

# overdispsim - Simulations of Overdispersion in SECR

Murray Efford

2025-05-19

## Contents

<b>Introduction</b>	<b>1</b>
<b>Dependence on other packages</b>	<b>2</b>
<b>Setup</b>	<b>2</b>
<b>Define populations</b>	<b>3</b>
Unconditional (Poisson $N(A)$ ) . . . . .	3
Conditional (fixed $N(A)$ ) . . . . .	5
<b>Examples using defined populations</b>	<b>5</b>
<b>Code for simulations in paper</b>	<b>5</b>
Sampling only . . . . .	6
Full model fits (suffix M) . . . . .	6
Conditional model fits (suffix MCL) . . . . .	6
<b>Summaries</b>	<b>7</b>
<b>Cohesion</b>	<b>9</b>
<b>References</b>	<b>9</b>
<b>Appendix 1. Example</b>	<b>10</b>
No model fit . . . . .	10
Model fit . . . . .	10
<b>Appendix 2. Simulation summaries</b>	<b>10</b>
No model . . . . .	10
Model fitted . . . . .	15
Model fitted, conditional likelihood . . . . .	19
Complete cohesion . . . . .	20
<b>Finally...</b>	<b>23</b>

## Introduction

This vignette explains some of the functions in the package **overdispsim** and provides code for simulations in the paper of Efford and Fletcher (2024 and later versions). **overdispsim** uses the R packages **secr** (Efford 2025a) and **secrdesign** (Efford 2025b) available from the CRAN repository. **overdispsim** itself is available on GitHub and Zenodo.

The research question concerns the effect of the distribution of animal activity centres (AC) on estimates from spatially explicit capture–recapture models, specifically the maximum-likelihood estimates of population density provided by R package **secr**.

The scenarios considered are a superset of those reported by Efford and Fletcher (2024). To reproduce the specific results in Efford and Fletcher (2024) see the file ‘Figures.R’ on Zenodo.

Three base processes are simulated for the distribution of AC:

1. log-Gaussian Cox process (LGCP),
2. Thomas cluster process, and
3. random binary habitat mosaic.

Each process is simulated both unconditionally and conditional on  $N(A)$  (i.e. fixed number in a buffered area  $A$ ). The unconditional simulations result in Poisson-distributed  $N(A)$ . Conditional simulations are identified by the suffix ‘f’. Thus there are 6 distinct processes (1, 2, 3, 1f, 2f, 3f). Each process is simulated over a range of parameter values.

This vignette provides both the code to run the simulations, given the R package **overdispersim**, and tables summarising the resulting simulations. Simulations are re-run if the variable ‘runsimsulations’ is set to TRUE; otherwise, summaries are based on previous simulations downloaded from Zenodo (Efford 2025c) or possibly a local directory.

## Dependence on other packages

Packages **secr** and **secrdesign** are used throughout for simulation and model fitting.

Functions **rLGCP** and **rThomas** from the **spatstat** package (Baddeley et al. 2015) are used where possible. The algorithm for **rLGCP** in **spatstat.random** changed in version 3.2.1 (Oct 2023) to avoid dependence on **RandomFields** (Schlather et al. 2015). Some simulations here were run with the earlier code and may differ (see warning in `?spatstat.random::rLGCP`). Conditional simulations with **rLGCP** and **rThomas** require **spatstat.random** version  $\geq 3.3.3.12$  and **secr** version  $\geq 5.2.2$ . These versions may be installed from R-universe if they are not yet on CRAN.

Package **zen4R** (Blondel 2024) is required to download simulations from Zenodo.

## Setup

```
library(overdispersim, quietly = TRUE)
options(digits = 5)
runsimsulations <- FALSE # set TRUE to run afresh
runexamples <- FALSE
nrepl_n <- 0 # 10000 zero causes load from localrepo or zenodoDOI
nrepl_M <- 0 # 1000
csvdir <- NULL # folder to save summaries as csv
zenodoDOI <- "10.5281/zenodo.15455288" # optionally retrieve all simulations
localrepo <- ""
```

For testing set `runsimsulations <- TRUE` and the number of replicates to a small number e.g. 5.

We next set some options by assigning variables to an environment (‘local’) that is available to other functions in the package, and incidentally keep a copy in a list named ‘localpar’.

The function is run with these arguments automatically when the package is attached (i.e. in `.onAttach`), so these are default settings. The actual number of cores is capped at the number available if that is less than the given `maxncores`.

```

localpar <- setparameters(
  lambda0    = 0.5,
  sigma      = 1.0,
  detectfn   = 'HHN',
  nooccasions = 5,
  traps      = make.grid(12,12, detector = 'proximity', spacing = 2.0),
  maskspacing = 0.5,
  maskbuffer = 4,
  N          = 256,
  maxncores  = 24
)

```

The list and environment include some derived quantities.

## Define populations

Start by specifying some vectors with varying levels of parameters:

```

Vlevels <- c(0, 0.125, 0.25, 0.5, 0.75, 1.0)      # LGCP variances
mulevels <- c(1,2,4,8,16,32)                      # expected number per cluster
Alevels <- c(0.0625, 0.125, 0.25, 0.5, 0.75, 1.0) # randomHabitat A

```

Next, for each process generate a list of arguments, one component for each combination of parameter values. Each component is ultimately passed internally to the **secl** function 'sim.popn' to simulate realisations of the distribution of AC.

## Unconditional (Poisson $N(A)$ )

```

eps <- spacing(localpar$mask)

# 1. LGCP
basepopargs1 <- list(
  D      = localpar$D,
  core   = localpar$mask,
  buffer = eps/2,
  model2D = "rLGCP",
  details = list(var = 0, scale = 5 * localpar$sigma,
                 eps = eps, saveLambda = TRUE))
popargs1 <- extend(basepopargs1,
  values = list(var = Vlevels, scale = c(2,5,10) * localpar$sigma))
t(sapply(popargs1, '[[', 'details'))

```

```

##      var  scale eps saveLambda
## [1,] 0      2     0.5 TRUE
## [2,] 0.125 2     0.5 TRUE
## [3,] 0.25  2     0.5 TRUE
## [4,] 0.5   2     0.5 TRUE
## [5,] 0.75  2     0.5 TRUE
## [6,] 1     2     0.5 TRUE
## [7,] 0     5     0.5 TRUE
## [8,] 0.125 5     0.5 TRUE
## [9,] 0.25  5     0.5 TRUE
## [10,] 0.5   5     0.5 TRUE
## [11,] 0.75  5     0.5 TRUE

```

```
## [12,] 1      5      0.5 TRUE
## [13,] 0      10     0.5 TRUE
## [14,] 0.125 10     0.5 TRUE
## [15,] 0.25  10     0.5 TRUE
## [16,] 0.5   10     0.5 TRUE
## [17,] 0.75  10     0.5 TRUE
## [18,] 1     10     0.5 TRUE
```

### # 2. Thomas clustering

```
basepopargs2 <- list(
  D      = localpar$D,
  core   = localpar$mask,
  buffer = eps/2,
  model2D = "rThomas",
  details = list(mu = 1, scale = localpar$sigma, eps = eps, saveLambda = TRUE))
# rThomas requires scale>0, so use tiny value instead 1e-4
popargs2 <- extend(basepopargs2,
  values = list(mu = mulevels, scale = c(1e-4, 1, 2, 4)*localpar$sigma))
t(sapply(popargs2, '[', 'details'))
```

```
##      mu scale eps saveLambda
## [1,] 1 1e-04 0.5 TRUE
## [2,] 2 1e-04 0.5 TRUE
## [3,] 4 1e-04 0.5 TRUE
## [4,] 8 1e-04 0.5 TRUE
## [5,] 16 1e-04 0.5 TRUE
## [6,] 32 1e-04 0.5 TRUE
## [7,] 1 1      0.5 TRUE
## [8,] 2 1      0.5 TRUE
## [9,] 4 1      0.5 TRUE
## [10,] 8 1      0.5 TRUE
## [11,] 16 1      0.5 TRUE
## [12,] 32 1      0.5 TRUE
## [13,] 1 2      0.5 TRUE
## [14,] 2 2      0.5 TRUE
## [15,] 4 2      0.5 TRUE
## [16,] 8 2      0.5 TRUE
## [17,] 16 2      0.5 TRUE
## [18,] 32 2      0.5 TRUE
## [19,] 1 4      0.5 TRUE
## [20,] 2 4      0.5 TRUE
## [21,] 4 4      0.5 TRUE
## [22,] 8 4      0.5 TRUE
## [23,] 16 4      0.5 TRUE
## [24,] 32 4      0.5 TRUE
```

### # 3. random habitat

```
basepopargs3 <- list(
  D      = randomDensity,
  core   = localpar$mask,
  buffer = 0,
  model2D = "IHP",
  details = list(D = localpar$D, p = 0.5, A = 0.25, rescale = TRUE))
popargs3 <- extend(basepopargs3,
  values = list(A = Alevels, p = c(0.25, 0.5)) )
```

```
t(sapply(popargs3, '[', 'details'))
```

```
##      D      p      A      rescale
## [1,] 2844.4 0.25 0.0625 TRUE
## [2,] 2844.4 0.25 0.125 TRUE
## [3,] 2844.4 0.25 0.25 TRUE
## [4,] 2844.4 0.25 0.5 TRUE
## [5,] 2844.4 0.25 0.75 TRUE
## [6,] 2844.4 0.25 1 TRUE
## [7,] 2844.4 0.5 0.0625 TRUE
## [8,] 2844.4 0.5 0.125 TRUE
## [9,] 2844.4 0.5 0.25 TRUE
## [10,] 2844.4 0.5 0.5 TRUE
## [11,] 2844.4 0.5 0.75 TRUE
## [12,] 2844.4 0.5 1 TRUE
```

## Conditional (fixed $N(A)$ )

Modify each of the preceding scenarios by telling `sim.popn` to use fixed  $N$ .

```
fix <- function(x) {
  x$Ndlist <- "fixed"
  x$Nbuffer <- localpar$N
  x
}

# 1f. Conditional LGCP requires spatstat.random >= v3.3.3.2 and secr >= 5.2.2
popargs1f <- lapply(popargs1, fix)

# 2f. Conditional Thomas process requires spatstat.random >= v3.3.3.12 and secr >= 5.2.2
popargs2f <- lapply(popargs2, fix)
# cf Bischof et al. fixed clusters (clone = "constant", scale = 0)

# 3f. random habitat, fixed-N
popargs3f <- lapply(popargs3, fix)
```

## Examples using defined populations

Rather than interrupt the flow here, refer to Appendix 1.

