`demes`: a Python package for pain-free specification of complex demography

Graham Gower[1,*], Jerome Kelleher[2,*], Aaron P. Ragsdale[3,*], and Kevin Thornton[4,*]

1

2

3

4

[*]Authors listed alphabetically

November 11, 2020

Abstract

come back to title and abstract later

Introduction

The ever-increasing amount of genetic sequencing data from genetically and geographically diverse species and populations has allowed us to infer complex demography and study life history at fine scales. An integral component to such population genetics studies is simulation. Software to either simulate whole genome sequences (Thornton, 2014, Kelleher et al., 2016, Haller and Messer, 2019, Adrion et al., 2020) or informative summary statistics of diversity (Gutenkunst et al., 2009, Kamm et al., 2017, Jouganous et al., 2017) have enabled the increasing complexity of genomic studies, with many software packages able to handle large sample sizes, many interacting populations, and deviations from random-mating models of panmixia.

The ability to simulate complex demographic scenarios, however, does not come without its own set of obstacles and frustrations. First of all, specifying models with many populations and parameters can be cumbersome and error-prone (e.g., (Ragsdale et al., 2020)). It is time-consuming and tedious to implement and then verify the correctness of such demographic models. Making matters worse, each software package typically has its own application programming interface (API), which makes it difficult to translate models between simulation methods.

Secondly, many software API are not particularly easy to read, especially when there are a large number of demographic events and parameters. This not only makes debugging difficult, but it can be difficult to even notice if a mistake has been made. A common interface for unambiguously specifying demographic models would reduce implementation errors and promote ease-of-use for many simulation software packages.

Finally, simulated or inferred demographic models are regularly shared in the literature, and for someone else to be able to reproduce or reimplement some demographic scenario, it needs to be described fully and without ambiguity. However, it is common for reported demographic models to lack clear or even complete descriptions, hindering reproducibility. A human-readable description for such models would facilitate their sharing and reimplementation.

Here, we present `demes`, a Python package for specifying complex demographic models in a way that is both human readable and can be directly passed to a growing number of simulation engines. `Demes` supports model specification in a high-level `YAML` format (Ben-Kiki et al., 2009) that is designed to be as easy as possible to implement, read, and share. Models are then represented in an unambiguous low-level format within Python that checks model validity and can then be used as the input for simulation software. The initial release of `demes` (version 1.0, described here) supports *static* demographic descriptions, that is, models with fixed parameters. Future releases will support [drawing parameters from distributions, for example for ABC-based inference, or describing "dynamic" models where parameters may be fit to data, e.g. `moments` or ∂a∂i].

Defining demographic models with `demes`

High-level model specification in `YAML`

- Short description of YAML language and why it's appropriate here
- General overview (1 paragraph) of defining demography in YAML
- A generic, commonly used, simple demographic model for illustration (e.g. IM)

Demographic Model 1: An IM model.

```yaml
description: A two-population isolation-with-migration model.
time_units: generations
demes:
  ancestral:
    description: The ancestral deme that splits into two child demes.
    initial_size: 5000
    end_time: 2000
  deme1:
    description: The first child deme.
    ancestors: ancestral
    initial_size: 1000
    final_size: 10000
  deme2:
    description: The second child deme.
    ancestors: ancestral
    initial_size: 2000
    final_size: 8000
migrations:
  symmetric:
  - demes: deme1, deme2
    rate: 1e-4
```

Python representation of demography as input for simulation software

- The Python API, reading in a YAML, and unambiguous and manipulable representation

Software support for `demes`

APR: at time of publication. TBD...

- `msprime`
- `stdpopsim`
- `fwdpy11`
- `moments`
- `∂a∂i`? should try to get Ryan on board
- `slim`? I guess that comes with `stdpopsim`

Specifying complex demography

APR: A few examples of complex demography - e.g. 1) the very common out-of-Africa model, 2) Denisovan admixture with a ton of populations and demographic events, 3) a non-human model, perhaps with selfing rates and rate changes

APR: Perhaps examples should be in the appendix, to keep the paper concise

Prospects and future directions

References

Jeffrey R Adrion, Christopher B Cole, Noah Dukler, Jared G Galloway, Ariella L Gladstein, Graham Gower, Christopher C Kyriazis, Aaron P Ragsdale, Georgia Tsambos, Franz Baumdicker, et al. A community-maintained standard library of population genetic models. *eLife*, 9:e54967, 2020.

Oren Ben-Kiki, Clark Evans, and Brian Ingerson. Yaml ain't markup language (yaml™) version 1.1. *Working Draft 2008-05*, 11, 2009.

Ryan N Gutenkunst, Ryan D Hernandez, Scott H Williamson, and Carlos D Bustamante. Inferring the joint demographic history of multiple populations from multidimensional snp frequency data. *PLoS genetics*, 5(10):e1000695, 2009.

Benjamin C Haller and Philipp W Messer. Slim 3: Forward genetic simulations beyond the wright–fisher model. *Molecular biology and evolution*, 36(3):632–637, 2019.

Julien Jouganous, Will Long, Aaron P Ragsdale, and Simon Gravel. Inferring the joint demographic history of multiple populations: beyond the diffusion approximation. *Genetics*, 206 (3):1549–1567, 2017.

John A Kamm, Jonathan Terhorst, and Yun S Song. Efficient computation of the joint sample frequency spectra for multiple populations. *Journal of Computational and Graphical Statistics*, 26(1):182–194, 2017.

Jerome Kelleher, Alison M Etheridge, and Gilean McVean. Efficient coalescent simulation and genealogical analysis for large sample sizes. *PLoS computational biology*, 12(5):e1004842, 2016.

Aaron P Ragsdale, Dominic Nelson, Simon Gravel, and Jerome Kelleher. Lessons learned from bugs in models of human history. *American Journal of Human Genetics*, 107(4):583–588, 2020.

Kevin R Thornton. A c++ template library for efficient forward-time population genetic simulation of large populations. *Genetics*, 198(1):157–166, 2014.

Appendix