



www.csiro.au

Risk in complex pathways: Propagation of risk via zero inflated distributions and mixing

Simon Barry and Xunguo Lin

Report Number: EP102408

Australian Centre of Excellence for Risk Analysis (ACERA)

Commercial In Confidence

Enquiries should be addressed to:

Dr Simon Barry
CSIRO Mathematics, Informatics and Statistics
CSIRO, GPO Box 664, Canberra, ACT 2601, Australia
Telephone : +61 2 6216 7157
Fax : +61 2 6216 7111
Email : Simon.Barry@csiro.au

Distribution List

Client	(1)
Publications Officer	(0)
Stream Leader	(0)
Authors	(2)

Copyright and Disclaimer

© CSIRO To the extent permitted by law, all rights are reserved and no part of this publication covered by copyright may be reproduced or copied in any form or by any means except with the written permission of CSIRO.

Important Notice

CSIRO advises that the information contained in this publication comprises general statements based on scientific research. The reader is advised and needs to be aware that such information may be incomplete or unable to be used in any specific situation. No reliance or actions must therefore be made on that information without seeking prior expert professional, scientific and technical advice. To the extent permitted by law, CSIRO (including its employees and consultants) excludes all liability to any person for any consequences, including but not limited to all losses, damages, costs, expenses and any other compensation, arising directly or indirectly from using this publication (in part or in whole) and any information or material contained in it.

Contents

1	Introduction	4
2	Pathway analysis	5
3	Models	7
3.1	Mixing	8
3.2	Inspection	9
4	Example	12
4.1	Introduction	12
4.2	Modelling and analysis	12
4.3	Summary	16
5	Discussion	16
6	Conclusions	18
7	Appendix A: cpathway - an R package	19
7.1	List of functions developed within the cpathway package	19
7.2	Function name: add.empirical	19
7.3	Function name: inspection	19
7.4	Function name: make.analytic	20
7.5	Function name: make.empirical	20
7.6	Function name: make.mix	20
7.7	Function name: plot.analytic	20
7.8	Function name: plot.empirical	20
7.9	Function name: simul	21
7.10	Function name: simul.analytic	21
7.11	Function name: simul.empirical	21
7.12	Function name: simul.mix	21
7.13	Function name: unmake.empirical	21
7.14	Example code	22
8	Appendix B: R code developed for cpathway	23
9	Acknowledgements	28

ABSTRACT

Trade in commodities is rarely simple. Biosecurity risk material occurs in a variety of aggregations such as farms, containers and cases in the importation pathway. Assessing either the effectiveness of mitigation methods or the risk of incursion often requires considering this context. This report provides tools to support this assessment. We have proposed a mathematical/statistical method to model complex pathways which often occur in quantitative risk assessments in biosecurity. The mathematical models we have developed in this report include pest infection, mixing and inspection. Monte Carlo simulation techniques were used to analyse different stages of mixing and inspection. A worked example is given to illustrate the proposed modelling and analysis techniques. All the required computing functions have been coded in R and bundled together to form an R package called `cpathway` for easy adoption by decision makers.

Keywords: risk assessment, biosecurity, complex risk pathway, simulation.

1 Introduction

Risk assessment methodology has wide application in government. The exact definition of risk assessment is problematic, but it typically involves identifying an end point which is undesirable and then modelling the likelihood of occurrence of this endpoint and the possible consequence if this endpoint occurs. The focus on a defined endpoint, and the structured process that is used to calculate the likelihoods (and possibly the consequences) are the major attractions of the methodology to policy.

Much of the methodology of risk assessment has strong roots in the engineering literature. A popular approach is the use of failure trees and probability techniques to analyse the likelihood and modes of failure. The likelihood of an endpoint such as the failure of a structure or device is routinely assessed via these approaches. More recently, the analysis of biosecurity risks has also been considered in this framework (Office International des Epizooties (OIE), 2004a,b; Michael and Podleckis, 2000; Gray et al., 1998). This occurred because of a desire to more transparently describe the logic of the assessment and to better characterise uncertainty. This report considers a particular aspect of this analysis that occurs when assessing the risk of importation of a commodity to a country.

In Australia, applications to import commodities are assessed to ensure that they do not impose an unacceptable risk to local industries and the environment. In certain cases a quantitative assessment may be performed to assess the particular risk. This assessment involves constructing a quantitative model that consists of the pathway through which the commodity is imported, inspected and distributed. Individual identified pests and diseases are assessed against this pathway to assess their risks.

The construction of a pathway typically involves deciding on a unit of analysis (Miller et al., 1993; Vose, 2008). For example, individual fruit, cases of fruit, or pallets of cases could be considered as an appropriate unit for analysis depending on which of the importation stages is the most critical. The choice of unit also has implications for the ease with which available data can be used and how the risk of individual units is aggregated into the overall risk. While the biosecurity risk assessment literature has identified the importance of choosing an appropriate unit to aid the formulation of the model, and the associated closeness of the model to the real world, it has not considered more complicated behavior that occurs in pathways. It has not considered modelling of complex pathways generally. However, this is important because this behaviour may strongly influence the likelihood of establishment and therefore have a direct influence on the overall assessment. Simple cases such as herds and individual animals have been considered by a number of authors (see for example Office International des Epizooties (OIE) (2004a,b)). Hartnett et al. (2007) alluded to an object oriented model that may consider this phenomenon, but it was not explicitly described.

This report considers the general modelling of more complex pathways in biosecurity risk assessments. It considers the logical basis of this analysis as well as the techniques for efficiently computing them. It first reviews the issues that arise in an import pathway in Section 2. In Section 3, it introduces an elaboration of the usual pathway model utilizing zero inflated distributions and mixtures, and sampling aggregates of them, to model risk. Section 4 contains a worked example. The technique is further discussed in Section 5 before the report is concluded in Section 6. In the Appendices, the descriptions of all the developed R functions of the `cpathway` will be given together with the R code.

2 Pathway analysis

Pathway analysis typically refers to the specification of a particular outcome (or endpoint) and the decomposition of the steps that need to occur for the outcome to occur (Vose, 2008). The steps are typically sequential in time and hopefully represent the key components of the process that leads to the outcome. An example from biosecurity is the probability that a disease spreads and establishes in a country because of the importation of particular goods. For this to happen it must occur on the farms the imported goods are sourced from. It must survive the packing and shipment process. It must survive inspection at the border and distribution. It must be transported to a region where it can establish, and it has to expose susceptible hosts in the region that it is imported to and then establish and spread. This model is closely related to the international standards for import risk assessment (Office International des Epizooties (OIE), 2004a,b), and is shown schematically in Figure 2.1.

This type of analysis has a long history in risk assessment, commencing in 1962 in relation to nuclear missile safety. The analysis typically assigns probabilities to each of the pathway steps which give the probability of infection/disinfection depending on the nature of the step. Probability distributions are assigned to these probabilities to express “uncertainty”. Standard calculation (typically Monte Carlo) is then used to predict model outcomes from the structure of the model and the uncertainty distributions. The model outcomes typically produce likelihoods that are used to quantify risk. This approach has a large number of proponents and has a sound basis in probability theory. Provided that the beliefs of the person or group carrying out the assessment are adequately represented by the probability distributions, the quantities used in the model are well defined, and the construction of the pathway reflects the real world beliefs about possibilities and plausibilities, then these probability calculations can be used to integrate the information in a consistent and interpretable way.

