populations are at carrying capacity $k = 1.27\,\text{females} \cdot \text{km}^{-2}$

---

[2] For consistency, we continue to quantify metapopulation abundance in terms of female otters. Also note that adult (i.e., mature) otters generally account for approximately 50% of total abundance (COSEWIC 2007)

(Gregr et al. 2008), and is associated with cumulative AO = $3\,954\,\mathrm{km}^2$ and cumulative EO = $73\,779\,\mathrm{km}^2$.

Regarding the possibly under-represented important patches on Haida Gwaii suggested by our analysis, `Pop 10` is the most important of the 19 patches on Haida Gwaii, but `Pop 10` is the 18th most important patch in BC. Also, `Pop 10` is the most important patch of the subset of patches that was not initially occupied. The inclusion of `Pop 10` in the metapopulation, which is small (e.g., AO = $8\,\mathrm{km}^2$) and has intermediate relative importance, causes EO to increase from 23 871 to $42\,849\,\mathrm{km}^2$. Two other small patches of relatively low importance cause EO to increase substantially: the 36th most important patch, `Pop 13` (EO from 108 344 to $130\,582\,\mathrm{km}^2$); and the 53rd most important patch, `Pop 80` (EO from 138 302 to $153\,759\,\mathrm{km}^2$). Thus, including these three small patches of relatively low importance in the metapopulation has a strong influence on EO due to their remote geographic location. Additionally, remote patches may help establish multiple locations, which help minimize impacts due to threats (Criteria B & D; IUCN 2010). In this context, a *location* is a geographically distinct area in which every individual could be impacted by a single threatening event, such as a catastrophic oil spill.

Consider a second application in which patch importance values help identify the number and location of patches required to achieve a target metapopulation EMA. For example, our analysis suggests that 18 patches are required to achieve a metapopulation EMA ≥ 750 female otters (Table 3). A few patches were required every iteration (e.g., `Pop 69`), but most patches were only required in a minority of the iterations (e.g., 3% of iterations for `Pop 67`). At carrying capacity, these 18 patches could support up to $N = 4\,819$ female otters, with cumulative AO = $3\,794\,\mathrm{km}^2$, and cumulative EO = $46\,479\,\mathrm{km}^2$.

6 Extensions

We demonstrated our analysis by ranking the relative importance of BC sea otter patches based on their influence on $\mathrm{P_{Pers}}$, and resolved ties by considering their influence on $\mathrm{N_{Min}}$. However, the **PatchImportance** code could be customized to suit individual requirements by considering alternative statistics, modifying our **R** code, and by incorporating other conservation tools.

6.1 Alternative statistics and modifications

The **PatchImportance** code could quantify patch importance according to alternative statistics, such as N, AO, EO, percent decline, probability of quasi-extinction (e.g., quasi-extinction threshold = 100 individuals), or time to extinction (Akçakaya 2005). The code could be modified to consider additional statistics, which could break patch importance ties instead of selecting a patch at random. Users could incorporate the aforementioned changes in the `statMat` matrix. It is noteworthy that patch importance

Table 3. Patches required for metapopulation expected minimum abundance ≥ 750 female British Columbia sea otters. Patches are ordered from top to bottom by the percentage of the `nIter` $= 200$ iterations that required the patch, $I_{\%}$, and then by median relative importance (not shown). Also indicated is whether the patch was initially occupied, $N_0 > 0$. Metapopulation statistics: female metapopulation abundance, N; area of occupancy (AO) as patch area; number of patches (NP); and extent of occurrence (EO) as minimum convex polygon. Units: AO and EO are in square kilometres (km^2). Note that N assumes that every patch is at carrying capacity, $k = 1.27\,\text{females} \cdot \text{km}^{-2}$ (Gregr et al. 2008).

					Cumulative			
Patch	$N_0 > 0$	$I_{\%}$	N	AO	NP	N	AO	EO
Pop 69	Yes	100	1208	952	1	1 208	952	952
Pop 47	Yes	100	1073	845	2	2 282	1 796	1 796
Pop 76	Yes	100	722	569	3	3 004	2 365	7 569
Pop 67	Yes	3	224	176	4	3 228	2 542	12 094
Pop 70	Yes	3	227	179	5	3 455	2 720	12 094
Pop 62	Yes	2	113	89	6	3 568	2 810	12 094
Pop 57	Yes	1	162	127	7	3 730	2 937	14 936
Pop 38	Yes	1	151	119	8	3 881	3 056	19 118
Pop 75	Yes	1	78	61	9	3 959	3 118	19 322
Pop 74	Yes	1	8	6	10	3 967	3 124	20 184
Pop 8	No	1	9	7	11	3 976	3 131	38 603
Pop 46	Yes	1	73	57	12	4 049	3 188	38 603
Pop 68	Yes	1	33	26	13	4 082	3 214	38 603
Pop 29	No	1	43	34	14	4 126	3 249	44 407
Pop 49	No	1	193	152	15	4 319	3 401	46 479
Pop 39	No	1	135	106	16	4 454	3 507	46 479
Pop 25	No	1	350	276	17	4 804	3 783	46 479
Pop 59	No	1	14	11	18	4 819	3 794	46 479

values may vary according to the statistics used to measure importance, the metapopulation dynamics parameter values, as well as the spatial scale and extent of the habitat suitability map.

A number of other potential modifications are possible, and we mention three of them here. First, users could quantify predictors (e.g., patch size, patch isolation) associated with important patches to develop predictions based on patch attributes (Vögeli et al. 2010). Quantifying predictor importance could also help identify thresholds (Fahrig 2001; Fagan et al. 2003) which could inform critical habitat identification, or other spatial planning initiatives.

Second, our code could be modified to incorporate prior knowledge regarding the ability or desire to protect specific patches. For example, some patches may be harder to protect due to their proximity to urban areas, while other patches may be easier to protect due to their proximity to existing protected areas (e.g., via expansion). Alternatively, the presence of multiple species at risk in some patches may support their inclusion in protected areas (e.g., key biodiversity areas; Langhammer et al. 2007).

Finally, the **PatchImportance** code could be used to evaluate the relative influence of patch quality by modifying patch characteristics (e.g., carrying capacity, links between habitat attributes, relative fecundity, relative survival). For example, consider a situation in which individuals in protected patches benefit from increased fitness (e.g., greater habitat quality leads to greater fecundity) compared to unprotected patches. For this approach, the **PatchImportance** algorithm could be adjusted to include every patch in the metapopulation, and iteratively increase habitat quality one patch at a time. Important patches would maintain their artificially increased fitness benefits to simulate their inclusion in a network of protected areas.

6.2 Incorporating additional conservation tools

The **PatchImportance** code could be used in conjunction with other conservation tools, such as **NetworkDistances** (Grinnell and Curtis 2011), and **Conefor Sensinode** (Saura and Torné 2009) to address some of the simplifying assumptions in **RAMAS**. As one example, **RAMAS Metapop** models that employ dispersal distance functions typically ignore the influence of barriers and other landscape attributes that influence dispersal rates. However, the effective distance between patches may not fall on a straight line for some species. Because migration rates and connectivity may influence patch importance values, it is critical to measure accurate distances among patches. Spatially restricted species, such as lotic fish, may be required to travel further than the Euclidean (i.e., straight line) distance between patches. In these cases, Euclidean distances may under-estimate effective inter-patch distances, which may affect simulated patch dynamics (Johst et al. 2002). Previously, we developed the **NetworkDistances** code to measure non-Euclidean inter-patch distances (Grinnell and Curtis 2011), which could modify the `Pairwise_distance` matrix. Because patch centroids remain constant among iterations, only one instance of **NetworkDistances** would be required to parameterize the **RAMAS Metapop** input file. Users could merge these two tools to quantify patch importance for spatially restricted species.

