Hierarchical spatial modelling for applied population and community ecology

Jeffrey W. Doser, Marc Kéry, Gesa von Hirschheydt

24-27 June 2024

Modelling big spatial data

Jeffrey W. Doser

24-27 June 2024

Bayesian spatial linear model

$$y(\boldsymbol{s}_j) \sim \mathrm{Normal}(\mu(\boldsymbol{s}_j), \tau^2)$$
$$\mu(\boldsymbol{s}_j) = \beta_0 + \beta_1 \cdot x_1(\boldsymbol{s}_j) + \mathrm{w}(\boldsymbol{s}_j)$$
$$\mathbf{w} \sim \mathrm{Multivariate\ Normal}(\mathbf{0}, \boldsymbol{C}(d, \sigma^2, \phi))$$

- Recall our use of Gaussian Processes (GPs) to model the spatial random effects
- Values of **w** are determined by the covariance matrix **C**.
- **C** is a J x J spatial covariance matrix (J is the number of spatial locations)

Covariance matrix with J = 5

$$\begin{bmatrix} k_{1,1} & k_{1,2} & k_{1,3} & k_{1,4} & k_{1,5} \\ k_{2,1} & k_{2,2} & k_{2,3} & k_{2,4} & k_{2,5} \\ k_{3,1} & k_{3,2} & k_{3,3} & k_{3,4} & k_{3,5} \\ k_{4,1} & k_{4,2} & k_{4,3} & k_{4,4} & k_{4,5} \\ k_{5,1} & k_{5,2} & k_{5,3} & k_{5,4} & k_{5,5} \end{bmatrix}$$

Covariance
between site 5 and
site 3

Covariance matrix with J = 5

$$\begin{bmatrix} k_{1,1} & k_{1,2} & k_{1,3} & k_{1,4} & k_{1,5} \\ k_{2,1} & k_{2,2} & k_{2,3} & k_{2,4} & k_{2,5} \\ k_{3,1} & k_{3,2} & k_{3,3} & k_{3,4} & k_{3,5} \\ k_{4,1} & k_{4,2} & k_{4,3} & k_{4,4} & k_{4,5} \\ k_{5,1} & k_{5,2} & k_{5,3} & k_{5,4} & k_{5,5} \end{bmatrix}$$

What happens when J = 200? 1,000? 100,000?

Covariance between site 5 and site 3

Gaussian process

- Flexible approach to account for spatial autocorrelation.

Gaussian process

- Flexible approach to account for spatial autocorrelation.
- But... becomes extremely slow as the number of sites increases.
- Not practical for data sets with hundreds of data points, let alone thousands.
- Computational bottleneck: dealing with a large, *dense* J x J matrix.
- The "big n" problem.

The "big n" problem underneath the hood

The "big n" problem underneath the hood

- Estimating the parameters in our model requires calculating:

$$p(\mathbf{w}) \propto -\frac{1}{2}\log(\boxed{\det(\boldsymbol{C})}) - \frac{1}{2}\mathbf{w}^\top \boxed{\boldsymbol{C}^{-1}}\mathbf{w}$$

The "big n" problem underneath the hood

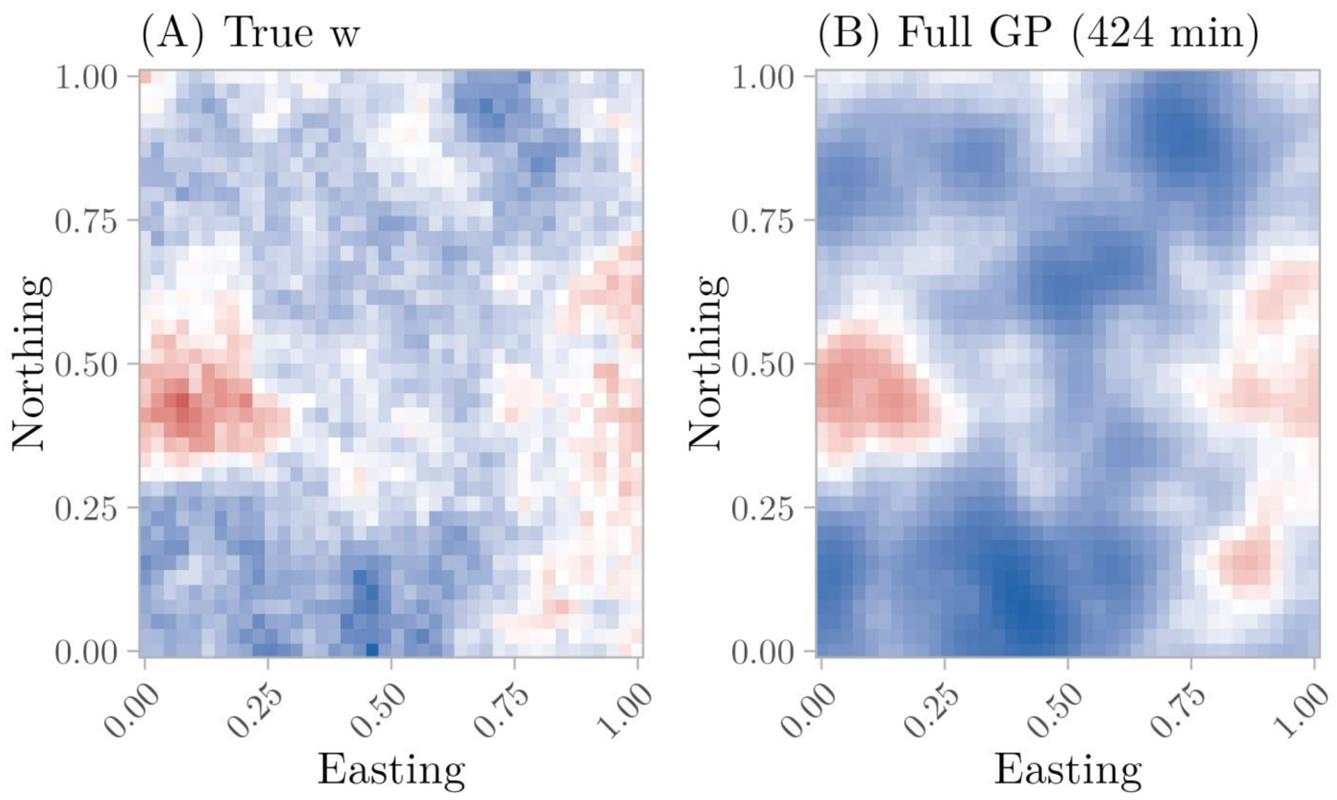
- Estimating the parameters in our model requires calculating:

$$p(\mathbf{w}) \propto -\frac{1}{2}\log(\boxed{\det(\boldsymbol{C})}) - \frac{1}{2}\mathbf{w}^\top \boxed{\boldsymbol{C}^{-1}}\mathbf{w}$$

- Storage requirements: $O(J^2)$ flops
- **C** is a dense matrix (i.e., there are no zeros)
- Computational complexity: $O(J^3)$ flops
- Computationally infeasible when J is even moderately large (e.g., 500)

Simulation comparison

- *J = 1600,* fit a single-species spatial occupancy model



Solutions to the "big n" problem

A Case Study Competition Among Methods for Analyzing Large Spatial Data

Matthew J. HEATON, Abhirup DATTA, Andrew O. FINLEY, Reinhard FURRER, Joseph GUINNESS, Rajarshi GUHANIYOGI, Florian GERBER, Robert B. GRAMACY, Dorit HAMMERLING, Matthias KATZFUSS, Finn LINDGREN, Douglas W. NYCHKA, Furong SUN, and Andrew ZAMMIT-MANGION

Solutions to the "big n" problem

Low-rank methods: reduces the dimensionality of the covariance matrix (e.g., analogous to how a PCA takes a set of variables and reduces it to a smaller set)

Solutions to the "big n" problem

Low-rank methods: reduces the dimensionality of the covariance matrix (e.g., analogous to how a PCA takes a set of variables and reduces it to a smaller set)

Sparse methods: strategically replace a lot of non-zero values in the covariance (or precision) matrix with 0s.

Solutions to the "big n" problem

Low-rank methods: reduces the dimensionality of the covariance matrix (e.g., analogous to how a PCA takes a set of variables and reduces it to a smaller set)

Sparse methods: strategically replace a lot of non-zero values in the covariance (or precision) matrix with 0s.

Algorithmic methods: focus more on improving algorithms to fit the model as opposed to building the model in a new way.

Solutions to the "big n" problem

Low-rank methods: reduces the dimensionality of the covariance matrix (e.g., analogous to how a PCA takes a set of variables and reduces it to a smaller set)

Sparse methods: strategically replace a lot of non-zero values in the covariance (or precision) matrix with 0s.

Algorithmic methods: focus more on improving algorithms to fit the model as opposed to building the model in a new way.

Nearest Neighbor Gaussian Processes (NNGPs)

- Datta et al. 2016, Finley et al. 2019

Nearest Neighbor Gaussian Processes (NNGPs)

- Datta et al. 2016, Finley et al. 2019
- Same parameters and interpretation as the full GP!

Nearest Neighbor Gaussian Processes (NNGPs)

- Datta et al. 2016, Finley et al. 2019
- Same parameters and interpretation as the full GP!
- Conceptually:
    1. Order the spatial locations (e.g., along the x-axis).

Nearest Neighbor Gaussian Processes (NNGPs)