It is the construction of an adequate stochastic representation of the pathway which is the focus of this report. A common phenomenon in these models is that people confuse the process and the empirical description of the process. A simple example is illustrative. In analysing the pathway in Figure 2.1 the standard approach as advocated by Vose (2008) and others is to decide on the unit of analysis. Imagine our choice is to consider individual fruit and to aggregate these fruit over the volume (number of fruit) of the trade. If we have probabilities associated with each step in the pathway we have implicitly defined a stochastic process for the generation of the risk. Any mathematical model of a physical process is just a mathematical abstraction of the process. Problems arise with the strong assumptions of independence between units (fruit). The model is also inconsistent with how risk is normally formed, transmitted and propagated. This is a particular danger when dealing with subject matter experts who are not used to expressing opinions quantitatively. If the quantities are not grounded in reality, miscommunication is almost inevitable.

Pathway analyses based on fixed units strike difficulties for a number of reasons. First, the efficacy of inspection processes depends on the infection status of other units in the consignment. If the unit of analysis is not the same as the consignment then significant difficulties will arise. Another issue is that the risk of an infected individual unit might depend on the infection status of nearby units. An example would be where insect pests are involved and there are allee effects. The final point is that significant errors can occur if using marginal probabilities over complex populations and processes. This is considered in a related report (Barry, 2010).

If one is tied to a simple pathway approach with simple probabilities, we need to consider

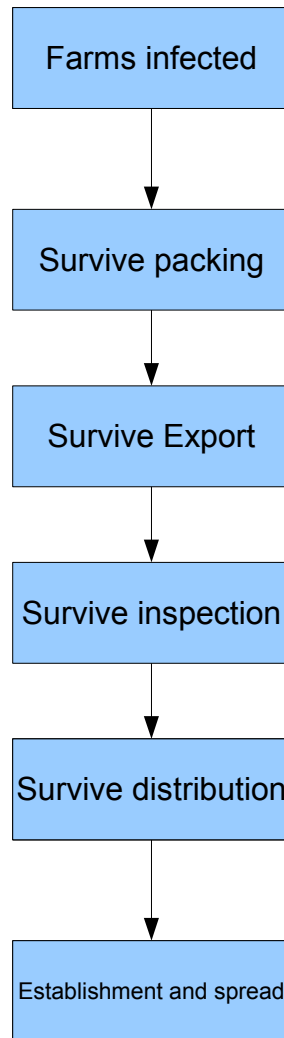


Figure 2.1: Hypothetical pathway.

alternative approaches and representations. If the reader is confused about why such a constraint should be applied to the analysis, it is useful to understand that the major proprietary tool in this area, @Risk¹, has a strong alignment towards models with this structure.

One possible avenue is to consider the trade over the specified period as a “bucket” of the commodities. The “probabilities” in the models are then interpreted as proportions, being the proportion of the commodity at each step that will be impacted by the nature of the step. In this case the expert is being asked to describe the hypothesised future trade in aggregate. This has the advantage of being a concrete concept that an individual can be comfortable with and it could potentially occur. Difficulties still arise. Each step needs to be conditioned correctly on the previous steps for the propagation of uncertainties to be valid. In addition, the model is marginal in the sense that it has no concept of clustering of risk units. This aspect is important because risk and the number of infected units are typically non linear due to allee effects between model variables. Thus the analysis needs to consider side analyses to take these things into account. This approach is considered in a related report (Barry, 2010) which also discusses the impact of ignoring clustering.

Alternatively, if we still want to respect the process we need to consider more realistic formulations. This is the focus of the remainder of the paper.

3 Models

The choice of an appropriate model is a trade-off between available data, understanding of process and the purpose of the model. We propose modelling the risk of units via a hierarchical distribution. If we consider a unit such as a tonne of a commodity, each unit is made up of multiple items (fruit, as an example). We propose modelling the infection status of the unit via a two-step process. First we have a probability p that the unit has any infected elements (fruit). Let this be denoted by the indicator y . Conditional on y , we model the proportion of the unit that is infected, α as some distribution $f(\alpha; \theta)$, with θ being the associated parameters which define the distribution. This approach provides the framework for moving between elements within the hierarchy. For a three level hierarchy the obvious extensions can be used.

An example distribution is shown in Figure 3.1. It shows the distribution of infection over the population of infected units, $f(\alpha; \theta)$ with α the proportion infected and θ the associated parameters which describe the distribution of α . In this case the distribution is from the Beta family and therefore θ is the two parameters of this distribution. From this figure it can be seen that the most likely amount of infection is 0.005 and very few units have infection rates above 0.06.

By having an explicit formulation for the structure of the units it is possible to coherently move between different units at each stage of the process, tracking the infection status appropriately. For example consider the pathway in the previous section. Initially the commodity (say, apples) is sourced from farms. These are packed into some unit, such as a bin and transported to a packing shed. At the packing shed inspections may occur and remedial actions applied to the bin. The apples are then packed into new units such as the pallet. These pallets are then imported and may be inspected, with selection and management actions occurring over the pallet, rather than the individual fruit. After this they may be repacked and distributed to end consumers.

¹www.palisade.com

In analysing these pathways there are three statistical issues that arise. These are the mixing that occurs when commodities are repackaged into new units, the effect of inspection on units and the pattern of distribution of the final risk after inspection. These will now be discussed in the following sections.

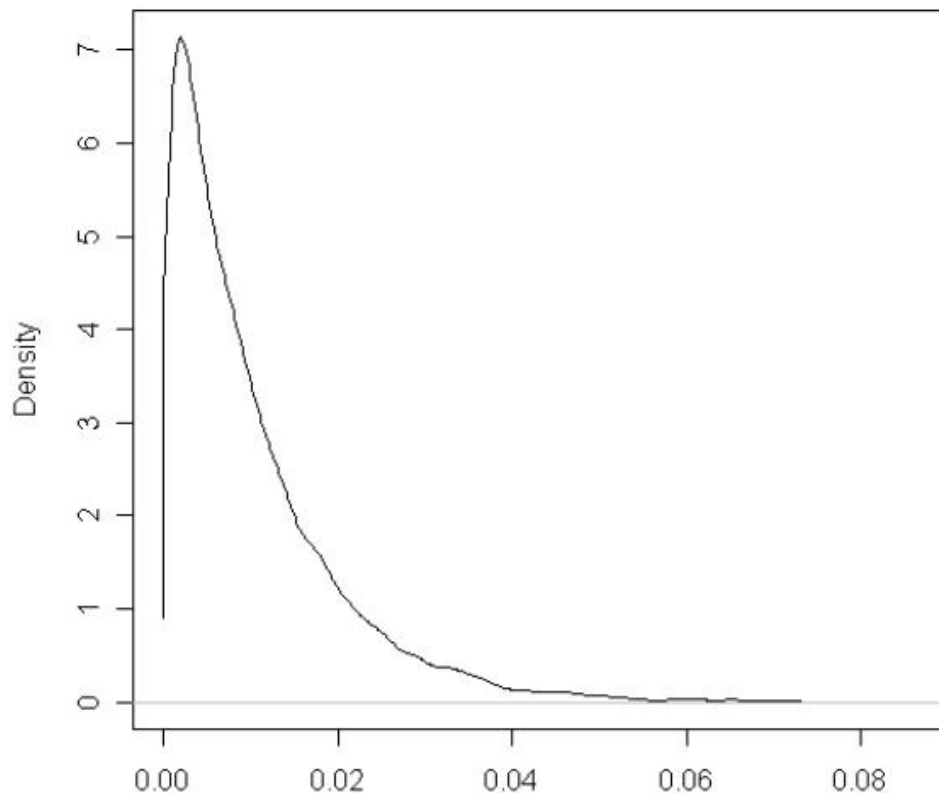


Figure 3.1: Distribution of infection rate within units.

