Bayesian multi-observation latent Gaussian models with inlabru

5th ISI Regional Statistics Conference, Valletta, Malta, 3-5 June 2026

Finn Lindgren, Finn.Lindgren@ed.ac.uk

School of Mathematics and Maxwell Institute for Mathematical Sciences, The University of Edinburgh

2026-06-05

Bayesian latent Gaussian process models

General latent Gaussian hierarchical model structure

$$
\begin{aligned}
\boldsymbol{\theta} &\sim p(\boldsymbol{\theta}) \\
\boldsymbol{x}|\boldsymbol{\theta} &\sim \mathrm{N}(\boldsymbol{\mu}_x(\boldsymbol{\theta}), \boldsymbol{Q}_x(\boldsymbol{\theta})^{-1}) \\
\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta} &\sim p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{\theta})
\end{aligned}
$$

Generalised additive models (GAMs) with Gaussian random fields (GRFs):

$$
\boldsymbol{x} = (\beta, \boldsymbol{u}_1, \dots, \boldsymbol{u}_K)
$$

$$
g(\mathrm{E}[y_i|\boldsymbol{x}, \boldsymbol{\theta}]) = \boldsymbol{\eta} = \boldsymbol{X}\beta + \sum_k f_k(z_{ik}; \boldsymbol{u}_k)
$$

and e.g. $y_i|\boldsymbol{x}, \boldsymbol{\theta} \sim \mathrm{N}(\boldsymbol{\eta}, \sigma_y^2)$ or $y_i|\boldsymbol{x}, \boldsymbol{\theta} \sim \mathrm{Po}(\exp(\eta_i))$

We want to estimate the parameters of the GRFs, $\boldsymbol{\theta}$, the GRF processes values $f_k(\cdot)$ at observed and unobserved locations, and quantify the uncertainty in these estimates.

The Matérn-Whittle-Markov GRF/SPDE/GMRF connection; mapping latent variables to functions

Each $f_k(\cdot)$ is a function of space, time, or a covariate, and is approximated by

$$f_k(z_{ik}; \boldsymbol{u}_k) = \sum_j \psi_{kj}(z_{ik}) u_{kj},$$

where $\psi_{kj}(z_{ik})$ are basis functions, e.g. finite element basis functions. Matérn fields are solutions to the spatial SPDE

$$(\kappa^2 - \nabla \cdot \nabla)^{\alpha/2}(\tau u(\boldsymbol{s})) = \kappa^\gamma \mathrm{d}W(\boldsymbol{s})$$

$$u(\boldsymbol{s}) \approx \sum_j \psi_j(\boldsymbol{s}) u_j, \boldsymbol{u} \sim \mathrm{N}(\boldsymbol{0}, \boldsymbol{Q}_u^{-1})$$

where $\boldsymbol{Q}$ is the precision matrix of the GRF/SPDE/GRMF representation.

When α is an integer, FEM yields a sparse matrix $\boldsymbol{Q}_u$, and $u(\boldsymbol{s})$ is a Markov random field (Lindgren et al, 2011).

For non-integers, $u(\boldsymbol{s})$ can be closely approximated by a sum of a few Markov processes (Bolin and Kirchner, 2020).

Parameter estimation and spatial prediction

$$p(\boldsymbol{\theta}|\boldsymbol{y}) \propto p(\boldsymbol{\theta})p(\boldsymbol{y}|\boldsymbol{\theta}) \approx \left. \frac{p(\boldsymbol{\theta})p(\boldsymbol{x}|\boldsymbol{\theta})p(\boldsymbol{y}|\boldsymbol{\theta},\boldsymbol{x})}{p_G(\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{y})} \right|_{\boldsymbol{x}=\boldsymbol{x}^*}$$

where $p_G(\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{y})$ is the Gaussian approximation to the conditional posterior density.

The INLA software uses numerical integration over $\boldsymbol{\theta}$ together with variational Bayes corrections $p_{GG}()$ of the Gaussian approximations to obtain the posterior marginal densities of $\boldsymbol{x}$:

$$\begin{aligned} p(\boldsymbol{x}|\boldsymbol{y}) &= \int p(\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{y})p(\boldsymbol{\theta}|\boldsymbol{y})\,\mathrm{d}\boldsymbol{\theta} \\ &\approx \sum_j p_{GG}(\boldsymbol{x}|\boldsymbol{\theta}^{(j)},\boldsymbol{y})p(\boldsymbol{\theta}^{(j)}|\boldsymbol{y})w_j \end{aligned}$$

