AMSI
VACATION
RESEARCH
SCHOLARSHIPS
2018-2019

An R package for occupancy model parameter estimation using the two-stage approach

Phillip Newbold

Supervised by Natalie Karavarsamis

La Trobe University

Vacation Research Scholarships are funded jointly by the Department of Education and Training and the Australian Mathematical Sciences Institute.

AMSI AUSTRALIAN MATHEMATICAL SCIENCES INSTITUTE

Australian Government
Department of Education and Training

Abstract

In the field of statistical ecology, occupancy models are an important method for measuring species dispersion. Occupancy models are dependant upon two parameters, the probability of occupancy and the probability of detection. These parameters have, until recently, been estimated simultaneously. A recently developed method allows for a two-stage approach to parameter estimation. In this approach, the probability of detection is estimated in the first stage, followed by the estimation of the probability of occupancy in the second stage. It was the aim of this research project to construct an `R` package that applies this two-stage method of parameter estimation to occupancy data. Using existing code, the package was constructed and successfully applied in the analysis of several data sets. Following further improvements, the package will be submitted to `CRAN` and made available to the research community.

1 Introduction

Measuring species dispersion is one of the key challenges of statistical ecology (Gimenez et al., 2014). Although abundance studies are often used to measure species dispersion, occupancy models constitute a more time and cost-efficient method (MacKenzie et al., 2002). In MacKenzie et al. (2002), Mackenzie introduced a general likelihood function for their occupancy model. Depending upon two parameters, namely probability of occupancy and detection, the likelihood function can be used to estimate the probability of presence of the species of interest at a specific site (MacKenzie et al., 2002). Estimates of these two parameters can be found simultaneously using numerical maximisation techniques, applied to the general likelihood function (MacKenzie et al., 2002). A recent method developed in Karavarsamis and Huggins (2018a) and Karavarsamis and Huggins (2018b), the two-stage approach, separates the likelihood function into two partial likelihood functions, which can then be maximised to independently find estimates for the probabilities of occupancy and detection.

The R (R Development Core Team, 2018) package `unmarked` (Fiske and Chandler, 2015) includes the `occu` function, which fits the occupancy model from MacKenzie et al.

(2002). However, this method of analysis is not always adequate (Karavarsamis and Huggins, 2018b), thus requiring the need for an alternative tool for researchers.

This research project aims to contribute to the field of statistical computation by developing an R package, named `twoStage`, that applies the method of parameter estimation from Karavarsamis and Huggins (2018a) and Karavarsamis and Huggins (2018b). The construction of the package will consist of taking the existing code used in Karavarsamis and Huggins (2018b) and Karavarsamis and Huggins (2018b) and develop accordingly to convert to an R package, including both their two-stage occupancy model and examples of its application. This package will give those in the field of statistical ecology an alternative tool for approximating the occupancy and detection probabilities to that which is currently available with `unmarked`. With neither method of analysis being superior to the other in all contexts, this package will provide another option if one approach to parameter estimation is not successful with a particular data set.

2 Likelihoods

Here we give likelihood functions for models fit in `twoStage`, details and derivations are from Karavarsamis and Huggins (2018a) and Karavarsamis and Huggins (2018b).

2.1 Homogeneous model

We have the probability of occupancy ψ at a particular site s=1,2,...,S, where S is the total number of sites visited. Each site was visited τ times, with the probability of detection, given the species was present at a particular site at a particular time, being p. The probability of an occupied site having at least one detection is $\theta = 1 - (1-p)^{\tau}$. We denote the number of sites with no detections as f_0, so the total number of sites with at least one detection is $O = S - f_0$. The total number of detections across all sites y is the sum of total detections for each particular site y_s. Thus, we can then write the full likelihood (MacKenzie et al., 2006) as

$$L(\psi, p) = \{\psi^{S-f_0} p^y (1-p)^{\tau O - y}\}(1-\psi\theta)^{f_0} \quad (1)$$

Let $\eta = \psi\theta$, then the log-likelihood (Karavarsamis and Huggins (2018a)) is

$$\ell(\eta, p) = f_0 \log(1-\eta) + (S - f_0)\log(\eta) \tag{2}$$

$$+ y\log(p) + (O\tau - y)\log(1-p) - O\log(\theta). \tag{3}$$

Equation 3, the conditional log-likelihood, is maximized by the maximum likelihood estimator of p, which we calculate with the `VGAM` package (Yee, 2010). Following Karavarsamis and Huggins (2018a) this estimate yields the estimate for occupancy $\widehat{\psi} = \widehat{\eta}/\widehat{\theta}$.