3.1 Mixing

At the packing shed, units from farms may be mixed with units from other farms. They are then packed into new units such as pallets.

In practice the level of mixing that occurs may vary from day to day and location to location. In any application a choice needs to be made about the level of abstraction of the mixing process that is incorporated in the model that can still produce interpretable results. Complicated models are typically difficult to parameterise and interpret, while simpler models may miss important features. The choice will depend on the analysis.

As the key building block we consider either deterministic or stochastic mixing of risk. With deterministic mixing we need to define a level of mixing. For example, if we assume that a final unit is made up of a fixed mixture of k input units we need to calculate the resulting infection distribution. The proportion of infected units in the mixture is

$$x = \frac{1}{k} \sum_i^k \pi_i y_i \alpha_i.$$

where the index k denotes the input units and π gives the mixing proportions, $\sum_i \pi_i = 1$. The distribution of this data for $k = 2$ and equal mixing is

$$Pr(x = 0) = (1 - p)^2$$

and

$$f(\alpha) = \frac{1}{p^2 + 2p(1 - p)} [p^2 f^*(\alpha) + 2p(1 - p)f(\alpha)]$$

with $f^*(\alpha)$ being the probability density of the sum

$$\frac{1}{2}\alpha_1 + \frac{1}{2}\alpha_2.$$

Note that p is the proportion of units that are infected, defined earlier.

The mixture distribution obviously becomes notationally more complicated as the number of mixing components increases, but the interpretation is straightforward. Practically, it is more general to consider using Monte Carlo methods to perform these calculations. As the range of the random variables is finite, $E[Y^2]$ is finite and it can be proved that all percentiles of the distribution can be approximated with arbitrary levels of uncertainty, controlled by the sample size chosen. Thus Monte Carlo methods are appropriate in this case.

The specification of the form of mixing will depend on the experts' opinions about the appropriate pattern for this mixing. Possible approaches are:

- Deterministic, as discussed above. This gives the greatest amount of mixing.
- Dirichlet mixing components, which use a Dirichlet distribution to specify the mixing proportions with its parameters determined by experts. This introduces extra variation which may exist in certain systems, and this variation may be significant.
- Multinomial mixing, where components come from a multi-nominal distribution over the input elements. In this case the mixing components are generated by a multinomial selection of units to mix.

The choice will depend on the problem at hand. Different choices will lead to greater or lesser amounts of mixing. In different contexts each may be plausible and reflect the process. The key issue is that it needs to be taken into account in some way.

These procedures have been implemented in R and are described in Section 7. Examples are given in Figure 3.2 and Figure 3.3.

3.2 Inspection

Inspection processes can be specified conditional on the level of infection and the action taken will depend on the situation at hand. The inspection process is defined as some function $[D|\alpha]$ where D is an indicator for whether the contaminant is detected. $D = 1$ if contamination is detected and 0 otherwise.

Detection of pests can lead to two main intervention strategies. In the first case, the entire cluster may be condemned and removed from the risk pathway. In this case, the effect of the intervention on the distribution of risk in the units is to modify the proportion of infected fruit by

$$p^* = p \times Pr(Destroyed) = p \int_0^1 [D|\alpha] f(\alpha) d\alpha$$

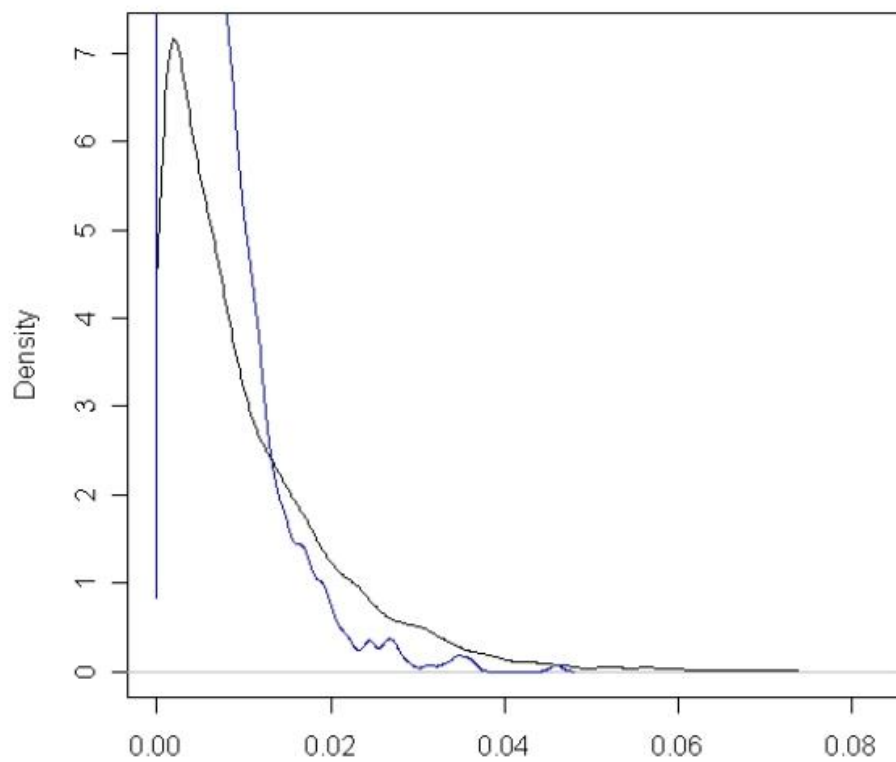


Figure 3.2: Mixing 50/50 deterministic mixing. Plotted curve is the density of α for the mixture. Variation in the mixture distribution occurs due to Monte Carlo errors.

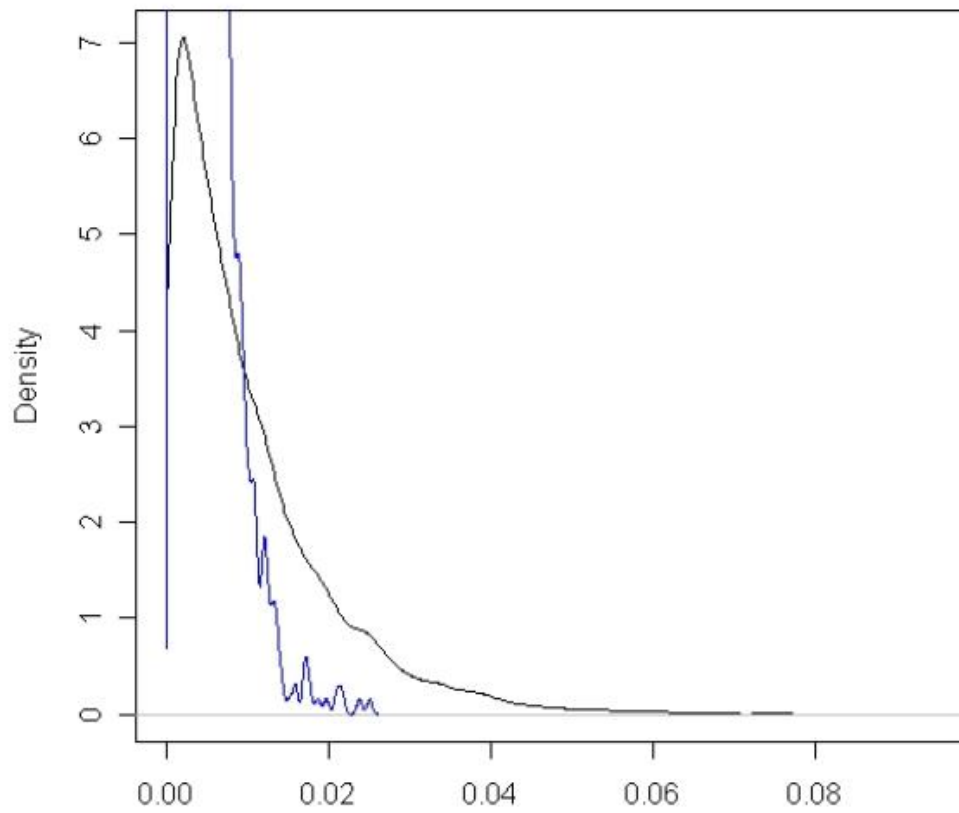


Figure 3.3: Mixing 25/25/25/25 multinomial distributions. Plotted curve is the density of α for the mixture.

and

$$f^*(\alpha) = \frac{f(\alpha)[D|\alpha]}{\int f(\alpha)[D|\alpha]d\alpha}.$$

In other words the proportion of units that are uninfected will increase and the distribution of infections will be modified. In the second strategy, detected units are removed but the unit remains. In this case further assumptions are required about the number of sub-units in the units to complete the analysis.