The inner core of the Integrated Nested Laplace method

- Latent Gaussian model structure (Bayesian GAMs with Gaussian process components)

$$\boldsymbol{\theta} \sim p(\boldsymbol{\theta}) \quad \text{(precision parameters)} \qquad \eta(\boldsymbol{s}, t) = \sum_{k=1}^{m} \psi_k(\boldsymbol{s}, t) u_k \quad \text{(predictor)}$$

$$\boldsymbol{u}|\boldsymbol{\theta} \sim \mathrm{N}[\boldsymbol{\mu}_u, \boldsymbol{Q}_u^{-1}] \quad \text{(latent field)} \qquad \boldsymbol{y}|\boldsymbol{\theta}, \boldsymbol{u} \sim p(\boldsymbol{y}|\boldsymbol{\theta}, \eta) \quad \text{(observations)}$$

- Conditional log-posterior mode ($\boldsymbol{\mu}_{u|y}$) and Hessian ($\boldsymbol{Q}_{u|y}$), for each $\boldsymbol{\theta}$, by iteration:

$$\boldsymbol{g}_y^* = -\left.\frac{\mathrm{d}}{\mathrm{d}\boldsymbol{u}} \log p(\boldsymbol{y}|\boldsymbol{\theta}, \eta)\right|_{\boldsymbol{u}=\boldsymbol{u}^*}$$

$$\boldsymbol{H}_y^* = -\left.\frac{\mathrm{d}^2}{\mathrm{d}\boldsymbol{u}\mathrm{d}\boldsymbol{u}^\top} \log p(\boldsymbol{y}|\boldsymbol{\theta}, \eta)\right|_{\boldsymbol{u}=\boldsymbol{u}^*}$$

$$\boldsymbol{Q}_{u|y} = \boldsymbol{Q}_u + \boldsymbol{H}_y^*$$

$$\boldsymbol{Q}_{u|y}(\boldsymbol{\mu}_{u|y} - \boldsymbol{\mu}_u) = \boldsymbol{Q}_u(\boldsymbol{u}^* - \boldsymbol{\mu}_u) - \boldsymbol{g}_y^*$$

General observation models; rarely direct observations

- Point-referenced data; additive noise, counts, presence-absence, etc.
- Aggregated data; spatial averages/totals, counts, presence-absence, etc.
- Point process data. Poisson process log-likelihood function:

$$-\int \lambda(\boldsymbol{s})\,\mathrm{d}\boldsymbol{s} + \sum_i \log[\lambda(\boldsymbol{y}_i)] \approx -\sum_j w_j \exp[\eta(\boldsymbol{s}_j)] + \sum_i \eta(\boldsymbol{y}_i)$$

  where $\{(\boldsymbol{s}_j, w_j)\}$ is a numerical integration scheme over the sampled region of space (Simpson et al, 2016, Biometrika)

- Semi-parametric densities, (animal) movement kernels, Hawkes processes, etc

Modern data analysis problems may involve multiple types of observations in a single model; each type may have a specialised method, but we need a general system for blending the information, which is provided by the general Bayesian framework.

Probabilistic latent Gaussian model specification

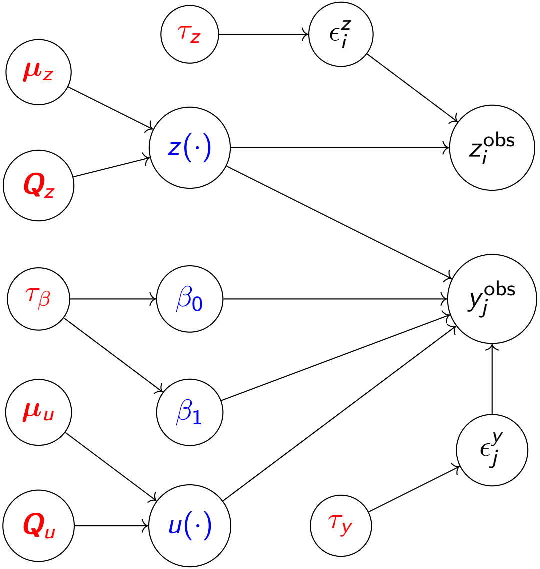
In plain INLA, the syntax mimics other R modelling packages such as `mgcv` and `lme4`, with formulae defining the predictor structure.

In `inlabru`, the latent components and observation models are defined separately, and combined into a full model definition. This allows a wide range of easy-to-specify extensions, model definitions, and flexible data wrangling.