Second, users could quantify patch importance based on connectivity statistics using the **Conefor Sensinode** software (Saura and Torné 2009). Briefly, **Conefor Sensinode** quantifies patch importance based on landscape connectivity indices using graph structures to represent patches and connections between patches (i.e., corridors). For example, quantifying importance based on connectivity might identify patches `Pop 1`, and `Pop 7` as important because they are critical to populating Haida Gwaii. Because **Conefor Sensinode** is fast to implement, **PatchImportance** could run **Conefor Sensinode** via a batch file to calculate connectivity statistics, which might resolve patch importance ties.

7 Conclusions

Identifying the most important habitat patches using a quantitative statistic is critical to designing protected areas when it is not feasible to protect the entire area of occupancy. We demonstrated our simulation approach to ranking patches according to their impact on metapopulation persistence and minimum abundance using the BC sea otter metapopulation as a case study. In this context, the inclusion of important patches causes the probability of metapopulation persistence (or metapopulation EMA) to increase more than the inclusion of less important patches. Our **PatchImportance** tool could be applied to BC sea otters and other species to provide science-based advice for spatial planning initiatives in BC. Further, users could modify our code to provide science-based advice on related conservation questions, and couple **PatchImportance** with other existing software.

8 Acknowledgements

The authors thank K. Kinnersley for facilitating their use of the High Performance Computing cluster at the Institute of Ocean Sciences, as well as I. R. Naujokaitis-Lewis and B. Davis for technical support. Funding support was provided by Fisheries and Oceans Canada's National Species at Risk Programme.

References

Akçakaya, H. R. 2005. RAMAS GIS: Linking spatial data with population viability analysis. Applied Biomathematics. URL `www.ramas.com/ramas.htm`. User manual for version 5

Akçakaya, H. R., Mills, G. and Doncaster, C. P. 2007. The role of metapopulations in conservation. In D. W. Macdonald and K. Service, eds., Key topics in conservation biology, chap. 5, 64–84. Blackwell Publishing. ISBN:978-1-4051-2249-8

Anderson, C. M. and LaBelle, R. P. 2000. Update of comparative occurrence rates for offshore oil spills. *Spill Science and Technology Bulletin* **6**(5/6): 303–321. doi: 10.1016/S1353-2561(01)00049-4

Battin, J. 2004. When good animals love bad habitats: Ecological traps and the conservation of animal populations. *Conservation Biology* **18**(6): 1482–1491. doi: 10.1111/j.1523-1739.2004.00417.x

Bivand, R. S. 2011. spdep: Spatial dependence: Weighting schemes, statistics and models. URL `http://CRAN.R-project.org/package=spdep`. With contributions by M. Altman, L. Anselin, R Assunção, O. Berke, A. Bernat, G. Blanchet, E. Blankmeyer, M. Carvalho, B. Christensen, Y. Chun, C. Dormann, S. Dray, R. Halbersma, E. Krainski, P. Legendre, N. Lewin-Koh, H. Li, J. Ma, G. Millo, W. Mueller, H. Ono, P. Peres-Neto, G. Piras, M. Reder, M. Tiefelsdorf, and D. Yu. R package version 0.5-43

Clarke, C. L. and Jamieson, G. S. 2006*a*. Identification of ecologically and biologically significant areas in the Pacific North Coast Integrated Management Area: Phase I – Identification of important areas. Canadian Technical Report of Fisheries and Aquatic Sciences 2678, Fisheries and Oceans Canada. URL `www.dfo-mpo.gc.ca/libraries-bibliotheques/tech-eng.htm`

Clarke, C. L. and Jamieson, G. S. 2006*b*. Identification of ecologically and biologically significant areas in the Pacific North Coast Integrated Management Area: Phase II – Final report. Canadian Technical Report of Fisheries and Aquatic Sciences 2686, Fisheries and Oceans Canada. URL `www.dfo-mpo.gc.ca/libraries-bibliotheques/tech-eng.htm`

COSEWIC (Committee on the Status of Endangered Wildlife in Canada). 2007. COSEWIC assessment and update status report on the sea otter *Enhydra lutris* in Canada. Technical Report, Canadian Wildlife Service and Environment Canada. URL `www.sararegistry.gc.ca/status`

COSEWIC (Committee on the Status of Endangered Wildlife in Canada). 2010. COSEWIC's assessment process and criteria. URL `www.cosewic.gc.ca`. Approved April 2010. Retrieved 1 March 2011

Curtis, J. M. R. and Naujokaitis-Lewis, I. R. 2008*a*. Sensitivity of population viability analysis to spatial and nonspatial parameters using GRIP. *Ecological Applications* **18**(4): 1002–1013. doi:10.1111/j.1523-1739.2008.01066.x

Curtis, J. M. R. and Naujokaitis-Lewis, I. R. 2008*b*. Source code for the program GRIP 1.0 (Generation of Random Input Parameters). URL `http://esapubs.org/archive/appl/A018/033/suppl-1.htm`. Ecological Archives: A018-033-S1 (Supplement)

Diamond, J. M. 1975. The island dilemma: Lessons of modern biogeographic studies for the design of nature reserves. *Biological Conservation* **7**(2): 129–146. doi:10.1016/0006-3207(75)90052-X

Dickins, D. F. 1995. The double hull issue and oil spill risk on the Pacific West coast. Technical report, Ministry of Environment, Lands and Parks. URL `www.env.gov.bc.ca`

Dudley, N., ed. 2008. Guidelines for applying protected area management categories. International Union for Conservation of Nature, Gland, Switzerland. URL `www.iucn.org/about/union/commissions/wcpa/`

Dwernychuk, L. W. and Boag, D. A. 1972. Ducks nesting in association with gulls—An ecological trap? *Canadian Journal of Zoology* **50**(5): 559–563. doi:10.1139/z72-076

Fagan, W. F., Fortin, M.-J. and Soykan, C. 2003. Integrating edge detection and dynamic modeling in quantitative analyses of ecological boundaries. *BioScience* **53**(8): 730–738. doi:10.1641/0006-3568(2003)053[0730:IEDADM]2.0.CO;2

Fahrig, L. 2001. How much habitat is enough? *Biological Conservation* **100**(1): 65–74. doi:10.1016/S0006-3207(00)00208-1

Fu, C., Wood, C. C. and Schweigert, J. 2004. Pacific herring (*Clupea pallasi*) in Canada: Generic framework for evaluating conservation limits and harvest strategies. In H. R. Akçakaya, M. A. Burgman, O. Kindvall, C. C. Wood, P. Sjögren-Gulve, J. Hatfield and M. A. McCarthy, eds., Species conservation and management: Case studies. Oxford University Press. ISBN:0-19-516646-9

Garshelis, D. L., Johnson, A. M. and Garshelis, J. A. 1984. Social organization of sea otters in Prince William Sound, Alaska. *Canadian Journal of Zoology* **62**(12): 2648–2658. doi:10.1139/z84-385

Gerber, L. R., Buenau, K. E. and VanBlaricom, G. 2004. Density dependence and risk of extinction in a small population of sea otters. *Biodiversity and Conservation* **13**: 2741–2757. doi:10.1007/s10531-004-2146-1

Gregr, E. J., Nichol, L. M., Watson, J. C., Ford, J. K. B. and Ellis, G. M. 2008. Estimating carrying capacity for sea otters in British Columbia. *Journal of Wildlife Management* **72**(2): 382–388. doi:10.2193/2006-518

Grinnell, M. H. and Curtis, J. M. R. 2011. User manual for NetworkDistances 1.0: Calculating network-wise distances between habitat patches for spatially restricted species. Canadian Technical Report of Fisheries and Aquatic Sciences 2960, Fisheries and Oceans Canada. URL `www.dfo-mpo.gc.ca/libraries-bibliotheques/tech-eng.htm`

Hatfield, T. 2009. Identification of critical habitat for sympatric stickleback species pairs and the Misty Lake parapatric stickleback species pair. Research Document 2009/056, Canadian Science Advisory Secretariat, Fisheries and Oceans Canada. URL `www.dfo-mpo.gc.ca/csas`