4 Example

4.1 Introduction

In this example we consider importing fruit originated from overseas farms which might be infested by a pest. We apply the complex pathway modelling and analysis techniques described in the previous sections to estimate the infection distribution at each stage of the importation process, including the final infection distribution in the distributed product after inspection.

This could be done as part of an integrated assessment of the risk posed by a particular trade. For example, a country might want, as part of a systems approach, to argue that the mixing of materials was a contributing factor to achieving an acceptable level of protection.

We will describe the modelling and analysis process step by step.

4.2 Modelling and analysis

Step 1: Scenario description Assume there are four fruit farms considered in this example with each of them producing the same kind of fruit. Fruit from these farms is harvested into bins and then transported into a central packing house. Within the packing house, fruit is boxed with fruit selected from bins supplied by the four farms in equal proportions. Each box contains 50 pieces of fruit. These boxes are then stacked onto pallets and put into larger containers before being shipped overseas. Pest inspections will be undertaken after importation before being distributed to final consumers. The inspections will be done by randomly selecting a box from a pallet within a container. This is an inspection of 50 individual pieces of fruit as each box contains 50 pieces of fruit and all fruit is individually inspected. How many boxes in total will be inspected depends on the intended size of inspections. Also assume that, if one or more pieces of fruit in a box is detected as infected during inspection, the whole pallet will be removed from the pathway.

What we are interested in calculating is the distribution of infection in the final product (fruit), given the original infection distribution assumed over the farms and given the way the fruit was handled (i.e., mixed) through the importation pathway. This information will give greater clarity to assessments of the threat posed by the trade.

Step 2: Construct the population distribution of rate of infestation of fruit across the farms

Assume each fruit farm has a probability p of being free from disease. Assuming available information suggests that the uncertainty in this can be represented by a Beta distribution with parameters $\alpha = 1000$ and $\beta = 10$, which has an expected value of $10/1010$. To put this into perspective this implies that for approximately every 1010 (=1000+10) of such fruit farms, 1000 farms will be uninfected.

Assume the conditional infection distribution for the level of infection in fruit from infected farms also has a Beta distribution with $\alpha = 1$ and $\beta = 100$. This implies a mean infection rate in infected farms of approximately 1% (i.e. 1/101). Note that the two Beta distributions have quite different interpretations. One represents uncertainty about the infection rate of farms. The other defines the distribution of the level of infection within a farm.

Under these two assumptions, we can construct the fruit infection distribution across the farms using the developed R functions. The function `make.analytic` will model the analytic population infection distribution with two Beta distributions described above (i.e., one for the farm and the other for the fruit within the farm); and the object-oriented `plot` function will execute another function `plot.analytic` which will display both the prior and the conditional densities in a 2 by 1 sub-plot fashion shown in Figure 4.1.

```
popdist <- make.analytic(1000,10,1,100)
plot(popdist)
```

The top graph of Figure 4.1 shows the proportion of farms free from infection within the total farms, that is, most (about 99%) of the farms are not infected. The bottom graph of Figure 4.1 is the conditional density plot given the prior distribution of farms not infected (i.e., the top graph). It shows that the majority of infection is no more than 5% (or 0.05), with the mean infection rate of 1%.

Step 3: Mixture of the four farms with equal probability This step will generate the population distribution of the mixtures based on the scenario given early, equal mixing from four farms. We assume that there is no secondary infection based on interactions between fruit.

The function `make.mix` will set up the distribution of the mixture with input parameter (0.25, 0.25, 0.25, 0.25) which defines an equal weighting in the mixture.

The object-oriented function `simul` will call `simul.mix`, in this case, and it will perform the simulation with required sample size (i.e., 100,000) and with relevant options (`pop=T` means it is a population based mixing; `random=T` means the mixing will be done randomly following a multinomial distribution from the fruit products from the four farms). Population mixing means that we will calculate the distribution over all packing sheds, each sourcing materials from the four farms. Technically it means we will resample from the farm infection distribution each time we simulate a value.

```
popmix <- make.mix(popdist, c(.25, .25, .25, .25))
test <- simul(popmix, n=100000, pop=T, random=T)
plot(test)
```

R calculations give the uninfected rate of the mixture as 97.356% (i.e. (100000-2644)/100000). Figure 4.2 shows the highest infection rate of infected units from the packing house is 3.66%. The infection rate within infected units is typically less than 1%. That is, in this particular mixing simulation, the infection rate in the mixture has been increased from the original 1% before mixing to 2.644% (i.e. 2644/100000) after the mixing. The shape of infection distribution has also been changed (see Figure 4.3 below). The increase is because the mixing leads to fruit from infected farms becoming distributed through the fruit from uninfected farms in the packing house, but at the same time, the infection rate in infected units is reduced as the total expected infection rate is constant.