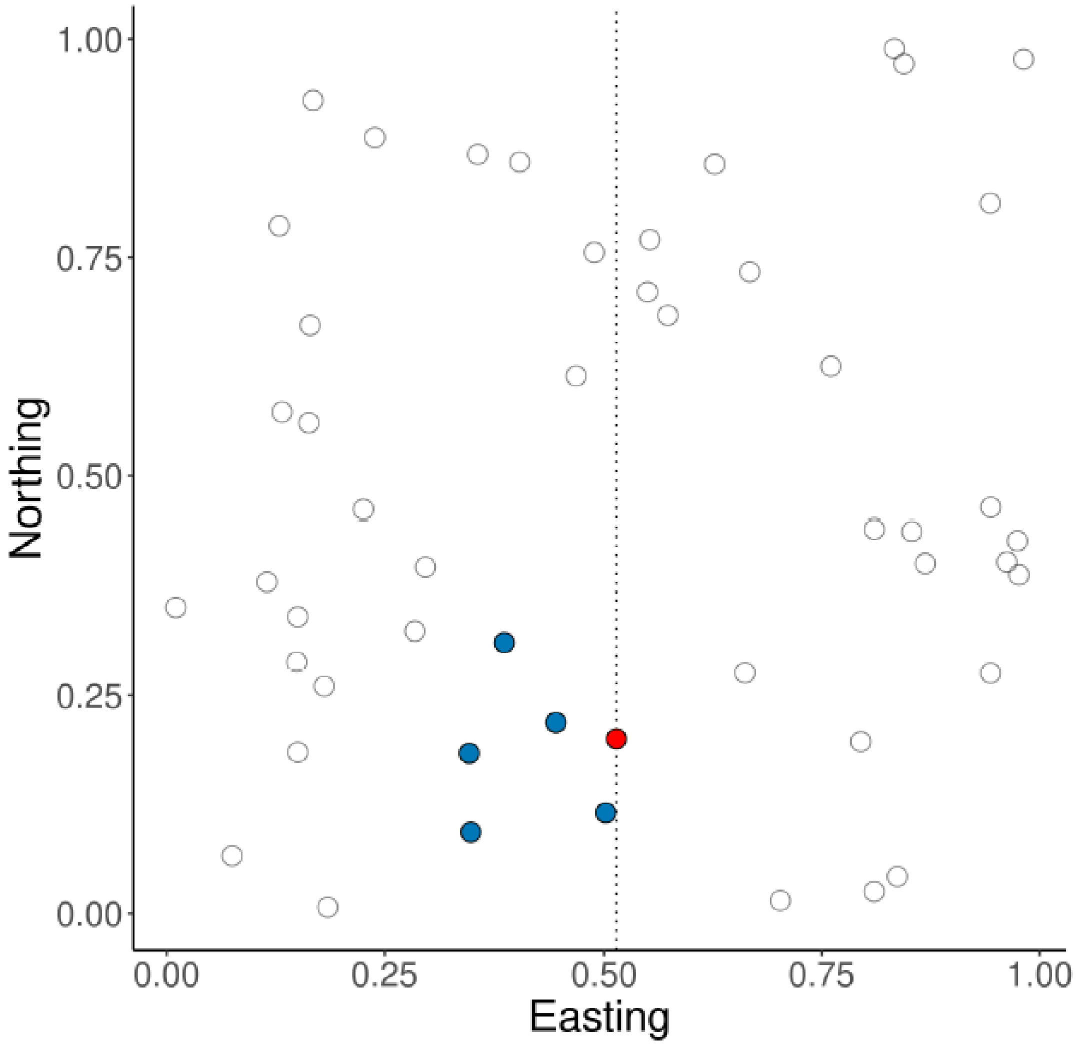
- Datta et al. 2016, Finley et al. 2019
- Same parameters and interpretation as the full GP!
- Conceptually:
  1. Order the spatial locations (e.g., along the x-axis).
  2. Determine the *m* nearest neighbors (subject to ordering) based on Euclidean (linear) distance.

Nearest Neighbor Gaussian Processes (NNGPs)

- Datta et al. 2016, Finley et al. 2019
- Same parameters and interpretation as the full GP!
- Conceptually:
  1. Order the spatial locations (e.g., along the x-axis).
  2. Determine the *m* nearest neighbors (subject to ordering) based on Euclidean (linear) distance.
  3. The spatial random effect at each site only depends on values of its *m* nearest neighbors and is conditionally independent of all other values.

