

spatPomp: An R package for spatiotemporal partially observed Markov process models

Kidus Asfaw¹, Joonha Park², Aaron A. King^{3,4}, and Edward L. Ionides¹✉

¹ University of Michigan, Department of Statistics. ² University of Kansas, Department of Mathematics.
³ University of Michigan, Departments of Ecology & Evolutionary Biology and Complex Systems. ⁴ Santa Fe Institute, Santa Fe, New Mexico. ✉ Corresponding author

DOI: [10.xxxxxx/draft](https://doi.org/10.xxxxxx/draft)

Software

- [Review](#) ✉
- [Repository](#) ✉
- [Archive](#) ✉

Editor: [Open Journals](#) ✉

Reviewers:

- [@openjournals](#)

Submitted: 01 January 1970

Published: unpublished

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#)).

Summary

The development of spatPomp was motivated by the goal of investigating dynamics arising from a collection of spatially distributed, interacting biological populations. The entire population, consisting of the union of these sub-populations over all the spatial locations, is called a metapopulation. Each sub-population may have its own structure, which could correspond to disease status in an epidemiological model or abundance of several species in an ecosystem model. The spatPomp package embeds this goal in a more general problem: inference for spatiotemporal partially observed Markov process (SpatPOMP) models. A POMP model consists of a latent Markov process model, together with a measurement model describing how the data arise from noisy and/or incomplete observation of this latent state. The latent Markov process may be constructed in discrete or continuous time, taking scalar or vector values in a discrete or continuous space. POMP models are also known as state space models, or hidden Markov models. A SpatPOMP model extends the POMP model formulation by adding an index set corresponding to spatial location, so that the state of the SpatPOMP is comprised of a value for each location. We say “unit” rather than “spatial location” to build our framework in the general context of an arbitrary index set. Measurements are made on each unit, and are assumed to depend only on the latent state value for that unit. The spatPomp R package provides a computational framework for modeling and statistical inference on SpatPOMP models.

Statement of Need

The spatPomp package provides statistical methodology for a broad class of nonlinear and non-Gaussian SpatPOMP models. This gives scientists the freedom to construct and analyze scientifically motivated mechanistic models. spatPomp emphasizes likelihood-based inference, using scalable Monte Carlo methods to evaluate and maximize the likelihood function. Previous approaches for evaluating the likelihood function for SpatPOMP models required specific model assumptions: linear Gaussian SpatPOMP models can be investigated using the Kalman filter ([Kalman, 1960](#)); SpatPOMP models with sufficiently minor deviations from linearity and Gaussianity can be effectively analyzed using the extended Kalman filter or the ensemble Kalman filter ([Evensen et al., 2022](#)). Likelihood evaluation for highly nonlinear low-dimensional POMP models can be carried using the particle filter, also known as sequential Monte Carlo ([Chopin & Papaspiliopoulos, 2020](#)). However, the particle filter suffers from a curse of dimensionality that makes it inapplicable on SpatPOMP models. Recent algorithmic developments have addressed this limitation, permitting consideration of the general class of nonlinear non-Gaussian SpatPOMP models. spatPomp provides implementations of such algorithms, including bagged particle filters ([Ionides et al., 2023](#)), block particle filters ([Ning & Ionides, 2023](#)), guided

43 particle filters (Park & Ionides, 2020), and ensemble Kalman filters (Evensen et al., 2022).

44 SpatPOMP models with high nonlinearity and stochasticity can arise when investigating the
45 ecological dynamics of a spatially distributed collection of interacting biological populations,
46 known as a metapopulation. Existing demonstrations of spatPomp have arisen from studying
47 the ecology of infectious diseases (J. Li et al., 2024; Wheeler et al., 2024; Zhang et al., 2022).
48 In such epidemiological settings, the state of each unit may be comprised of the abundance
49 of a pathogen species, a host species, and perhaps also a vector species. We anticipate that
50 epidemiology will continue to be a major application area for SpatPOMP models. However,
51 this is a general model class with potential applications across the biological and social sciences,
52 healthcare, engineering, industry and government.

53 The spatPomp package is designed for researchers who aim to develop scientifically plausible
54 dynamic models to describe spatiotemporal systems. The package assists with the application
55 of existing models, modification of such models, or the development of entirely new models. It
56 provides methodologies to carry out statistical inference on these models, involving parameter
57 estimation, model selection, and model criticism. It focuses on algorithms with the plug-and-
58 play property, meaning that the dynamic model can be specified by code to simulate the latent
59 process for this model. A consequence of the plug-and-play property is that the data analyst is
60 not required to provide explicit specification of transition probabilities. This makes spatPomp a
61 flexible tool to assist model development.