Now, let the number surveys to be done after the initial detection at site s be given by b_s, then $b = \sum_{s=1}^{S} b_s$. Also, let $a = O\tau - O - b$, which gives

$$L(\eta, p) = (1-\eta)^{f_0}\eta^{S-f_0} \tag{4}$$

$$\times \frac{p^O(1-p)^a}{\theta^O} \tag{5}$$

$$\times p^{y-O}(1-p)^{b-(y-O)}. \tag{6}$$

From the partial likelihood, Equation 6, we find the estimate of p, $\widehat{p} = (y - O)/b$. The variance is $Var(\widehat{p}) = \widehat{p}(1-\widehat{p})/b$. Using $(1-\eta)^{f_0}\eta^S - f_0$, we find the estimate for occupancy, $\widehat{\psi} = (S - f_0)/S\widehat{\theta}$ (Karavarsamis and Huggins, 2018a).

The variance for occupancy is given by

$$\text{Var}\left(\widetilde{\psi}\right) \approx \left(\frac{\psi(1-\psi\theta)}{S\theta} + \psi^2\right)\frac{\tau^2(1-p)^{2(\tau-1)}}{\theta^2}\frac{p(1-p)}{b} + \frac{\psi(1-\psi\theta)}{S\theta}. \tag{7}$$

2.2 Heterogeneous model

Now we introduce the heterogeneous model as developed in Karavarsamis and Huggins (2018b). Due to the inclusion of site and time varying covariates in the heterogeneous model we must now introduce some more notation, while maintaining the definitions from section 2.1. The detection probability p may now vary depending on time so we now denote the probability of species detection given it is present at a site s on a survey occasion j as p_{sj}, or simply p_s if detection does not vary with time. The probability of a site s being occupied is ψ_s, this does not vary with time as we assume that once a site is occupied it remains occupied. The expression for θ for a specific site is $\theta_s = 1 - \prod_{j=1}^{\tau}(1 - p_{sj})$.

With the introduction of covariates, occupancy is now written as a function of a vector of site dependant covariates, x_s, and a vector of coefficients, $\boldsymbol{\alpha}$, that is $\psi_s = h(x_s^T \boldsymbol{\alpha})$. Detection can be dependant upon site and time varying covariates so it is now written as a function of a vector of covariates, u_{sj}, where s is the site being visited and j is the visiting occasion, and a vector of coefficients $\boldsymbol{\beta}$, so $p_{sj} = h(u_{sj}^T \boldsymbol{\beta})$. As is mentioned in Karavarsamis and Huggins (2018b), the function h is usually the logistic function. For a time dependant model we define $\boldsymbol{p}_s = (p_{s1}, \ldots, p_{s\tau})^T$ and for a time independent model we define $\boldsymbol{p}_s = p_s$.

Each site contributes to the full likelihood equation, where the contribution of a particular site s is given by

$$L_s(\psi_s, \boldsymbol{p}_s) = (1 - \psi_s \theta_s)^{z_s} \left\{ \psi_s \prod_{j=1}^{\tau} p_{sj}^{y_{sj}} (1 - p_{sj})^{1 - y_{sj}} \right\}^{1 - z_s}. \quad (8)$$

Similarly to the process in section 2.1 the full likelihood contribution from site s can be rewritten with η_s $(= \psi_s \theta_s)$ (Karavarsamis and Huggins (2018b)), as

$$\begin{aligned} L(\eta_s, \boldsymbol{p}_s) &= (1 - \eta_s)^{z_s} \eta_s^{1 - z_s} \times \left\{ \frac{\prod_{j=1}^{\tau} p_{sj}^{y_{sj}} (1 - p_{sj})^{1 - y_{sj}}}{\theta_s} \right\}^{1 - z_s} \\ &= L_1(\eta_s) L_2(\boldsymbol{p}_s). \end{aligned} \quad (9)$$

Using this decomposition we can first estimate $\boldsymbol{\beta}$ and use this to estimate $\boldsymbol{\alpha}$. When the detection probability is time independent the conditional likelihood for site s is simply

$$L_2(\boldsymbol{\beta}) = \prod_{s=1}^{O} \frac{p_s^{y_s} (1 - p_s)^{\tau - y_s}}{\theta_s}, \quad (10)$$