## Code for simulations in paper

Here we list the commands used to generate the main results. Sampling with each process (1, 2, 3, 1f, 2f, 3f) and parameter set is initially run many times without fitting a model in order to approximate the expected number of detected individuals etc. Then each sampling scenario is simulated and a model is fitted by maximising the full SECR likelihood (1M, 2M, 3M, 1Mf, 2Mf, 3Mf). A subset of scenarios is further simulated fitting only the conditional SECR likelihood (1MCL, 2MCL, 3MCL).

The function `run_all` from **overdispsim** is a wrapper for **secr** and **secrdesign** simulation functions that uses settings in the local environment. The argument `'extractfn'` of **secrdesign::run.scenarios** is a different function for each of the three groups of simulations; these functions are defined in **overdispsim**.

Each of these takes substantial time to run (hours). They should be run separately and saved for later processing (i.e. summary).

Table 1. Simulations.

Code	AC distribution	Fit	extractfn
1	LGCP	none	extract_n
2	Thomas process	none	extract_n
3	random habitat	none	extract_n
1f	fixed-N(A) LGCP	none	extract_n
2f	fixed-N(A) Thomas process	none	extract_n
3f	fixed-N(A) random habitat	none	extract_n
1M	LGCP	full likelihood	extract_M
2M	Thomas process	full likelihood	extract_M
3M	random habitat	full likelihood	extract_M
1fM	fixed-N(A) LGCP	full likelihood	extract_M
2fM	fixed-N(A) Thomas process	full likelihood	extract_M
3fM	fixed-N(A) random habitat	full likelihood	extract_M
1MCL	LGCP	conditional likelihood	extract_MCL
2MCL	Thomas process	conditional likelihood	extract_MCL
3MCL	random habitat	conditional likelihood	extract_MCL

## Sampling only

Here we generate SECR samples, count the number of detected individuals  $n$  and compute the  $\hat{c}$  measure of overdispersion using the known detection parameters. Models are not fitted.

```
# Poisson N(A)
sims1 <- run_all(nrepl_n, popargs1, fit = FALSE)
sims2 <- run_all(nrepl_n, popargs2, fit = FALSE)
sims3 <- run_all(nrepl_n, popargs3, fit = FALSE)

# Fixed N(A)
sims1f <- run_all(nrepl_n, popargs1f, fit = FALSE)
sims2f <- run_all(nrepl_n, popargs2f, fit = FALSE)
sims3f <- run_all(nrepl_n, popargs3f, fit = FALSE)
```

## Full model fits (suffix M)

Here we select a subset of the defined population scenarios.

```
# Poisson N(A)
sims1M <- run_all(nrepl_M, popargs1[13:18], fit = TRUE)
sims2M <- run_all(nrepl_M, popargs2[13:18], fit = TRUE,
  start = list(D = 3000, lambda0 = 0.4, sigma = 2.2))
sims3M <- run_all(nrepl_M, popargs3[7:12], fit = TRUE)

# Fixed N(A)
sims1fM <- run_all(nrepl_M, popargs1f[13:18], fit = TRUE, distribution = "binomial",
  start = list(D = 5000))
sims2fM <- run_all(nrepl_M, popargs2f[13:18], fit = TRUE, distribution = "binomial")
sims3fM <- run_all(nrepl_M, popargs3f[7:12], fit = TRUE, distribution = "binomial")
```

## Conditional model fits (suffix MCL)

The SECR model was fitted by maximising the conditional likelihood as a convenient way in **secr** to simulate the coverage of confidence intervals for the effective sampling area  $a(\hat{\theta})$  and the proportion of variance due to

$n$  and  $a(\hat{\theta})$ .

```
# Poisson N
sims1MCL <- run_all(nrepl_M, popargs1[13:18], fit = TRUE, CL = TRUE)
sims2MCL <- run_all(nrepl_M, popargs2[13:18], fit = TRUE, CL = TRUE,
                    start = list(D = 3000, lambda0 = 0.4, sigma = 2.2))
sims3MCL <- run_all(nrepl_M, popargs3[7:12], fit = TRUE, CL = TRUE)
```

## Summaries

Simulation results have been archived on Zenodo (Efford 2025c). If necessary, we retrieve them with the R package **zen4R** (Blondel 2024).

```
tmpfolder <- tempdir()
if (file.exists(localrepo)) {
  files_to_copy <- list.files(localrepo, full.names = TRUE)
  file.copy(files_to_copy, tmpfolder)
  reloadedfrom <- localrepo
} else {
  if (!requireNamespace('zen4R')) stop ("Package zen4R is required to download from Zenodo")
  zen4R::download_zenodo(zenodoDOI, path = tmpfolder, quiet = TRUE)
  reloadedfrom <- zenodoDOI
}
files <- list.files(path = tmpfolder, full.names = TRUE)
files <- files[grepl('.rdata', tolower(files))]
reloaded <- unlist(lapply(files, load, envir = .GlobalEnv))
required <- c('sims1', 'sims2', 'sims3',
              'sims1f', 'sims2f', 'sims3f',
              'sims1M', 'sims2M', 'sims3M',
              'sims1fM', 'sims2fM', 'sims3fM',
              'sims1MCL', 'sims2MCL', 'sims3MCL',
              'sims2C', 'sims2CCL', 'sims2Cf', 'sims2CfCL'
              )
if (!all(required %in% reloaded)) {
  warning ("Could not reload all required simulations")
} else {
  cat ("successfully reloaded all required files from ", reloadedfrom, "\n")
}
```

```
## successfully reloaded all required files from 10.5281/zenodo.15455288
```

Now summarise the simulations. The summary tables are printed in Appendix 2.

```
# no fit
sum1 <- summary_n(sims1)
sum2 <- summary_n(sims2)
sum3 <- summary_n(sims3)
sum1f <- summary_n(sims1f)
sum2f <- summary_n(sims2f)
sum3f <- summary_n(sims3f)

# vs detection-weighted local density
sum1M <- summary_M(sims1M)
sum2M <- summary_M(sims2M)
sum3M <- summary_M(sims3M)
sum1fM <- summary_M(sims1fM)
```

```

sum2fM <- summary_M(sims2fM)
sum3fM <- summary_M(sims3fM)

# vs global density
sum1MT <- summary_M(sims1M, true = localpar$D)
sum2MT <- summary_M(sims2M, true = localpar$D)
sum3MT <- summary_M(sims3M, true = localpar$D)
sum1fMT <- summary_M(sims1fM, true = localpar$D)
sum2fMT <- summary_M(sims2fM, true = localpar$D)
sum3fMT <- summary_M(sims3fM, true = localpar$D)

# conditional likelihood (for COV(a) etc.)
sum1MCL <- summary_MCL(sims1MCL, true = localpar$D)
sum2MCL <- summary_MCL(sims2MCL, true = localpar$D)
sum3MCL <- summary_MCL(sims3MCL, true = localpar$D)

```

Summaries are optionally output to .csv:

```

if (!is.null(csvdir)) {

  # no fit
  write.csv(sum1, file= paste0(csvdir, '/sum1.csv'))
  write.csv(sum2, file= paste0(csvdir, '/sum2.csv'))
  write.csv(sum3, file= paste0(csvdir, '/sum3.csv'))

  # no fit, fixed N
  write.csv(sum1f, file= paste0(csvdir, '/sum1f.csv'))
  write.csv(sum2f, file= paste0(csvdir, '/sum2f.csv'))
  write.csv(sum3f, file= paste0(csvdir, '/sum3f.csv'))

  # fitted model vs detection-weighted local density
  write.csv(sum1M, file= paste0(csvdir, '/sum1M.csv'))
  write.csv(sum2M, file= paste0(csvdir, '/sum2M.csv'))
  write.csv(sum3M, file= paste0(csvdir, '/sum3M.csv'))

  # fitted model vs detection-weighted local density, fixed N
  write.csv(sum1fM, file= paste0(csvdir, '/sum1fM.csv'))
  write.csv(sum2fM, file= paste0(csvdir, '/sum2fM.csv'))
  write.csv(sum3fM, file= paste0(csvdir, '/sum3fM.csv'))

  # fitted model vs global density
  write.csv(sum1MT, file= paste0(csvdir, '/sum1MT.csv'))
  write.csv(sum2MT, file= paste0(csvdir, '/sum2MT.csv'))
  write.csv(sum3MT, file= paste0(csvdir, '/sum3MT.csv'))

  # fitted model vs global density, fixed N
  write.csv(sum1fMT, file= paste0(csvdir, '/sum1fMT.csv'))
  write.csv(sum2fMT, file= paste0(csvdir, '/sum2fMT.csv'))
  write.csv(sum3fMT, file= paste0(csvdir, '/sum3fMT.csv'))

  # conditional likelihood (for COV(a) etc.)
  write.csv(sum1MCL, file= paste0(csvdir, '/sum1MCL.csv'))
  write.csv(sum2MCL, file= paste0(csvdir, '/sum2MCL.csv'))
  write.csv(sum3MCL, file= paste0(csvdir, '/sum3MCL.csv'))
}

```



```
}
```

## Cohesion

Additional simulations were performed with overdispersion in the *detection* process. A novel function (`sim.cohesion`) is defined in **overdispim** for simulating detection with variable within-cluster cohesion, ranging from none ( $\gamma = 0$ ) to complete ( $\gamma = 1$ ) (Bischof et al. 2020). This applies *only* to clustered AC.