62 The spatPomp package builds on pomp (King et al., 2016) which is a successful software
63 package for low-dimensional POMP models. Other packages with similar capabilities to pomp
64 include nimble (Michaud et al., 2021), LiBBi (Murray, 2015) and mcstate with odin and
65 dust (FitzJohn et al., 2020). All these packages enable plug-and-play inference based on
66 sequential Monte Carlo. Markov chain Monte Carlo packages, such as stan, have been found
67 to be effective for inference on some POMP models (M. Li et al., 2018) but they lack the
68 plug-and-play property. Perhaps for that reason, sequential Monte Carlo methods have found
69 broader applicability for this model class. We are not aware of alternative packages to spatPomp
70 that provide statistically efficient, plug-and-play inference for the general class of SpatPOMP
71 models.

72 Package Design

73 The spatPomp package provides a standardized interface between SpatPOMP models and
74 statistical inference methods. This approach is designed to provide an environment for data
75 analysis using existing algorithms as well as the development of new algorithms. New methods
76 can readily be tested on existing models, since the models have defined operations (such as
77 simulation, or evaluation of the measurement density) that the methods can access. For the
78 same reason, new models can readily be investigated using a range of methods. Currently, all
79 the methods in spatPomp have the plug-and-play property, i.e., they require a simulator for the
80 SpatPOMP model under investigation, but not an evaluator of its transitions densities. The
81 functionality of spatPomp permits specification of transition probabilities, so it is possible to
82 implement algorithms without the plug-and-play property. However, based on the development
83 trajectory of pomp (King et al., 2016), we anticipate that most use of spatPomp will focus on
84 plug-and-play methods.

85 The development of spatPomp has focused on likelihood-based inference. However, the
86 framework also permits Bayesian inference and consideration of non-likelihood-based model
87 fitting criteria.

88 A distance may be defined between units, and algorithms may assume that distant units have
89 only weak interactions. Such assumptions may involve a bias/variance tradeoff specific to
90 the choice of model and the choice of inference algorithm. Therefore, it may be beneficial to
91 evaluate various different algorithms when investigating a specific model of scientific interest.

92 The inter-operability of methods across models, provided by spatPomp, facilitates consideration
93 of a range of methods.

94 Resources

95 The spatPomp website (<https://spatPomp-org.github.io/spatPomp/>) provides links to various
96 resources for users and developers of the package. This includes the following.

- 97 1. An extended tutorial (Asfaw et al., 2024) introduces the mathematical framework behind
98 spatPomp, describes the software implementation of this framework, provides pseudocode
99 for various algorithms included in the package, and illustrates some basic usage. Section
100 2 explains the elementary methods used to access properties of the spatPomp model
101 class. These elementary methods are the building blocks available to developers for
102 implementing complex algorithms acting on spatPomp models.
- 103 2. A tutorial provided as a supplement to (Ning & Ionides, 2023) focuses specifically on
104 the iterated block particle filter algorithm. This is available at [https://spatPomp-org.
105 github.io/spatPomp/vignettes/ibpf.pdf](https://spatPomp-org.github.io/spatPomp/vignettes/ibpf.pdf).
- 106 3. A numerical comparison of spatiotemporal filtering methods by Ionides et al. (2023),
107 carried out using spatPomp, has source code available at [https://github.com/ionides/
108 bagged_filters](https://github.com/ionides/bagged_filters).
- 109 4. A spatiotemporal data analysis of cholera transmission in Haiti (Wheeler et al., 2024),
110 carried out using spatPomp, has source code available at [https://github.com/jeswheeler/
111 haiti_article](https://github.com/jeswheeler/haiti_article).
- 112 5. A spatiotemporal data analysis of COVID-19 transmission in China (J. Li et al., 2024),
113 carried out using spatPomp, has source code available at [https://github.com/jifanli/
114 metapop_article](https://github.com/jifanli/metapop_article).

115 Acknowledgments

116 This work was supported by National Science Foundation grants DMS-1761603 and DMS-
117 1646108, and National Institutes of Health grants 1-U54-GM111274, 1-U01-GM110712 and
118 1-R01-AI143852. We recognize those who have participated in the development and testing
119 of spatPomp, especially Allister Ho, Zhuoxun Jiang, Jifan Li, Patricia Ning, Eduardo Ochoa,
120 Rahul Subramanian and Jesse Wheeler. We are grateful to John Lees, Ben Bolker, and the
121 editors at The Journal of Open Source Software for their constructive feedback.