- List of latent components, each with a Gaussian process definition, and a mapping between locations/covariate values, the latent variables, and the resulting "effect".
- List of observation models, each with a likelihood function linking a predictor expression to the observation distribution, and a mapping between the latent components/effects and the predictor.

The `inlabru` model specification is a type of probabilistic programming, and is increasingly using automatic differentiation techniques to compute Jacobians.

Basic joint model example; misaligned covariate/response measurements



$$(\tau_z, \boldsymbol{\mu}_z, \boldsymbol{Q}_z) \sim p(\tau_z)p(\boldsymbol{\mu}_z)p(\boldsymbol{Q}_z)$$

$$z(\cdot)\text{-coefficients} \sim \mathrm{N}(\boldsymbol{\mu}_z, \boldsymbol{Q}_z{}^{-1})$$

$$\epsilon_i^z \sim \mathrm{N}(0, \tau_z{}^{-1})$$

$$z_i^{\text{obs}} = z(\boldsymbol{s}_i) + \epsilon_i^z,\ \boldsymbol{s}_i \in \mathcal{S}_z$$

$$y_j^{\text{obs}} = \beta_0 + \beta_1 z(\boldsymbol{s}_j) + u(\boldsymbol{s}_j) + \epsilon_j^y,\ \boldsymbol{s}_j \in \mathcal{S}_y$$

$$\epsilon_i^y \sim \mathrm{N}(0, \tau_y{}^{-1})$$

$$u(\cdot)\text{-coefficients} \sim \mathrm{N}(\boldsymbol{\mu}_u, \boldsymbol{Q}_u{}^{-1})$$

$$\beta_0, \beta_1 \sim \mathrm{N}(0, \tau_\beta{}^{-1})$$

$$\tau_y \sim p(\tau_y),\ \tau_\beta \text{ fixed}$$

$$(\boldsymbol{\mu}_u, \boldsymbol{Q}_u) \sim p(\boldsymbol{\mu}_u)p(\boldsymbol{Q}_u)$$

Basic joint models and model mis-specification handling

- Joint models allow uncertainty propagation between sub-models:

$$
\begin{aligned}
z_i^{\text{obs}} &= z(\boldsymbol{s}_i) + \epsilon_i^z, \quad z(\cdot) \sim \text{GRF} \\
y_j^{\text{obs}} &= \beta_0 + \beta_1 z(\boldsymbol{s}_j) + u(\boldsymbol{s}_j) + \epsilon_j^y, \quad \beta_0, \beta_1 \sim \text{N},\ u(\cdot) \sim \text{GRF}
\end{aligned}
$$

  Oops, $\beta_1 z(\boldsymbol{s})$ is nonlinear in the latent variables! We'll deal with that in a moment.

- Two-step approaches (e.g. fit covariate model, then fit main model) can avoid improper feedback problems, but need to propagate uncertainty:

$$
\begin{aligned}
z_i^{\text{obs}} &= z(\boldsymbol{s}_i) + \epsilon_i^z \\
z^{\text{post}} &\sim \text{N}(\boldsymbol{\mu}_{z|z^{\text{obs}}}, \boldsymbol{Q}_{z|z^{\text{obs}}}^{-1}) \quad \text{(convenient posterior approximation)} \\
y_j^{\text{obs}} &= \beta_0 + \beta_1 z^{\text{post}}(\boldsymbol{s}_j) + u(\boldsymbol{s}_j) + \epsilon_j^y
\end{aligned}
$$

- Since each observation model links to the same latent representations, misalignment of the covariate and response measurements is handled by the modelling principle.

Linearisation iteration

- For nonlinear predictor models, inlabru linearises $\eta(\{\boldsymbol{u}_k\})$ around a linearisation point $\{\boldsymbol{u}_k^*\}$
- INLA only sees the linearised model given by

$$
\begin{aligned}
\overline{\boldsymbol{\eta}}(\{\boldsymbol{u}_k\}) &= \boldsymbol{\eta}(\{\boldsymbol{u}_k^*\}, \{\boldsymbol{f}_k^* = \boldsymbol{f}_k(\boldsymbol{u}_k^*)\}) + \sum_k \frac{\mathrm{d}\boldsymbol{\eta}}{\mathrm{d}\boldsymbol{u}_k}(\boldsymbol{u}_k - \boldsymbol{u}_k^*) \\
&= \left[\boldsymbol{\eta}(\{\boldsymbol{u}_k^*\}, \{\boldsymbol{f}_k^* = \boldsymbol{f}_k(\boldsymbol{u}_k^*)\}) - \sum_k \frac{\mathrm{d}\boldsymbol{\eta}}{\mathrm{d}\boldsymbol{u}_k}\boldsymbol{u}_k^*\right] + \sum_k \frac{\mathrm{d}\boldsymbol{\eta}}{\mathrm{d}\boldsymbol{u}_k}\boldsymbol{u}_k
\end{aligned}
$$

- After each INLA run, the linearisation point is updated to approach the new conditional posterior mode, and the process is repeated until convergence.