IUCN (International Union for the Conservation of Nature). 2010. Guidelines for using the IUCN Red List categories and criteria. Version 8.1, IUCN Species Survival Commission. URL `www.iucnredlist.org`. Prepared by the Standards and Petitions Subcommittee in March 2010

Johst, K., Brandl, R. and Eber, S. 2002. Metapopulation persistence in dynamic landscapes: The role of dispersal distance. *Oikos* **98**(2): 263–270. doi:10.1034/j.1600-0706.2002.980208.x

Jordán, F., Báldi, A., Orci, K.-M., Rácz, I. and Varga, Z. 2003. Characterizing the importance of habitat patches and corridors in maintaining the landscape connectivity of a *Pholidoptera transsylvanica* (Orthoptera) metapopulation. *Landscape Ecology* **18**(1): 83–92. doi:10.1023/A:1022958003528

Krkosek, M., Lauzon-Guay, J. and Lewis, M. A. 2007. Relating dispersal and range expansion of California sea otters. *Theoretical Population Biology* **71**(4): 401–407. doi:10.1016/j.tpb.2007.01.008

Langhammer, P. F., Bakarr, M. I., Bennun, L. A., Brooks, T. M., Clay, R. P., Darwall, W., De Silva, N., Edgar, G. J., Eken, G., Fishpool, L. D. C., da Fonseca, G. A. B., Foster, M. N., Knox, D. H., Matiku, P., Radford, E. A., Rodrigues, A. S. L., Salaman, P., Sechrest, W. and Tordoff, A. W. 2007. Identification and gap analysis of key biodiversity areas: Targets for comprehensive protected area systems. Best Practices Protected Area Guidelines Series 15, International Union for the Conservation of Nature. URL `www.iucn.org/publications`. Gland, Switzerland

Li, H. and Wu, J. 2004. Use and misuse of landscape indices. *Landscape Ecology* **19**(4): 389–399. doi:10.1023/B:LAND.0000030441.15628.d6

Loughlin, T. R. 1980. Home range and territoriality of sea otters near Monterey, California. *The Journal of Wildlife Management* **44**(3): 576–582. URL `www.jstor.org/stable/3808005`

Mazet, J. A. K., Gardner, I. A., Jessup, D. A. and Lowenstine, L. J. 2001. Effects of petroleum on mink applied as a model for reproductive success in sea otters. *Journal of Wildlife Diseases* **37**(4): 686–692. URL `www.jwildlifedis.org/content/37/4/686.abstract`

McCarthy, M. A. and Thompson, C. 2001. Expected minimum population size as a measure of threat. *Animal Conservation* **4**(4): 351–355. doi:10.1017/S136794300100141X

McLeod, E., Salm, R., Green, A. and Almany, J. 2009. Designing marine protected area networks to address the impacts of climate change. *Frontiers in Ecology and the Environment* **7**(7): 362–370. doi:10.1890/070211

Nichol, L. 2007. Recovery potential assesment for sea otters (*Enhydra lutris*) in Canada. Research Document 2007/034, Canadian Science Advisory Secretariat, Fisheries and Oceans Canada. URL `www.dfo-mpo.gc.ca/csas`

Nichol, L. M., Boogaards, M. D. and Abernethy, R. 2009. Recent trends in the abundance and distribution of sea otters (*Enhydra lutris*) in British Columbia. Research Document 2009/016, Canadian Science Advisory Secretariat, Fisheries and Oceans Canada. URL `www.dfo-mpo.gc.ca/csas`

Pascual-Hortal, L. and Saura, S. 2006. Comparison and development of new graph-based landscape connectivity indices: Towards the priorization of habitat patches and corridors for conservation. *Landscape Ecology* **21**(7): 959–967. doi:10.1007/s10980-006-0013-z

Ralls, K., Eagle, T. C. and Siniff, D. B. 1996. Movement and spatial use patterns of California sea otters. *Canadian Journal of Zoology* **74**: 1841–1849. doi:10.1139/z96-207

RDCT (R Development Core Team). 2011. R: A language and environment for statistical computing. URL `www.R-project.org`. R Foundation for Statistical Computing. Vienna, Austria. R version 2.14.0

Rosenfeld, J. S. and Hatfield, T. 2006. Information needs for assessing critical habitat of freshwater fish. *Canadian Journal of Fisheries and Aquatic Sciences* **63**(3): 683–698. doi:10.1139/f05-242

SARA (Species at Risk Act). 2011. The Species at Risk Act: An Act respecting the protection of wildlife species at risk in Canada. URL `www.sararegistry.gc.ca`. Canada Gazette Part III, Vol. 25, No. 3. Retrieved 30 November 2011

Saura, S. and Torné, J. 2009. Conefor Sensinode 2.2: A software package for quantifying the importance of habitat patches for landscape connectivity. *Environmental Modelling and Software* **24**(1): 135–139. doi:10.1016/j.envsoft.2008.05.005

Tinker, M. T., Doak, D. F. and Estes, J. A. 2008. Using demography and movement behavior to predict range expansion of the southern sea otter. *Ecological Applications* **18**(7): 1781–1794. URL `www.jstor.org/stable/40062251`

Tinker, M. T., Doak, D. F., Estes, J. A., Hatfield, B. B., Staedler, M. M. and Bodkin, J. L. 2006. Incorporating diverse data and realistic complexity into demographic estimation procedures for sea otters. *Ecological Applications* **16**(6): 2293–2312. URL `www.jstor.org/stable/40061959`

Urban, D. and Keitt, T. 2001. Landscape connectivity: A graph-theoretic perspective. *Ecology* **82**(5): 1205–1218. doi:10.1890/0012-9658(2001)082[1205:LCAGTP]2.0.CO;2

Vögeli, M., Serrano, D., Pacios, F. and Tella, J. L. 2010. The relative importance of patch habitat quality and landscape attributes on a declining steppe-bird metapopulation. *Biological Conservation* **143**(5): 1057–1067. doi:10.1016/j.biocon.2009.12.040

Watson, J. C., Ellis, G. M., Smith, T. G. and Ford, J. K. B. 1997. Updated status of the sea otter, *Enhydra lutris*, in Canada. *The Canadian Field-Naturalist* **111**(2): 277–286. URL `www.biodiversitylibrary.org`

Williams, J. C., ReVelle, C. S. and Levin, S. A. 2005. Spatial attributes and reserve design models: A review. *Environmental Modeling and Assessment* **10**(3): 163–181. doi:10.1007/s10666-005-9007-5

WPD (Wine Project Developers). 2010. Wine: Wine is not an emulator. URL `www.winehq.org`. Version 1.2.2

Appendix

Electronic copies of the **PatchImportance** code, `PatchImportance.R` (Listing 3), and the **RAMAS Metapop** input file for British Columbia (BC) sea otters, `seaotter.mp`, are available from the authors upon request. We can also provide electronic copies of the batch file, `batch.txt` (Listing 1), and the two end-of-line conversion files, `beforeMP` and `afterMP` (Listing 2). Quantifying the relative importance of the 80 BC sea otter patches requires significant time; metapopulations with fewer patches are faster to implement. For example, the BC herring sample file available with the **RAMAS Metapop** installation, `PacificHerring.mp`, has five patches (based on Fu et al. 2004). Note that BC herring patch names must follow the **GRIP** naming convention (Subsection 4.1).

