UNDER REVIEW

RESCIENCE C

Reproducing "Fluctuation Domains"

Carl Boettiger[1, ID]
[1]University of California, Berkeley, 130 Mulford Hall, Berkeley, CA 94720-3114 USA

Edited by
(Editor)

Received
—

Published
—

DOI
—

Abstract A stochastic ensemble simulation of evolutionary dynamics has been successfully reproduced. Simulation code was in C and could be called from wrappers implemented as an R package. Reproduction required minor modification to the configuration files for compatibility with newer versions of the GNU Scientific Library.

A replication of Boettiger2010.

Historical context

In this analysis I seek to reproduce the numerical results of (Boettiger, Dushoff, and Weitz 2010), which focuses on the evolutionary trajectory of species trait under selection, as predicted by the so-called "Canonical Equation" of adaptive dynamics (Dieckmann and Law 1996). Theoretical work I present in the paper predicts the emergence of a 'fluctuation enhancement' regime far from an evolutionary stable point, and suggests that the evolution of traits which start in this distant region would deviate substantially from the expected evolutionary path predicted by the canonical equation. In particular, the distribution of such trajectories could be bimodal, making the "canonical" path far from the most likely path. The details of this distribution cannot be solved for analytically however, so in this paper I resorted to an replicate exact stochastic simulations of the evolutionary model using a Gillespie algorithm (Gillespie 1977).

I implemented Gillespie algorithm for the evolutionary model in C code, and wrapped the C code in an R package structure to make it easier to run and easier to plot results. The C code depended on one standard C library, the GNU Scientific Library, which has evolved considerably (with one major version change) since that time. The original source code was 'published' (initially only on a colleague's website, as stated in the paper, though later copies were submitted to formal archives, as discussed below). That original copy was distributed under a GPL v3 licence, though I later replaced that with a CC0 license when submitting the version to a data archive (Dryad) which required public domain declarations.

Retrieval of software

I have changed laptops and the organizational schemes I use for my documents several times since publishing the original paper. Though I still have my original copy of the code for the manuscript sitting around somewhere in a compressed, encrypted archive blob that now lives on Amazon Glacier, accessing it there would have been complex and demanding. Fortunately, in 2011 I uploaded a copy of this code to a dedicated repository on GitHub at https://github.com/cboettig/fluctuationDomains. I rely on this copy for the remainder of this reproducibility effort.

The published paper itself states that the code is available at the personal website of one of

Correspondence should be addressed to Carl Boettiger (cboettig@berkeley.edu)
The authors have declared that no competing interests exists.
Code is available at https://github.com/cboettig/fluctationDomains..

my co-authors, with URL http://ecotheory.biology.gatech.edu/downloads/. Fortunately, this URL still resolves (my colleague is still at the same institution), and does indeed contain a downloadable zip file with a copy of the code. In 2012, I had uploaded a copy of this code as a supplement to Dryad, but both on my CV at the time and on my colleague's website we had recorded only the Dryad handle URL, http://datadryad.org/handle/10255/dryad.37625, which no longer resolves. That handle can be resolved by the WayBackMachine, which doesn't contain the code archive but does reveal the DOI, http://dx.doi.org/10.5061/dryad.j8n0p7vc, which successfully resolves to the Dryad copy. The code can also be found by searching the Dryad web interface for my surname (though it is listed as "data" and not "software." In 2017, I had also imported the GitHub copy into the Zenodo data repository where it was again assigned a DOI and is identified as software in the metadata. I note that the package was never submitted to the Central R Archive Network, CRAN.

Notably, these versions are not entirely the same. The GitHub/Zenodo/Dryad version has the addition of a `configure.ac` script for autoconf, intended to make the package more portable in assisting with the linkage to the GSL library, which is missing from the original deposit (on the GeorgiaTech website and Dryad). The GitHub version also lacked a copy of the `inst/` sub-directory, which contained a copy of the manuscript `.tex` file, original manuscript `.pdf` figures, and the R scripts that were supposed to generate these figures. (Though both versions include a `demo` directory that included similar code required to generate the figures, as discussed in the next section.)

Execution

I had taken what I thought at the time would be appropriate steps to facilitate reproducibility, stating at the end of the manuscript:

> In the spirit of scientifically reproducible research(Gentleman and Lang, 2007; Schwabet al., 2000; Stodden, 2009), we freely provide all the source code required to replicate the simulations and figures shown in the text. Though the numerical simulations are written in C for computational efficiency, we provide a user interface and documentation by releasing all the code, figures, text, and examples as a software package for the widely used and freely available R statistical computing language. This package can be downloaded from http://ecotheory.biology.gatech.edu/downloads/.

Unfortunately, in attempting execution I ran into two significant obstacles that would have likely stopped a typical researcher unfamiliar with the details of the project.