When the detection is time dependant the conditional likelihood for site s is

$$L_2(\boldsymbol{p}_s) = \frac{\prod_{j=1}^{\tau} p_{sj}^{y_{sj}} (1 - p_{sj})^{1 - y_{sj}}}{\theta_s}. \quad (11)$$

Both $L_2(\boldsymbol{\beta})$ and $L_2(\boldsymbol{p}_s)$ can be maximised using the `VGAM` package in `R`. Once the estimate for detection is found this way, the iterative weighted least squares (IWLS) method can be used along with these estimates, to estimate $\boldsymbol{\alpha}$. Proof of the applicability of the IWLS method is shown in Karavarsamis and Huggins (2018b).

The methods for estimating standard errors for occupancy and detection in the heterogeneous case are lengthy and as such won't be outlined here. However, they can be found in Karavarsamis and Huggins (2018b).

3 The `twoStage` package

It was the aim of this research project to construct an `R` package that implements this two-stage method for occupancy parameter estimation. This was done with the use of pre-existing code developed by Natalie Karavarsamis. This code was developed into a package, which we named `twoStage`, with the use of `Rtools`[1], the collection of tools for constructing packages in `R`, and the package `devtools` (Hadley Wickham and team (2018)).

To apply the two-stage method of occupancy parameter estimation with the `twoStage` package, a user interacts with four main functions; `model.calc`, `est.time.homog`, `est.time.homog.int` and `est.time.varying`. Figure 1 shows the algorithm that the package follows to implement the two-stage method. What follows now is an overview of the main functions and the functions they call at each stage of the process, along with some data formatting information for users.

3.1 Data formatting and `model.calc`

For a user to analyse their data with the `twoStage` package their data must first be formatted as a data frame, where the detection matrix is contained in the first τ rows, then followed by the site varying covariates and lastly the time varying covariates. Time dependent covariates should have a character root then the visit number. For example, the matrix of measurements for a time varying covariate '`timecov`' would have column names: `timecov1`,`timecov2`,`timecov3`,etc..

[1]https://cran.r-project.org/bin/windows/Rtools/

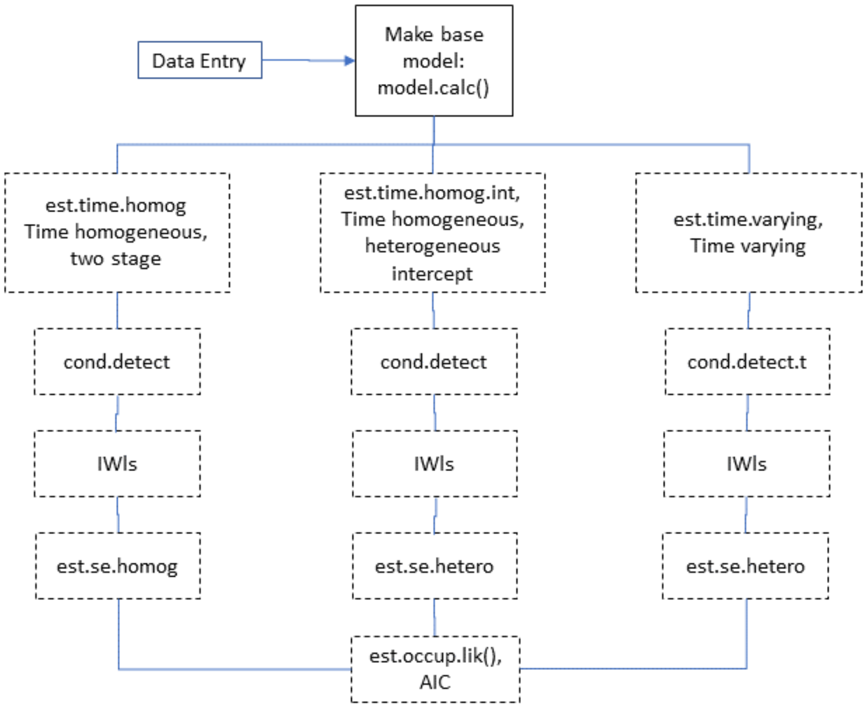

Figure 1: Flowchart of `twoStage` algorithm

As an example of a correctly formatted data frame we present, in table 1, the first three rows of a brook trout occupancy study data-set (available on the James Peterson[2] website), used in an example later in this report. The first three columns make up the detection matrix, the column titled 'Ele' is the site dependant covariate measurements and the final three columns are the time dependant covariate measurements.