Table 2. Simulations with complete within-cluster cohesion.

Code	AC distribution	Fit	extractfn
2C	Thomas process	none	extract_n
2Cf	fixed-N(A) Thomas process	none	extract_n
2CCL	Thomas process	conditional likelihood	extract_M
2CfCL	fixed-N(A) Thomas process	conditional likelihood	extract_M

```
detargs <- list (savepopn = TRUE, gamma = 1) # complete cohesion
# No fit
sims2C <- run_all(nrepl_n, popargs2, CH.function = "sim.cohesion", detargs = detargs,
  fit = FALSE)
sims2Cf <- run_all(nrepl_n, popargs2f, CH.function = "sim.cohesion", detargs = detargs,
  fit = FALSE)
# Conditional likelihood fit (effect of cohesion on coverage of a-hat)
sims2CCL <- run_all(nrepl_M, popargs2[1:6], CH.function = "sim.cohesion", detargs = detargs,
  fit = TRUE, CL = TRUE)
sims2CfCL <- run_all(nrepl_M, popargs2f[1:6], CH.function = "sim.cohesion", detargs = detargs,
  fit = TRUE, CL = TRUE)

sum2C <- summary_n(sims2C)
sum2Cf <- summary_n(sims2Cf)
sum2CCL <- summary_MCL(sims2CCL, true = localpar$D)
sum2CfCL <- summary_MCL(sims2CfCL, true = localpar$D)
```

## References

- Baddeley, A., Rubak, E., and Turner, R. (2015) *Spatial Point Patterns: Methodology and Applications with R*. Chapman and Hall/CRC Press, London.
- Bischof, R., Dupont, P., Milleret, C., Chipperfield, J., and Royle, J. A. (2020) Consequences of ignoring group association in spatial capture–recapture analysis. *Wildlife Biology* wlb.00649. DOI 10.2981/wlb.00649
- Blondel, E. (2024) zen4R: Interface to ‘Zenodo’ REST API. R package version 0.10. <https://CRAN.R-project.org/package=zen4R/>
- Borchers, D. L. and Efford, M. G. (2008) Spatially explicit maximum likelihood methods for capture–recapture studies. *Biometrics* **64**, 377–385.
- Efford, M. G. (2025a) secr: Spatially explicit capture–recapture models. R package version 5.2.2. <https://CRAN.R-project.org/package=secr/>
- Efford, M. G. (2025b) openCR: Spatially explicit capture–recapture models. R package version 2.9.3. <https://CRAN.R-project.org/package=openCR/>

Efford, M. G. (2025c) Simulations of overdispersed activity centres in spatially explicit capture-recapture [Data set]. Zenodo. <https://doi.org/10.5281/zenodo.14873669>

Efford, M. G. and Fletcher, D. (2024) The effect of spatial overdispersion on confidence intervals for population density estimated by spatially explicit capture-recapture. *bioRxiv* <https://doi.org/10.1101/2024.03.12.584742>.

Saura, S., and Martínez-Millán, J. (2000) Landscape patterns simulation with a modified random clusters method. *Landscape Ecology* **15**, 661–678.

Schlather, M., Malinowski, A., Menck, P. J., Oesting, M., and Stokorb, K. (2015) Analysis, simulation and prediction of multivariate random fields with package RandomFields. *Journal of Statistical Software* **63**, 1–25. <https://www.jstatsoft.org/v63/i08/>

## Appendix 1. Example

This is an aside, intended to show the code in action.

First select a subset of cluster (Thomas process) population scenarios and extract the parameter levels for later use.

```
supply(popargs2, '[' , 'details')[1:2,] # display parameter values
pops <- popargs2[13:18]                # select mu = 1..32, scale = 2
```

And plot an example with  $\mu = 32$  and  $\text{scale} = 2$ :

```
set.seed(12345)
pop <- do.call(sim.popn, popargs2[[18]])
plot(pop) # AC
plot(localpar$traps, add = TRUE) # detectors
```

Further execution of the example is suppressed by default, but can be cut-and-pasted into an R session.

### No model fit

Messages are suppressed.

```
sims <- run_all(nrepl = 100, pops, fit = FALSE)
summary_n(sims)
```

### Model fit

Using the same population scenarios -

```
simsM <- run_all(nrepl = 10, pops, fit = TRUE)
summary_M(simsM)
```

## Appendix 2. Simulation summaries

### No model

Column	Description
[var, scale etc.]	Parameters specific to the scenarios
varration	Ratio $\text{var}(n) / \text{mean}(n)$
chatF	Mean ‘Fletcher-chat’ for count $n_k$ (individuals per detector)
chatW	Mean ‘Wedderburn-chat’ for count $n_k$ (individuals per detector)
Nsim	number of datasets simulated

Column	Description
N	mean number of individuals in simulated population
varN	variance of number of individuals in simulated population
n	mean number of individuals detected
varn	variance of number of individuals detected
localD	mean of detection-weighted density
varlocalD	variance of detection-weighted density
VRlocalD	variance ratio from localD : $(\text{varlocalD}/\text{localD}^2 + 1/\text{En}) / (1/\text{En})$

sum1 # *LGCP Poisson N(A)*

##		var	scale	var	variation	chatF	chatW	Nsim	N	varN	n	varn	localD
##	[1,]	0.000	2	0.99245	1.0236	1.0235	10000	255.88	258.74	180.64	179.27	2844.4	
##	[2,]	0.125	2	1.68665	1.1656	1.1679	10000	256.17	443.26	180.86	305.04	2845.7	
##	[3,]	0.250	2	2.46564	1.3213	1.3299	10000	256.56	652.35	181.19	446.76	2849.1	
##	[4,]	0.500	2	4.03615	1.6491	1.6700	10000	255.91	1081.94	180.82	729.82	2842.6	
##	[5,]	0.750	2	5.62483	2.0036	2.0478	10000	256.11	1514.58	180.65	1016.13	2841.0	
##	[6,]	1.000	2	7.61201	2.4179	2.5222	10000	256.71	2087.58	181.19	1379.22	2851.0	
##	[7,]	0.000	5	0.99076	1.0216	1.0210	10000	256.11	254.15	180.71	179.04	2844.4	
##	[8,]	0.125	5	4.20222	1.2630	1.2684	10000	255.79	1166.92	180.65	759.13	2844.8	
##	[9,]	0.250	5	7.49596	1.5160	1.5349	10000	256.26	2108.89	180.72	1354.64	2847.6	
##	[10,]	0.500	5	14.08327	2.0564	2.1461	10000	256.65	4009.15	181.08	2550.26	2848.5	
##	[11,]	0.750	5	22.49622	2.6528	2.9097	10000	257.53	6392.31	182.11	4096.83	2867.5	
##	[12,]	1.000	5	31.36740	3.3012	3.7434	10000	255.81	8878.25	180.17	5651.36	2840.1	
##	[13,]	0.000	10	1.02480	1.0207	1.0197	10000	255.86	262.32	180.52	185.00	2844.4	
##	[14,]	0.125	10	8.49390	1.3160	1.3254	10000	256.39	2555.69	180.97	1537.10	2846.2	
##	[15,]	0.250	10	16.14817	1.6140	1.6444	10000	256.09	4912.57	180.65	2917.14	2842.4	
##	[16,]	0.500	10	33.38755	2.2638	2.4123	10000	255.63	10113.74	180.28	6019.12	2838.9	
##	[17,]	0.750	10	50.46441	2.9385	3.2699	10000	253.18	15161.13	178.56	9010.90	2815.8	
##	[18,]	1.000	10	75.04071	3.7679	4.5916	10000	256.99	22816.81	181.41	13613.44	2856.2	
##		varlocalD		VRlocalD									
##	[1,]		0	1.0000									
##	[2,]		31769	1.7089									
##	[3,]		66398	2.4780									
##	[4,]		136368	4.0495									
##	[5,]		206878	5.6313									
##	[6,]		298914	7.6450									
##	[7,]		0	1.0000									
##	[8,]		144762	4.2323									
##	[9,]		292922	7.5272									
##	[10,]		594044	14.2292									
##	[11,]		982467	22.5895									
##	[12,]		1355672	31.3683									
##	[13,]		0	1.0000									
##	[14,]		334931	8.4706									
##	[15,]		673894	16.0722									
##	[16,]		1455533	33.6346									
##	[17,]		2195648	51.0380									
##	[18,]		3321925	74.5787									

