Probabilistic graph models for landscape genetics

Brook G. Milligan
Department of Biology
New Mexico State University
Las Cruces, New Mexico 88003 USA
brook@nmsu.edu

July 4, 2016

Abstract

Progress in landscape genetics depends on a strong conceptual foundation and the means of identifying mechanistic connnections between environmental factors, landscape features, and genetic or genomic variation. Many existing approaches and much of the software commonly in use was developed for population genetics or statistics and is not entirely appropriate for landscape genetics. Probabilistic graph models provide a statistically rigorous and flexible means of constructing models directly applicable to landscape genetics. Sophisticated software also exists for the analysis of graph models. However, much of that software does not handle the types data used by landscape geneticis, model structures involving autoregressive spatial interaction between variables, or the scale of landscape genetics problems. Thus, an important priority for the field is to develop suitably flexible software tools for graph models that overcome these problems and allow landscape geneticists to explore meaningfully mechanistic and flexible models. We are developing such a library and applying it to examples in landscape genetics.

One recurring theme in the landscape genetics literature is that progress is limited by available analytical methods (Balkenhol et al., 2009, 2016a; Guillot et al., 2009). In part this derives from the fact that many of the available analytical tools and much of the usable software were originally developed for population genetics or even broader statistical applications. They may include assumptions or be applicable to data that are not completely appropriate for landscape genetics studies. Because of this gap, there is no consensus in the literature regarding how to approach landscape genetics analysis (Balkenhol et al., 2016a). Indeed, the most extreme view is that a comprehensive theory for landscape genetics is lacking and that more focus must be given to mechanistic understanding of the influence of landscapes and environments on genetic and genomic variation (Balkenhol et al., 2016b). Development of a more comprehensive theory will come

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.2225v1 | CC BY 4.0 Open Access | rec: 4 Jul 2016, publ: 4 Jul 2016

in part from an improved foundation of computational tools, especially open source ones, allowing explicit and flexible modeling.

This brief review focuses on three themes. First, it identifies the types of models most likely to contribute to advancement of a comprehensive theory of landscape genetics, improved mechanistic understanding, and better predictive power upon which, for example, conservation policy and management can be based. Second, it considers a set of open source software upon which such models for landscape genetics might be based. All of these turn out to have significant limitations. Consequently, it also suggests characteristics that will be essential for the ongoing development of models and computational tools most likely to advance landscape genetics.

Landscape genetics and Bayesian inference The prevailing challenge in landscape genetics is identifying the mechanisms by which landscape and environmental factors influence genetic and genomic variation. More precisely, the central question in landscape genetics is the following: given data on intraspecific genetic variation across landscapes (or waterscapes; Manel and Holderegger (2013); Selkoe et al. (2016)), what inferences are possible regarding the functional mechanisms and factors causing that variation? Framing the question in this way emphasizes the inherent connection between the science of landscape genetics and the nature of Bayesian inference.

Bayesian models in landscape genetics The natural connection between landscape genetics and Bayesian inference has led to the development of a variety of widely used Bayesian analysis methods. Although originally designed for population genetics, the most widely used is Structure, which identifies putative populations and assigns individuals to them (Pritchard et al., 2000). Applications of Structure in landscape genetics are dominated by either identifying putative populations that are subsequently compared with geographic locations or identifying migrant individuals whose genetic assignment is inconsistent with their geographic location. Neither of these applications addresses directly the core question of landscape genetics. A second set of Bayesian models applied to landscape genetics seeks to identify population clusters by modeling allele frequency distributions in a spatially explicit way (Chen et al., 2007; Guillot et al., 2005a,b). More recently, Bayesian models that explicitly relate environmental gradients to spatially explicit allele frequency distributions have been developed (Coop et al., 2010; Frichot et al., 2013). One element is common to all of these models and associated software: each one covers a particular type of model and provides very limited opportunity for exploring related models or for expanding their scope. This is a serious limitation for a scientific field that repeatedly asserts that more mechanistic models and a stronger theoretical foundation is essential (Balkenhol et al., 2016b).

Probabilistic graph models This gap is not for lack of a general statistical framework that is completely applicable. Probabilistic graph models (Bishop, 2006; Koller and Friedman, 2009) are the means of describing and analyzing a broad range of models and sophisticated software exists

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.2225v1 | CC BY 4.0 Open Access | rec: 4 Jul 2016, publ: 4 Jul 2016

to handle them. Indeed, the model underlying Structure (Pritchard et al., 2000) is an early contribution to latent factor analysis, a field that now finds application broadly in machine learning, artificial intelligence, and document and image processing, as well as landscape genetics (Frichot et al., 2013). Probabilistic graph models are composed of random variables (vertices) and relationships between them (edges), and are completely distinct from the graph theory applied to landscape genetics (Murphy et al., 2016). The primary advantage of probabilistic graph models is that complex and realisticly mechanistic models can be constructed, and the model structure can be manipulated easily to explore alternatives. Thus, there is great scope for constructing general theories. For many applications, Bayesian networks represented by directed acyclic graphs (DAGs) or Markov random fields represented by undirected graphs are sufficient; landscape genetics models, however, often require more general types of graphs to accommodate, for example, spatially autoregressive relationships among random variables.