y01	y02	y03	Ele	CSA1	CSA2	CSA3
0	0	0	4266.14	2.87	2.61	2.731
1	0	0	4038.82	0.88	2.54	1.161
0	0	0	2031.58	1.50	0.96	1.521

Table 1: Example data frame layout

Once the data is formatted correctly, a model can be specified by the user to the function `model.calc`, with arguments; `form.occup`, `form.detect`, `t.root`, `tau` and `data`. `form.occup` and `form.detect` signify the site varying covariates to be included in the modelling of occupancy and detection respectively, allowing `model.calc` to organise these

[2] http://people.oregonstate.edu/~peterjam/occupancy_workshop/hands_on.html

covariates into design matrices. `t.root` gives the character root of the name of all time dependant covariates. `tau` is the number of survey occasions (τ in the likelihood functions discussed in section 2) and `data` is the data frame, formatted as outlined above. The output of `model.calc` is saved in an object named `Model.out` which is the basis for all further analysis.

3.2 Analysis functions

Once the user has prepared the data for analysis with the `model.calc` function they must now choose the analysis function they will use to analyse the data. The choice of analysis function is dependant upon whether they wish to include time and/or site varying covariates in the modelling, as well as whether they wish to allow the intercept of the model of detection to vary with time.

All three analysis functions follow a similar algorithm for applying the two-stage method to data, as is shown in figure 1. In the first stage of the two-stage method, the detection coefficients are estimated with one of two functions; `cond.detect` or `cond.detect.t`. These functions prepare the input for the `vglm` function (from `VGAM`) that fits a vector generalised linear model[3]. To estimate the occupancy coefficients, the second stage of the two-stage method, all three analysis functions call the `iwls` function. This function applies the IWLS method to estimate the occupancy coefficients, using the detection coefficient estimates found in the first stage. The IWLS method is used as it was shown, in Karavarsamis and Huggins (2018b), to be the most numerically stable method for occupancy estimation.

Following the estimation of the detection and occupancy coefficients, the standard errors are estimated with either the `est.se.homog` or `est.se.hetero` functions. These functions follow methods of standard error estimation found in Karavarsamis and Huggins (2018b). Then the AIC values, t-values and p-values are calculated for the coefficients. The AIC values give the user a convenient method for comparing models fit with various combinations of the covariates present in their data. The analysis functions give outputs similar in form to figure 2, which lists the estimates of the coefficients, along with their

[3]More information on vector generalised linear models and the workings of the `VGAM` package can be found in Yee et al. (2015)

standard errors, t-values and p-values. These analysis functions are listed in table 2 and are described below.

Analysis function	Used for
`est.time.homog`	Time homogeneous covariates and intercept
`est.time.homog.int`	Time homogeneous covariates with heterogeneous intercept
`est.time.varying`	Time varying covariates

Table 2: occupancy report functions

3.2.1 `est.time.homog`

This analysis function is used when the user desires the fitting of the homogeneous model (section 2.1), or the time independent site heterogeneous model (section 2.2).

To use this function for fitting a homogeneous model the user must input `1` for the `form.occup` and `form.detect` arguments of the `model.calc function`. `est.time.homog` ignores any time dependant covariates, so the `t.root` argument for `model.calc` can be set to `NULL`. To estimate the detection coefficients `est.time.homog` calls `cond.detect`, which, as there are no time dependant covariates and the intercept is assumed to be time independent, uses the `posbinomial` family in the `vglm` function for detection coefficient estimation (equation 3). Due to the homogeneity of the model being fitted, the standard error estimations are calculated with the `est.se.homog` function.

If the user wished to fit a site heterogeneous model, they must add any site varying covariates they wish to include in the modelling of occupancy and detection to the `form.occup` and `form.detect` arguments of the `model.calc` function. Again, to estimate the detection coefficients, `cond.detect` uses the `posbinomial` family for the `vglm` function (equation 10). As the covariates and intercept are time homogeneous, the standard error estimates are calculated with the `est.se.homog` function.