Listing 3. The **PatchImportance** code (`PatchImportance.R` version 1.0) is written in the programming language **R** (RDCT 2011).

```
################################################################################
#
# Authors:        Janelle M. R. Curtis and Matthew H. Grinnell
# Affiliation:    Pacific Biological Station, Fisheries and Oceans Canada
# Research group: Conservation Biology Section (Janelle M. R. Curtis)
# Contacts:       e-mail: janelle.curtis@dfo-mpo.gc.ca | tel: (250) 756.7157
#                 e-mail: matt.grinnell@dfo-mpo.gc.ca | tel: (250) 756.7326
# Project:        Quantify patch importance based on metapopulation persistence
#                 and minimum abundance
# Code name:      PatchImportance.R
# Code version:   1.0
# Date started:   2008-04-01  (yyyy-mm-dd)
# Date finished:  2012-02-20  (yyyy-mm-dd)
#
# Goal: Quantify the relative importance of habitat patches based on their
# influence on the probability of metapopulation persistence, breaking ties
# based on the expected minimum metapopulation abundance. That is to say,
# compared to less important patches, the inclusion of more important patches
# in the metapopulation increases the probability of metapopulation persistence
# (or the expected minimum metapopulation abundance) by a larger amount.
#
# Requirements: In addition to this code, at least two files are required in
# the working directory: 'batch.txt'; and the metapopulation input file (e.g.,
# 'seaotter.mp'). Two programmes must be installed: R[1]; and RAMAS Metapop[2].
# Note that non-Windows operating systems require two additional files,
# 'beforeMP' and 'afterMP', to convert end-of-line characters between dos and
# unix, as well as an additional programme, WineHQ[3], to run RAMAS Metapop.
# Read the PatchImportance user manual[4] for more implementation details
# analysis, and an example using the British Columbia sea otter metapopulation.
#
# Notes: Please contact the authors if you have questions, comments,
# suggestions, or concerns regarding the code. We are attempting to keep track
# of this code's use; please cite the user manual[4] and contact the authors if
# you use PatchImportance for research. Note that PatchImportance comes with
# absolutely no warranty.
#
# References:
# [1] RDCT (R Development Core Team). 2011. R: A language and environment for
#     statistical computing. URL www.R-project.org. R foundation for
#     Statistical Computing. Vienna, Austria. R version 2.14.0
# [2] Akcakaya, H. R. 2005. RAMAS GIS: Linking spatial data with population
#     viability analysis. Applied Biomathematics. URL www.ramas.com. User
#     manual for version 5
# [3] WPD (Wine Project Developers). 2010. WineHQ: Wine is not an emulator.
#     URL www.winehq.org. Version 1.2.2
```

```
# [4] Grinnell, M. H. and Curtis, J. M. R. 2012. User manual for
#     PatchImportance 1.0: Quantifying relative habitat patch importance based
#     on metapopulation persistence and minimum abundance. Can. Tech. Rep.
#     Fish. Aquat. Sci. 2977: vi + 41 p.
#
###############################################################################

##########################################
##### Start file 'PatchImportance.R' #####
##########################################

# Housekeeping
rm( list=ls( ) )       # Clear the workspace
graphics.off( )        # Turn graphics off
gc( )                  # Empty the trash
sTime <- Sys.time( )  # Start the timer

########################################
##### Start user-defined variables #####
########################################

# RAMAS Metapop file name
mpFile <- "seaotter.mp"

# Number of years
nYr <- 100

# Number of replications per iteration
nRep <- 50

# Number of iterations
nIter <- 200

# Save simulation output?
doSave <- TRUE

# RAMAS Metapop drive letter (if using WINE)
wineDrive <- "Z:"

######################################
##### End user-defined variables #####
######################################

# Remove files/directories from previous runs
unlink( x=c( "Abundance.pdf", "Persistence.pdf", "RelativeImportance.pdf",
        "ranks.txt", "pProbs.txt", "nAbunds.txt", "RelativeImportance.csv",
        "pbs.std*", "OutputData"), recursive=TRUE )
# Create a directory to hold output data (only used if doSave)
dir.create( path="OutputData" )

# Get the required files, operating system dependent
ifelse( .Platform$OS.type == "windows",
    reqFiles <- c( mpFile, "batch.txt" ),
    reqFiles <- c( mpFile, "batch.txt", "beforeMP", "afterMP" ) )
# Check if there are missing files, and error if they are missing
if( !all(reqFiles %in% list.files()) ) {
  # If so, stop
  stop( "Ensure the working directory has the required files.", call.=FALSE )
}  # End if there are missing files

# Ensure inputs are integers that satisfy constraints
nYr <- as.integer( nYr + 0.5 )
if( nYr < 1 | nYr > 500 ) {
  # Get offending value
  nYrSt <- nYr
```

```
  # Reset to allowed value
  ifelse( nYr < 1, nYr <- 1, nYr <- 500 )
  # Warning
  warning( "Bad value: variable 'nYr' changed from ", nYrSt," to ", nYr, ".",
      call.=FALSE )
}
nRep <- as.integer( nRep + 0.5 )
if( nRep < 4 | nRep > 10000 ) {
  # Get offending value
  nRepSt <- nRep
  # Reset to allowed value
  ifelse( nRep < 4, nRep <- 4, nRep <- 10000 )
  # Warning
  warning( "Bad value: variable 'nRep' changed from ", nRepSt, " to ", nRep,
      ".", call.=FALSE )
}
nIter <- as.integer( nIter + 0.5 )
if( nIter < 1 ) {
  # Get offending value
  nIterSt <- nIter
  # Reset to allowed value
  nIter <- 1
  # Warning
  warning( "Bad value: variable 'nIter' changed from ", nIterSt, " to ", nIter,
      ".", call.=FALSE )
}

# Get the first item from each line in the original mp file
firstItemMP <- scan( file=mpFile, skip=0, sep=",", what="character",
    quiet=TRUE, flush=TRUE, blank.lines.skip=FALSE )

# Count the number of patches (ignore the header, and after the pop matrix)
CountPatches <- function( dat ) {
  # Get lines corresponding to patch names
  vec <- grep( "Pop ", dat, perl=TRUE )
  # Ignore the first 6 lines (the header)
  vec <- vec[vec > 6]
  # Get the first number and build a sequence of length 'vec'
  seqVec <- seq( from=vec[1], by=1, length.out=length(vec) )
  # Count the number of elements that match
  nMatch <- length( which(vec == seqVec) )
  # If there are less than 2 patches, stop
  if( nMatch < 2 )  stop( "Require >=2 patches, named 'Pop 1', 'Pop 2', ...",
        call.=FALSE )
  # Return the number of matches
  return( nMatch )
}  # End CountConsecutive function
num <- CountPatches( dat=firstItemMP )

# Print messages
cat( "Input file '", mpFile, "' with ", num, " patches: nYr=", nYr, "; nRep=",
    nRep, "; and nIter=", nIter, ".\n",
    "The PatchImportance algorithm will call RAMAS Metapop ",
    nIter * (num * (num + 1) ) / 2, " times.\n", sep="" )
if( !doSave )
  cat( "Intermediate input and output files will not be saved to disk.\n" )

# Count the number of ties resolved using the second statistic, and random
nSecond <- 0
nRandom <- 0

# Start loop over nIter
for( q in 1:nIter ) {