First, the formally archived copy (GitHub/Dryad/Zenodo version) could not be installed directly due in fact to the autoconf `configure` script, which incorrectly states that it cannot find a version of the GNU Scientific Library (GSL) `>=1.8` (the version against which the package was originally built – the current stable version is 2.6, see http://git.savannah.gnu.org/cgit/gsl.git/tree/NEWS for the changelog). Ironically, the original GeorgiaTech archive version installed directly for me on both Mac and Linux platforms without any additional work (the default library linking flags in the provided `Makevars` file of the original source were sufficient.) For instance, on a standard Ubuntu 18.04 (bionic) instance with `r-base`, and `libgsl-dev` installed from the `apt` repos, it was sufficient for me to run:

```
wget http://ecotheory.biology.gatech.edu/sites/default/files/adpaper_0.2-8.tar.gz
R CMD INSTALL adpaper_0.2-8.tar.gz
```

The same strategy worked on MacOSX (Catalina) with `gsl` installed via Homebrew. I was able to update the original autoconf script by recalling that it had been adapted from another package, the `gsl` R package https://cran.r-project.org/web/packages/gsl/, which also binds against the GSL library. Working from the autoconf files provided

there, I could successfully get the package to configure and compile. This highlights a potential risk in changes that are intended to make the package more portable but also add complexity. More generally, while the package has no dependency on external R packages (beyond those default packages provided as part of base R: `stats`, `utils`, `graphics`, and `grDevices`, this issue illustrates the familiar reproducibility challenge that is created by changing versions of dependencies and system libraries (GSL in this case). System libraries against which a package is compiled are particularly pernicious, because versioning of these components is both platform dependent and separate from the R package manager process, and thus cannot be addressed merely with package version tracking in tools such as `packrat` (Ushey et al. 2018) or `renv` (Ushey 2019). Today, containerized approaches such as Docker (see Boettiger 2015; Boettiger and Eddelbuettel 2017) provide a solution to this system-dependency issue that was not available a decade ago.

The second challenge was in figuring out just what commands had to be run to duplicate the figures. As mentioned above, the GitHub version of the source-code did not have the manuscript source, and even the version that did have the manuscript source did not write the manuscript itself as a dynamic document (e.g. in Sweave (Xie 2015)). In that version, files `fig1.R` and `fig2.R` are were included in the same sub-directory as the `pdf` figures referenced by the `tex` version of the manuscript, but these files contained only the code required to actually generate the figures themselves once the result data had been pre-computed and saved as `.txt` files. The result data files had also not been archived separately. Fortunately, re-reading my own documentation, I was able to deduce that these text files were generated by running the function `logistic()` with the parameter settings given in the caption of Fig 2 in the manuscript.

Reproducibility — Running the `logistic()` function provided in the code, I was able to replicate the key results of the paper. A few qualifications here are necessary. The paper consists of two core results – a theoretical (i.e. derived analytically without computational component) which predicts the existence of 'fluctuation enhancement regimes' which can create substantial deviations from the dynamics predicted by the accepted theory at the time (that is, from the so-called canonical equation of adaptive dynamics), and a second result which relies on a computationally intensive stochastic ensemble simulation to illustrate these deviations arising for dynamics that begin in the "fluctuation enhancement" regime. These results are summarized by figures 1 and 2 of the original paper, which I reproduce here.

Reproducing figure 1 worked directly from the provided code, which merely plots equations derived from a little stochastic calculus as presented in the paper and appendix:

```
library(fluctuationDomains)
## Figure 1 function definitions:
logistic_curve = function(x){-x*exp(-x*x/2)}
chemostat_curve <- function(x, d = 0.1, q = 0.1){
  (q*x-d/(1-x*d))* (d/x-2*d**2)
}
branching_curve <- function(x, SIGMA2_K = 2, SIGMA2_C = 1){
  -exp( (x^2/2)*(4/SIGMA2_C -1/SIGMA2_K) ) *
    (SIGMA2_C+exp(2*x^2/SIGMA2_C)*SIGMA2_C-2*SIGMA2_K)*x /
    ( (1+exp(2*x^2/SIGMA2_C)^2)*SIGMA2_C*SIGMA2_K)
}

landscape(logistic_curve, -3, 3)
landscape(chemostat_curve, 1, 9)
landscape(branching_curve, -.8, .8)
```