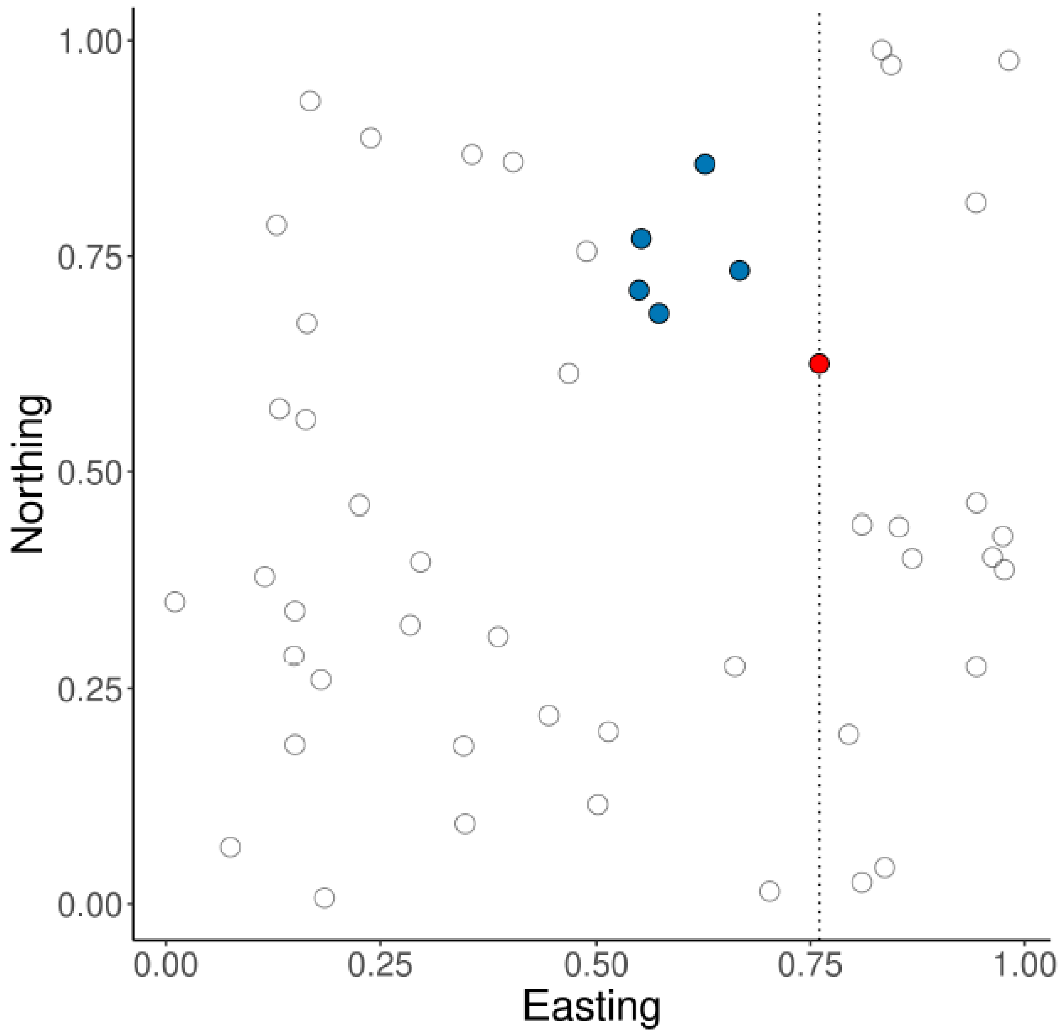
Choosing the neighbors

- `spOccupancy` and `spAbundance` order sites along the horizontal axis (i.e., Easting)
- Example: NNGP with 5 neighbors
- Red point denotes the current site
- Blue points denote sites in the "neighbor set"



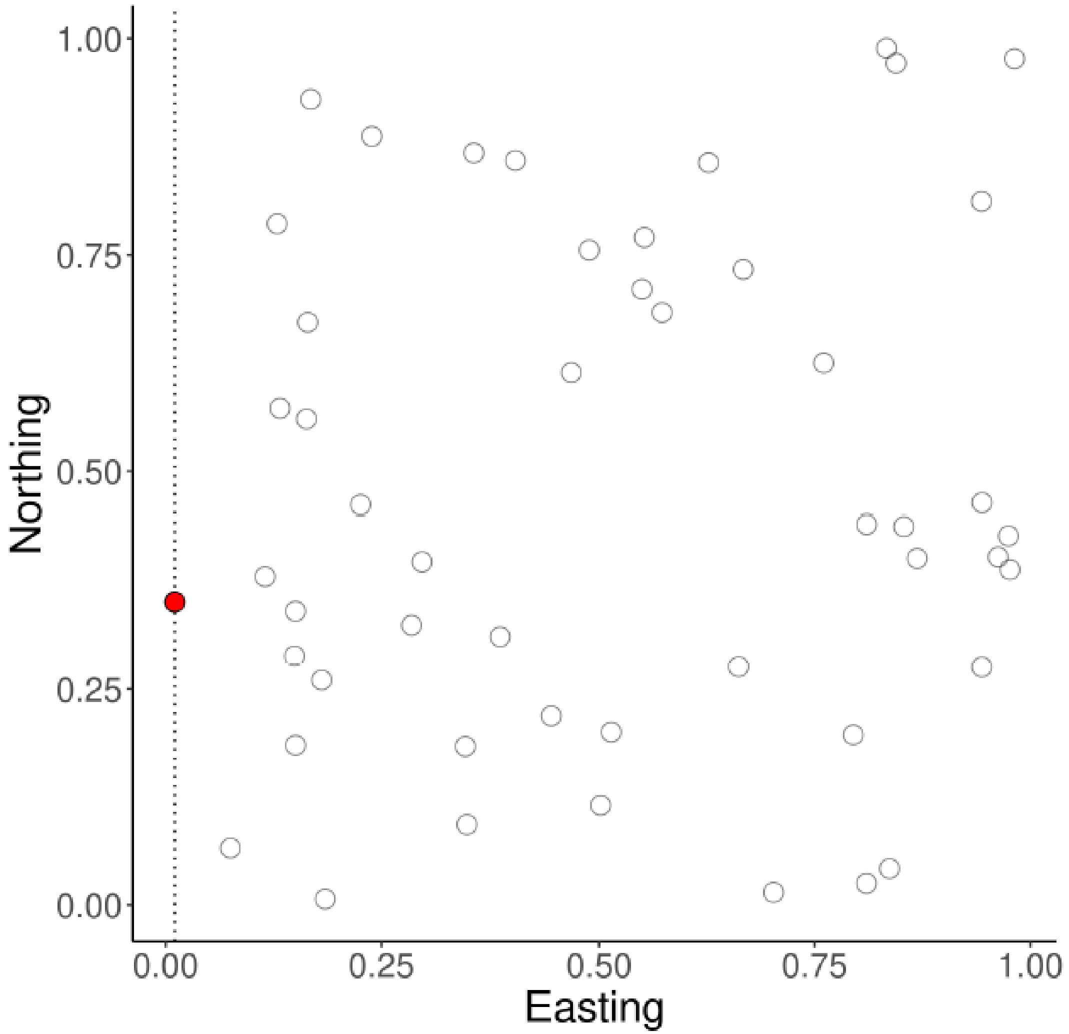
Choosing the neighbors

- `spOccupancy` and `spAbundance` order sites along the horizontal axis (i.e., Easting)
- Example: NNGP with 5 neighbors
- Red point denotes the current site
- Blue points denote sites in the "neighbor set"