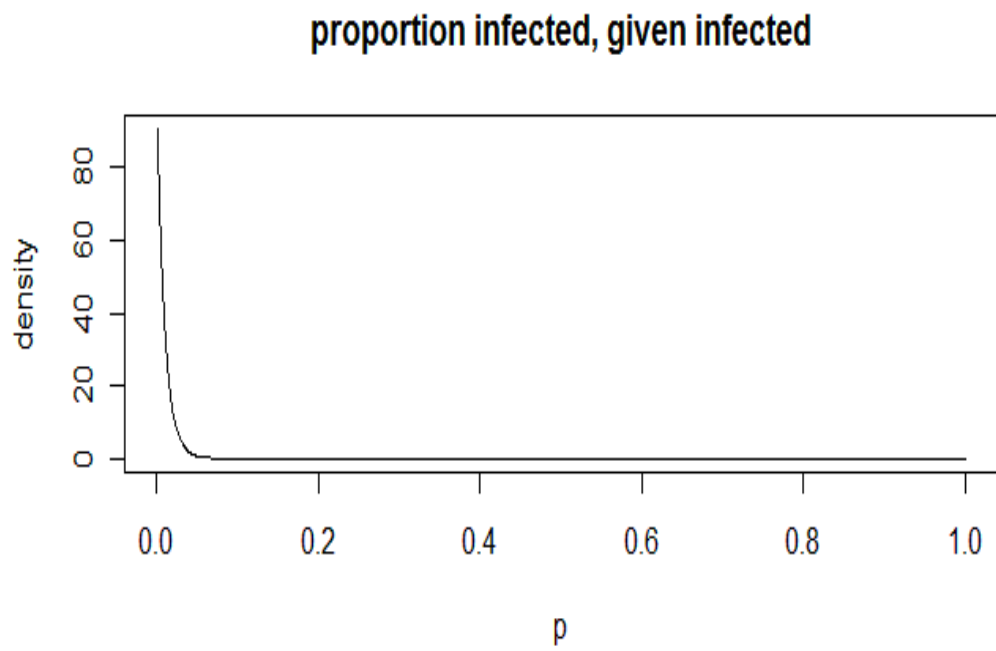
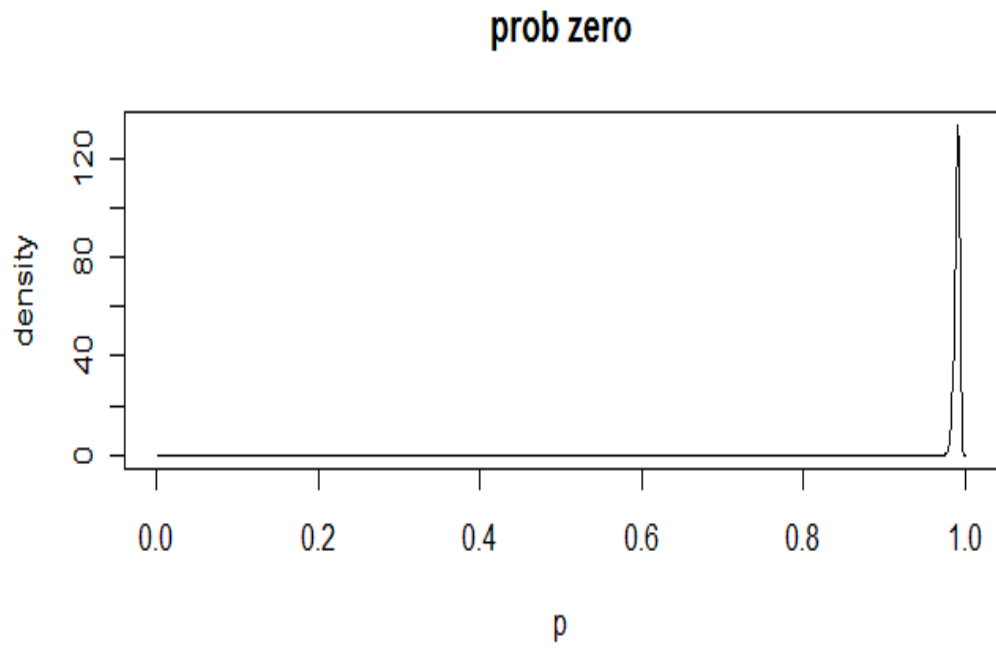


Figure 4.1: Population distribution.

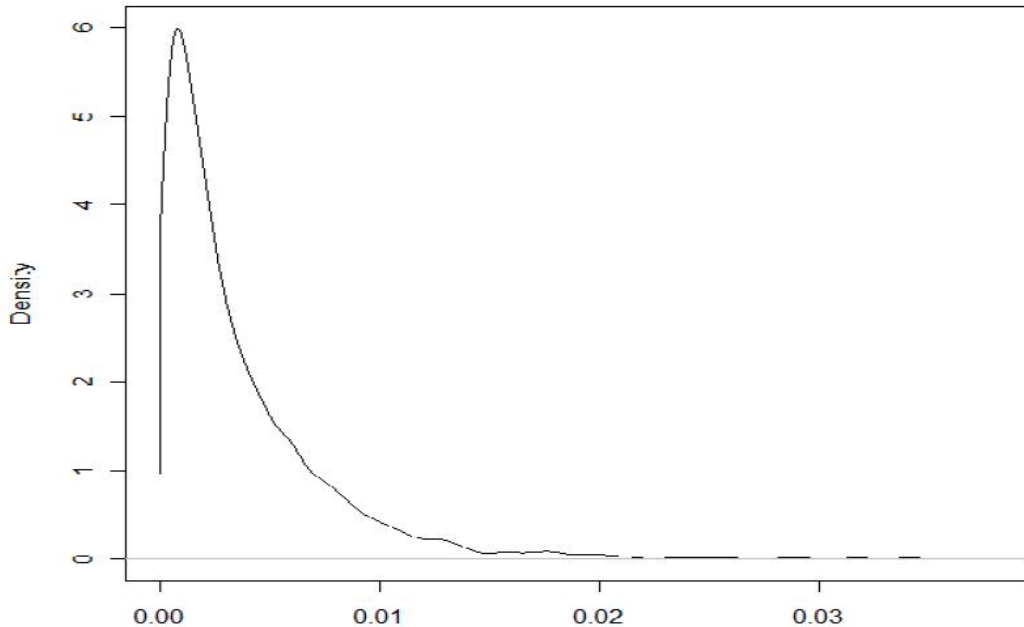


Figure 4.2: Density plot of mixture.

If we want to see the difference in terms of the infection distribution of the population between the original (before mixing) and the after mixing units, we can add the bottom plot of Figure 4.1 (the original infection population density) into Figure 4.2 (the mixture density) by using the function `add.empirical`.

```
add.empirical(popdist)
```

The combined plot is shown in Figure 4.3. The changed shape of the two distributions is obvious, and the mixing produces more homogeneous units.

Step 4: Inspections after importation We have assumed that pest inspections will be done after importation. We are interested here in calculating how many infected fruit will escape from inspections, because those infected fruit represent a biosecurity threat to the industry and the environment.

We already know that the fruit has been boxed so the inspection for each selected box will result in one of the following cases:

- One piece of fruit infected,*
- Two pieces of fruit infected,*
- ...*
- Fifty pieces of fruit infected, or*
- No infection occurred.*

Assume p_i is the infection probability of fruit i in the mixture, then $1 - p_i$ is the probability that fruit i is not infected. Hence, $(1 - p_i)^{50}$ is the probability that all 50 pieces of fruit are not infected. Therefore, $q = 1 - (1 - p_i)^{50}$ is the probability that at least one piece of fruit in a box has been detected by inspection. Here, p_i is given by Figure 4.2.

The following code will run such inspection and plot the density of those undetected, but

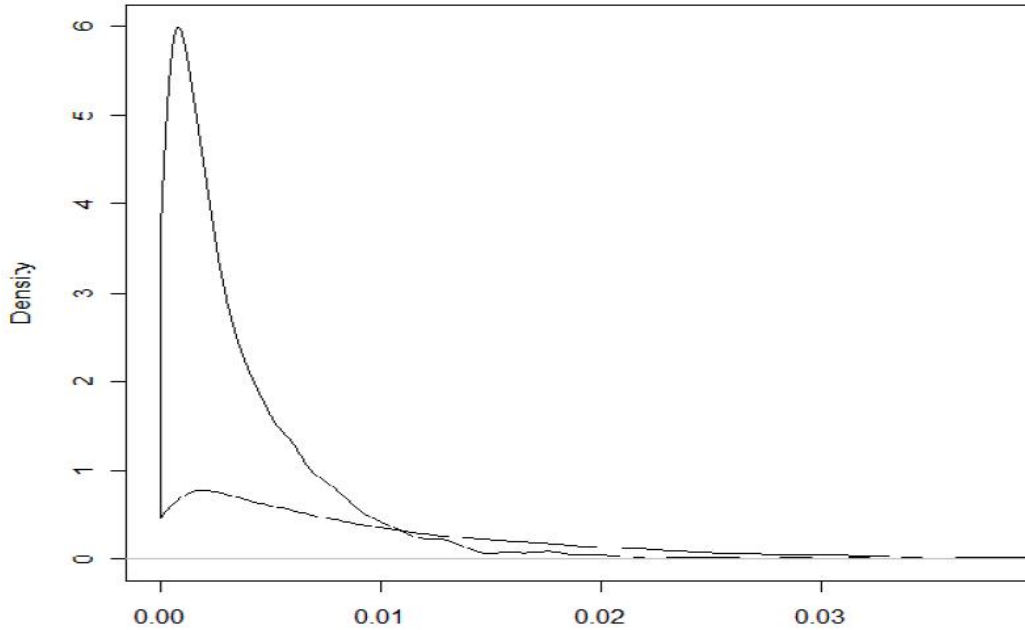


Figure 4.3: Comparing the infection population distribution of the mixture (the upper curve) with the original before mixing (the lower curve).