  # Vectors to fill in later -- could include other stats here if desired
  ranks <- vector( )  # Ranked patches
```

```
pProbs <- 0          # Cumulative persistence probability
nAbunds <- 0         # Minimum metapopulation abundance

# Loop over the number of patches the determine next most influential patch
for( m in 1:num )  {
  # Remove files from previous runs
  unlink( x=c("batch_file.bat", "rep_*.mp", "METABAT.REC", "Metapop.RES",
          "rep_*.SCL", "Abund*.txt", "FinalStageN*.txt", "Harvest*.txt",
          "Int*.txt", "Loc*.txt", "MetapopOcc*.txt", "Quasi*.txt",
          "Ter*.txt") )
  # Vector of 1:num
  V_Numberpops <- 1:num
  # Exclude patches that are already identified in ranks
  if( length(ranks) != 0 )  V_Numberpops <- V_Numberpops[-ranks]

  # Loop through replicate simulations to include an additional patch
  for( y in 1:length(V_Numberpops) ) {
    # Read in the original PVA input file to reference lines
    inputFile <- readLines( mpFile, n=-1 )
    # Find the line that specifies the number of stages modeled
    N_stages <- scan( mpFile, what="list", skip=9, nlines=1, quiet=TRUE )
    N_stages <- as.numeric( N_stages[1] )
    # Read the "batch.txt" file used to create the RAMAS Metapop batch file
    Start <- readLines( "batch.txt" )
    # Update the input file name
    filename <- paste( "rep", "_", y, ".mp", sep="" )

    # Create the RAMAS Metapop batch file (operating system dependent)
    ifelse( .Platform$OS.type == "windows",
        batch_rep <- paste( Start[1], " ", Start[2], " ", Start[3], " ",
            Start[4], "rep_", y, ".mp", Start[5], sep="" ),
        batch_rep <- paste( Start[1], " ", Start[3], " ", Start[4], "rep_",
            y, ".mp", Start[5], sep="" ) )
    # Write the RAMAS Metapop batch file
    write( batch_rep, file="batch_file.bat", append=TRUE )
    # Ensure results files correspond to appropriate input files
    write( paste("rename Abund.txt ", "Abund_", y, ".txt", sep=""),
        "batch_file.bat", append=TRUE )
    write( paste("rename FinalStageN.txt ", "FinalStageN_", y, ".txt",
            sep=""), "batch_file.bat", append=TRUE )
    write( paste("rename Harvest.txt ", "Harvest_", y, ".txt", sep=""),
        "batch_file.bat", append=TRUE )
    write( paste("rename HarvestRisk.txt ", "HarvestRisk_", y, ".txt",
            sep=""), "batch_file.bat", append=TRUE )
    write( paste("rename IntExpRisk.txt ", "IntExpRisk_", y, ".txt", sep=""),
        "batch_file.bat", append=TRUE )
    write( paste("rename IntExtRisk.txt ", "IntExtRisk_", y, ".txt", sep=""),
        "batch_file.bat", append=TRUE )
    write( paste("rename IntPerDec.txt ", "IntPerDec_", y, ".txt", sep=""),
        "batch_file.bat", append=TRUE )
    write( paste("rename LocalOcc.txt ", "LocalOcc_", y, ".txt", sep=""),
        "batch_file.bat", append=TRUE )
    write( paste("rename LocExtDur.txt ", "LocExtDur_", y, ".txt", sep=""),
        "batch_file.bat", append=TRUE )
    write( paste("rename MetapopOcc.txt ", "MetapopOcc_", y, ".txt", sep=""),
        "batch_file.bat", append=TRUE )
    write( paste("rename QuasiExp.txt ", "QuasiExp_", y, ".txt", sep=""),
        "batch_file.bat", append=TRUE )
    write( paste("rename QuasiExt.txt ", "QuasiExt_", y, ".txt", sep=""),
        "batch_file.bat", append=TRUE )
    write( paste("rename TerExpRisk.txt ", "TerExpRisk_", y, ".txt", sep=""),
        "batch_file.bat", append=TRUE )
    write( paste("rename TerExtRisk.txt ", "TerExtRisk_", y, ".txt", sep=""),
        "batch_file.bat", append=TRUE )
    write( paste("rename TerPerDec.txt ", "TerPerDec_", y, ".txt", sep=""),
```

```
        "batch_file.bat", append=TRUE )

    # Get the line number in the input file that has "Migration" on it
    Migration <- grep( "Migration", inputFile, perl=TRUE )
    # Get the line number in input file that has "Correlation" on it
    Correlation <- grep( "Correlation", inputFile, perl=TRUE )
    # Calculate the number of rows in the dispersal matrix
    Nrows_dispersal_matrix <- Correlation - Migration - 3
    # Get the line number in the input file that has "Constraints Matrix"
    Line_Constraints <- grep( "Constraints", inputFile, perl=TRUE )
    # Sample a new number of patches based on the original number
    Npops <- m
    # Calculate the number of patches that need to be added
    Diff <- num - Npops
    # Get the first line of the population dataframe and subtract 1
    firstNpop <- grep( "Pop ", firstItemMP, perl=TRUE )[1] - 1
    # Read in the patch dataframe
    pop <- read.table( mpFile, skip=firstNpop, sep=",", nrow=num )
    # Read in the total number of individuals in the original patches
    # to calculate % decline at t = 100 and t = 20
    Total_N_original <- sum( pop$V4 )
    # Create patch names for all patches
    pop$V1 <- as.character( 1:num )
    # Write patch names
    PopNames <- as.list( paste("pop", 1:num, sep="") )
    # Matrix to keep track of which patches are included
    ifelse( y == 1,
        Pop_vector <- rep( 0, num ),
        Pop_vector <- rbind( Pop_vector, rep(0, num) ) )

    # Read in the dispersal matrix, if it is there
    if( Nrows_dispersal_matrix > 0 ) {
      Dispersal_matrix <- matrix( scan(mpFile, sep=",",
              nlines=(Correlation - Migration - 3), skip=Migration+2,
              quiet=TRUE), ncol=(Correlation - Migration - 2), byrow=TRUE )
      # Remove the last column (NAs)
      Dispersal_matrix <- Dispersal_matrix[, -ncol(Dispersal_matrix)]
    } else {  # Otherwise, create an empty one
      Dispersal_matrix <- matrix( 0, num, num )
    }
    # Read in the correlation matrix, or create one
    D_autofill <- as.logical( scan(mpFile, what="list", skip=Correlation[1],
            nlines=1, quiet=TRUE) )
    if( any(D_autofill) ) {
      Correlation_matrix <- matrix( 0, num, num )
    } else {
      Correlation_matrix <- matrix( NA, num, num )
      corDat <- scan( mpFile, sep=",", quiet=TRUE, nlines=num,
          skip=Correlation + 2 )
      Correlation_matrix[upper.tri(Correlation_matrix, diag=TRUE)] <-
          na.omit( corDat )
      Correlation_matrix[lower.tri(Correlation_matrix, diag=TRUE)] <-
          na.omit( rev(corDat) )
    }

    # Read in the stage-specific initial abundances
    Initial_Abundances <- matrix( scan(mpFile, nlines=num, quiet=TRUE,
            skip=Line_Constraints + (N_stages*3 + 3)), ncol=N_stages,
        byrow=TRUE )

    # Get the line on which the number of stages matrices is listed
    Line_N_stage_matrices <- grep( "stage matrix", inputFile, perl=TRUE )
    # Read the stage and standard deviation matrices, and their descriptions
    N_stage_matrices <- scan( mpFile, skip=Line_N_stage_matrices - 1,
        nlines=1, what="list", quiet=TRUE )
```

```
    # The number of stage and standard deviation matrices
    N_matrices <- as.numeric( N_stage_matrices[1] )
    # Read in the 4-line descriptions of the Stage_matrices and save as list
    Description_Stage_matrix <- vector( "list", N_matrices )
    Stage_matrices <- list( )
    for( i in 1:N_matrices ) {
      Description_Stage_matrix[[i]] <- scan( mpFile, quiet=TRUE, nlines=4,