3.2.2 `est.time.homog.int`

This analysis function is similar to `est.time.homog`, in that it fits both homogeneous (section 2.1) and site varying models (section 2.2). However, `est.time.homog.int` is different to `est.time.homog` in that it allows the intercept of the model of detection to be time dependant.

When using the `est.time.homog.int` function to analyse data the inputs are similar to those for `est.time.homog`, described in section 3.2.1. To accommodate the time varying intercept, the input for the `int` argument of the `cond.detect` function in `est.time.homog.int` is set as `TRUE` instead of `FALSE`. This leads the `cond.detect` function, when estimating the detection coefficients with the `vglm` function, to use the `posbernoulli.t` family, with the argument `parallel.t=FALSE~1`. This specifies the time dependence of the intercept and time independence of the site varying covariates. Due to the time dependence of the intercept, the standard errors are found with the `est.se.hetero` function.

3.2.3 `est.time.varying`

Finally, this function allows the user to fit time dependant heterogeneous models (section 2.2). It also allows for the inclusion of site dependant covariates and/or a time dependant intercept.

To use this function to fit a model including time dependent covariates the user must include, in the `t.root` argument of the `model.calc` function, the name of all time dependant covariates to be included in the model. Any site dependant covariates the user wishes to include can also be input in the `form.occup` and `form.detect` arguments, as was done in section 3.2.1 for the `est.time.homog` function. The `est.time.varying` function also has an argument `int`, where the user can input `TRUE` or `FALSE` to signify an intercept that is time dependant or constant respectively.

To calculate the detection coefficient estimates, the `est.time.varying` function calls the `cond.detect.t` function. Similarly to the `cond.detect` function, `cond.detect.t` prepares an input for the `vglm` function. In this case, only the sites with at least one detection are included in the estimation of the detection coefficients (equation 11). Due to the inclusion of time dependent covariates, the input that `cond.detect.t` prepares for the `vglm` function is more complex. As there is a requirement for further arguments for time dependant model fitting with `vglm`. New variables are constructed for each time varying covariate, which are used in the `xij` argument of the `vglm` function. This argument, `xij`, indicates that the covariates vary for each site visit. The new variables take the form `timecova.tij`, `timecovb.tij`, `timecovc.tij`, ..., where 'timecova, timecovb, timecovc, ...' are the time dependant covariates included in the model. cond.detect.t uses the posbernoulli.t family in the vglm function, as each detection occasion is an independent Bernoulli trial due to detection being time dependant. The users input for the int argument stipulates whether the posbernoulli.t family has the (parallel.t=FALSE∼1) argument indicating a time dependant intercept or the (parallel.t=FALSE∼0) argument indicating a constant intercept. As we assume occupancy is time independent, the method for calculating the occupancy coefficient estimate is still the IWLS method, applied with the iwls function.

For the est.time.varying function, the standard errors are calculated using the est.se.hetero function. This is due to the time dependence of the included covariates, as well as the possible time dependence of the intercept.

4 Applications

We will now see how this package can be applied to two different data sets. We first look at a Coosa bass data set, where we use both est.time.homog and est.time.homog to estimate the occupancy and detection coefficients. We then look at a Brook trout data set, where, due to the time dependant covariates included in the data set, we use est.time.varying to estimate the occupancy and detection coefficients.

Each of the analysis functions give an occup.report class object as their output. This object contains four slots; desc, ests, AIC.beta and AIC.alpha. desc is the description of what model is being fitted. ests gives the matrix of values found for the coefficient estimates (est), along with the standard errors (se), t-values (t) and p-values (p) of those estimates. The occupancy coefficient estimates are given first, followed by the detection coefficient estimates. AIC.beta and AIC.alpha give the AIC value for the detection and occupancy coefficient estimates respectively.

4.1 Coosa bass application

Here we will look at an example data set that is site heterogeneous but time homogeneous. The data set is Coosa bass data (available on the James Peterson[4] website). For this

[4]http://people.oregonstate.edu/∼peterjam/occupancy_workshop/hands_on.html

```
> est.time.homog(Model.out,method = 'iWls')
An object of class "occup.report"
Slot "desc":
[1] "Time homogeneous covariates and intercept; two stage"

Slot "ests":
                   est        se          t           p
(Intercept)  3.9411440 1.7506899  2.2511947 0.024373205
Mag          3.5700437 1.2763532  2.7970656 0.005156906
CV          -0.1425384 0.6804286 -0.2094832 0.834071059
(Intercept)  1.2490291 0.4421258  2.8250535 0.004727273
Mag         -0.2792275 0.2148791 -1.2994634 0.193784950
CV           0.2215304 0.2730449  0.8113335 0.417174205

Slot "AIC.beta":
[1] 170.1675

Slot "AIC.alpha":
[1] 28.84258
```