Non-linear predictor linearisation

Additive predictors have simple Jacobians:

$$\boldsymbol{\eta} = \boldsymbol{X}\beta + \sum_k \boldsymbol{f}_k(z_{ik}; \boldsymbol{u}_k)$$

$$\frac{\mathrm{d}\boldsymbol{\eta}}{\mathrm{d}\boldsymbol{\beta}} = \boldsymbol{X}, \qquad \frac{\mathrm{d}\boldsymbol{\eta}}{\mathrm{d}\boldsymbol{u}_k} = \frac{\mathrm{d}\boldsymbol{f}_k}{\mathrm{d}\boldsymbol{u}_{\boldsymbol{k}}}$$

inlabru allows more general expressions:

$$\boldsymbol{\eta} = \boldsymbol{\eta}(\{\boldsymbol{u}_k\}, \{\boldsymbol{f}_k = \boldsymbol{f}_k(\boldsymbol{u}_k)\})$$

$$\frac{\mathrm{d}\boldsymbol{\eta}}{\mathrm{d}\boldsymbol{u}_k} = \frac{\partial\boldsymbol{\eta}}{\partial\boldsymbol{u}_k} + \frac{\partial\boldsymbol{\eta}}{\partial\boldsymbol{f}_k}\frac{\mathrm{d}\boldsymbol{f}_k}{\mathrm{d}\boldsymbol{u}_k}$$

The internals will soon support the further extension

$$\boldsymbol{\eta} = h\left[\widetilde{\boldsymbol{\eta}}(\{\boldsymbol{u}_k\}, \{\boldsymbol{f}_i = \boldsymbol{f}_i(\{\boldsymbol{u}_j\})\})\right]$$

$$\frac{\mathrm{d}\boldsymbol{\eta}}{\mathrm{d}\boldsymbol{u}_k} = \frac{\partial h}{\partial\widetilde{\boldsymbol{\eta}}}\left[\frac{\partial\widetilde{\boldsymbol{\eta}}}{\partial\boldsymbol{u}_k} + \sum_i \frac{\partial\widetilde{\boldsymbol{\eta}}}{\partial\boldsymbol{f}_i}\frac{\mathrm{d}\boldsymbol{f}_i}{\mathrm{d}\boldsymbol{u}_k}\right]$$

Linearisation computation implementation

The "pandemic" reimplementation introduced the mapper composition system:

- explicit Jacobians and the chain rule within each component: $\frac{\mathrm{d}\boldsymbol{f}_k}{\mathrm{d}\boldsymbol{u}_k}$
- Explicit Jacobians for additive predictors
- Numerical derivatives for general predictor expressions, with the chain rule connecting to the components
- Speedups for rowwise data/effect calculations

New "fullchain" extension (to appear in version 2.15.0):

- Allow a post-transformation step, $h(\widetilde{\boldsymbol{\eta}})$, with a Jacobian $\frac{\partial h}{\partial \widetilde{\boldsymbol{\eta}}}$
- This allows e.g. non-linear aggregation of additive models to use the chain rule to compute the Jacobian.
- Improved rowwise derivatives

Beyond generalised additive models

- Spatially consistent modelling of aggregated counts:

$$\lambda(\boldsymbol{s}) = \exp\left[\beta_0 + \beta_1 z(\boldsymbol{s}) + u(\boldsymbol{s})\right] \quad \text{(point process intensity)}$$