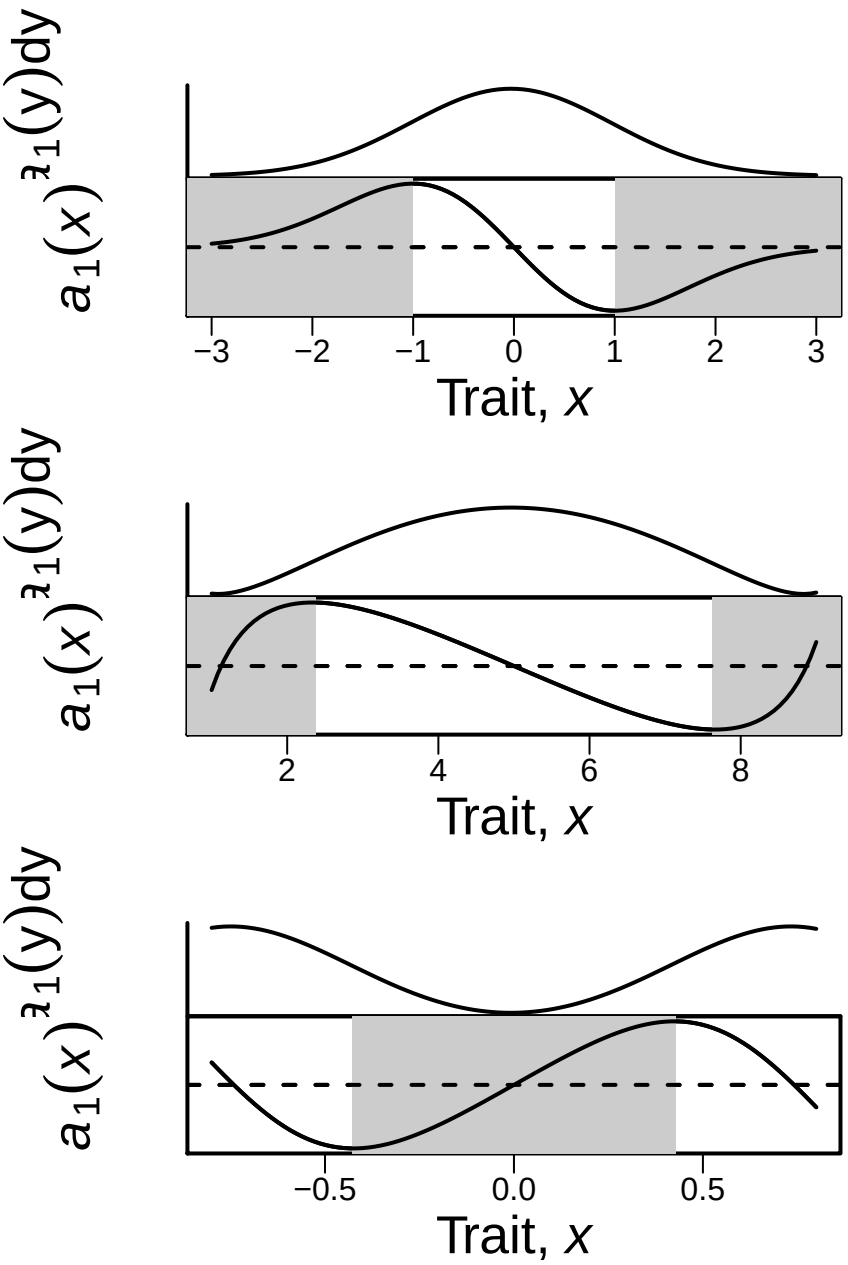

Figure 1. The three panels of figure 1, showing examples of evolutionary landscapes and the corresponding fluctuation domains for three classic models (logistic, chemostat, branching model). This figure is largely theoretical, and so easily replicates Figure 1 of the original paper. The first panel showing the logistic model depicts the fluctuation regimes for the model parameters used in the numerical analysis.

Figure 2 of the original paper has the core computational results demonstrating the consequences of these fluctuation regimes. The code for this analysis was implemented as pure C code, which can be run in stand-alone mode and generates three output text files. The R package uses the `.C` interface to pass parameters to the C function (`Rlogistic`) which runs the ensemble and writes its output as text. The C function does not expose a random seed, so re-running the analysis creates results which are not bit-wise identical each time. Of course the scientific results are expected to be independent of the seed anyway, so it is perhaps reassuring that as long as the ensemble of replicates is large enough, the same pattern can be reproduced quite closely. The original manuscript uses ensembles of 100,000 replicates, which I have reproduced here. A possible source of confusion is that the code defaults to only 100 replicates if the ensemble size is not explicitly verified – which is an ensemble small enough to still see random deviations from the predicted results. I probably set that lower default to facilitate faster testing. Likewise, the MAXTIME parameter has to be extended from the default setting to allow the third initial condition set to run sufficiently long.