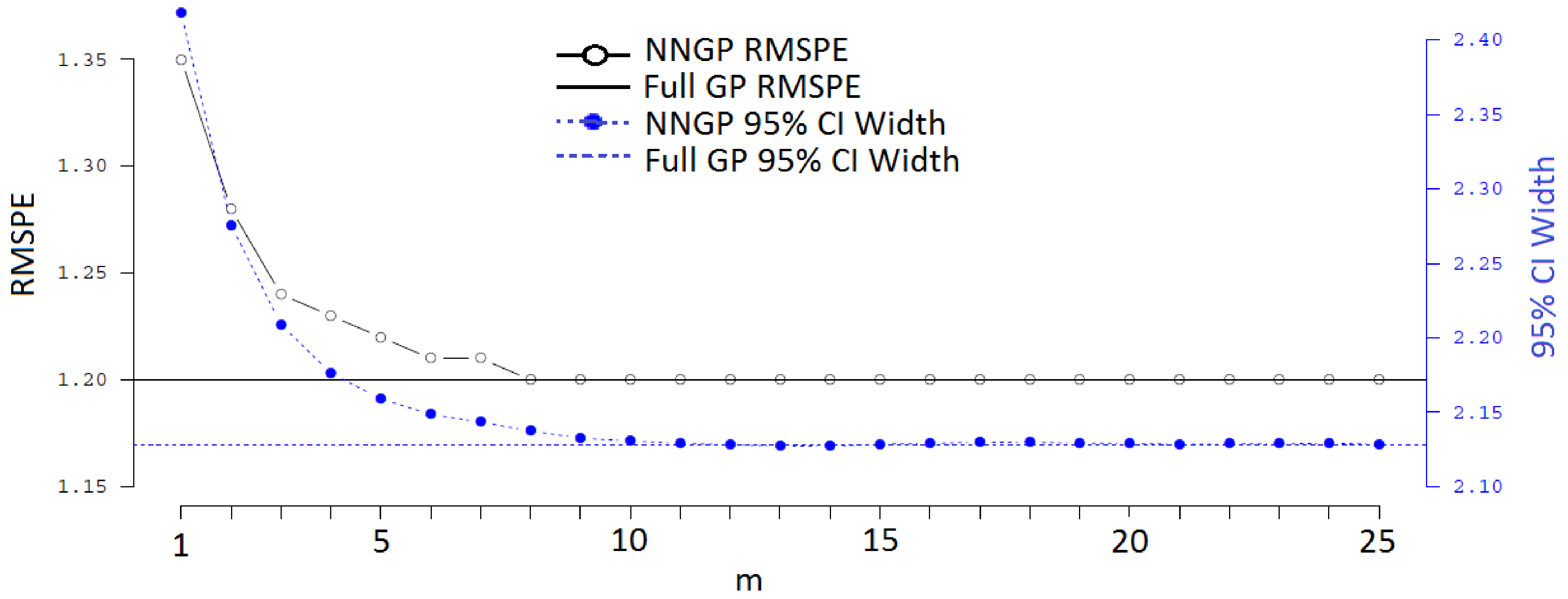


Choosing the neighbors

- `spOccupancy` and `spAbundance` order sites along the horizontal axis (i.e., Easting)
- Example: NNGP with 5 neighbors
- Red point denotes the current site
- Blue points denote sites in the "neighbor set"