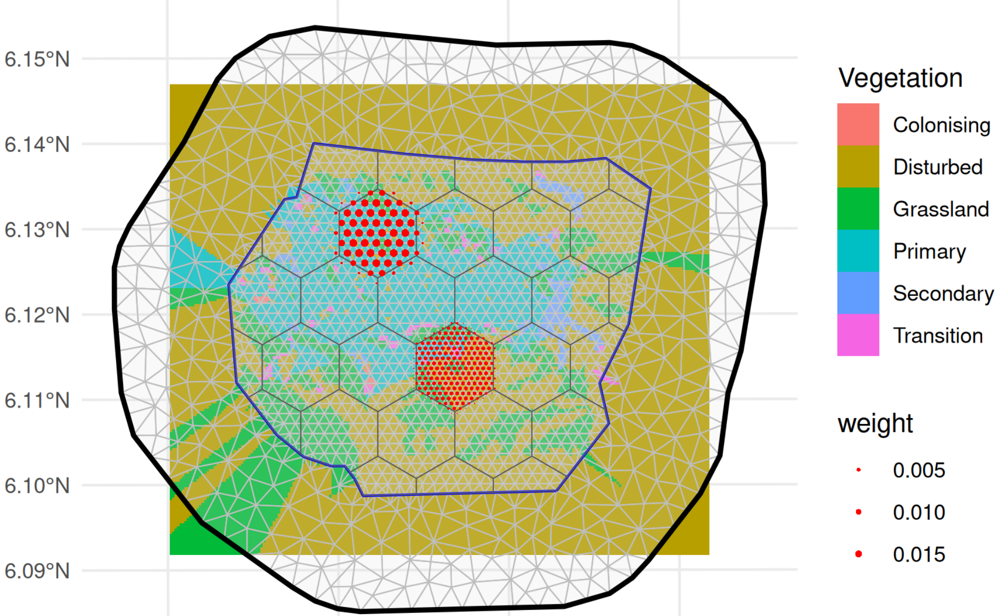
$$y_j^{\text{obs}} \sim \text{Poisson}\left(\Lambda_j = \int_{\mathcal{B}_j} \lambda(\boldsymbol{s})\, \mathrm{d}\boldsymbol{s}\right)$$

- Although $\lambda(\boldsymbol{s})$ is log-additive, the response expectation Λ_j is not:

$$\begin{aligned}\log \Lambda_j &= \log\left[\int_{\mathcal{B}_j} \exp\left[\beta_0 + \beta_1 z(\boldsymbol{s}) + u(\boldsymbol{s})\right] \mathrm{d}\boldsymbol{s}\right] \\ &\approx \log\left[\sum_{k=1}^{m_j} w_{j,k} \exp\left[\beta_0 + \beta_1 z(\boldsymbol{s}_{j,k}) + u(\boldsymbol{s}_{j,k})\right]\right]\end{aligned}$$

  where $\{\boldsymbol{s}_{j,k}, w_{j,k}\}$ is a numerical integration scheme over $\mathcal{B}_j$.
- Note: The integration scheme needs to be fine enough to capture the variation in both $z(\cdot)$ and $u(\cdot)$

Field representation, covariate resolution, and integration schemes



Non-linear predictors in point process models

The original motivation for the `inlabru` package was ecological transect distance sampling, requiring a model for imperfect detections:

$$\lambda_{\text{apparent}}(\boldsymbol{s};\boldsymbol{u},\boldsymbol{v}) = \lambda(\boldsymbol{s};\boldsymbol{u})h(\boldsymbol{s};\boldsymbol{v}),$$

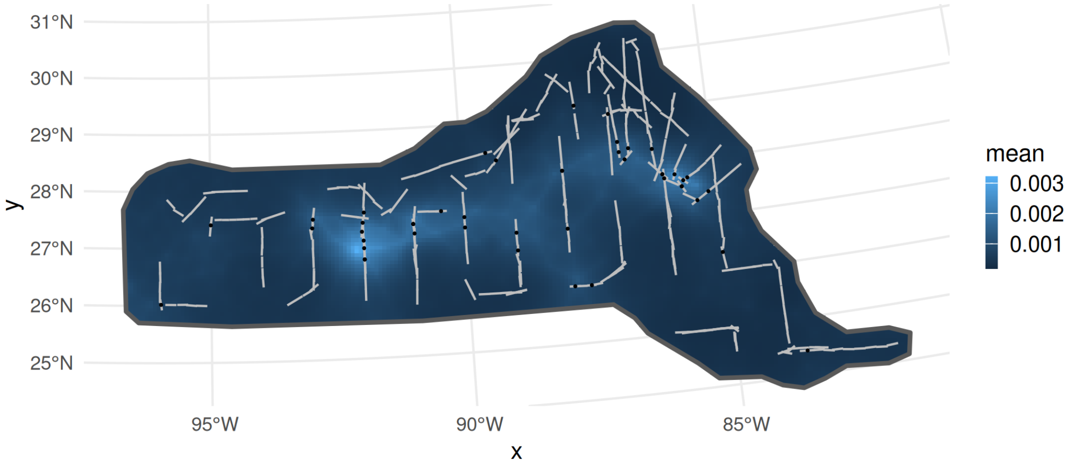
where $h(\boldsymbol{s};\boldsymbol{v})$ is the detection probability for a point located at $\boldsymbol{s}$, and $\boldsymbol{v}$ is a vector of parameters for the detection function, such as the *hazard rate* model:

$$h(\boldsymbol{s};\boldsymbol{v}) = 1 - \exp\left[-\left(\frac{\text{distance}[\boldsymbol{s},\text{transect}]}{\phi_1(v_1)}\right)^{-\phi_2(v_2)}\right] \quad \text{(quantile transformations } \phi_1 \text{ and } \phi_2\text{)}$$

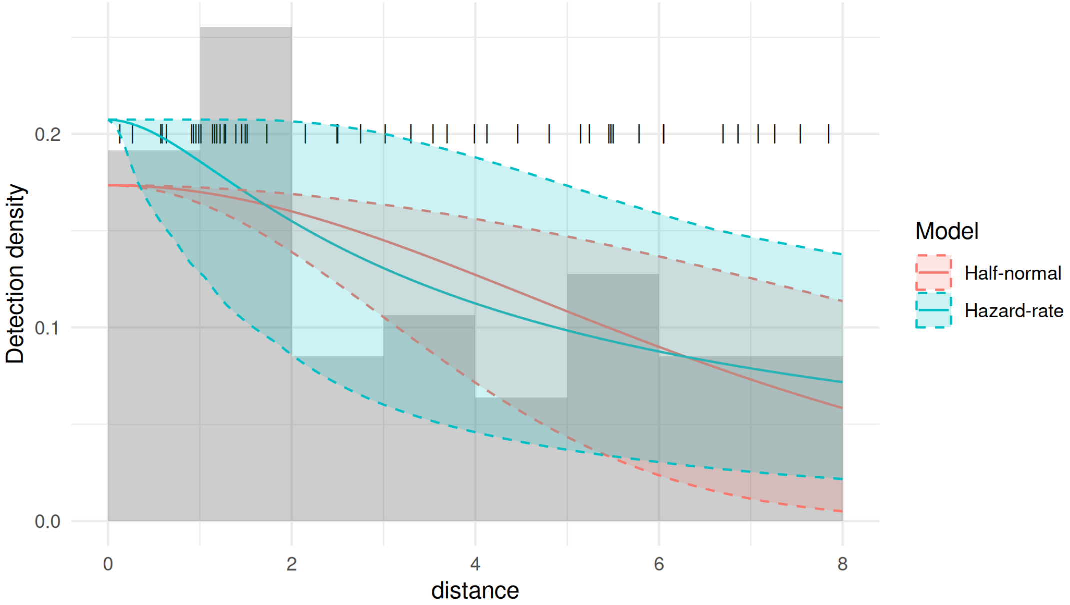
The `inlabru` package solves this by iterating the INLA method on a linearisation w.r.t. $\boldsymbol{u}$ and $\boldsymbol{v}$ of the non-linear predictor

$$\eta(\boldsymbol{s};\boldsymbol{u},\boldsymbol{v}) = \log[\lambda(\boldsymbol{s};\boldsymbol{u})] + \log[h(\boldsymbol{s};\boldsymbol{v})].$$

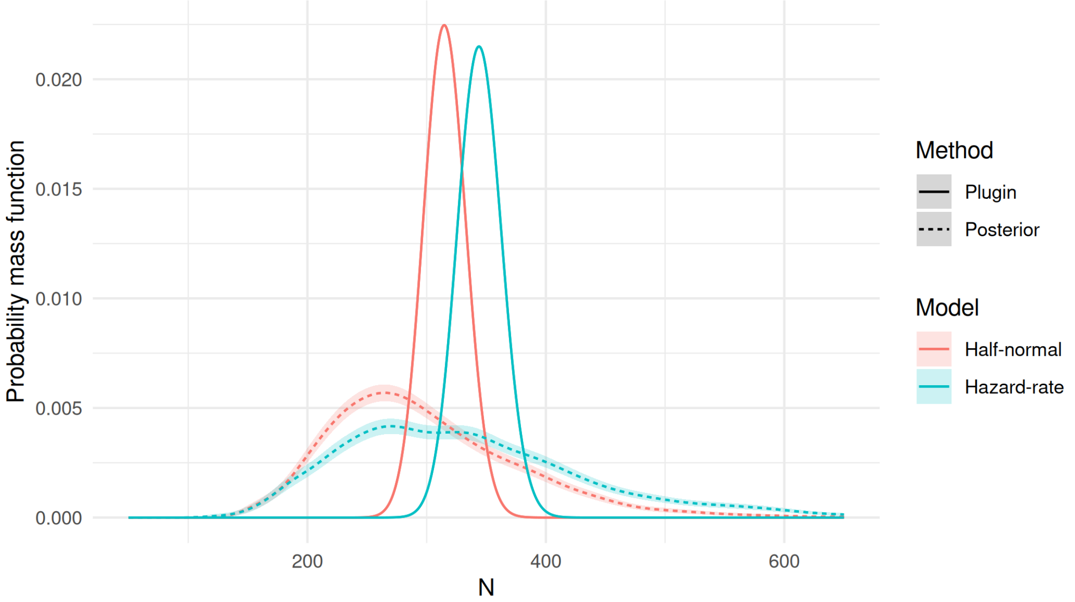
Dolphin group detection; estimated density field



