Steven T. Hoekman. 2021. Multi-observer methods for estimating uncertain species identification. *Ecosphere*.

Metadata S3: R v. 4.10 computer code for simulation study of multi-observer methods

---

Author

Steven T. Hoekman
Wild Ginger Consulting
P. O. Box 182
Langley, WA, USA 98260
steven.hoekman@protonmail.com

File list (found within folder 'DataS3-R_Code_for_Simulations')

R code files

`generate_simulation_data.R`	Function for generating simulated survey data
`optimization_functions.R`	Functions for likelihood optimization
`supplemental_functions.R`	Supplemental functions for data analysis and summary
`simulations_r_code.R`	R code for conducting simulation analyses
`likelihood_equations.R`	Generates pre-computed values for computations

Example simulation input/output files

`omnibus_m_22_example.csv`	Example simulation input file
`omnibus_m_22_example_output.csv`	Example simulation output file

Dependencies (required, except as noted below)

R packages (available at <https://cran.r-project.org/web/packages/>)

`plyr`	Programming tools, data manipulation
`dplyr`	Data frame manipulation
`doSNOW`	Back end for parallel code execution
`mrds`	Delta method for arbitrary functions
`extraDistr`	Probability distributions
`nnet`	Only needed for estimating observed proportions
`gtools`	Only needed for generating new `likelihood_equations` files

Files description

`generate_simulation_data.R`

Function '`generate.simulation.data.f`' accepts user-specified inputs defining model structure and true parameter values and returns simulated survey data.

`optimization_functions.R`

Functions that accept input of simulated survey data for multi-observation method (MOM) and single-observation method (SOM) models and returns the minimized −log(likelihood) (plus any penalty terms), estimated parameter values, and the Hessian matrix.

`supplemental_functions.R`

Functions providing supplemental services, such as: drawing random numbers for simulations; likelihood computations; and formatting, summary, and error-checking of data and output.

`simulations_r_code.R`

R computer code for conducting simulation analyses, including loading required R packages and input files, executing statistical analyses, and summarizing output.

`likelihood_equations.R`

Functions for generating R list objects '`likelihood.equations`' containing pre-computed values for likelihood equations calculating probabilities for observed groups. List objects required for simulation analyses are provided in DataS2-Input_Files_for_Simulations.

`omnibus_m_22_example.csv`

An example .csv format file specifying simulation inputs for MOM models with 2 observation states, 2 true species states. Includes distinct models with groups of size one, homogeneous groups, and heterogeneous groups. See MetadataS2.pdf for specifications for input files.

`omnibus_m_22_example_output.csv`

An example .csv format file containing output from simulation analyses for the example input file above. These results can be replicated by executing the R command "set.seed(1859)" prior to executing simulation code to specify the seed value for generating random numbers.

Additional to general documentation below, each included .R file contains detailed documentation comments in code describing specific code, functions, and variables.

Simulation engine description

The 'omnibus', 'covariate', 'distinct observer' statistical simulations for MOM and SOM models described in the companion article and Appendix S3, code version 1.1.0 updated to run in the R statistical computing environment version 4.1 (R Development Core Team 2021) using the files provided here, in conjunction with input files provided in DataS2-Input_Files_for_Simulations. For the simulation engine used to conduct analyses in the companion article, see release version 1.0.0 of the simulation engine (archived at Zenodo https://doi.org/10.5281/zenodo.4738002). To conduct these or other user-specified simulations, R code in `simulations_r_code.R` requires:

- Functions provided in files `generate_simulation_data.R`, `optimization_functions.R`, and `supplemental_functions.R`
- A .csv format file specifying simulation inputs (see MetadataS2.pdf)
- The '`likelihood.equations`' list objects providing pre-computed values (provided in DataS2-Input_Files_for_Simulations)
- R packages listed in the "Required R packages" section at the beginning of `simulations_r_code.R`

The simulation engine supports:

- One or no primary observers and 1-4 secondary observers
- Secondary observers with identical or distinct classification probabilities
- One (SOM models) or 2 (MOM models) observation methods
- Groups of size 1, homogeneous groups, and heterogeneous groups
- Two to four true species states and observation states (without covariates)
- Heterogeneous groups composed of individuals of 2 true species states
- Alternative models for heterogeneous groups: the "constant" model (eq. 7 in the main article) and "encounter" model (eqs. S3, S4 in Appendix S3), which may be necessary for models with a covariate predicting true species probabilities
- Covariates predicting true species probabilities and classification probabilities
    - With a covariate predicting true species probabilities *or* classification probabilities, 2-3 true species states and observation states
    - With a covariate predicting true species probabilities *or* classification probabilities, boot-strapped estimates of standard errors and confidence intervals for overall mean probabilities
    - With separate covariates predicting true species probabilities *and* classification probabilities, 2 true species states and 2 observation states
- Un-modeled heterogeneity in data arising from a covariate predicting true species probabilities, a covariate predicting classification probabilities, or distinct classification probabilities among secondary observers
- Estimation of observed species proportions (assuming no misidentification) using multinomial logit models

Simulation inputs

All MOM and SOM models use the R 'optim' function to minimize the –log(likelihood) using the L-BFGS-B optimization method of Byrd et al. (1995), which allows box constraints placing lower and upper bounds on values of individual parameters. Appropriate lower box constraints are generated automatically. Upper box constraints default to 'Inf' (i.e., infinity or no constraint), but can be specified for classification probabilities by users as described in MetadataS2.pdf.

Additional to box constraints, penalty functions in model optimization functions constrain optimization by adding penalty terms to the –log(likelihood) if true species probabilities or heterogeneous group probabilities take inadmissible or unrealistic values. MetadataS1.pdf provides descriptions and examples of penalty functions.

Five sets of initial parameter values are automatically generated for each distinct model. For each simulation replicate, each set is used in succession until achieving acceptable optimization. Replicates that never successfully optimize are removed from simulation output.

Simulation inputs specified in the .csv format files are summarized in data frame 'sim_profiles', which is used by function 'generate.simulation.data.f' to produce list object 'sim_data', comprised of 4 elements. The first element 'survey_data' is a data frame with simulated survey data and, if applicable, fields for covariate values and key values (Table 1). These key values either match observed groups for each observer to a key table of unique observed groups in 'observed_group_key' (list element two, Table 2), or match values for a binned covariate predicting true species probabilities to a key table of unique covariate values in 'psi_key' (list element three). If a binned covariate predicting classification probabilities is present, key values for observed groups correspond to unique combinations of observed groups and covariate values. Key tables can substantially reduce computational load by limiting computation of probabilities to unique keyed items.

The final list element 'mean_probability' contains two vectors giving overall mean values (from the true model) of true species probabilities and classification probabilities across all groups for models with covariates predicting these probabilities. Named vector elements are 'psi_[*b*]' and 'theta.[*observer*].[*ab*]', where bracketed suffixes denote true species state *b*, observation state *a*, and observer identity for primary 'p' and secondary observers "s1" to "s4".

Simulation outputs

List 'model_results' stores statistical output for each simulation replicate from optimization functions, consisting of: parameter estimates ('par'), the −log(likelihood) + any penalty term(s) ('value'), and the Hessian matrix for estimation of the variance-covariance matrix ('hessian').

Data frame 'output_global' contains summarized statistical output, with rows giving results for each distinct model. The left-most fields give simulation inputs (MetadataS2.pdf provides descriptions of fields and naming conventions fir all inputs and parameters). To the right are statistics for estimated parameters (Table 3). Field 'rep.tru' is the count of replicates successfully optimizing for each distinct model. For simulations including predictive covariates, output

includes estimates of overall means for true species or classification probabilities (see Appendix S3) and associated mean error, root mean squared error, and 95% confidence interval coverage. If inputs specify bootstrap resamples, output for these overall means also includes bootstrapped estimates of standard errors and 95% confidence interval coverage (see Appendix S3).

For simulations including un-modeled heterogeneity in data, '`output_global`' includes fields for the true parameter values used for data generation and field 'heterogeneity', which takes value 1 for 'covariate" simulations with an un-modeled predictive covariate and takes value 2 for "distinct observer" simulations with un-modeled differences in classification probabilities among secondary observers.

Upon completion, reports on simulation speed and any error messages are printed to the console window, and output is written to the hard drive in .csv file format with "_output" appended to the input file name. For lengthy simulations, output is periodically written to the hard drive.