infected, fruit.

```
plot.empirical(inspection(test, 50))
```

Figure 4.4 shows that within the original 2,644 infected samples, 2,220 of them have escaped inspections. Thus there is a decrease in the proportion of units that are infected. The maximum infection rate in infected units now has reduced from 3.66% to about 2.90%, while the sample size has also been reduced from the original 100,000 to 99,576 (i.e., 424 infected samples have been removed after the inspection).

4.3 Summary

Using this simple but practical example, we have demonstrated the modelling and analysis methods we have proposed in this report for the complex pathways of commodity importation. This approach considers infection rates on farms, mixing and inspection. It can be seen that the developed R functions can be easily used to model and analyse the infection densities throughout the complex pathways of importation. Using this software tool we can also experiment and evaluate different mixing and inspection methods in terms of their impacts on the final infection distribution.

5 Discussion

A valid criticism of many risk assessments is that the complexity of the model far exceeds the quality of the available data. This will be a key issue in this case. In considering this, it is also important to be mindful that simple models can mislead if they do not capture key phenomena. The scale and aggregation of trade is a key aspect of the risk and should not be dismissed lightly.

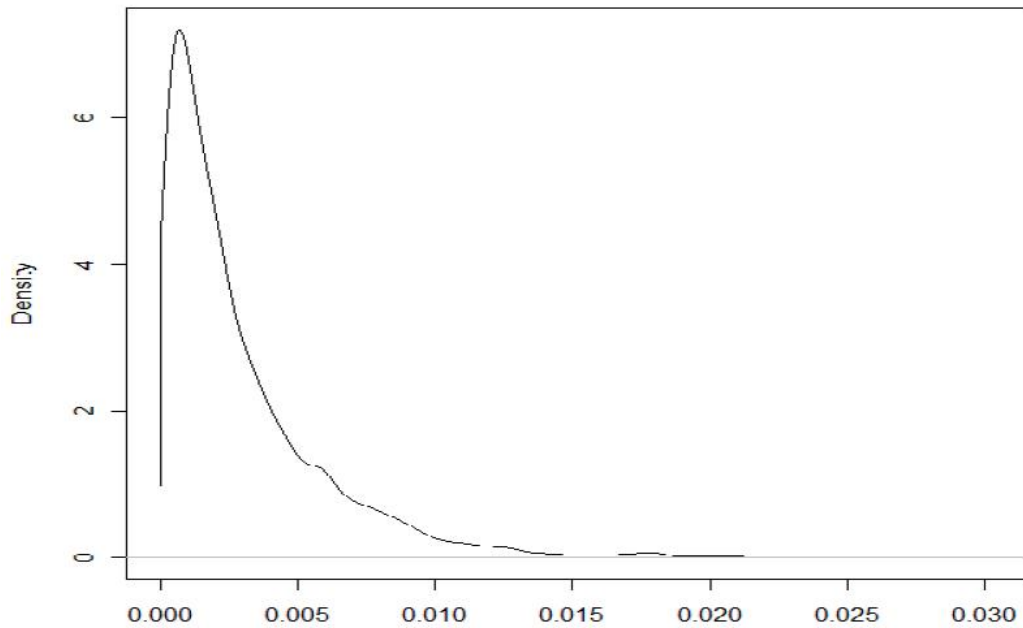


Figure 4.4: The density of infected but undetected fruit after the inspection.

The additional overheads that this model imposes on a user are potentially outweighed by the additional clarity of the answer provided by the approach. It is typically easier for experts to describe units in terms of p (the probability that the unit has any infected fruit) and α (the proportion of the unit that is infected) rather than the more complicated problems that occur when constructing a useful model marginal to the statistical structure. Marginal in this case means that experts need to consider the possible effects “in their head”, rather than considering the possible assumptions and their logical implications transparently.

Note that the mixing process homogenizes the population. With large amounts of mixing the infection status of material becomes uniform, and for rare event the Poisson distribution becomes appropriate. This could be useful in defining analyses which respect mixing but can be parameterised more simply.

Arguments involving mixing will only impact on certain biosecurity decisions. While the potential use of mixing as a primary mitigation strategy is low, it may become useful as part of a systems approach. With the withdrawal of certain fumigation approaches there is a need to find alternatives that are sustainable. For pests with thresholds to establishment, the mixing approach could provide a component of an integrated approach. For an import risk assessment it could demonstrate that the risks remain unacceptably high.

A limitation of this approach is that it cannot be implemented easily into Microsoft Excel, which is the standard software used in most organisations. The approaches considered here are more simply implemented as simulation techniques rather than analytic calculations but this should not be seen as an impediment to its application.

6 Conclusions

We have proposed a mathematical/statistical method to model complex pathways that often occur in quantitative risk assessments in biosecurity. Particularly, these complex pathways are often presented in food importation scenarios such as imported fruits originated from multiple supplies and/or farms in overseas. The mathematical models we have developed in this report including:

1. Pest infection models, which can further be categorised by one of the following three types:
 - (a) analytical model which is based on a Beta distribution;
 - (b) empirical model which is essentially a “model free” approach, that is, simulating directly from user supplied data; and
 - (c) mix model which uses a multinomial distribution approach.
2. Product mixing model.
3. Final inspection model.

A work-through example was given to illustrate the proposed modelling and analysis techniques. All the required computing functions have been coded in R and bundled together to form an R package called `cpathway` for easy adoption by the general public.

7 Appendix A: cpathway - an R package

In this section, we will describe the developed R package called `cpathway` which stands for complex pathways. The `cpathway` package contains all the required R functions to carry out the complex pathway modelling and the analysis described in the previous sections. The package was generated under R version 2.10.0. It does not require any other R packages to run apart from the standard R. The zipped package is currently named `cpathway_1-0.zip` and it can be obtained by contacting any one of the authors. After the trial process, the authors may submit it to the CRAN depository for possible use by the general public. The zipped package contains both the developed functions and their documentations in both the text and the HTML format.

7.1 List of functions developed within the cpathway package

Table 7.1 lists (in alphabetical order) all the functions developed for the modelling and analysis of complex pathways.

Function name
<code>add.empirical</code>
<code>inspection</code>
<code>make.analytic</code>
<code>make.empirical</code>
<code>make.mix</code>
<code>plot.analytic</code>
<code>plot.empirical</code>
<code>simul</code>
<code>simul.analytic</code>
<code>simul.empirical</code>
<code>simul.mix</code>
<code>unmake.empirical</code>

Table 7.1: List of R functions developed for `cpathway` package.

7.2 Function name: `add.empirical`

Description: add an empirical distribution onto the current plot.

Input(s): x – input data which is going to be added on to the current plot;

`col` – defines the line colour with 1 = black (default), 2 = red, 3 = green, 4 = blue, 5 = cyan.

Output: plotting x onto the current graph.

7.3 Function name: `inspection`

Description: conduct pest inspection and return those undetected (but infected originally).