Open-source probabilistic graph models While probabilistic graph models applied to landscape genetics do not generally harness their full flexibility, there exist modeling software that does. The most widely used is based upon the BUGS language for describing graph models, and includes WinBUGS, OpenBugs (Lunn et al., 2009) and JAGS (Plummer, 2015). The BUGS language allows textual description of general graph models that include a broad range of distributions. The textual description is translated into executable code, a process that introduces some of the limitations common to this type of modeling software. First, the flexibility of possible applications is limited by the features of the BUGS language. A limited range of data types, generally scalars and vectors or matrices constructed from them, is available, only data structures describable in the language may be used, and algorithms are limited to those already programmed. Second, the scale of models is limited by the execution environment provided by the implementation. Despite the inherent flexibility of graph models in general, both of these limitations are barriers to convenient development of landscape genetics models that leverage the flexibility of graph models. While genetic data can be recoded in the form of only integers or real numbers, it is tedious and error-prone to do so; thus, the limited data types available create needless barriers. A landscape genetics model might include thousands or millions of random variables within it; consider, for example, a model of population allele freqencies and environmental factors across a landscape grid of 1000×1000 pixels. This puts severe stress on models that cannot harness the full power of multithreading, distributed multiprocessing, and careful memory management. Being limited by the BUGS language, these programs provide no capacity for modelers to address these issues. Another general graph modeling system is Stan, named for Stanislaw Ulam, an inventor of Monte Carlo approaches to inference (Carpenter et al., 2015; Gelman et al., 2015). Although more flexible in some ways than BUGS, Stan suffers from some of the same limitations that reduce its applicability to landscape genetics. It has the same limited data types and the execution environment is likewise limited by the Stan language. In addition to these two major classes of graph modeling software, a broad range of more specialized software systems is also available; many of these are summarized by Murphy (2014). Some are open source and may have potential for landscape genet-

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.2225v1 | CC BY 4.0 Open Access | rec: 4 Jul 2016, publ: 4 Jul 2016

Name	Graph types	Primitive variables	Preprocessing	Implementation language	Reference
Darwin	FGs	scalars	compiled	C++	Gould (2015)
HYDRA	DAGs, MRFs, FGs, HMMs	Java classes	compiled	Java	Warmes (2013)
Infer.NET	FGs	C# classes	compiled	C#	Minka et al. (2014)
JAGS	DAGs	scalars	interpreted	C++	Plummer (2016)
JavaBayes	DAGs	scalars	interpreted	Java	Cozman (2001)
libDAI	FGs	discrete	compiled	C++	Mooiji (2015)
Mocapy++	DAGs, HMMs	C++ classes	compiled	C++	Antonov et al. (2015)
Nimble	DAGs	scalar	interpreted	C++	de Valpine et al. (2016)
OpenBUGS	DAGs	scalar	interpreted	Component Pascal	Thomas (2009)
OpenGM	DAGs, MRFs, FGs	discrete	compiled	C++	OpenGM (2015)
PNL	DAGs, MRFs	C++ classes	compiled	C++	Sysoyev et al. (2013)
RISO	DAGs	Java classes	compiled	Java	Dodier (2012)
Stan		scalars	interpreted	C++	Stan Development Team (2016)
Vibes	DAGs	scalar	compiled	Java	Winn (2004)

Table 1: A selection of open source software tools for analyzing probabilistic graph models. Type of graphs include directed acyclic graphs (DAGs), Markov random fields (MRFs), factor graphs (FGs) hidden Markov models (HMMs), and Gaussian Markov models (GMMs).

ics applications (Table 1). It is clear from the Table 1, however, that even beyond their specialized nature and general inaccessability to landscape geneticists, these tools also suffer from many of the same limitations. They often handle a more limited range of graphs than is needed for landscape genetics, the data types are not well suited to landscape genetics, or their execution environments are limiting. Overall, landscape geneticists interested in developing a strong conceptual foundation for the field based upon the expressive power, flexibility, and statistical rigor of probabilistic graph models are faced with a fundamental challenge. Existing frameworks such as provided by BUGS, JAGS, and Stan offer much flexibility and power but are designed for types of graphs, random variables, and data types that are not ideally suited to landscape genetics. Other software libraries may suffer from these same limitations but in addition are much more difficult to program and well beyond the reach of typical landscape geneticists.