          skip=Line_N_stage_matrices + ((i - 1)*4) + (i - 1)*N_stages,
          sep= ",", what="list" )
      Stage_matrices[[i]] <- matrix( scan(mpFile, quiet=TRUE,
              skip=Line_N_stage_matrices + 4 + ((i-1)*4) + ((i-1)*N_stages),
              nlines=N_stages), N_stages, N_stages, byrow=TRUE )
    }

    # Reference the location of the standard deviation matrix
    Line_N_stdev_matrices <- grep( "st.dev. matrix", inputFile, perl=TRUE )
    # Get the information on this line
    N_stdev_matrices <- scan( mpFile, skip=Line_N_stdev_matrices - 1,
      nlines=1, what="list", quiet=TRUE )
    # Read in the 1-line descriptions of the Stdev_matrices as a list
    Description_stdev_matrix <- vector( "list", N_matrices )
    Stdev_matrices <- list( )
    for( i in 1:N_matrices ) {
      Description_stdev_matrix[[i]] <- scan( mpFile, sep= ",", quiet=TRUE,
          skip=Line_N_stdev_matrices + ((i - 1)) + (i - 1)*N_stages,
          nlines=1, what="list" )
      Stdev_matrices[[i]] <- matrix( scan(mpFile, quiet=TRUE,
              skip=Line_N_stdev_matrices + 1 + (i - 1) + ((i - 1)*N_stages),
              nlines=N_stages), N_stages, N_stages, byrow=TRUE )
    }

    # Assign 1 when patch is removed, 0 when patch is retained
    Pops_removed <- sort( V_Numberpops[-y], decreasing=TRUE )
    if( m != num ) {
      ifelse( y == 1,
          Pop_vector[Pops_removed] <- 1,
          Pop_vector[y, Pops_removed] <- 1 )
    }
    # Remove the selected patches from the patch dataframe
    if( m != num )  pop <- pop[-Pops_removed, ]
    # Remove selected patches from the dispersal matrix
    if( m != num )  Dispersal_matrix <-
          Dispersal_matrix[-Pops_removed, -Pops_removed]
    if( Npops != 1 ) {
      # Read in the dispersal-distance parameters
      M_parameters <- scan( mpFile, skip=(Migration[1] + 1), nlines=1,
          sep=",", quiet=TRUE )
      # Create a matrix of pairwise geographic distances (Euclidean)
      Pairwise_distance <- matrix( 0, Npops, Npops )
      for( i in 1:Npops ) {
        for( j in 1:Npops )  Pairwise_distance[i, j] <-
              sqrt( (pop[i, 2] - pop[j, 2])^2 +  (pop[i, 3] - pop[j, 3])^2 )
      }

      # Dispersal
      if( M_parameters[1] != 0 ) {
        # Get dispersal and constrain
        Dispersal_matrix <- M_parameters[1] *
            exp( (-Pairwise_distance^M_parameters[3]) / M_parameters[2] )
        if( M_parameters[4] > 0 )
          Dispersal_matrix[Pairwise_distance > M_parameters[4]] <- 0
        # Constrain between 0 and 1
        Dispersal_matrix[Dispersal_matrix < 0] <- 0.0
        Dispersal_matrix[Dispersal_matrix > 1] <- 1.0
      } else {  # Otherwise, sample dispersal from a random uniform
```

```
        Dispersal_matrix <- matrix( runif(1, 0, 1), nrow=Npops, ncol=Npops )
        # Set diagonal to zero
        diag( Dispersal_matrix ) <- 0
        # Enforce a maximum total dispersal rate of 1 from each patch
        colSum1 <- sum( Dispersal_matrix[, 1] )
        if( colSum1 > 1 ) Dispersal_matrix <- Dispersal_matrix / colSum1
      }
      # Keep track of dispersal rates for calculating statistics
      Lower_logical <- lower.tri( Dispersal_matrix )
      # Round values in the dispersal matrix are rounded to 7 decimal places
      Dispersal_matrix <- round( Dispersal_matrix, 7 )
      # Add the extra column of "," at the end of each matrix row for
      # writing to the replicate simulation input file
      Dispersal_matrix <- cbind( Dispersal_matrix, rep("", Npops),
          deparse.level=0 )
    }  # End if Npops != 1

    # Remove the selected patches from the correlation matrix
    if( m != num )  Correlation_matrix <-
          Correlation_matrix[-Pops_removed, -Pops_removed]
    # Read in the correlation-distance function parameters
    C_parameters <- scan( mpFile, skip=Correlation[1] + 1, nlines=1, sep=",",
        quiet=TRUE )
    if( Npops != 1 ) {
      # Constant correlation if no distance function specified
      if( sum(Correlation_matrix, na.rm=TRUE) ==
          length(Correlation_matrix[, 1]) & sum(C_parameters <= 0.1) ) {
        Correlation_matrix <- matrix( runif(1, 0, 1), nrow=Npops,
            ncol=Npops )
      } else {  # If there are correlation parameters
        # Adjust correlations according to new distances among patch pairs
        Correlation_matrix <- C_parameters[1] *
            exp( (-Pairwise_distance^C_parameters[3]) / C_parameters[2] )
        # Ensure diagonal elements on the correlation matrix are 1
        diag( Correlation_matrix ) <- 1
        # Constrain individual elements between 0 and 1
        Correlation_matrix[Correlation_matrix < 0] <- 0.0
        Correlation_matrix[Correlation_matrix > 1] <- 1.0
      }
      # Ensure correlation coefficients are rounded to 6 decimal places
      Correlation_matrix <- round( Correlation_matrix, digits=6 )
    }  # End if Npops != 1
    # Add column of "," at the end of each rows
    Correlation_matrix <- cbind( Correlation_matrix, rep("", Npops),
        deparse.level=0 )

    # Remove the selected patches from the Initial Abundances matrix
    if( m != num ) {
      ifelse( N_stages > 1,
          Initial_Abundances <- Initial_Abundances[-Pops_removed, ],
          Initial_Abundances <- Initial_Abundances[-Pops_removed] )
    }

    # Create an interim replicate simulation file 'pref_*.mp'
    write( inputFile[1:6], file=filename, append=FALSE )
    # Write the number of stochastic runs within each replicate simulation
    write( nRep, file=filename, append=TRUE )
    # Write the number of time steps
    write( nYr, file=filename, append=TRUE )
    write( inputFile[9:28], file=filename, append=TRUE )
    # Write what happens when population size falls below local threshold
    write( "count in total", file=filename, append=TRUE )
    write( inputFile[30:firstNpop], file=filename, append=TRUE )
    # Write the new population dataframe
    write.table( pop, file=filename, append=TRUE, sep=",", row.names=FALSE,
```

```
          col.names=FALSE, na="", quote=FALSE )
      # Write "Migration" on a line
      write( "Migration", file=filename, append=TRUE )
      # Write the line after "Migration"
      write( "TRUE", file=filename, append=TRUE )
      # Write Dispersal_distance function parameters
      write( inputFile[Migration + 2], file=filename, append=TRUE )
      # Write "Correlation"
      write( "Correlation", file=filename, append=TRUE )
      # Write the line after "Correlation"
      write( "TRUE", file=filename, append=TRUE )
      # Write Correlation_distance function parameters
      write( inputFile[Correlation + 2], file=filename, append=TRUE )
      # Write the stage and standard deviation information
      write( inputFile[Line_N_stage_matrices:(Line_Constraints - 1)],
          file=filename, append=TRUE )
      # Write the Constraints Matrix, relative dispersal indices and
      # catastrophe multipliers
      write( inputFile[Line_Constraints:(Line_Constraints + (N_stages*3 + 3))],
          file=filename, append=TRUE )
      # Write Initial_Abundance matrix