Input(s): x – data object from the mixing simulation;

t – boxsize (i.e., t pieces of fruit per box).

Output: a list includes the distribution of undetected but infected goods.

7.4 Function name: `make.analytic`

Description: set up population distribution parameters (four parameters for Beta distributions) of a farm.

Input(s): `p.alpha` – 1st parameter of a Beta distribution to define the probability a farm is uninfected, hence, the prior information;
`p.beta` – 2nd parameter of the Beta distribution to define the probability that a farm is uninfected;
`quant.alpha` – 1st parameter of a Beta distribution that defines the proportion of the fruit infected given that at least one item is infected;
`quant.beta` – 2nd parameter of a Beta distribution that defines the proportion of the fruit infected given that at least one item is infected.

Output: a list that contains two pairs of Beta parameters and a class attribute which specifies the current approach is ‘analytic’.

7.5 Function name: `make.empirical`

Description: turns raw data into summary information (including to find the number of zeros).

Input(s): x – raw data in vector format.

Output: a list that contains the number of zeros in x , the number of non-zeros and the non-zero part of x .

7.6 Function name: `make.mix`

Description: set up a mix distribution.

Input(s): `dist` – the parameters for the mixture distribution generated by `make.analytic`;
`weight` – the weights for the mixing. The number of elements in the vector specifies the number of components in the mixture. The weights are used in simulation and their interpretation depends on the flags set there.

Output: a list containing the weights, the distribution parameters for individual farms before mixing and the class attribute.

7.7 Function name: `plot.analytic`

Description: plotting analytic distributions on a 2 by 1 sub-plot.

Input(s): x – a list contains two pairs of the Beta parameters constructed by `make.analytic`;
`...` – other parameters inherited from the generic plot function.

Output: The top density plot is the prior distribution made by the 1st pair of parameters of the Beta distribution; the bottom density plot is the proportion infected made by the 2nd pair of parameters of the Beta distribution.

7.8 Function name: `plot.empirical`

Description: plot the data using R function `Density` - Kernel smoothing method.

Input(s): x – raw data to be fitted by a Kernel smoothing density.

Output: a density plot.

7.9 Function name: `simul`

Description: a generic simulation function, depending on the class, it will call one of `{simul.analytic; simul.empirical; simul.mix}`.

Input(s): x – a simulation object;
... – other possible parameters carried over.

Output: execute one of `{simul.analytic; simul.empirical; simul.mix}` depending on class attribute.

7.10 Function name: `simul.analytic`

Description: conduct simulation based on an analytic distribution.

Input(s): x – the analytic object;
 n – the number of observations;
`lock` – whether values generated are conditional on a single draw (`lock=T`) or marginalised over the uncertainty (`lock=F`).

Output: a vector of Beta variables based on parameters given.

7.11 Function name: `simul.empirical`

Description: conduct simulation based on an empirical distribution.

Input(s): x – the empirical object;
 n – the number of simulations;
... – other possible parameters carried over.

Output: a vector of empirical random variables redraw from given dataset.

7.12 Function name: `simul.mix`

Description: do mixing simulation.

Input(s): x – the mix object;
 n – the number of observations;
`random` – whether the mixing weights are fixed (`random=F`) or random (`random=T`);
`pop` – whether the sample is from the population of units after mixing (`pop=T`) (i.e. marginal to mixing) or from a particular mixture (`pop=F`).

Output: a list containing mixing simulation results.

7.13 Function name: `unmake.empirical`

Description: a reverse operation as `make.empirical`, i.e., put those zeros back into the original dataset.

Input(s): x – the dataset with zeros have been removed by `make.empirical`.

Output: a vector with zeros and non-zeros as originally presented in x .

7.14 Example code

The following R code produced the example with plots in Section 4.

```
# example code

# Construct the population distribution of fruit
# in any of these four farms
popdist <- make.analytic(1000,10,1,100)
plot(popdist)

# Mixture of the four farms with equal probability
popmix <- make.mix(popdist, c(.25,.25,.25,.25))
test <- simul(popmix, n=100000, pop=T, random=F)
plot(test)

# Add the original population density from a single farm
add.empirical(popdist, col=2)

# Inspection after importation
plot.empirical(inspection(test,50))
```


8 Appendix B: R code developed for cpathway

```
# R functions developed for complex pathways
# developed by Simon Barry
# maintained by Xunguo Lin
# 2010
#
#
#####
# make.analytic - set up population distribution parameters
# (4 Beta distribution parameters) of a farm
#
# input:
#
# p.alpha, p.beta - a pair of parameters for Beta distribution to define
# the probability that a farm is un-infected.
# Hence it is the prior information.
#
# quant.alpha, quant.beta - a pair of parameters for Beta distribution that
# defines the proportion of the fruit infected
# given that at least one item is infected
#####

make.analytic <- function(p.alpha, p.beta, quant.alpha, quant.beta) {

result <- list(p.beta.a=p.alpha, p.beta.b=p.beta, quant.a=quant.alpha, quant.b=quant.beta)
class(result) <- "analytic"
result

}

#####
# make.empirical - turns raw data into summary information (inc finding 0s)
#
# input:
#
# x - raw data
#
# output:
#
# p - proportion of zeros
# x - non-zero data
# nx - number of non-zero data
# n - number of data points
#####

make.empirical <- function(x){

n <- length(x)
ind <- x==0
p <- sum(ind)/n
result <- list(p=p, x=x[!ind], nx=n-sum(ind), n=n)
class(result) <- "empirical"
result

}

#####
# unmake.empirical - a reverse operation as make.empirical,
# i.e. put those 0s back into the original dataset
# input:
#
# x - the dataset with 0s have been removed by make.empirical
#####

unmake.empirical <- function(x){

result<-c(rep(0,x$n))

}
```

```

if(x$p<1) result[1:x$nx]<-x$x

result

}

#####
# make.mix - set up a mix distribution
#
# input:
#
# dist - the mixture distribution parameters generated by make.analytic
#
# weight- the weights for the mixing. The number of elements in the vector specifies
#         the number of components in the mixture. The weights are used in simulation
#         and their interpretation depends on the flags set there.
#
# output:
#
# a list contains - weights, Beta parameters, simulated (using beta para) values
#                 which has the same length as weights
#
# Note, the object returned by the function contains a single realisation
# of the mixture. This could be useful in certain simulations
#####

make.mix <- function(dist, weight) {

result <- list(weights=weight, dist=dist, x=simul(dist, n=length(weight), F))
class(result) <- "mix"
result

}

#####
# plot.analytic - plot analytic (object) distribution on a 2 by 1 sub-plot
#
# input:
#
# x - a list contains 2 pairs of the Beta parameters constructed by make.analytic
#
# plots:
#
# the top density plot is the prior distribution made by the 1st pair of para
# the bottom density plot is the proportion infected made by the 2nd pair of para
#####

plot.analytic <- function(x,...) {

oldpars <- par()
par(mfrow=c(2,1))

plot(1:1000/1000,dbeta(1:1000/1000, x$p.beta.a, x$p.beta.b), type="l", ylab="density", xlab="p")

title("prob zero")

plot(1:1000/1000,dbeta(1:1000/1000, x$quant.a, x$quant.b), type="l", ylab="density", xlab="p")

title("proportion infected, given infected")

par(oldpars)

}

#####
# plot.empirical - plot the data using kernel smoothing method
#####