Designing a probabilistic graph model for landscape genetics What then is the ideal design of a software system intended to harness the power, flexibility, and rigor of probabilistic graph models applied to landscape genetics? First and foremost, it must support a full range of relevant graph types, which in particular means not being limited to directed acyclic graphs. Second, it must support a full range of useful data types that landscape geneticists work with; in addition to simple scalars, vectors, and matrices, these include named alleles and genotypes, loci and chromosomes, spatial data of various sorts, and geographic locations. Ideally, user-defined or third-party data types should be easy to accommodate. Third, the algorithms available should be extensible to allow improved efficiency as needed. Fourth, the execution environment should not be limited to that encapsulated within a single, predefined program. This is especially important for landscape genetics models that may well encompass thousands or millions of random variables. Finally, all of

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.2225v1 | CC BY 4.0 Open Access | rec: 4 Jul 2016, publ: 4 Jul 2016

this power and flexibility must be abstracted enough that a full spectrum of landscape geneticists can create simple models easily, test alternative and biologically relevant models flexibly, and improve upon the models and algorithms as needed. It is little surprise that existing software tools are unable to meet these stringent demands; they are largely conflicting and impossible to resolve without advanced software design. The most likely path forward (Lunn et al., 2009) leverages the power of C++ to present high-level abstractions based upon embedded domain specific languages (de Guzman and Kaiser, 2016; Niebler, 2016) assembled with expression templates (Niebler, 2016; Veldhuizen, 1995) from highly reusable generic components (Stepanov and Rose, 2014). Although beyond the scope of this paper, we are following these design principles to implement a software library intended to provide the expressive power and computational performance demanded for advancing a coherent conceptual foundation for landscape genetics.

Conclusion Landscape genetics suffers greatly from the absence of an analytical foundation that encourages development of a mechanistic understanding of the impact of environmental and landscape factors on genetic and genomic variation (Balkenhol et al., 2016a). This stems in part from the adoption of software tools and methods originally developed for other purposes. There exist well-established concepts and statistical approaches associated with probabilitistic graph models that are ideally suited as the needed foundation for landscape genetics. Unfortunately, the associated software tools cannot be borrowed directly, because they are limited in ways that do not accommodate the needs of landscape geneticists. One priority that would directly advance the field and resolve these problems is the development of probabilistic graph model tools that do apply to landscape genetics. Despite the inherent difficulty of this task, we have developed a suitable library and are beginning to apply it to landscape genetics.

References

Antonov, L., M. Paluszewski, and T. Hamelrcyk. May 7 2015. Mocapy++. `https://sourceforge.net/projects/mocapy/`.

Balkenhol, N., S. A. Cushman, A. T. Storfer, and L. P. Waits, editors. 2016a. *Landscape genetics: concepts, methods, applications*. Wiley Blackwell.

Balkenhol, N., S. A. Cushman, L. P. Waits, and A. Storfer. 2016b. Current status, future opportunities, and remaining challenges in landscape genetics. In Balkenhol, N., S. A. Cushman, A. T. Storfer, and L. P. Waits, editors, *Landscape genetics: concepts, methods, applications*, chapter 14, pages 247–255. Wiley Blackwell.

Balkenhol, N., F. Gugerli, S. A. Cushman, L. P. Waits, A. Coulon, J. W. Arntzen, R. Holderegger, H. H. Wagner, and Participants of the Landscape Genetics Research Agenda Workshop 2007. 2009. Identifying future research needs in landscape genetics: where to from here? *Landscape Ecology*, 24:455–463.

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.2225v1 | CC BY 4.0 Open Access | rec: 4 Jul 2016, publ: 4 Jul 2016

Bishop, C. M. 2006. *Pattern recognition and machine learning*. Springer.

Carpenter, B., D. Lee, A. Gelman, B. Goodrich, J. Guo, M. Hoffman, M. Betancourt, P. Li, M. A. Brubaker, and A. Riddell. 2015. Stan: a probabilistic programming language. *Journal of Statistical Software*.

Chen, C., E. Durand, F. Forbes, and O. François. 2007. Bayesian clustering algorithms ascertaining spatial population structure: a new computer program and a comparison study. *Molecular Ecology Notes*, 7:747–756.

Coop, G., D. Witonsky, A. Di Rienzo, and J. K. Pritchard. 2010. Using environmental correlations to identify loci underlying local adaptation. *Genetics*, 185:1411–1423.

Cozman, F. G. January 1 2001. JavaBayes: Bayesian networks in Java. `http://www.cs.cmu.edu/~javabayes/index.html`.

de Guzman, J. and H. Kaiser. 2016. The Boost Spirit library. `http://www.boost.org/doc/libs/1_61_0/libs/spirit/doc/html/index.html`.

de Valpine, P., D. Turek, C. Paciorek, D. Temple Lang, and R. Bodik. May 27 2016. Nimble: numerical inference for hierarchical models using Bayesian and likelihood estimation. `https://bids.berkeley.edu/research/nimble-numerical-inference-hierarchical-models-using-bayesian-and-likelih`

Dodier, R. December 15 2012. RISO: distributed belief networks. `https://sourceforge.net/projects/riso/`.