Dolphin group detection; estimated detection density



Dolphin group detection; estimated total count



Joint inference with observation-coupling

- The detectability of an animal group may depend on its size
- For illustration, assume groups appear independently, and have conditionally independent sizes
- The size pdf is $p_z(z|\boldsymbol{s}, w)$, where w is a parameter for the size distribution
- The detection function is $h(\boldsymbol{s}, z; \boldsymbol{v})$
- New apparent intensity function for the joint point process:

$$\lambda_{\text{apparent}}(\boldsymbol{s}, z; \boldsymbol{u}, \boldsymbol{v}, w) = \lambda(\boldsymbol{s}; \boldsymbol{u}) p_z(z|\boldsymbol{s}, w) h(\boldsymbol{s}, z; \boldsymbol{v})$$

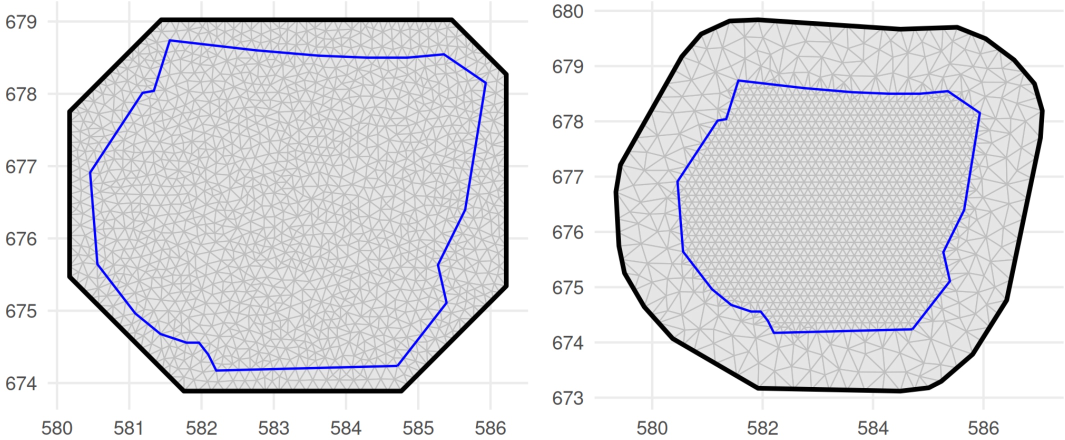
- Note: The likelihood depends on $p_z(z|\boldsymbol{s}, w)$ where no group has been observed, since $h()$ couples the point process for the groups and the size distribution.
- See Houldcroft et al (Ecography, 2024) for a practical illustration

Note: This model construction greatly simplifies how the animals form groups.