      write.table( Initial_Abundances, filename, append=TRUE,
          col.names=FALSE, row.names=FALSE)
      # Read and write in the information from the "stages menu"
      write( inputFile[(Line_Constraints + 1 + (N_stages*3 + 3) + num):
                  (Line_Constraints + (N_stages*3 + 3) + num + 5*N_stages)],
          file=filename, append=TRUE )
      # Ensure there is no population management modeled
      write( "0 (pop mgmnt)", file=filename, append=TRUE )
      # Write a generic extinction threshold of zero
      write( "0", file=filename, append=TRUE )
      # Write a generic explosion threshold of zero
      write( "0", file=filename, append=TRUE )
      # Write the timestep datum as specified in the original input file
      Mgmnt <- grep( "mgmnt", inputFile, perl=TRUE )
      N_Mgmnt <- scan( mpFile, what="list", skip=Mgmnt - 1, nlines=1,
          quiet=TRUE )
      N_Mgmnt <- as.numeric( N_Mgmnt[1] )
      write( inputFile[Mgmnt + N_Mgmnt + 1 + 2], file=filename, append=TRUE )
      # Write end-of-file
      write( "-End of file-", file=filename, append=TRUE )
    }  # End y loop over V_Numberpops

    # Run the RAMAS Metapop batch file (operating system dependent)
    if( .Platform$OS.type == "windows" ) {
      system( "batch_file.bat", wait=TRUE, show.output.on.console=FALSE )
    } else {
      system( paste(getwd(), "beforeMP", sep=.Platform$file.sep), wait=TRUE )
      system( paste("wine wineconsole ", wineDrive, getwd(), "/batch_file.bat",
             sep=""), wait=TRUE )
      system( paste(getwd(), "afterMP", sep=.Platform$file.sep), wait=TRUE )
    }

    # Matrix to hold probability and abundance, and other stats if desired
    statMat <- matrix( NA, nrow=y, ncol=2 )
    colnames( statMat ) <- c( "pExt", "nAbun" )
    # Loop through replicate simulations and collect results
    for( p in 1:y ) {
      # Get the file with interval extinction risk data
      intExtRisk <- paste( "IntExtRisk", "_", p, ".txt", sep="" )
      # Read the entire file
      extFile <- readLines( intExtRisk )
      # Get the line with expected minimum metapopulation abundance
      firstLine <- grep( "Expected minimum abundance", extFile )
      # If the line can't be found, error
```

```
      if( length(firstLine) == 0 )  stop( "Check file '", intExtRisk,
            "'; unable to reference the required line.", call.=FALSE )
      # Get the matrix of extinction data
      extList <- scan( intExtRisk, skip=firstLine + 2, quiet=TRUE, flush=TRUE,
          nlines=1, what=list(thresh=0, prob=0) )
      # First, get cumulative probability of metapopulation extinction
      ifelse( extList$thresh > 0,  # Note: if threshold > 0, prob = 0.0
          statMat[p, "pExt"] <- 0.0,
          statMat[p, "pExt"] <- extList$prob )
      # Get expected minimum metapopulation abundance line
      minAbunList <- scan( intExtRisk, skip=firstLine - 1, quiet=TRUE,
          flush=TRUE, nlines=1, what=list(char="", minAbun=0), sep="=" )
      # Then, get expected minimum metapopulation abundance
      statMat[p, "nAbun"] <- minAbunList$minAbun
    }  # End p loop over replicate simulations
    # Bind extinction probability with information regarding included patches
    extDF <- data.frame( statMat, matrix(Pop_vector, ncol=num) )

    # Start vectors to hold stats -- could include other stats if desired
    popID <- vector( )  # Patch ID
    pPers <- vector( )   # Metapopulation persistence probability
    nAbun <- vector( )   # Minimum metapopulation abundance
    # Loop over patches that haven't been identified as important
    for( i in (1:num)[!1:num %in% ranks] ) {
      # Get the row that include the patch
      pred <- subset( extDF, extDF[i + ncol(statMat)] == 0 )
      # Stop if there are too many rows
      if( nrow(pred) > 1 )  stop( "Too many rows in 'pred'.", call.=FALSE )
      # Get statistics
      popID[i] <- i              # Patch ID
      pPers[i] <- 1 - pred$pExt  # Probability of metapop persistence
      nAbun[i] <- pred$nAbun     # Minimum metapopulation abundance
    }  # End i loop over patches
    # Get a table of patch ID and statistics
    impMat <- na.omit( data.frame(popID=popID, pPers=pPers, nAbun=nAbun) )

    # Get the vector of persistence values
    vecPers <- impMat$pPers
    # Get the maximum prob of persistence
    maxPers <- max( vecPers, na.rm=TRUE )
    # If there is more than one maximum (ties!)
    if( length(which(vecPers == maxPers)) > 1 ) {
      # Get the patches with the maximum pPers
      maxPersPops <- impMat[impMat$pPers == maxPers, ]
      # Get the vector of abundances
      vecAbun <- maxPersPops$nAbun
      # Get the maximum of minimum abundance
      maxAbun <- max( vecAbun, na.rm=TRUE )
      # If there is more than one maximum (more ties!)
      if( length(which(vecAbun == maxAbun)) > 1 ) {
        # Get the patches with maximum persistence and abundance
        maxPersAbunPops <- impMat$popID[ impMat$pPers == maxPers &
                impMat$nAbun == maxAbun ]
        # Break the tie by selecting a patch at random; could use a third stat
        ranks[m] <- sample( x=maxPersAbunPops, size=1 )
        # Update the number of random tie breaks
        nRandom <- nRandom + 1
      } else {  # End if there is more than one in maxAbun, else
        # Get the patch with the maximum abundance
        ranks[m] <- impMat$popID[impMat$pPers == maxPers &
                impMat$nAbun == maxAbun]
        # Update the number of secondary tie breaks
        nSecond <- nSecond + 1
      }  # End procedure if there is only one in maxThresh
    } else {  # End if more than one in maxPers, else
```

```
      # Get the patch with the highest prob of persistence
      ranks[m] <- impMat$popID[impMat$pPers == maxPers]
    }  # End procedure for ranking patches
    # Record persistence and abundance; could include other stats
    pProbs[m] <- maxPers
    nAbunds[m] <- impMat$nAbun[impMat$popID == ranks[m]]

    # Save output, if specified
    if( doSave ) {
      # Get list of required files -- could include other stats if desired
      mpFiles <- list.files( pattern="^rep.+[.]mp$" )
      extFiles <- list.files( pattern="^IntExtRisk.+[.]txt$" )
      # Create a subdirectory to hold output data
      newDir <- paste( "OutputData/output.", q, ".", m, sep="" )
      dir.create( path=newDir )
      # Copy desired files
      file.copy( from=c(mpFiles, extFiles), to=newDir )
    }  # End if doSave
  }  # End m loop over num

  # Write output data to text files -- write additional stats if included
  write( ranks, file="ranks.txt", append=TRUE, sep=",", ncolumns=num )
  write( pProbs, file="pProbs.txt", append=TRUE, sep=",", ncolumns=num )
  write( nAbunds, file="nAbunds.txt", append=TRUE, sep=",", ncolumns=num )
  # Print progress message
  cat( "Finished iteration ", q, " of ", nIter, ": " , sep="" )
  print( Sys.time( ) - sTime )
}  # End q loop over nIter

# Calculate and plot relative patch importance
CalcPatchImp <- function( dat1, dat2, dat3 ) {
  # Set up a matrix to hold importance values
  mat1 <- matrix( NA, nrow=ncol(dat1), ncol=nrow(dat1) )