Figure 2: Input and output of analysis of Coosa bass data with `est.time.homog` function

example there are four surveying occasions and two site varying covariates 'Mag' and 'CV'. The data frame is given the name 'data' and is prepared for analysis with the following input into the `model.calc` function:

> `Model.out=model.calc(form.occup=~Mag + CV, form.detect=~Mag + CV, t.root=NULL, data=data, tau=4)`.

In this case the `form.occup` and `form.detect` functions are set as the two site varying covariates and the `t.root` argument is set to `NULL`, as there are no time varying covariates. `data` is set to the data frame's name, 'data' in this case, and `tau` is set to `4` as there were four surveying occasions (τ from section 2). As there are no time varying covariates, we have the choice of two occupancy report functions to call depending on which model we would like to fit; `est.time.homog` and `est.time.homog.int`.

For this case, where there are only site dependant covariates, `est.time.homog` uses the conditional likelihood (Equation 10) to find site specific detection estimates with the `cond.detect` function and then uses the IWLS method for occupancy coefficient estimation with the `iwls` function. The `R` input for `est.time.homog` is shown in figure 2, along with its output, listing the estimates found for the occupancy and detection coefficients.

The `est.time.homog.int` function is used when we want the model to include a time

```
> est.time.homog.int(Model.out, method = 'iWls')
An object of class "occup.report"
Slot "desc":
[1] "Time homogeneous covariates and time varying intercept"

Slot "ests":
                     est        se          t           p
(Intercept)    3.9307484 1.7390936  2.2602282 0.023807090
Mag            3.5629078 1.2678906  2.8101067 0.004952508
CV            -0.1413454 0.6795834 -0.2079883 0.835238074
(Intercept):1  0.7504151 0.5289575  1.4186680 0.155995822
(Intercept):2  1.6323972 0.5956242  2.7406496 0.006131786
(Intercept):3  1.1376618 0.5505052  2.0665778 0.038773958
(Intercept):4  1.6323972 0.5956242  2.7406496 0.006131786
Mag           -0.2848894 0.2170699 -1.3124319 0.189374448
CV             0.2260902 0.2758204  0.8197008 0.412386691

Slot "AIC.beta":
[1] 172.4851

Slot "AIC.alpha":
[1] 28.82628
```

Figure 3: Input and output of analysis of Coosa bass data by `est.time.homog.int` function

dependant intercept. This function uses the same algorithm for analysing the data as `est.time.homog`, except for the use of `posbernoulli.t` in the vglm function when calculating the detection coefficient estimate, and the use of `est.se.hetero` for calculating the standard errors (as described in section 3.22). The input for `est.time.homog.int` is shown in Figure 3, along with its output. In this case, we can see that there are four estimates given for the detection intercept. This is because the intercept was taken to be time dependant, therefore giving separate estimates for each of the four surveying occasions.

4.2 Brook trout application

We will now look at an example with a data set that is both site and time heterogeneous. The data set used is the Brook trout data, which has three surveying occasions, one site varying covariate 'Ele' and one time varying covariate 'CSA'. It has been formatted as a data frame as described in Section 4.1 and given the name 'data'. We first prepare the data for analysis by calling the `model.calc` function with input:

```
> Model.out=model.calc(form.occup=~Ele, form.detect=~Ele,
t.root=list("CSA"), data=data, tau=3).
```

In both `form.occup` and `form.detect` we include ‘Ele’, the only site varying covariate in the data set. In this case ‘CSA’ is the only time dependant covariate, which is measured on each surveying occasion. Thus, the `t.root` argument is a list containing only ‘CSA’. The argument `data` is set to the name we have given the data frame, ‘data’ in this example. Finally the argument `tau` is set as the number of surveying occasions, which in this case is `3`. The analysis function used in this example is `est.time.varying`, due to the inclusion of the time dependant covariate ‘CSA’.