Joint inference with observation-coupling; code example

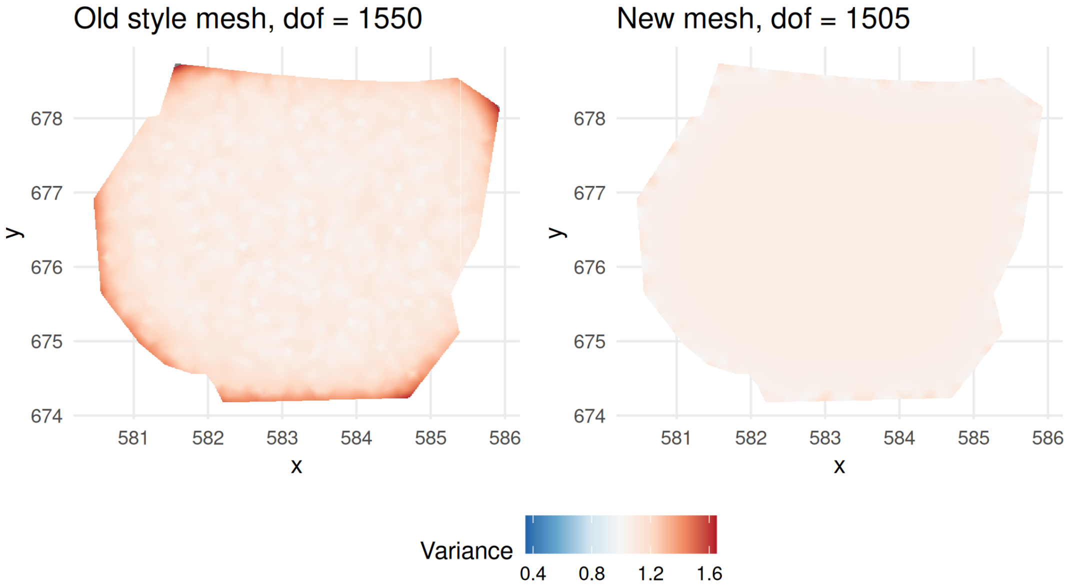
```
fit <- bru(
  components = ~ 0 + Intercept(1) + field(geometry, model = matern) +
    sigma(1, prec.linear = 1, marginal = bm_marginal(qexp, pexp, dexp, rate = 1 /
    shape(1, prec.linear = 1, marginal = bm_marginal(qgamma, pgamma, dgamma, shape
    size_int(1) + size_field(geometry, model = matern) + [...],
  bru_obs(
    geometry + distance + size ~
      Intercept + field + log_hr(distance, size, sigma, shape, [...]) + log(2)
      dgeom(size - 1L, prob = plogis(size_int + size_field), log = TRUE) -
        pgeom(max_size - 1L, prob = plogis(size_int + size_field), log.p = TRUE),
    data = mexdolphin_sf$points,
    domain = list(
      geometry = mesh,
      distance = fm_mesh_1d(seq(0, 8, length.out = 30)),
      size = seq_len(max_size)
    ),
    samplers = samplers,
    family = "cp"
  )
)
```

“Don’t use a non-uniform FEM meshes just because it’s allowed”



Plain `fm_mesh_2d()` vs `fm_mesh_2d(loc = fm_hexagon_lattice(region), boundary = list(region, fm_nonconvex_hull(region, ...))`

Variances



Old style mesh, dof = 1550
New mesh, dof = 1505
678
677
676
675
674
y
581
582
583
584
585
586
x
Variance
0.4
0.8
1.2
1.6

Summary, challenges, and software FAQs

- The Bayesian framework and INLA/inlabru provides a flexible system for joint modelling of misaligned data, with proper uncertainty propagation.
- Build models of the underlying phenomena, with each observation type having its own mapping, instead of trying to "align" incompatible data supports.
- How to automate uncertainty propagation in multi-step approaches, e.g. for aggregated data problems (work with Man Ho Suen and Stephen Jun Villejo)
- Multi-predictor observation models, e.g. location/scale (in progress)
- Transformation models (in progress more slowly)
- Always check the inlabru NEWS page for bug fixes and new features
- Q: Can I do X? A: "Probably, but the developers have limited time/funding to help".
- Q: Should I do X? A: "Maybe, but not without careful consideration of the model and data implications, often unrelated to the software itself".

References (subset)

- The SPDE approach for Gaussian and non-Gaussian fields: 10 years and still running (Lindgren et al, 2022, Spatial Statistics) https://doi.org/10.1016/j.spasta.2022.100599
- Going off grid: computationally efficient inference for log-Gaussian Cox processes (Simpson et al, 2016, Biometrika) https://doi.org/10.1093/biomet/asv064
- A diffusion-based spatio-temporal extension of Gaussian Matérn fields (Lindgren et al, 2024, SORT) https://doi.org/10.57645/20.8080.02.13
- A data fusion model for meteorological data using the INLA-SPDE method (Stephen Jun Villejo et al, 2025, JRSS-C)
- Coherent Disaggregation and Uncertainty Quantification for Spatially Misaligned Data (Man Ho Suen et al, 2026, Environmetrics) https://arxiv.org/abs/2502.10584
- Documentation and examples: `R-INLA` (https://www.r-inla.org/), `inlabru`, `fmesher` (https://inlabru-org.github.io/inlabru/ and `.../fmesher/`)