sum2 # *Thomas Poisson N(A)*

##		mu	scale	var	variation	chatF	chatW	Nsim	N	varN	n	varn	localD
##	[1,]	1	1e-04	1.9515	1.5942	1.5979	10000	255.97	525.71	180.77	352.77	2846.1	

```

## [2,] 2 1e-04 2.8197 2.1514 2.1543 10000 256.26 759.87 180.64 509.34 2844.4
## [3,] 4 1e-04 4.6758 3.2979 3.3087 10000 255.89 1264.75 180.56 844.26 2840.0
## [4,] 8 1e-04 8.4963 5.5645 5.5728 10000 255.52 2375.23 180.39 1532.60 2841.3
## [5,] 16 1e-04 15.7522 10.1470 10.1935 10000 256.85 4370.05 181.45 2858.29 2855.6
## [6,] 32 1e-04 30.8829 19.7672 19.4955 9988 257.89 8499.70 182.20 5626.97 2873.5
## [7,] 1 1e+00 1.8845 1.3501 1.3517 10000 256.10 491.33 180.71 340.55 2845.8
## [8,] 2 1e+00 2.8045 1.6905 1.6928 10000 255.97 741.55 180.69 506.74 2843.7
## [9,] 4 1e+00 4.4429 2.3584 2.3670 10000 256.11 1187.99 180.92 803.84 2845.6
## [10,] 8 1e+00 8.0496 3.6882 3.7002 10000 256.15 2172.81 181.08 1457.60 2851.1
## [11,] 16 1e+00 14.9667 6.3834 6.3463 10000 255.51 4036.53 180.19 2696.87 2836.0
## [12,] 32 1e+00 29.0646 12.6467 11.7497 9993 255.41 7834.09 180.06 5233.44 2835.9
## [13,] 1 2e+00 1.7829 1.1637 1.1603 10000 255.69 468.04 180.29 321.43 2840.8
## [14,] 2 2e+00 2.5815 1.3139 1.3178 10000 256.45 686.65 181.02 467.31 2848.2
## [15,] 4 2e+00 4.3688 1.6088 1.6143 10000 256.31 1166.32 180.93 790.46 2845.5
## [16,] 8 2e+00 7.4237 2.1859 2.1763 10000 255.58 2019.52 180.27 1338.29 2841.2
## [17,] 16 2e+00 13.7286 3.4020 3.3624 10000 255.64 3739.45 180.12 2472.85 2839.0
## [18,] 32 2e+00 27.1992 5.9977 5.7396 9999 256.96 7381.74 181.25 4929.97 2851.9
## [19,] 1 4e+00 1.6799 1.0680 1.0687 10000 256.30 447.67 180.88 303.86 2848.7
## [20,] 2 4e+00 2.3468 1.1123 1.1122 10000 256.21 630.24 180.80 424.30 2845.2
## [21,] 4 4e+00 3.7444 1.2023 1.2031 10000 256.20 1016.82 181.00 677.71 2848.0
## [22,] 8 4e+00 6.3141 1.3877 1.3814 10000 256.30 1724.33 180.84 1141.87 2846.2
## [23,] 16 4e+00 11.4893 1.7612 1.7380 10000 256.07 3194.12 180.85 2077.86 2848.8
## [24,] 32 4e+00 22.5649 2.6180 2.4849 10000 257.24 6322.40 181.65 4098.83 2861.0
##      varlocalD VRlocalD
## [1,] 41457 1.9248
## [2,] 80883 2.8065
## [3,] 164846 4.6931
## [4,] 335048 8.4992
## [5,] 662692 15.6847
## [6,] 1338423 30.2892
## [7,] 40093 1.8946
## [8,] 80718 2.8036
## [9,] 153265 4.4200
## [10,] 313790 7.9752
## [11,] 618597 14.8978
## [12,] 1244243 28.9563
## [13,] 35576 1.7966
## [14,] 72455 2.6139
## [15,] 149770 4.3425
## [16,] 287085 7.4263
## [17,] 564954 13.6654
## [18,] 1175481 27.1160
## [19,] 30660 1.6827
## [20,] 59222 2.3219
## [21,] 118003 3.6288
## [22,] 235601 6.2552
## [23,] 464264 11.3367
## [24,] 971294 22.4419

```

```
sum3 # Random habitat Poisson N(A)
```

```

##      p      A varration  chatF  chatW  Nsim      N      varN      n      varn localD
## [1,] 0.25 0.0625 10.1904 5.7224 5.5442 10000 244.55 2669.49 172.23 1755.11 2711.8
## [2,] 0.25 0.1250 5.3219 3.2687 3.1591 10000 246.45 1394.87 173.62 923.98 2733.3
## [3,] 0.25 0.2500 2.9727 2.0106 1.9656 10000 250.52 772.83 176.53 524.77 2783.0

```

```

## [4,] 0.25 0.5000    1.6728  1.3531  1.3548 10000 256.20   439.67 180.81   302.45 2846.2
## [5,] 0.25 0.7500    1.2025  1.1264  1.1334 10000 257.60   311.23 181.86   218.69 2864.5
## [6,] 0.25 1.0000    1.0210  1.0206  1.0205 10000 256.10   255.65 180.78   184.57 2844.4
## [7,] 0.50 0.0625   82.0432 14.6344 15.4264 10000 243.85 21086.70 171.72 14088.11 2700.5
## [8,] 0.50 0.1250   38.9567  7.9088  7.7991 10000 245.06  9960.17 172.45  6718.08 2714.9
## [9,] 0.50 0.2500   17.7696  4.1351  4.0193 10000 250.56  4648.48 176.77  3141.19 2785.4
## [10,] 0.50 0.5000    6.3848  2.0811  2.0437 10000 256.04  1707.66 180.69  1153.69 2847.1
## [11,] 0.50 0.7500    2.8202  1.3666  1.3645 10000 257.99   751.02 182.32   514.19 2865.6
## [12,] 0.50 1.0000    1.0090  1.0221  1.0214 10000 255.84   257.99 180.57   182.19 2844.4
##      varlocalD VRlocalD
## [1,] 394297.8  10.6884
## [2,] 186531.9   5.5116
## [3,]  87158.9   3.0334
## [4,]  29269.0   1.6529
## [5,]   9469.7   1.2085
## [6,]    0.0    1.0000
## [7,] 3435483.1  86.1249
## [8,] 1624819.1  40.8331
## [9,]  739692.8  18.2270
## [10,] 245677.3   6.4765
## [11,]  81988.9   2.8041
## [12,]    0.0    1.0000

```

```
sumlf # LGCP fixed N(A)
```

```

##      var scale varration  chatF  chatW  Nsim  N varN      n      varn localD varlocalD
## [1,] 0.000      2  0.29472 1.0090 1.0087 10000 256    0 180.66  53.244 2844.4      0
## [2,] 0.125      2  0.36986 1.1449 1.1459 10000 256    0 180.70  66.835 2841.4    19261
## [3,] 0.250      2  0.43128 1.2885 1.2891 10000 256    0 180.60  77.887 2840.1    29111
## [4,] 0.500      2  0.56122 1.5800 1.5809 10000 256    0 180.43 101.262 2836.1    41958
## [5,] 0.750      2  0.74428 1.9168 1.9223 10000 256    0 180.61 134.424 2837.6    51114
## [6,] 1.000      2  0.90422 2.2630 2.2705 10000 256    0 180.21 162.954 2831.9    60226
## [7,] 0.000      5  0.29611 1.0119 1.0120 10000 256    0 180.71  53.509 2844.4      0
## [8,] 0.125      5  0.42046 1.1997 1.1969 10000 256    0 180.44  75.866 2837.7    35588
## [9,] 0.250      5  0.55801 1.3968 1.3957 10000 256    0 180.43 100.683 2837.5    44841
## [10,] 0.500      5  0.85296 1.8135 1.8089 10000 256    0 179.97 153.507 2826.8    59986
## [11,] 0.750      5  1.16479 2.2458 2.2374 10000 256    0 179.50 209.082 2817.2    74005
## [12,] 1.000      5  1.47639 2.7350 2.7313 10000 256    0 179.51 265.032 2818.1    90358
## [13,] 0.000     10  0.29366 1.0068 1.0065 10000 256    0 180.65  53.049 2844.4      0
## [14,] 0.125     10  0.42407 1.1820 1.1795 10000 256    0 180.47  76.534 2836.6    39128
## [15,] 0.250     10  0.58069 1.3537 1.3485 10000 256    0 180.13 104.600 2832.0    46074
## [16,] 0.500     10  0.90157 1.7221 1.7074 10000 256    0 179.53 161.863 2823.2    61645
## [17,] 0.750     10  1.18220 2.1031 2.0815 10000 256    0 179.23 211.891 2816.2    73503
## [18,] 1.000     10  1.56242 2.4980 2.4598 10000 256    0 178.50 278.896 2806.9    90877
##      VRlocalD
## [1,]  1.0000
## [2,]  1.4311
## [3,]  1.6521
## [4,]  1.9426
## [5,]  2.1471
## [6,]  2.3570
## [7,]  1.0000
## [8,]  1.7986
## [9,]  2.0064
## [10,] 2.3564