We have chosen for this example to keep the intercept constant, so the `int` argument in the `est.time.varying` function is set to `FALSE`. This being the case `est.time.varying` calls `cond.detect.t`, which uses the `(parallel.t=FALSE~0)` argument for the `posbernoulli.t` family in the input for the `vglm` function. As described in section 3.2.3, this estimates the detection coefficients with a constant intercept (equation 11). The `iwls` function then estimates the occupancy coefficients using the IWLS method. The input for the `est.time.varying` function, along with its output, is shown in figure 4.This output includes an estimate for the newly constructed time dependant variable `CSA.tij`. This was constructed by the `cond.detect.t` function, to allow the `vglm` function to fit a model that includes time varying covariates (as described in section 3.2.3). An estimate for this variables coefficient is not given for occupancy, as we assume occupancy to be time dependant.

5 Conclusion and future directions

It was the aim of this project to construct a package that applies the two-stage approach to occupancy model parameter estimation developed in Karavarsamis and Huggins (2018a) and Karavarsamis and Huggins (2018b). The resulting package, named `twoStage`, gives statistical ecologists a simple and useful method for applying this two-stage approach. The package provides an alternate to the already commonly used `unmarked` package, which is not adequate in all applications (Karavarsamis and Huggins, 2018b). The package was constructed from existing code used in Karavarsamis and Huggins (2018a) and

```
> est.time.varying(Model.out, int=FALSE, method="iWls")
An object of class "occup.report"
Slot "desc":
[1] "Time varying covariates and constant intercept"

Slot "ests":
                      est           se          t           p
(Intercept) -3.8715972774 1.2754622463 -3.0354464 0.002401799
Ele          0.0013046359 0.0004076897  3.2000705 0.001373940
(Intercept) -0.1608646150 1.2397342176 -0.1297573 0.896758415
Ele          0.0004002955 0.0003198167  1.2516404 0.210700943
CSA.tij     -0.7437817947 0.2873110370 -2.5887686 0.009631979

Slot "AIC.beta":
[1] 119.8384

Slot "AIC.alpha":
[1] 82.07872
```

Figure 4: Output from analysis of Brook trout data with `est.time.varying` function

Karavarsamis and Huggins (2018b).

Following further improvements, the package will be submitted to `CRAN`. This includes further troubleshooting and software testing, to identify possible constraints on data sets that it can be applied to.

Acknowledgements

The author would like to thank Dr. Natalie Karavarsamis for her help and guidance during this research project. The author would also like to thank the Department of Education and the Australian Mathematical Sciences Institute for the opportunity to experience life as a researcher.

References

Fiske, I. J. and Chandler, R. B. (2015). *Overview of unmarked: an R package for the analysis of data from unmarked animals*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0.

Gimenez, O., Blanc, L., Besnard, A., Pradel, R., Doherty-Jr, P. F., Marboutin, E., and Choquet, R. (2014). Fitting occupancy models with E-SURGE: hidden Markov modelling of presence-absence data. *Methods in Ecology and Evolution*, 5:592 – 597.

Hadley Wickham, Jim Hester, W. C. R. and team, R. C. (2018). *devtools: Tools to Make Developing R Packages Easier*. R package version 2.0.1.

Karavarsamis, N. and Huggins, R. (2018a). A two-stage approach to the analysis of occupancy data I. The Homogeneous case (analysis of occupancy data). Arxiv ID: 1804.08694.

Karavarsamis, N. and Huggins, R. (2018b). Two-stage approaches to the analysis of occupancy data ii. the heterogeneous model and conditional likelihood. *Computational Statistics and Data Analysis*. DOI: https://doi.org/10.1016/j.csda.2018.09.009.

MacKenzie, D. I., Nichols, J., Royle, J., Pollock, K., Bailey, L., and Hines, J. (2006). *Occupancy Estimation and Modeling Inferring Patterns and Dynamics of Species Occurrence*. Elsevier, San Diego, CA.

MacKenzie, D. I., Nichols, J. D., Lachman, G. B., Droege, S., Royle, J., and Langtimm, C. A. (2002). Estimating site occupancy rates when detection probabilities are less than one. *Ecology*, 83(8):2248–2255.

R Development Core Team (2018). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0.

Yee, T. W. (2010). The VGAM package for categorical data analysis. *Journal of Statistical Software*, 32:1–34.

Yee, T. W., Stoklosa, J., and Huggins, R. M. (2015). The VGAM package for capture–recapture data using the conditional likelihood. *Journal of Statistical Software*, 65:1–33.