122 References

- 123 Asfaw, K., Park, J., King, A. A., & Ionides, E. L. (2024). A tutorial on spatiotemporal partially
124 observed Markov process models via the R package spatPomp. *arXiv:2101.01157v4*.
125 <https://doi.org/10.48550/arXiv.2101.01157>
- 126 Chopin, N., & Papaspiliopoulos, O. (2020). *An introduction to sequential Monte Carlo*.
127 Springer. <https://doi.org/10.1007/978-3-030-47845-2>
- 128 Evensen, G., Vossepoel, F. C., & Van Leeuwen, P. J. (2022). *Data assimilation fundamentals:
129 A unified formulation of the state and parameter estimation problem*. Springer Nature.
130 <https://doi.org/10.1007/978-3-030-96709-3>
- 131 FitzJohn, R. G., Knock, E. S., Whittles, L. K., Perez-Guzman, P. N., Bhatia, S., Guntoro,
132 F., Watson, O. J., Whittaker, C., Ferguson, N. M., Cori, A., Baguelin, M., & Lees, J. A.
133 (2020). Reproducible parallel inference and simulation of stochastic state space models

- 134 using odin, dust, and mcstate. *Wellcome Open Research*, 5. [https://doi.org/10.12688/](https://doi.org/10.12688/wellcomeopenres.16466.2)
135 [wellcomeopenres.16466.2](https://doi.org/10.12688/wellcomeopenres.16466.2)
- 136 Ionides, E. L., Asfaw, K., Park, J., & King, A. A. (2023). Bagged filters for partially observed
137 interacting systems. *Journal of the American Statistical Association*, 118, 1078–1089.
138 <https://doi.org/10.1080/01621459.2021.1974867>
- 139 Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of*
140 *Basic Engineering*, 82, 35–45. <https://doi.org/10.1115/1.3662552>
- 141 King, A. A., Nguyen, D., & Ionides, E. L. (2016). Statistical inference for partially observed
142 Markov processes via the R package pomp. *Journal of Statistical Software*, 69, 1–43.
143 <https://doi.org/10.18637/jss.v069.i12>
- 144 Li, J., Ionides, E. L., King, A. A., Pascual, M., & Ning, N. (2024). Inference on spatiotemporal
145 dynamics for coupled biological populations. *Journal of the Royal Society Interface*, 21(216),
146 20240217. <https://doi.org/10.1098/rsif.2024.0217>
- 147 Li, M., Dushoff, J., & Bolker, B. M. (2018). Fitting mechanistic epidemic models to data:
148 A comparison of simple Markov chain Monte Carlo approaches. *Statistical Methods in*
149 *Medical Research*, 27(7), 1956–1967. <https://doi.org/10.1177/0962280217747054>
- 150 Michaud, N., Valpine, P. de, Turek, D., Paciorek, C. J., & Nguyen, D. (2021). Sequential
151 Monte Carlo methods in the nimble and nimbleSMC R packages. *Journal of Statistical*
152 *Software*, 100, 1–39. <https://doi.org/10.18637/jss.v100.i03>
- 153 Murray, L. M. (2015). Bayesian state-space modelling on high-performance hardware using
154 LibBi. *Journal of Statistical Software*, 67(10), 1–36. <https://doi.org/10.18637/jss.v067.i10>
- 155 Ning, N., & Ionides, E. L. (2023). Iterated block particle filter for high-dimensional parameter
156 learning: Beating the curse of dimensionality. *Journal of Machine Learning Research*, 24,
157 1–76. <https://doi.org/10.48550/arXiv.2110.10745>
- 158 Park, J., & Ionides, E. L. (2020). Inference on high-dimensional implicit dynamic models
159 using a guided intermediate resampling filter. *Statistics and Computing*, 30, 1497–1522.
160 <https://doi.org/10.1007/s11222-020-09957-3>
- 161 Wheeler, J., Rosengart, A., Jiang, Z., Tan, K., Treutle, N., & Ionides, E. L. (2024).
162 Informing policy via dynamic models: Cholera in Haiti. *PLOS Computational Biology*, 20,
163 e1012032. <https://doi.org/10.1371/journal.pcbi.1012032>
- 164 Zhang, B., Huang, W., Pei, S., Zeng, J., Shen, W., Wang, D., Wang, G., Chen, T., Yang,
165 L., Cheng, P., Wang, D., Shu, Y., & Du, X. (2022). Mechanisms for the circulation of
166 influenza A (H3N2) in China: A spatiotemporal modelling study. *PLOS Pathogens*, 18(12),
167 e1011046. <https://doi.org/10.1371/journal.ppat.1011046>