```

```
## [11,] 2.6849
## [12,] 3.0558
## [13,] 1.0000
## [14,] 1.8787
## [15,] 2.0381
## [16,] 2.3975
## [17,] 2.6747
## [18,] 3.0843
```

```
sum2f # Thomas process fixed N(A)
```

```
##      mu scale varration chatF chatW Nsim  N varN      n      varn localD varlocalD
## [1,] 1      2      0.36914 1.1459 1.1461 10000 256      0 180.65 66.683 2845.2      21293
## [2,] 2      2      0.43387 1.2782 1.2783 10000 256      0 180.72 78.407 2844.8      30640
## [3,] 4      2      0.58498 1.5456 1.5468 10000 256      0 180.66 105.685 2847.5      42370
## [4,] 8      2      0.85208 2.0814 2.0825 10000 256      0 180.64 153.918 2846.5      58968
## [5,] 16     2      1.43674 3.1561 3.1518 10000 256      0 180.36 259.131 2846.2      86831
## [6,] 32     2      2.60334 5.2964 5.2967 10000 256      0 180.48 469.863 2845.5      138297
##      VRlocalD
## [1,] 1.4753
## [2,] 1.6841
## [3,] 1.9442
## [4,] 2.3150
## [5,] 2.9369
## [6,] 4.0863
```

```
sum3f # Random habitat fixed N(A)
```

```
##      p      A varration chatF chatW Nsim  N varN      n      varn localD
## [1,] 0.25 0.0625 2.26640 5.9074 5.9198 10000 256      0 179.94 407.823 2702.5
## [2,] 0.25 0.1250 1.23438 3.3139 3.3161 10000 256      0 180.28 222.540 2734.8
## [3,] 0.25 0.2500 0.68621 1.9983 1.9966 10000 256      0 180.37 123.769 2778.7
## [4,] 0.25 0.5000 0.41695 1.3399 1.3390 10000 256      0 180.64 75.317 2845.1
## [5,] 0.25 0.7500 0.33455 1.1153 1.1159 10000 256      0 180.82 60.493 2865.6
## [6,] 0.25 1.0000 0.28844 1.0098 1.0099 10000 256      0 180.67 52.112 2844.4
## [7,] 0.50 0.0625 7.91690 15.0237 14.6784 10000 256      0 174.22 1379.295 2705.4
## [8,] 0.50 0.1250 4.00903 7.9921 7.8031 10000 256      0 176.81 708.833 2734.8
## [9,] 0.50 0.2500 1.87218 4.0732 4.0103 10000 256      0 178.93 334.986 2780.9
## [10,] 0.50 0.5000 0.81551 2.0284 2.0108 10000 256      0 180.17 146.928 2844.3
## [11,] 0.50 0.7500 0.46354 1.3426 1.3374 10000 256      0 180.42 83.633 2862.5
## [12,] 0.50 1.0000 0.29686 1.0097 1.0083 10000 256      0 180.46 53.572 2844.4
##      varlocalD VRlocalD
## [1,] 405161.3 11.0243
## [2,] 189515.2 5.5786
## [3,] 85666.7 3.0048
## [4,] 29171.9 1.6512
## [5,] 9406.1 1.2070
## [6,] 0.0 1.0000
## [7,] 3338345.8 83.4196
## [8,] 1682055.3 41.6374
## [9,] 723218.1 17.8984
## [10,] 245287.9 6.4785
## [11,] 82804.1 2.8261
## [12,] 0.0 1.0000
```

## Model fitted

Column	Description
[var, scale etc.]	Parameters specific to the scenarios
n	mean number of individuals detected
N	mean number of individuals in simulated population
nvalid	number of successful simulations
estimate	mean estimated density
SE.estimate	mean SE of estimate
RSE	ratio of preceding
trueD	true density; either detection-weighted (default) or global as specified by the 'true' argument
RB	estimated relative bias relative to trueD
seRB	SE of RB
COV	unadjusted coverage of 95% interval relative to trueD
COVF	adjusted coverage
chatF	mean 'Fletcher-chat' for count $n_k$ (individuals per detector)
varration	Ratio var(n) / mean(n)

Summarise relative to local density –

sum1M # *LGCP Poisson N(A)*

##	var	scale	n	N	nvalid	estimate	SE.estimate	RSE	trueD	RB
## [1,]	0.000	10	180.41	255.85	1000	2839.7	212.93	0.074857	2844.4	-0.00165807
## [2,]	0.125	10	180.28	255.56	1000	2839.1	211.83	0.075970	2845.1	-0.00254059
## [3,]	0.250	10	178.61	254.10	1000	2813.4	209.68	0.077459	2828.8	-0.00556826
## [4,]	0.500	10	180.48	256.00	1000	2842.6	208.67	0.079521	2856.5	-0.00482499
## [5,]	0.750	10	178.91	253.83	1000	2818.0	205.94	0.082149	2822.0	-0.00335182
## [6,]	1.000	10	182.52	258.94	1000	2874.7	206.05	0.084302	2877.3	0.00079733
##	seRB	COV	COVF	chatF	varration					
## [1,]	0.0023270	0.949	0.955	1.0028	0.95833					
## [2,]	0.0024735	0.946	0.962	1.1682	8.94715					
## [3,]	0.0024919	0.955	0.980	1.3598	18.45895					
## [4,]	0.0025272	0.950	0.984	1.7308	35.68167					
## [5,]	0.0025814	0.958	0.988	2.0670	50.03398					
## [6,]	0.0026938	0.954	0.992	2.5461	68.79657					

sum2M # *Thomas Poisson N(A)*

##	mu	scale	n	N	nvalid	estimate	SE.estimate	RSE	trueD	RB
## [1,]	1	2	180.78	256.33	1000	2848.5	213.25	0.075030	2848.5	0.00003581
## [2,]	2	2	180.14	254.89	1000	2835.2	212.51	0.075336	2832.7	0.00069200
## [3,]	4	2	181.92	257.38	1000	2865.1	213.47	0.075198	2864.8	0.00016706
## [4,]	8	2	180.42	255.41	1000	2842.4	212.14	0.075760	2853.2	-0.00401549
## [5,]	16	2	180.29	256.52	1000	2840.7	211.01	0.077462	2840.3	0.00029217
## [6,]	32	2	180.59	254.65	1000	2843.8	208.91	0.080513	2848.6	-0.00240334
##	seRB	COV	COVF	chatF	varration					
## [1,]	0.0024017	0.952	0.966	1.1440	1.7964					
## [2,]	0.0023799	0.938	0.964	1.2668	2.6201					
## [3,]	0.0023800	0.954	0.986	1.5516	4.2845					
## [4,]	0.0023878	0.951	0.995	2.0623	7.6649					
## [5,]	0.0024258	0.954	0.998	3.1192	14.8511					
## [6,]	0.0025481	0.960	0.999	5.1320	27.4153					

```
sum3M # Random habitat Poisson N(A)
```

##	p	A	n	N	nvalid	estimate	SE.estimate	RSE	trueD	RB
## [1,]	0.5	0.0625	173.59	246.59	1000	2733.8	198.29	0.092845	2732.6	0.00344502
## [2,]	0.5	0.1250	176.92	249.66	1000	2785.4	205.34	0.082412	2775.6	0.00093416
## [3,]	0.5	0.2500	177.19	251.34	1000	2789.3	208.64	0.078612	2784.5	0.00197242
## [4,]	0.5	0.5000	179.97	254.85	1000	2834.0	211.89	0.076073	2831.0	0.00055428
## [5,]	0.5	0.7500	181.95	257.93	1000	2865.7	213.66	0.075016	2863.9	0.00043249
## [6,]	0.5	1.0000	180.39	255.56	1000	2840.7	213.00	0.074884	2844.4	-0.00131566
##	seRB	COV	COVF	chatF	varration					
## [1,]	0.0031768	0.950	1.000	12.5557	79.9847					
## [2,]	0.0027466	0.943	1.000	7.1100	41.2310					
## [3,]	0.0025757	0.944	0.999	3.7833	17.3613					
## [4,]	0.0024379	0.943	0.988	1.9548	6.7732					
## [5,]	0.0024436	0.945	0.967	1.3330	3.0225					
## [6,]	0.0024268	0.945	0.950	1.0084	1.0495					

```
sum1fM # LGCP fixed N(A)
```

##	var	scale	n	N	nvalid	estimate	SE.estimate	RSE	trueD	RB
## [1,]	0.000	10	180.53	256	1000	2836.2	117.44	0.041289	2844.4	-0.00288660
## [2,]	0.125	10	180.55	256	1000	2837.1	117.49	0.041736	2826.4	0.00737808
## [3,]	0.250	10	180.94	256	1000	2842.6	117.56	0.041569	2841.7	0.00427704
## [4,]	0.500	10	179.57	256	1000	2821.5	117.15	0.041746	2823.0	0.00374049
## [5,]	0.750	10	178.63	256	1000	2806.0	116.76	0.041696	2819.2	-0.00067265
## [6,]	1.000	10	177.98	256	1000	2796.6	116.61	0.041932	2801.9	0.00196996
##	seRB	COV	COVF	chatF	varration					
## [1,]	0.0012856	0.955	0.960	1.0058	0.28361					
## [2,]	0.0023048	0.727	0.765	1.1722	0.43590					
## [3,]	0.0023200	0.731	0.792	1.3535	0.53914					
## [4,]	0.0024148	0.721	0.821	1.6926	0.82003					
## [5,]	0.0023429	0.738	0.877	2.0995	1.16323					
## [6,]	0.0023323	0.742	0.913	2.4541	1.47605					

```
sum2fM # Thomas fixed N(A)
```