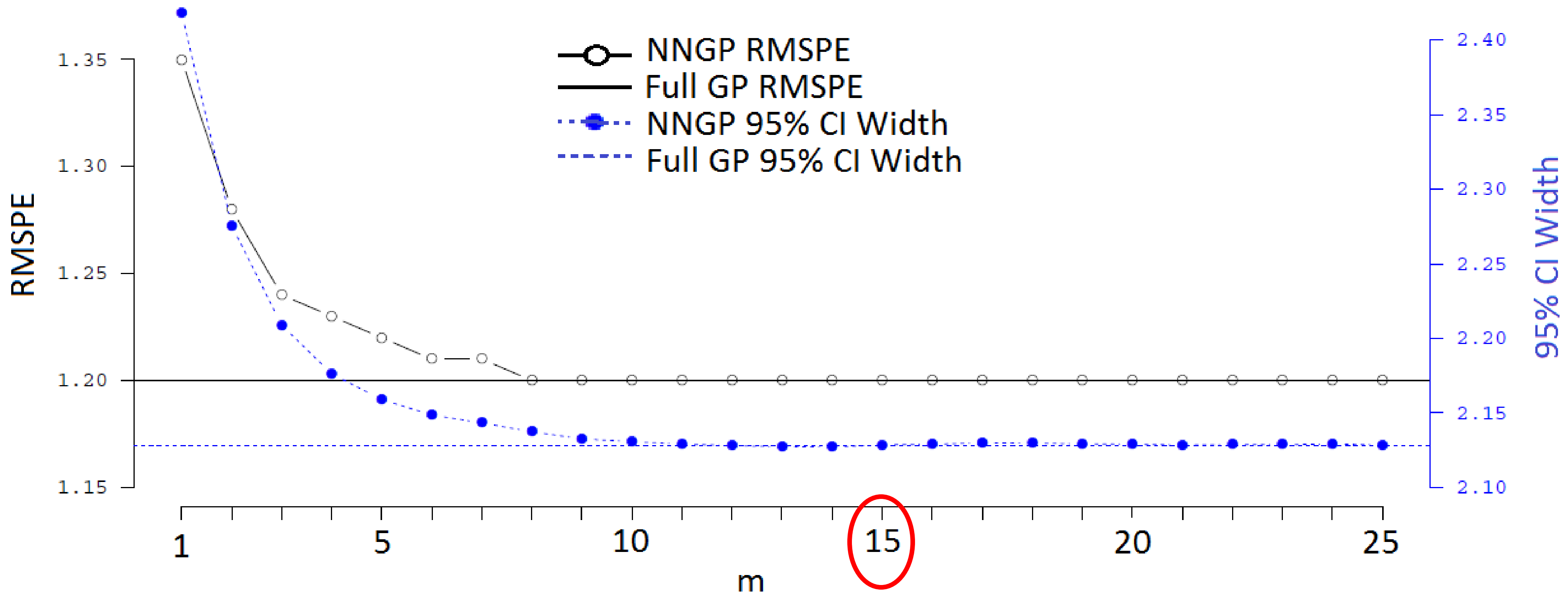


How many neighbors?



Datta et al. (2016) JASA

How many neighbors?



- m= 15 neighbors is often adequate (software default).
- Can compare smaller m using WAIC.

Datta et al. (2016) JASA

NNGPs in more detail

NNGPs in more detail

- Use a *sparse* matrix to approximate the dense Gaussian Process covariance matrix **C** (Datta et al. 2016).
- Based on rewriting the GP for **w** as a product of conditional densities.

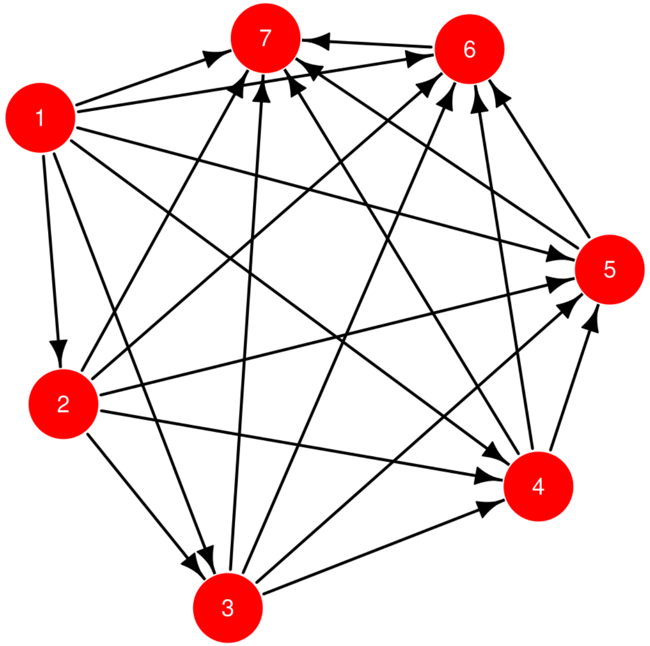
NNGPs in more detail

- Use a *sparse* matrix to approximate the dense Gaussian Process covariance matrix **C** (Datta et al. 2016).
- Based on rewriting the GP for **w** as a product of conditional densities.
- $\mathbf{w} \sim \mathrm{Normal}(\mathbf{0}, \boldsymbol{C})$ is equivalent to

$$p(w_1)p(w_2 \mid w_1)p(w_3 \mid w_1, w_2) \cdots p(w_n \mid w_1, w_2, \ldots, w_{n-1})$$

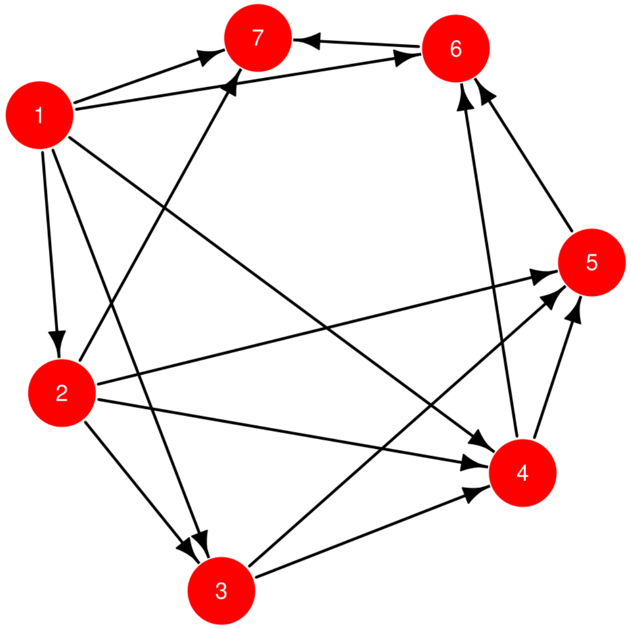
Introducing sparsity via graphical models