```

```

plot.empirical <- function(x) {

val <- density(x$x, from=0.00001)
val$x <- c(0,val$x)
val$y <- c(x$p,val$y*(1-x$p))
plot(val, main="")

}

#####
# add.empirical - adds an empirical distribution onto the current plot
# This function is normally used to compare 2 curves
# input:
#
# x - input data which is going to be added on to the current plot
# col - define the colour with 1 = black (default), 2 = red, 3 = green,
# 4 = blue, 5 = cyan
#####

add.empirical <- function(x, col=1) {

if(class(x)=="analytic") x <- make.empirical(simul(x,1000000))

val <- density(x$x, from=0.00001)
val$x <- c(0,val$x)
val$y <- c(x$p,val$y*(1-x$p))
lines(val, col=col)

}

#####
# simul - generic simulation function, depending on the class,
# it will call one of
#
# simul.analytic
# simul.empirical
# simul.mix
#####

simul <- function(x,...){

UseMethod("simul")

}

#####
# simul.analytic - do simulation based on analytic distribution
#
# input:
#
# x - the analytic object
# n - the number of observations
# lock - whether values generated are conditional on a single draw (lock=T)
# or marginalised over the uncertainty (lock=F)
#####

simul.analytic <- function(x,n=1,lock=F){

result <- rep(0,n)

if(lock) {
p <- rbeta(1, x$p.beta.a, x$p.beta.b)
ind <- p > runif(1)
if (!ind) result <- rep(rbeta(1,x$quant.a, x$quant.b),n)
} else {
p <- rbeta(n, x$p.beta.a, x$p.beta.b)
}
}

```

```

ind <- p > runif(n)
result[!ind] <- rbeta(n-sum(ind),x$quant.a, x$quant.b)
}

result

}

#####
# simul.empirical - do simulation based on empirical distribution

#
# input:
#
# x - the empirical object
# n - the number of simulations
#####

simul.empirical <- function(x,n=1,...) {

result<-rbinom(n, 1, 1-x$p)

if(sum(result>0)) {

result[result>0]<-sample(x$x, size=sum(result), replace=T)

}

result

}

#####
# simul.mix - do mixing simulation
#
# input:
#
# x      - the mix object
# n      - the number of observations
# random - whether the mixing weights are fixed (random=F) or random (random =T)
# pop    - whether the sample is from the population of units after mixing (pop=T)
#         (ie marginal to mixing) or from a particular mixture (pop=F)
#
#
#####

simul.mix <- function(x,n=1, random=F, pop=T){

if(!pop) {

if(!random) {
  result<-rep(sum(x$weights*x$x), n)
}
else {
  new.weights<-t(rmultinom(n,length(x$weight), x$weight))/length(x$weight)
  result<-as.vector(new.weights%**x$x)
}
} else #pop
{
if(!random) {

result<-matrix(simul(x$dist,n=length(x$weight)*n), nrow=n)**% x$weights
} else #random
{
new.weights<-t(rmultinom(n,length(x$weight), x$weight))/length(x$weight)

result<-as.vector(apply(new.weights*simul(x$dist,n=length(x$weight)*n),1,sum))
}
}
}

```

```

result<-make.empirical(result)
result

}

#####
# inspection - conduct pest inspection of fruit and return
#   those undetected (but infected originally)
# input:
#
# x - data object from the mixing simulation
# t - boxsize (i.e., t pieces of fruit per box)
#####

inspection <- function(x, t=20){

pdetect <- 1-(1-x$x)^t
index <- rbinom(x$nx,1,pdetect)
result <- x
result$x <- x$x[!as.logical(index)]
result$n <- x$n-sum(index)
result$nx <- x$nx-sum(index)
result

}

```

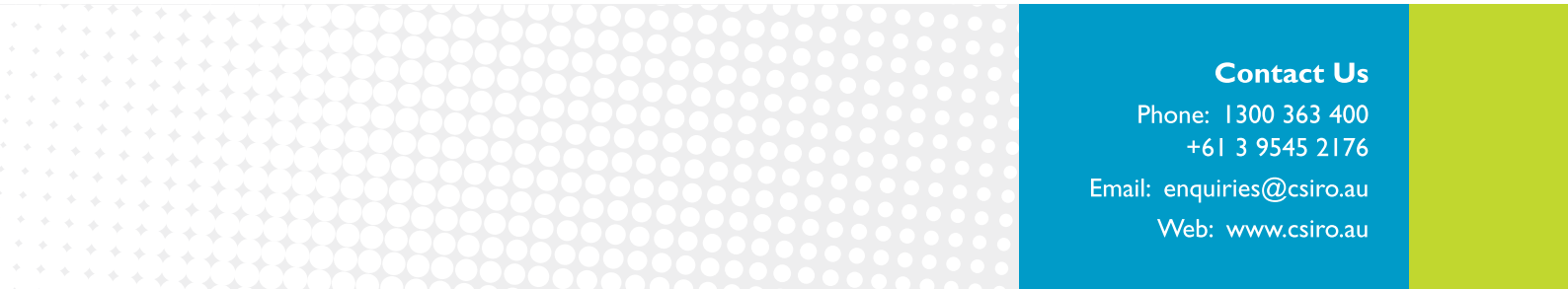
References

- Barry, S. C. (2010). Putting the quantitative into qualitative import risk assessment. Technical report, CSIRO Report for the Australian Center of Excellence for Risk Analysis Project 0705.
- Gray, G. M., Allen, J. C., Burmaster, D. E., Gage, S. H., Hammitt, J. K., Kaplan, S., Keeney, R. L., Morse, J. G., North, D. W., Nyrop, J. P., Stahevitch, A., and Williams, R. (1998). Principles for conduct of pest risk analyses: Report of an expert workshop. *Risk Analysis*, 18(8):773–780.
- Hartnett, E., Adkin, A., Seaman, M., Cooper, J., Watson, E., Coburn, H., England, T., Marrooney, C., Cox, A., and Wooldridge, M. (2007). A quantitative assessment of the risks from illegally imported meat contaminated with foot and mouth disease virus to Great Britain. *Risk Analysis*, 27(1):187 – 202. complex pathways.
- Michael, J. F. and Podleckis, E. V. (2000). *Quantitative methods for conservation biology*, chapter Likelihood of introducing nonindigenous organism with agricultural commodities: probabilistic estimation, pages 77–95. Springer-Verlag.
- Miller, L., McElvaine, M. D., McDowell, R. M., and Ahl, A. S. (1993). Developing a quantitative risk assessment process. *Scientific and Technical Review - International Office of Epizootics*, 12:1153–1164.
- Office International des Epizooties (OIE) (2004a). *Handbook on import risk analysis for animals and animal products: Volume 1. Introduction and qualitative risk analysis*. Office International des Epizooties. no ref.
- Office International des Epizooties (OIE) (2004b). *Handbook on import risk analysis for animals and animal products: Volume 2. Quantitative risk assessment*. Office International des Epizooties. no ref.
- Vose, D. (2008). *Risk analysis: a quantitative guide*. John Wiley & Sons, 3rd edition.

9 Acknowledgements

This report is the product of the Australian Centre of Excellence for Risk Analysis (ACERA). In preparing this report, the authors acknowledge the financial and other support provided by the Department Of Agriculture, Fisheries and Forestry (DAFF) and the University of Melbourne.

This report has been improved by Keith Hayes and two anonymous reviewers. Any remaining errors are our own.



Contact Us

Phone: 1300 363 400
+61 3 9545 2176

Email: enquiries@csiro.au

Web: www.csiro.au

Your CSIRO

Australia is founding its future on science and innovation. Its national science agency, CSIRO, is a powerhouse of ideas, technologies and skills for building prosperity, growth, health and sustainability. It serves governments, industries, business and communities across the nation.