##	mu	scale	n	N	nvalid	estimate	SE.estimate	RSE	trueD	RB
## [1,]	1	2	180.87	256	1000	2843.4	117.73	0.041468	2846.3	1.1849e-03
## [2,]	2	2	181.10	256	1000	2846.2	117.69	0.041404	2851.7	7.4964e-04
## [3,]	4	2	179.95	256	1000	2829.4	117.45	0.041522	2841.7	-7.7040e-04
## [4,]	8	2	180.21	256	1000	2832.1	117.36	0.041490	2843.2	-5.4217e-04
## [5,]	16	2	180.75	256	1000	2842.0	117.66	0.041625	2845.2	2.1757e-03
## [6,]	32	2	181.17	256	1000	2845.5	117.42	0.041555	2857.5	-6.0318e-05
##	seRB	COV	COVF	chatF	varration					
## [1,]	0.0018688	0.843	0.864	1.1391	0.36090					
## [2,]	0.0020438	0.785	0.831	1.2770	0.47138					
## [3,]	0.0022310	0.764	0.844	1.5261	0.60383					
## [4,]	0.0022181	0.776	0.899	2.0714	0.86592					
## [5,]	0.0022865	0.746	0.938	3.0704	1.43629					
## [6,]	0.0024376	0.727	0.977	5.2926	2.57225					

```
sum3fM # Random habitat fixed N(A)
```

##	p	A	n	N	nvalid	estimate	SE.estimate	RSE	trueD	RB
## [1,]	0.5	0.0625	175.42	256	636	2706.5	114.51	0.042322	2840.7	-0.031199
## [2,]	0.5	0.1250	176.53	256	749	2764.8	115.62	0.040939	2860.6	-0.032500



```
## [3,] 0.5 0.2500 179.87 256      872    2812.9      116.93 0.040584 2909.3 -0.029165
## [4,] 0.5 0.5000 179.25 256      985    2813.8      116.91 0.040369 2903.4 -0.030488
## [5,] 0.5 0.7500 180.57 256      999    2838.4      117.61 0.040273 2923.7 -0.029139
## [6,] 0.5 1.0000 181.25 256     1000    2847.7      117.68 0.040212 2928.1 -0.027488
##          seRB      COV  COVF    chatF varration
## [1,] 0.00122320 0.95440 1.000 12.3827   8.13516
## [2,] 0.00092689 0.97196 1.000  7.1206   3.96755
## [3,] 0.00082460 0.99083 1.000  3.8178   1.90863
## [4,] 0.00071098 0.98376 1.000  2.0137   0.80661
## [5,] 0.00068789 0.99399 0.998  1.3329   0.46176
## [6,] 0.00069386 0.98800 0.992  1.0008   0.30812
```

Summarise relative to global density –

sum1MT # *LGCP Poisson N(A)*

```
##          var scale      n      N nvalid estimate SE.estimate      RSE  trueD      RB
## [1,] 0.000      10 180.41 255.85    1000    2839.7      212.93 0.074857 2844.4 -0.00165807
## [2,] 0.125      10 180.28 255.56    1000    2839.1      211.83 0.074470 2844.4 -0.00188017
## [3,] 0.250      10 178.61 254.10    1000    2813.4      209.68 0.073716 2844.4 -0.01090800
## [4,] 0.500      10 180.48 256.00    1000    2842.6      208.67 0.073361 2844.4 -0.00063536
## [5,] 0.750      10 178.91 253.83    1000    2818.0      205.94 0.072402 2844.4 -0.00929904
## [6,] 1.000      10 182.52 258.94    1000    2874.7      206.05 0.072439 2844.4  0.01064051
##          seRB      COV  COVF    chatF varration
## [1,] 0.0023270 0.949 0.955 1.0028   0.95833
## [2,] 0.0070575 0.515 0.544 1.1682   8.94715
## [3,] 0.0100575 0.357 0.418 1.3598  18.45895
## [4,] 0.0140470 0.272 0.362 1.7308  35.68167
## [5,] 0.0165683 0.238 0.325 2.0670  50.03398
## [6,] 0.0196247 0.217 0.326 2.5461  68.79657
```

sum2MT # *Thomas Poisson N(A)*

```
##          mu scale      n      N nvalid estimate SE.estimate      RSE  trueD      RB
## [1,] 1      2 180.78 256.33    1000    2848.5      213.25 0.074972 2844.4  0.00141947
## [2,] 2      2 180.14 254.89    1000    2835.2      212.51 0.074709 2844.4 -0.00324955
## [3,] 4      2 181.92 257.38    1000    2865.1      213.47 0.075048 2844.4  0.00726087
## [4,] 8      2 180.42 255.41    1000    2842.4      212.14 0.074582 2844.4 -0.00072018
## [5,] 16     2 180.29 256.52    1000    2840.7      211.01 0.074184 2844.4 -0.00131664
## [6,] 32     2 180.59 254.65    1000    2843.8      208.91 0.073446 2844.4 -0.00023791
##          seRB      COV  COVF    chatF varration
## [1,] 0.0031668 0.860 0.891 1.1440   1.7964
## [2,] 0.0038008 0.784 0.838 1.2668   2.6201
## [3,] 0.0049038 0.672 0.778 1.5516   4.2845
## [4,] 0.0065224 0.517 0.684 2.0623   7.6649
## [5,] 0.0090643 0.383 0.624 3.1192  14.8511
## [6,] 0.0123188 0.282 0.619 5.1320  27.4153
```

sum3MT # *Random habitat Poisson N(A)*

```
##          p      A      n      N nvalid estimate SE.estimate      RSE  trueD      RB
## [1,] 0.5 0.0625 173.59 246.59    1000    2733.8      198.29 0.069711 2844.4 -0.0388946
## [2,] 0.5 0.1250 176.92 249.66    1000    2785.4      205.34 0.072191 2844.4 -0.0207409
## [3,] 0.5 0.2500 177.19 251.34    1000    2789.3      208.64 0.073349 2844.4 -0.0193934
## [4,] 0.5 0.5000 179.97 254.85    1000    2834.0      211.89 0.074492 2844.4 -0.0036849
## [5,] 0.5 0.7500 181.95 257.93    1000    2865.7      213.66 0.075116 2844.4  0.0074584
## [6,] 0.5 1.0000 180.39 255.56    1000    2840.7      213.00 0.074884 2844.4 -0.0013157
```

```
##          seRB   COV   COVF   chatF varration
## [1,] 0.0206043 0.160 0.561 12.5557   79.9847
## [2,] 0.0149577 0.230 0.601  7.1100   41.2310
## [3,] 0.0097142 0.354 0.628  3.7833   17.3613
## [4,] 0.0061204 0.545 0.710  1.9548    6.7732
## [5,] 0.0041230 0.730 0.795  1.3330    3.0225
## [6,] 0.0024268 0.945 0.950  1.0084    1.0495
```

sum1fMT # *LGCP Poisson N(A)*

```
##          var scale      n   N nvalid estimate SE.estimate      RSE  trueD      RB
## [1,] 0.000      10 180.53 256   1000   2836.2      117.44 0.041289 2844.4 -0.00288660
## [2,] 0.125      10 180.55 256   1000   2837.1      117.49 0.041304 2844.4 -0.00259471
## [3,] 0.250      10 180.94 256   1000   2842.6      117.56 0.041329 2844.4 -0.00065155
## [4,] 0.500      10 179.57 256   1000   2821.5      117.15 0.041184 2844.4 -0.00806882
## [5,] 0.750      10 178.63 256   1000   2806.0      116.76 0.041048 2844.4 -0.01350721
## [6,] 1.000      10 177.98 256   1000   2796.6      116.61 0.040997 2844.4 -0.01680326
##          seRB   COV   COVF   chatF varration
## [1,] 0.0012856 0.955 0.960 1.0058   0.28361
## [2,] 0.0015732 0.893 0.916 1.1722   0.43590
## [3,] 0.0017458 0.864 0.902 1.3535   0.53914
## [4,] 0.0021492 0.766 0.875 1.6926   0.82003
## [5,] 0.0025528 0.691 0.857 2.0995   1.16323
## [6,] 0.0028613 0.623 0.810 2.4541   1.47605
```

sum2fMT # *Thomas Poisson N(A)*

```
##          mu scale      n   N nvalid estimate SE.estimate      RSE  trueD      RB
## [1,]  1      2 180.87 256   1000   2843.4      117.73 0.041388 2844.4 -0.00035378
## [2,]  2      2 181.10 256   1000   2846.2      117.69 0.041374 2844.4  0.00062985
## [3,]  4      2 179.95 256   1000   2829.4      117.45 0.041291 2844.4 -0.00528330
## [4,]  8      2 180.21 256   1000   2832.1      117.36 0.041260 2844.4 -0.00434261
## [5,] 16      2 180.75 256   1000   2842.0      117.66 0.041364 2844.4 -0.00086732
## [6,] 32      2 181.17 256   1000   2845.5      117.42 0.041281 2844.4  0.00035765
##          seRB   COV   COVF   chatF varration
## [1,] 0.0014377 0.925 0.946 1.1391   0.36090
## [2,] 0.0016430 0.887 0.918 1.2770   0.47138
## [3,] 0.0018490 0.829 0.902 1.5261   0.60383
## [4,] 0.0022333 0.746 0.902 2.0714   0.86592
## [5,] 0.0028425 0.636 0.884 3.0704   1.43629
## [6,] 0.0038039 0.505 0.880 5.2926   2.57225
```

sum3fMT # *Random habitat Poisson N(A)*

```
##          p      A      n   N nvalid estimate SE.estimate      RSE  trueD      RB
## [1,] 0.5 0.0625 175.42 256   636   2706.5      114.51 0.040257 2844.4 -0.048506
## [2,] 0.5 0.1250 176.53 256   749   2764.8      115.62 0.040649 2844.4 -0.027992
## [3,] 0.5 0.2500 179.87 256   872   2812.9      116.93 0.041107 2844.4 -0.011100
## [4,] 0.5 0.5000 179.25 256   985   2813.8      116.91 0.041100 2844.4 -0.010771
## [5,] 0.5 0.7500 180.57 256   999   2838.4      117.61 0.041349 2844.4 -0.002136
## [6,] 0.5 1.0000 181.25 256  1000   2847.7      117.68 0.041370 2844.4  0.001151
##          seRB   COV   COVF   chatF varration
## [1,] 0.0083433 0.28302 0.81289 12.3827   8.13516
## [2,] 0.0053235 0.45527 0.84112  7.1206   3.96755
## [3,] 0.0034914 0.57913 0.86812  3.8178   1.90863
## [4,] 0.0021316 0.76954 0.91675  2.0137   0.80661
```

```
## [5,] 0.0016157 0.89790 0.93994 1.3329 0.46176
## [6,] 0.0013360 0.94800 0.95200 1.0008 0.30812
```