Introducing sparsity via graphical models



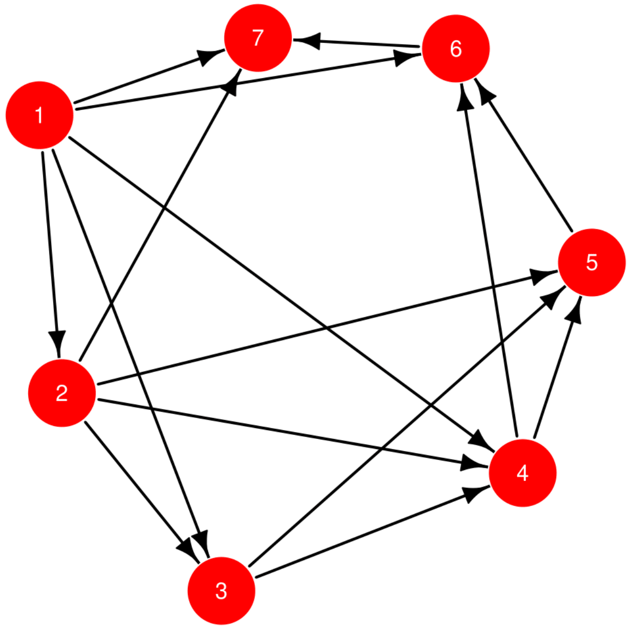
$$p(w_1)p(w_2 \mid w_1)p(w_3 \mid w_1, w_2)p(w_4 \mid w_1, w_2, w_3)$$
$$\times\; p(w_5 \mid w_1, w_2, w_3, w_4)p(w_6 \mid w_1, w_2, \ldots, w_5)$$
$$\times\; p(w_7 \mid w_1, w_2, \ldots, w_6)\;.$$

Introducing sparsity via graphical models



$$p(w_1)p(w_2 \mid w_1)p(w_3 \mid w_1, w_2)p(w_4 \mid w_1, w_2, w_3)$$
$$\times\ p(w_5 \mid \cancel{w_1}, w_2, w_3, w_4)p(w_6 \mid w_1, \cancel{w_2}, \cancel{w_3}, w_4, w_5)$$
$$\times\ p(w_7 \mid w_1, w_2, \cancel{w_3}, \cancel{w_4}, \cancel{w_5}, w_6)$$

Introducing sparsity via graphical models



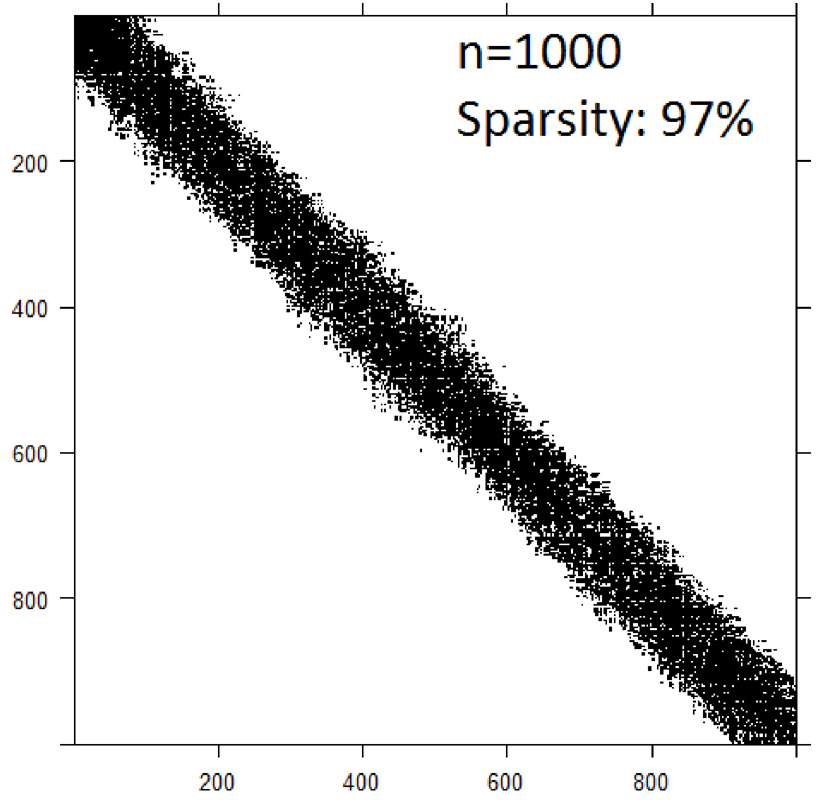
Size of neighbor set is $\leq$ m

$$p(w_1)p(w_2 \mid w_1)p(w_3 \mid w_1, w_2)p(w_4 \mid w_1, w_2, w_3)$$
$$\times\, p(w_5 \mid \cancel{w_1}, w_2, w_3, w_4)p(w_6 \mid w_1, \cancel{w_2}, \cancel{w_3}, w_4, w_5)$$
$$\times\, p(w_7 \mid w_1, w_2, \cancel{w_3}, \cancel{w_4}, \cancel{w_5}, w_6)$$