Workflow for conducting simulation analyses

To execute simulation analyses:

1) Place the .R format files within DataS3 in the R working directory.
2) Place the .RData and .csv format files within DataS2-Input_Files_for_Simulations in the R working directory.
3) In an R editor or integrated R development environment (e.g., *RStudio* by RStudio, Inc., Boston, Massachusetts, USA, <https://rstudio.com>), open files `generate_simulation_data.R`, `optimization_functions.R`, and `supplemental_functions.R`. Execute R code in these files to load these functions into the R environment.
4) Open file `simulations_r_code.R`, navigate to the 'Required R packages' section, and install and load these R packages.
5) In the following code section, execute R code to register the 'doSNOW' parallel execution backend.
    - Parallel processing can dramatically increase speed of simulations.
6) Select a simulation input .csv file provided in DataS2.
    - Alternatively, use the example file `omnibus_m_22_example.csv` provided here, or create your own input file.
7) In the 'Import simulation profiles' section, specify the name of the input file. Execute code in this section to define the profile(s) for simulation analyses and automatically load the appropriate 'likelihood_equations' list.
    - I recommend executing code in this section prior to each simulation analysis (except loading '`likelihood.equations`' only as necessary).
8) Conduct simulation analyses by executing R code the appropriate code section (below)

Simulation code sections:

• ‘MOM/SOM model simulations’

Conducts “omnibus,” “covariate,” and “distinct observer” simulations for MOM and SOM models *without* un-modeled heterogeneity in data. Without covariates, supports ≤4 true species states and ≤4 observation states (more states are possible by altering input to function ‘`theta4.f`’). Supports ≤3 states with one covariate, and supports 2 states with two covariates.

• ‘MOM/SOM model simulations: Un-modeled heterogeneity (covariates)’

Conducts “covariate” simulations for MOM and SOM models *with* un-modeled heterogeneity in data arising from a covariate predicting true species probabilities ***or*** classification probabilities. Supports ≤3 true species states and observation states.

• ‘MOM/SOM model simulations: Un-modeled heterogeneity (observers)’

Conducts “distinct observer” simulations for MOM and SOM models *with* un-modeled heterogeneity in data arising from differing classification probabilities among secondary observers. Supports ≤3 true species states and observation states; does not support covariates.

• ‘Multinomial model simulations: Estimating observed species proportions’

For MOM models in ‘omnibus’ simulations, estimates observed species proportions using a multinomial logit model assuming no species misidentification. Supports ≤4 true species states and observation states; does not support covariates or SOM models. Output file includes field ‘observed_p’ with value of 1. Requires R package `nnet`.

`Likelihood_equations.R`

Most users will not need to generate new '`likelihood.equations`' lists, unless those in DataS2 do not support sufficient group sizes. Computations of conditional probabilities for observed groups increase rapidly with group size, especially for heterogeneous groups. File `likelihood_equations.R` contains code for creating lists '`likelihood.equations`', which contain pre-computed values of equations for calculating probabilities in a computationally efficient form. Lists are specific to the number of observation states *A* and true species states *B*, and lists cover a range of group sizes *g*. Requires R package `gtools`.

Saved list objects provided in DataS2-Input_Files_for_Simulations follow the naming convention 'likelihood_equations_A[#]B[#]G[#].RData', where [#] denotes an integer for *A*, *B*, or maximum group size (with group sizes ranging from 1 to the maximum).

Lists contain 3 types of named elements:

1) 'true.combinations.g.[size]', where [size] ranges from 1 to maximum group size

   An integer matrix for each group size *g* containing possible combinations of true species states *b* on separate rows, with columns denoting true state of each individual.

2) 'true.permutations.count.g.[size]', where [size] ranges from 1 to maximum group size

   An integer vector for each group size *g* with counts (*via* the multinomial coefficient) of possible permutations for each true group. Vector elements correspond to rows describing true groups for the same group size in (1) above.

3) 'observed.[*observed group*]', where [*observed group*] denotes counts of individuals in each observation state 1 to A (integers separated by decimals).

   Data frame summarizing equations giving the conditional probability of an observed group, with columns:

   a) 'index'

      Integer corresponding to rows for true groups of the same size in (1) above.

   b) 'coefficient'

      Integer to be multiplied by the probability terms in (c) below to account for the count of possible permutations (*K*) of observation states giving rise to the classifications in (c).

   c) '1.1', '2.1', . . .

      Integers (separated by decimals) in each of *A* * *B* columns named '*a*.*b*' give counts of individuals in an observed group belonging to true species state *b* classified to observation state *a*.