## Model fitted, conditional likelihood

Column	Description
[var, scale etc.]	Parameters specific to the scenarios
n	mean number of individuals detected
nvalid	number of successful simulations
estimate	mean estimated density
SE.estimate	mean SE of estimate
RSE	ratio of preceding
trueD	true density; either detection-weighted (default) or global as specified by the 'true' argument
RB	estimated relative bias relative to trueD
COV	unadjusted coverage of 95% interval relative to trueD
chatF	mean 'Fletcher-chat' for count $n_k$ (individuals per detector)
varration	Ratio var(n) / mean(n)
a	mean a-hat effective sampling area
SEa	SE of a-hat
RBa	RB(a-hat)
RSEa	RSE(a-hat)
COVa	coverage of 95% CI for a-hat
pCVn	fraction of var(D-hat) attributable to var(n)

### sum1MCL # *LGCP Poisson N(A)*

```
##          var scale      n nvalid estimate SE.estimate      RSE trueD      RB      COV
## [1,] 0.000      10 180.41   1000   2839.7      212.71 0.075106 2844.4 0.00165462 0.949
## [2,] 0.125      10 180.28   1000   2839.1      211.61 0.076330 2844.4 0.00187665 0.515
## [3,] 0.250      10 178.61   1000   2813.4      209.45 0.078064 2844.4 0.01090464 0.357
## [4,] 0.500      10 180.48   1000   2842.6      208.44 0.080050 2844.4 0.00063189 0.272
## [5,] 0.750      10 178.91   1000   2818.0      205.69 0.082677 2844.4 0.00929566 0.238
## [6,] 1.000      10 182.52   1000   2874.7      205.79 0.084337 2844.4 -0.01064404 0.216
##          chatF varration      a      SEa      RBa      RSEa      COVa      pCVn
## [1,] 0.99571   0.95833 0.063535 0.00055086 -0.00015474 0.0086737 0.956 0.98659
## [2,] 1.15995   8.94715 0.063513 0.00056111 0.00020169 0.0088384 0.946 0.98653
## [3,] 1.35020  18.45895 0.063492 0.00058084 0.00053530 0.0091530 0.944 0.98621
## [4,] 1.71866  35.68167 0.063493 0.00059641 0.00050757 0.0093983 0.953 0.98620
## [5,] 2.05242  50.03398 0.063489 0.00061907 0.00057851 0.0097581 0.950 0.98610
## [6,] 2.52815  68.79657 0.063491 0.00063768 0.00054452 0.0100531 0.961 0.98586
```

### sum2MCL # *Thomas Poisson N(A)*

```
##          mu scale      n nvalid estimate SE.estimate      RSE trueD      RB      COV      chatF
## [1,] 1      2 180.78   1000   2848.5      213.04 0.075165 2844.4 -0.00142354 0.860 1.1359
## [2,] 2      2 180.14   1000   2835.2      212.29 0.075425 2844.4 0.00324558 0.784 1.2579
## [3,] 4      2 181.92   1000   2865.1      213.26 0.075313 2844.4 -0.00726474 0.671 1.5407
## [4,] 8      2 180.42   1000   2842.4      211.93 0.076208 2844.4 0.00071646 0.516 2.0478
## [5,] 16     2 180.29   1000   2840.7      210.79 0.077573 2844.4 0.00131319 0.382 3.0973
## [6,] 32     2 180.59   1000   2843.8      208.67 0.080924 2844.4 0.00023491 0.282 5.0959
##          varration      a      SEa      RBa      RSEa      COVa      pCVn
## [1,] 1.7964 0.063470 0.00055321 0.00087121 0.0087201 0.950 0.98648
```

```
## [2,] 2.6201 0.063538 0.00055272 -0.00018870 0.0087032 0.951 0.98662
## [3,] 4.2845 0.063501 0.00055623 0.00038337 0.0087633 0.947 0.98640
## [4,] 7.6649 0.063482 0.00056422 0.00068177 0.0088915 0.943 0.98634
## [5,] 14.8511 0.063472 0.00057905 0.00085095 0.0091275 0.958 0.98613
## [6,] 27.4153 0.063513 0.00062685 0.00020532 0.0098765 0.948 0.98541
```

```
sum3MCL # Random habitat Poisson N(A)
```

```
##      p      A      n nvalid estimate SE.estimate      RSE trueD      RB      COV
## [1,] 0.5 0.0625 170.47    1000    2684.1      196.21 0.092636 2844.4 0.05638411 0.165
## [2,] 0.5 0.1250 175.64    1000    2765.6      204.51 0.083118 2844.4 0.02770617 0.226
## [3,] 0.5 0.2500 179.40    1000    2823.9      209.58 0.078271 2844.4 0.00723175 0.345
## [4,] 0.5 0.5000 180.79    1000    2847.2      212.26 0.075915 2844.4 -0.00097145 0.565
## [5,] 0.5 0.7500 181.55    1000    2858.2      213.16 0.075161 2844.4 -0.00484925 0.765
## [6,] 0.5 1.0000 180.51    1000    2842.3      212.85 0.075092 2844.4 0.00073771 0.956
##      chatF varration      a      SEa      RBa      RSEa COVa      pCVn
## [1,] 12.40657 78.65684 0.063502 0.00073237 3.7137e-04 0.0115561 0.919 0.98453
## [2,] 7.02766 38.29193 0.063509 0.00063511 2.5739e-04 0.0100162 0.922 0.98559
## [3,] 3.77776 18.04944 0.063538 0.00058288 -1.9502e-04 0.0091783 0.938 0.98623
## [4,] 1.93655 6.27424 0.063506 0.00056026 3.0983e-04 0.0088258 0.954 0.98642
## [5,] 1.30621 2.70141 0.063526 0.00055286 1.5263e-06 0.0087070 0.944 0.98651
## [6,] 0.99349 0.98934 0.063515 0.00055202 1.7452e-04 0.0086953 0.948 0.98651
```