Sparsity via NNGP

$$\mathrm{Normal}(\mathbf{w} \mid \mathbf{0}, \boldsymbol{C}) \approx \mathrm{Normal}(\mathbf{w} \mid \mathbf{0}, \tilde{\boldsymbol{C}})$$

NNGP-derived covariance matrix



$\tilde{C}^{-1}$

Bayesian spatial NNGP linear model

$$y(\boldsymbol{s}_j) \sim \text{Normal}(\mu(\boldsymbol{s}_j), \tau^2)$$
$$\mu(\boldsymbol{s}_j) = \beta_0 + \beta_1 \cdot x_1(\boldsymbol{s}_j) + \text{w}(\boldsymbol{s}_j)$$
$$\mathbf{w} \sim \text{Multivariate Normal}(\mathbf{0}, \tilde{\boldsymbol{C}}(d, \sigma^2, \phi))$$

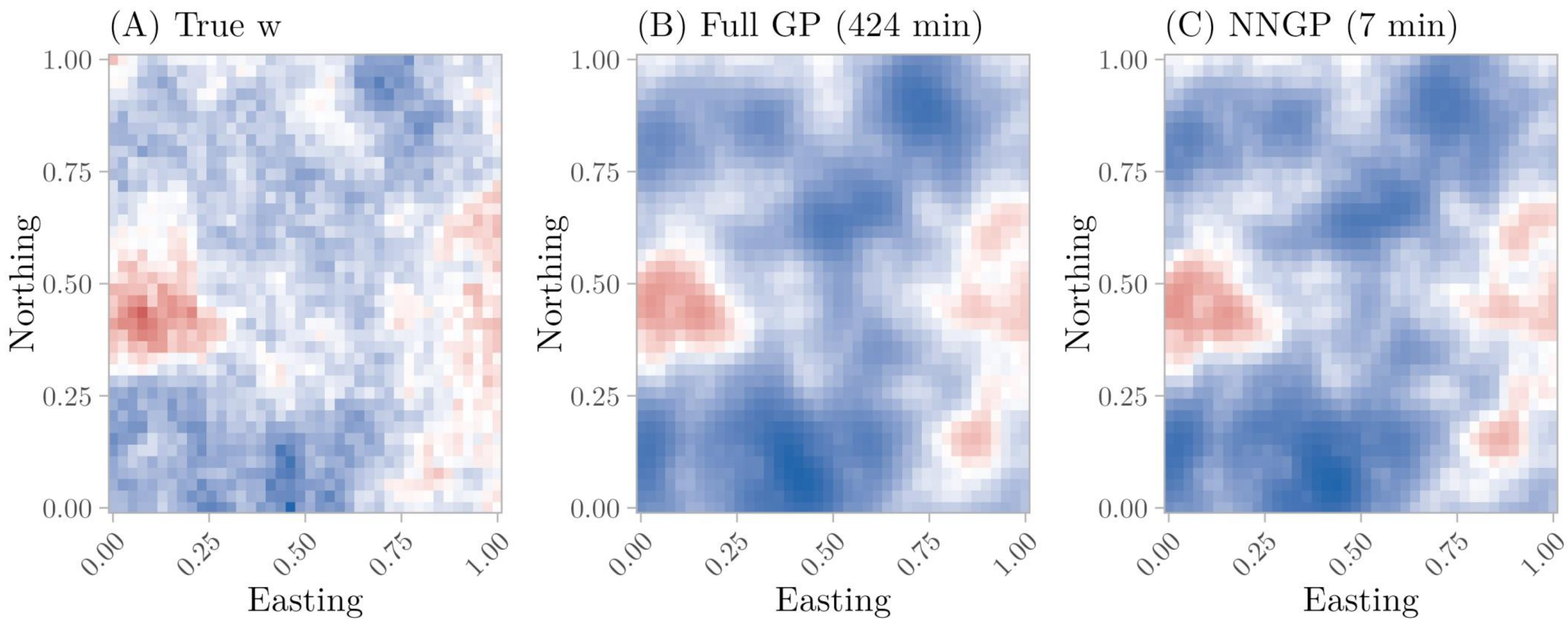
- All we do is replace the covariance matrix from the GP with the NNGP-derived covariance matrix. Otherwise, it's exactly the same!

Storage and computation

- Storage
  - Never need to store J x J distance matrix.
  - Stores smaller m x m matrices (m is the number of neighbors).
  - Total storage requirements: $O(Jm^2)$.
- Computation:
  - Only involves inverting small m x m matrices.
  - $O(Jm^3)$ flops.
- Since m << J, NNGP offers great scalability for big spatial data.

Simulation comparison

- *J= 1600,* fit a single-species spatial occupancy model



Simulation comparison

3a-gp-sim-example.R
3b-nngp-sim-example.R

- *J= 1600,* fit a single-species spatial occupancy model