  # Add row names
  rownames( mat1 ) <- paste( "Pop", 1:num, sep=" " )
  # Set up identical matrices to hold probability and incremental probability
  mat2 <- mat1
  mat3 <- mat1
  # Loop over patches, and assemble patch statistics by row
  for( p in 1:nrow(mat1) ) {
    # Find out which columns the patch is in (get rank)
    mat1[p, ] <- which( dat1 == p, arr.ind=TRUE )[ ,"col"]
    # Get the probability of persistence
    mat2[p, ] <- dat2[which(dat1 == p, arr.ind=TRUE)]
    # Get the incremental increase in the probability of persistence
    mat3[p, ] <- dat3[which(dat1 == p, arr.ind=TRUE)]
  }  # End loop over patches
  # Switch so important patches (low ranks) have high importance values
  mat1 <- max( mat1 ) - mat1 + 1
  # Re-scale between 0.00 and 1.00
  mat1 <- ( mat1 - min(mat1) ) / ( max(mat1) - min(mat1) )
  # Calculate some statistics for each patch
  relImp <- data.frame( row.names=NULL,
      # Patch names
      patch=rownames(mat1),
      # Relative importance
      imp025=apply(X=mat1, MARGIN=1, FUN=quantile, probs=0.025),   # 0.025
      imp25=apply(X=mat1, MARGIN=1, FUN=quantile, probs=0.25),    # 0.25
      impMed=apply(X=mat1, MARGIN=1, FUN=quantile, probs=0.5),    # Median
      imp75=apply(X=mat1, MARGIN=1, FUN=quantile, probs=0.75),    # 0.75
      imp975=apply(X=mat1, MARGIN=1, FUN=quantile, probs=0.975),   # 0.975
      # Probability of metapopulation persistence
      prob025=apply(X=mat2, MARGIN=1, FUN=quantile, probs=0.025),   # 0.025
      prob25=apply(X=mat2, MARGIN=1, FUN=quantile, probs=0.25),    # 0.25
      probMed=apply(X=mat2, MARGIN=1, FUN=quantile, probs=0.5),    # Median
```

```
    prob75=apply(X=mat2, MARGIN=1, FUN=quantile, probs=0.75),     # 0.75
    prob975=apply(X=mat2, MARGIN=1, FUN=quantile, probs=0.975),   # 0.975
    # Threshold abundance
    abun025=apply(X=mat3, MARGIN=1, FUN=quantile, probs=0.025),   # 0.025
    abun25=apply(X=mat3, MARGIN=1, FUN=quantile, probs=0.25),     # 0.25
    abunMed=apply(X=mat3, MARGIN=1, FUN=quantile, probs=0.5),     # Median
    abun75=apply(X=mat3, MARGIN=1, FUN=quantile, probs=0.75),     # 0.75
    abun975=apply(X=mat3, MARGIN=1, FUN=quantile, probs=0.975) )  # 0.975
# Get the order of importance; by median, then 95th percentile range
impOrd <- order( relImp$impMed, relImp$imp975, relImp$imp025 )
# Order by the relative importance
relImp <- relImp[impOrd, ]
# Write to a csv
write.table( relImp, file="RelativeImportance.csv",
    col.names=TRUE, sep=",", row.names=FALSE, append=FALSE )

# Plot relative patch importance
pdf( height=9.75, width=8, file="RelativeImportance.pdf" )
# Set graph area
par( oma=c(0, 1.5, 0, 0), mar=c(3.55, 3, 1.75, 0.1) )
# Plot using a Cleveland dot plot
dotchart( x=relImp$impMed, labels=relImp$patch, pch=19, cex.axis=1, cex=0.7,
    xlim=c(0, 1), ann=FALSE )
# Labels
mtext( side=1, line=2.35, "Relative importance" )
mtext( side=2, line=3.35, "Patch name" )
mtext( side=3, line=0.5, font=2, paste("RAMAS Metapop file '", mpFile, "'",
        sep="") )
# Vertical line at 0.5
abline( v=0.5, lwd=1, lty="dashed" )
# Start loop over rows
for( i in 1:nrow(relImp) ) {
  # Add line for 95% range
  segments( x0=relImp$imp025[i], y0=i, x1=relImp$imp975[i], y1=i, lwd=1 )
  # Add a grey rectangle for 50% range
  rect( xleft=relImp$imp25[i], ybottom=i-0.25, xright=relImp$imp75[i],
      ytop=i+0.25, col=grey(0.75), border=NA )
  # Re-plot dots
  points( x=relImp$impMed[i], y=i, pch=19, cex=1 )
  # Add vertical lines for end of 95% range
  segments( x0=relImp$imp025[i], y0=i+0.15, x1=relImp$imp025[i], y1=i-0.15,
      lwd=1.5 )
  segments( x0=relImp$imp975[i], y0=i+0.15, x1=relImp$imp975[i], y1=i-0.15,
      lwd=1.5 )
}  # End i loop over rows
# Close the pdf
dev.off( )

# Plot probability of metapopulation persistence
pdf( height=6, width=6, file="Persistence.pdf" )
# Set up the plot area
par( mar=c(3.6, 3.5, 1.5, 0.1) )
# Plot the median
plot( x=1:num, y=apply(X=dat2, MARGIN=2, FUN=median), ylim=c(0, 1), lwd=3,
    type="l", ann=FALSE )
mtext( side=1, line=2.5, "Number of patches", cex=1.25 )
mtext( side=2, line=2.5, "Probability of metapopulation persistence",
    cex=1.25 )
mtext( side=3, line=0.5, paste("RAMAS Metapop file '", mpFile, "'", sep=""),
    cex=1.25 )
# Add 50% polygon
polygon( x=c(1:num, num:1), border=NA, col=rgb(0, 0, 0, 0.35),
    y=c(apply(X=dat2, MARGIN=2, FUN=quantile, probs=0.25),
        rev(apply(X=dat2, MARGIN=2, FUN=quantile, probs=0.75))) )
# Add 95% lines
```

```
  lines( x=1:num, y=apply( X=dat2, MARGIN=2, FUN=quantile, probs=0.025 ),
      col="black", lwd=1 )
  lines( x=1:num, y=apply( X=dat2, MARGIN=2, FUN=quantile, probs=0.975 ),
      col="black", lwd=1 )
  # Turn the device off
  dev.off( )

  # Plot expected minimum metapopulation abundance
  pdf( height=6, width=6, file="Abundance.pdf" )
  # Set up the plot area
  par( mar=c(3.6, 3.5, 1.5, 0.1) )
  # Determine the upper range
  upper95 <- apply( X=dat3, MARGIN=2, FUN=quantile, probs=0.975 )
  # Plot the median
  plot( x=1:num, y=apply(X=dat3, MARGIN=2, FUN=median), lwd=3, type="l",
      ylim=c(0, max(upper95)), ann=FALSE )
  mtext( side=1, line=2.5, "Number of patches", cex=1.25 )
  mtext( side=2, line=2.5, "Metapopulation expected minimum abundance",
      cex=1.25 )
  mtext( side=3, line=0.5, paste("RAMAS Metapop file '", mpFile, "'", sep=""),
      cex=1.25 )
  # Add 50% polygon
  polygon( x=c(1:num, num:1), border=NA, col=rgb(0, 0, 0, 0.35),
      y=c(apply(X=dat3, MARGIN=2, FUN=quantile, probs=0.25),
          rev(apply(X=dat3, MARGIN=2, FUN=quantile, probs=0.75))) )
  # Add 95% lines
  lines( x=1:num, y=apply(X=dat3, MARGIN=2, FUN=quantile, probs=0.025),
      col="black", lwd=1 )
  lines( x=1:num, y=upper95, col="black", lwd=1 )
  # Turn the device off
  dev.off( )

  # Return each patch's relative importance
  return( list(stats=relImp, imps=mat1, probs=mat2, abunds=mat3) )
}  # End CalcPatchImp function

# Calculate and plot relative patch importance
patchImp <- CalcPatchImp( dat1=read.table(file="ranks.txt", sep=","),
    dat2=read.table(file="pProbs.txt", sep=","),
    dat3=read.table(file="nAbunds.txt", sep=",") )

# Remove old output files from the directory
unlink( x=c("batch_file.bat", "rep_*.mp", "METABAT.REC", "Metapop.RES",
        "rep_*.SCL", "Abund*.txt", "FinalStageN*.txt", "Harvest*.txt",
        "Int*.txt", "Loc*.txt", "MetapopOcc*.txt", "Quasi*.txt", "Ter*.txt") )

# Print end of file message and elapsed time
cat( "End of file PatchImportance.R: ", sep="" )
print( Sys.time( ) - sTime )

# Messages if ties broken using secondary statistic, or random
if( nSecond > 0 )  message( "Note: ", nSecond,
      " tie(s) resolved using the secondary statistic (",
      format((nSecond*100) / ((num-1)*nIter), digits=3), "%)." )
if( nRandom > 0 )  message( "Note: ", nRandom,
      " tie(s) resolved by selecting a patch at random (",
      format((nRandom*100) / ((num-1)*nIter), digits=3), "%)." )

###########################################
##### End of file 'PatchImportance.R' #####
###########################################
```

Index