## Complete cohesion

Thomas cluster process

```
sum2C # Poisson N(A)
```

```
##      mu scale varration      chatF      chatW Nsim      N      varN      n      varn localD
## [1,] 1 1e-04 1.9714 2.0406 2.0442 10000 255.92 501.92 180.69 356.21 2843.4
## [2,] 2 1e-04 2.9922 3.0495 3.0670 10000 256.46 782.01 180.79 540.97 2848.5
## [3,] 4 1e-04 4.9533 5.0723 5.0756 10000 255.19 1280.16 180.34 893.30 2838.3
## [4,] 8 1e-04 9.0612 9.1509 9.1809 10000 256.34 2319.68 180.72 1637.56 2848.3
## [5,] 16 1e-04 17.1158 17.3421 17.2080 10000 253.74 4326.44 179.45 3071.43 2820.8
## [6,] 32 1e-04 32.4194 33.9636 33.9090 9967 257.06 8260.15 180.92 5865.30 2859.7
## [7,] 1 1e+00 1.9803 2.0301 2.0290 10000 255.63 478.52 180.08 356.61 2841.1
## [8,] 2 1e+00 2.9434 3.0515 3.0500 10000 255.29 728.99 179.61 528.66 2837.1
## [9,] 4 1e+00 4.9435 5.0711 5.0751 10000 255.82 1187.97 180.07 890.20 2839.7
## [10,] 8 1e+00 8.9155 9.1458 9.1541 10000 256.03 2145.97 180.33 1607.78 2844.1
## [11,] 16 1e+00 16.6991 17.2916 17.2731 10000 256.49 4052.56 180.25 3010.02 2846.3
## [12,] 32 1e+00 32.7225 33.7846 33.2691 9964 256.08 7661.98 179.19 5863.46 2842.4
## [13,] 1 2e+00 1.9161 2.0211 1.9946 10000 255.98 470.24 176.30 337.80 2840.2
## [14,] 2 2e+00 2.9886 3.0199 2.9836 10000 255.91 721.92 176.16 526.47 2844.3
## [15,] 4 2e+00 4.9052 5.0275 4.9783 10000 256.19 1137.59 176.85 867.47 2848.8
## [16,] 8 2e+00 8.9506 9.0482 8.9707 10000 256.61 2042.35 176.85 1582.89 2850.5
## [17,] 16 2e+00 16.5316 17.1257 16.8396 10000 255.42 3720.44 175.64 2903.65 2835.8
## [18,] 32 2e+00 31.8944 33.4905 32.8890 9965 256.75 7074.87 176.20 5619.66 2852.3
## [19,] 1 4e+00 1.8816 2.0085 1.8253 10000 256.22 440.62 160.70 302.38 2845.5
## [20,] 2 4e+00 2.7680 2.9329 2.6728 10000 256.20 608.46 160.76 444.98 2846.7
## [21,] 4 4e+00 4.7211 4.8095 4.3736 10000 255.93 1007.97 160.53 757.86 2846.5
## [22,] 8 4e+00 8.4540 8.5647 7.7899 10000 255.96 1765.85 160.60 1357.74 2846.4
## [23,] 16 4e+00 15.5473 16.1004 14.6217 10000 255.49 3241.52 160.30 2492.29 2839.5
## [24,] 32 4e+00 30.0134 31.2622 28.1783 9972 257.05 6115.98 160.48 4816.58 2853.4
##      varlocalD VRlocalD
## [1,] 41629 1.9304
```

```
## [2,]      82656    2.8408
## [3,]     165929    4.7218
## [4,]     334375    8.4475
## [5,]     652945   15.8283
## [6,]    1300729   29.7413
## [7,]      39704    1.8888
## [8,]      79503    2.7847
## [9,]     155408    4.4825
## [10,]    313076    7.9939
## [11,]    622576   14.8864
## [12,]   1231872   28.5515
## [13,]     35758    1.8010
## [14,]     73344    2.6382
## [15,]    142657    4.1762
## [16,]    293555    7.5281
## [17,]    568685   13.7777
## [18,]   1135388   26.2181
## [19,]     29197    1.6516
## [20,]     57743    2.2875
## [21,]    120780    3.6934
## [22,]    244349    6.4497
## [23,]    477183   11.6941
## [24,]    931136   21.6652
```

```
sum2Cf    # fixed N(A)
```

```
##      mu scale varration   chatF   chatW Nsim  N varN      n      varn localD varlocalD
## [1,]  1 1e-04  0.58924  2.0292  2.0278 10000 256    0 180.54  106.380 2846.1    25755
## [2,]  2 1e-04  0.89261  3.0374  3.0393 10000 256    0 180.63  161.231 2846.6    40384
## [3,]  4 1e-04  1.47404  5.0449  5.0525 10000 256    0 180.70  266.359 2849.0    63596
## [4,]  8 1e-04  2.65284  9.0685  9.0925 10000 256    0 180.93  479.967 2850.2   104436
## [5,] 16 1e-04  4.98912 17.1378 17.0805 10000 256    0 179.92  897.636 2840.5   188417
## [6,] 32 1e-04  9.67104 33.2156 33.2628 10000 256    0 180.62 1746.787 2853.4   343799
## [7,]  1 1e+00  0.55110  2.0102  2.0092 10000 256    0 180.31   99.367 2846.8    23167
## [8,]  2 1e+00  0.77483  3.0217  3.0199 10000 256    0 180.21  139.632 2843.1    35139
## [9,]  4 1e+00  1.26138  5.0446  5.0453 10000 256    0 180.40  227.557 2846.8    50756
## [10,] 8 1e+00  2.17174  9.0731  9.0736 10000 256    0 180.22  391.394 2847.0    77807
## [11,] 16 1e+00  4.10780 17.1478 17.1136 10000 256    0 179.59  737.721 2839.0   128838
## [12,] 32 1e+00  7.99488 33.1655 33.0066 10000 256    0 179.83 1437.712 2847.1   219923
## [13,]  1 2e+00  0.48443  1.9249  1.9154 10000 256    0 179.05   86.737 2846.7    20602
## [14,]  2 2e+00  0.70217  2.8855  2.8755 10000 256    0 179.13  125.779 2847.9    31113
## [15,]  4 2e+00  1.12666  4.8686  4.8486 10000 256    0 178.92  201.577 2844.0    42535
## [16,]  8 2e+00  1.92840  8.8594  8.7881 10000 256    0 178.03  343.316 2847.9    58624
## [17,] 16 2e+00  3.52786 16.8071 16.6270 10000 256    0 177.70  626.892 2847.2    87609
## [18,] 32 2e+00  7.09988 32.8090 32.1553 10000 256    0 176.59 1253.801 2844.7   141560
## [19,]  1 4e+00  0.47417  1.8839  1.8539 10000 256    0 176.15   83.525 2844.5    17642
## [20,]  2 4e+00  0.64272  2.7892  2.7427 10000 256    0 175.91  113.063 2846.1    25688
## [21,]  4 4e+00  0.97093  4.6870  4.5955 10000 256    0 175.48  170.383 2846.9    33945
## [22,]  8 4e+00  1.73923  8.6250  8.3802 10000 256    0 173.81  302.298 2843.6    44001
## [23,] 16 4e+00  3.09755 16.3679 15.7848 10000 256    0 172.01  532.822 2848.1    54732
## [24,] 32 4e+00  6.45844 31.7910 29.9547 10000 256    0 167.96 1084.766 2841.6    76268
##      VRlocalD
## [1,]    1.5745
## [2,]    1.9005
## [3,]    2.4157
```

```
## [4,] 3.3229
## [5,] 5.2196
## [6,] 8.6300
## [7,] 1.5165
## [8,] 1.7855
## [9,] 2.1317
## [10,] 2.7346
## [11,] 3.8884
## [12,] 5.9024
## [13,] 1.4594
## [14,] 1.6931
## [15,] 1.9502
## [16,] 2.3061
## [17,] 2.9528
## [18,] 4.1609
## [19,] 1.3940
## [20,] 1.5730
## [21,] 1.7568
## [22,] 1.9832
## [23,] 2.2192
## [24,] 2.7067
```

```
sum2CCL # Poisson  $N(A)$ , model fitted by maximising conditional likelihood
```

##	mu	scale	n	nvalid	estimate	SE.estimate	RSE	trueD	RB	COV
## [1,]	1	1e-04	180.76	1000	2850.5	213.22	0.075200	2844.4	-0.0021367	0.8360
## [2,]	2	1e-04	179.36	1000	2826.3	212.06	0.075678	2844.4	0.0063846	0.7290
## [3,]	4	1e-04	178.85	1000	2824.1	211.90	0.076214	2844.4	0.0071628	0.5870
## [4,]	8	1e-04	179.54	1000	2840.1	212.21	0.076903	2844.4	0.0015156	0.4560
## [5,]	16	1e-04	181.53	1000	2881.9	213.72	0.077795	2844.4	-0.0131552	0.3790
## [6,]	32	1e-04	184.82	987	2965.3	217.53	0.081205	2844.4	-0.0424851	0.2766
##	chatF	varration	a	SEa	RBa	RSEa	COVa	pCVn		
## [1,]	1.9739	1.9227	0.063423	0.00055790	0.00161431	0.0088051	0.83100	0.98613		
## [2,]	2.9824	2.9929	0.063480	0.00055960	0.00071431	0.0088282	0.75900	0.98615		
## [3,]	4.9483	5.4891	0.063358	0.00057051	0.00264306	0.0090273	0.64600	0.98555		
## [4,]	8.8896	9.8519	0.063238	0.00058835	0.00452230	0.0093613	0.49700	0.98433		
## [5,]	16.7772	16.2780	0.063035	0.00062604	0.00773051	0.0101110	0.36300	0.98102		
## [6,]	32.7160	32.0965	0.062445	0.00071643	0.01701753	0.0121070	0.27052	0.97237		

```
sum2CfCL # fixed  $N(A)$ , model fitted by maximising conditional likelihood
```

##	mu	scale	n	nvalid	estimate	SE.estimate	RSE	trueD	RB	COV	chatF
## [1,]	1	2	179.14	1000	2822.8	212.28	0.075306	2844.4	0.0075961	0.996	1.9135
## [2,]	2	2	178.23	1000	2808.2	211.69	0.075548	2844.4	0.0127534	0.974	2.8475
## [3,]	4	2	178.35	1000	2814.1	212.08	0.075618	2844.4	0.0106765	0.922	4.7972
## [4,]	8	2	179.01	1000	2831.7	213.04	0.075642	2844.4	0.0044680	0.828	8.8168
## [5,]	16	2	177.55	1000	2816.7	212.84	0.076367	2844.4	0.0097519	0.689	16.6758
## [6,]	32	2	175.51	1000	2815.5	214.98	0.077974	2844.4	0.0101690	0.529	32.1971
##	varration	a	SEa	RBa	RSEa	COVa	pCVn				
## [1,]	0.49980	0.063469	0.00055324	0.00088721	0.0087248	0.858	0.98643				
## [2,]	0.77999	0.063482	0.00055583	0.00068254	0.0087684	0.756	0.98631				
## [3,]	1.18750	0.063405	0.00056545	0.00190234	0.0089411	0.614	0.98562				
## [4,]	1.91385	0.063260	0.00057844	0.00417795	0.0091875	0.482	0.98453				
## [5,]	3.59820	0.063125	0.00060059	0.00630514	0.0096317	0.344	0.98222				
## [6,]	6.79155	0.062534	0.00067588	0.01560910	0.0112573	0.286	0.97376				

## Finally...

```
# appears to fail when knitting to pdf, so suppress in that case  
# unlink(tmpfolder, recursive = TRUE)
```