Frichot, E., S. D. Schoville, G. Bouchard, and O. François. 2013. Testing for associations between loci and environmental gradients using latent factor mixed models. *Molecular Biology and Evolution*.

Gelman, A., D. Lee, and J. Guo. 2015. Stan: A probabilistic programming language for Bayesian inference and optimization.

Gould, S. 2015. DARWIN: a framework for machine learning and computer vision research and development. `http://drwn.anu.edu.au`.

Guillot, G., A. Estoup, F. Mortier, and J. F. Cosson. 2005a. A spatial statistical model for landscape genetics. *Genetics*, 170:1261–1280.

Guillot, G., R. Leblois, A. Coulon, and A. C. Frantz. 2009. Statistical methods in spatial genetics. *Molecular Ecology*, 18:4734–4756.

Guillot, G., F. Mortier, and A. Estoup. 2005b. GENELAND: a computer package for landscape genetics. *Molecular Ecology Notes*, 5:712–715.

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.2225v1 | CC BY 4.0 Open Access | rec: 4 Jul 2016, publ: 4 Jul 2016

Koller, D. and N. Friedman. 2009. *Probabilistic graphical models: principles and techniques*. MIT Press, Cambridge, MA.

Lunn, D., D. Spiegelhalter, A. Thomas, and N. Best. 2009. The BUGS project: evolution, critique and future directions. *Statistics in Medicine*, 28:30493067.

Manel, S. and R. Holderegger. 2013. Ten years of landscape genetics. *Trends in Ecology and Evolution*, 28:614–621.

Minka, T., J. Winn, J. Guiver, S. Webster, Y. Zaykov, B. Yangel, A. Spengler, and J. Bronskill. 2014. Infer.NET 2.6. `http://research.microsoft.com/en-us/um/cambridge/projects/infernet/default.aspx`.

Mooiji, J. September 24 2015. libDAI: a free and open source C++ library for discrete approximate inference in graphical models. `https://staff.fnwi.uva.nl/j.m.mooij/libdai/`.

Murphy, K. June 16 2014. Software packages for graphical models. `https://www.cs.ubc.ca/~murphyk/Software/bnsoft.html`.

Murphy, M., R. Dyer, and S. A. Cushman. 2016. Graph theory and network models in landscape genetics. In Balkenhol, N., S. A. Cushman, A. T. Storfer, and L. P. Waits, editors, *Landscape genetics: concepts, methods, applications*, chapter 10, pages 165–179. Wiley Blackwell.

Niebler, E. 2016. The Boost Proto library. `http://www.boost.org/doc/libs/1_61_0/doc/html/proto.html`.

OpenGM. September 23 2015. OpenGM. `http://hciweb2.iwr.uni-heidelberg.de/opengm/`.

Plummer, M. October 1 2015. JAGS version 4.0.0 user manual. Technical report.

Plummer, M. January 16 2016. JAGS. `http://mcmc-jags.sourceforge.net`.

Pritchard, J. K., M. Stephens, and P. Donnelly. 2000. Inference of population structure using multilocus genotype data. *Genetics*, 155:945–959.

Selkoe, K. A., K. T. Scribner, and H. M. Galindo. 2016. Waterscape genetics—applications of landscape genetics to rivers, lakes, and seas. In Balkenhol, N., S. A. Cushman, A. T. Storfer, and L. P. Waits, editors, *Landscape genetics: concepts, methods, applications*, chapter 13, pages 220–246. Wiley Blackwell.

Stan Development Team. 2016. The Stan math library, version 2.10.0. `http://mc-stan.org`.

Stepanov, A. A. and D. E. Rose. 2014. *From Mathematics to Generic Programming*. Addison-Wesley, first edition.

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.2225v1 | CC BY 4.0 Open Access | rec: 4 Jul 2016, publ: 4 Jul 2016

Sysoyev, A. V., B. Milch, G. R. Bradski, and D. Dash. April 16 2013. Probabilistic networks library. https://sourceforge.net/projects/openpnl/.

Thomas, A. December 14 2009. OpenBugs. http://www.openbugs.net/w/FrontPage.

Veldhuizen, T. June 1995. Expression templates. *C++ Report*, 7(5):26–31.

Warmes, G. October 23 2013. HYDRA MCMC library. https://sourceforge.net/projects/hydra-mcmc/.

Winn, J. 2004. Variational inference for Bayesian networks. http://vibes.sourceforge.net.