Rows in each data frame in (3) correspond to permutations of observation states possibly arising from the true group specified by 'index'. Each integer in an '*a.b*' column specifies counts of individuals of true species state *b* classified to observation state *a*; this integer value gives the exponent for the corresponding classification probability. All probability terms specified in each row are multiplied by multinomial 'coefficient' *K*. Rows with the same 'index' value are summed, and the overall probability for the observed group is the product of these summed probabilities across all index values. These data frames are a computationally efficient method of organizing equations for conditional classification probabilities, where are illustrated in the main article in Table 2 under the 'Observed group **y**' columns. For an observed group, matrix rows correspond to each group of multiplied terms within a table cell, with the 'coefficient' *K* multiplied by these terms and integers in columns '*a.b*' representing exponents on each term. Data frame rows with identical 'index' values are the summed terms within a table cell, while groups of rows with different 'index' values represent terms in each table cell on different rows within a column.

Workflow for generating 'likelihood.equations' list objects

Time to produce these list files depends on user-specified inputs and computing power, but increases rapidly with larger group sizes.

1) In an R editor or integrated R development environment, open the R script file `likelihood_equations.R`
2) Install and load packages in the "Required R packages" section
3) In the following section, execute R code to register the 'doSNOW' parallel execution backend
4) Execute the code in the "Required supporting functions" section
5) In "User-specified inputs", specify the true species states *B*, observation states *A*, and range of group sizes *g*
6) Execute all code in the "Generate a list" section

    • Depending on your computer specs and user inputs, you may have time to prepare: tea, dinner, your dissertation, etc.

7) After inspection, rename list to '`likelihood.equations`" and save to the hard drive

---

Literature Cited

Byrd, R. H., Lu, P., Nocedal, J. and Zhu, C. 1995. A limited memory algorithm for bound constrained optimization. SIAM Journal on Scientific Computing. 16:1190–1208.

R Development Core Team. 2021. R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. www.r-project.org

Table 1. Naming conventions and description of fields in the data frame '`survey_data`', which contains simulated survey data. Suffixes (in brackets, separated by underscores) denote observer identities and observation state *a*.

Field name	Field values	Description
y_[observer]_[*a*]	Integer ≥0	Counts of murrelets classified by survey observers to observation state *a*. Text suffix [observer] denotes primary observer 'p' or secondary observers 's1', 's2', and 's3', and integer suffix [*a*] denotes observation state *a*.
covariate_theta	Numeric	Group-level covariate predicting classification probabilities θ
covariate_psi	Numeric	Group-level covariate predicting true species probabilities ψ
group_size	Integer ≥1	Count of individuals in a group *g*
id	Integer ≥1	Index for simulation replicates
count	Integer ≥1	Count of each distinct observation history
y_[observer]_key	Integer ≥1	Key value indexing unique observed groups for each observer. Text suffix [observer] denotes primary 'p' or secondary observers 's1', 's2', and 's3'. Key values match those in data frame '`observed_group_key`' (Table 2).
psi_key	Integer ≥1	Key value indexing unique (binned) values of covariate predicting true species probabilities ψ. Key values match those in data frame '`psi_key`'.

Table 2. Naming conventions and description of fields in the data frame '`observed_group_key`', which contains key values indexing unique observed groups. Unique observed groups are matched to observed groups in data frame '`survey data`' (Table 1) using matching key values in fields 'y_[observer]_key'. A suffix (in brackets, separated by an underscore) denotes observation state *a*.

Field name	Field values	Description
A_[*a*]	Integer ≥0	Count of individuals classified to the observation state *a*, specified by integer suffix [*a*]
covariate_theta	Numeric	Unique values (binned) of covariate predicting classification probabilities
key	Integer ≥1	Key value indexing unique observed groups
B_states	Integer ≥1	Number of possible combinations of true groups for the observed group
sum	Integer ≥0	Count of classifications in the observed group

Table 3. Prefixes describing summary statistics for parameters in simulation output. MetadataS2.pdf provides descriptions and naming conventions for parameters.

Prefix	Description
m.	Mean
sd.	Standard deviation
se.	Mean standard error
cv.	Coefficient of variation (SE/mean)
me.	Mean error
rm.	Root mean squared error
ci.	95% confidence interval coverage
ci.lg.	95% confidence interval coverage for derived parameters (e.g., overall means for probabilities predicted by values of group-level covariates), with confidence intervals estimated in logit scale using estimated standard error (see eqs. S6-S8 in Appendix S3)
se.bt.	Mean standard error estimated from bootstrap resamples
ci.btlg.	95% confidence interval coverage, with confidence interval estimated in logit scale using bootstrapped estimates of standard error