The code was never parallelized, but runs with a small memory footprint by streaming simulation data out (e.g. to disk) over `fprint`. Replicating all results takes about an hour of computational time on a Intel-i7 processor.

```
out = logistic(Xo = 1,  ENSEMBLES = 10^5, MAX_TIME = 3000)
```

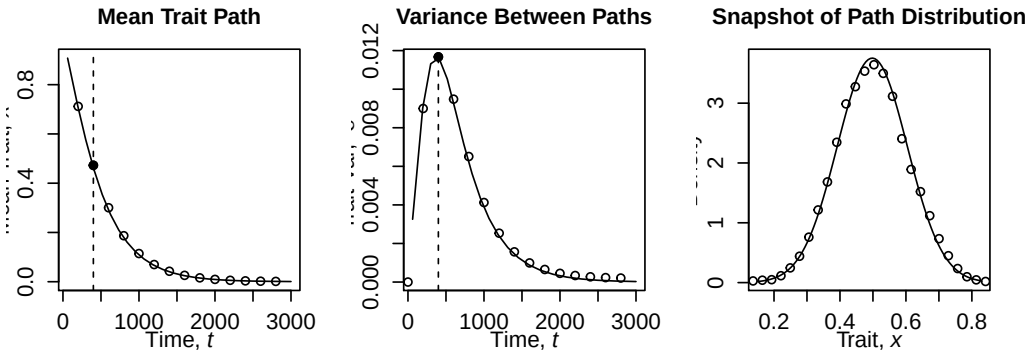

Figure 2. A starting condition of Xo = 1, at the edge of the fluctuation dissipation regime. The whole evolutionary trajectory thus occurs within the dissipation regime, and mean dynamics closely match that of the canonical equation, and show a tight normal distribution around the expected path. This matches the trajectories seen figure 2 of the original manuscript.

```
out = logistic(Xo = 2,  ENSEMBLES = 10^5, MAX_TIME = 5000)
```

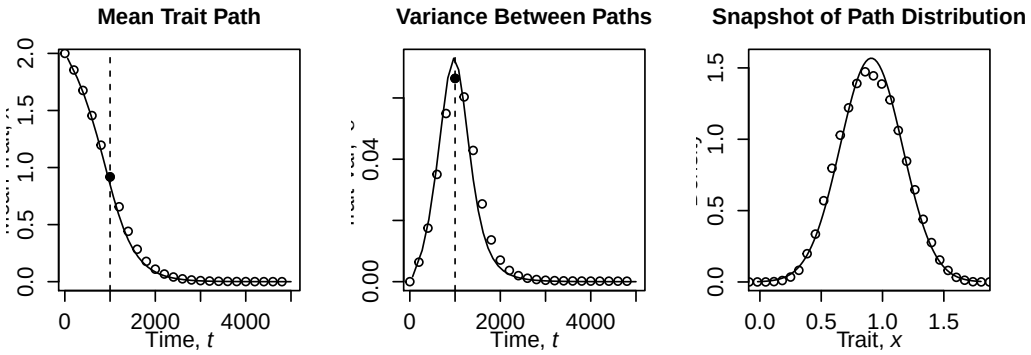

Figure 3. A starting condition of Xo = 2, matching the trajectories seen figure 2 of the original manuscript. Deviations from the canonical path are visible, but the distribution is not yet bimodal.

```
out = logistic(Xo = 3,  ENSEMBLES = 10^5, MAX_TIME = 10000)
```

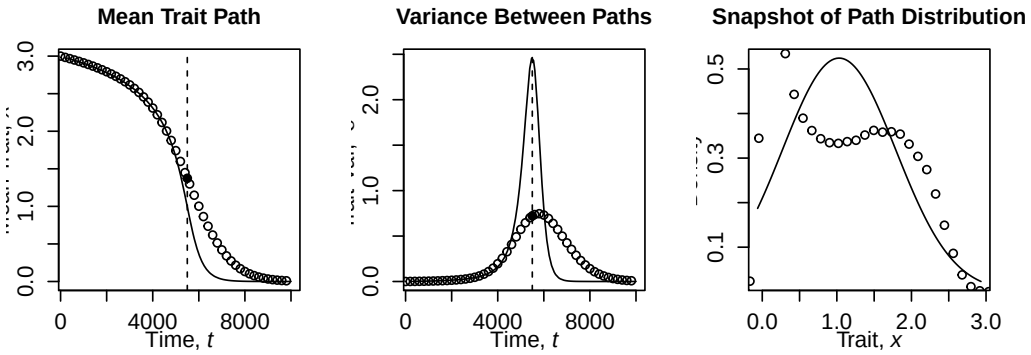

Figure 4. Starting deep in the predicted fluctuation enhancement regime, Xo = 3, the resulting distribution of trajectories is bimodal, just as shown in the original figure 2 of the original manuscript.

On the R package compendium — My original paper acknowledges the use of the R package compendium approach (Gentleman and Temple Lang 2007) for reproducible research, and idea later developed in Marwick, Boettiger, and Mullen (2018). While a decade ago the package as created would pass R's built-in `R CMD check` routine, standards have since evolved and the current implementation would not do so:

- The C methods are expected to bind R's random number generator engines, and not use their own methods, which allows them to coordinate with standard R methods for setting seed and so forth.
- Data should be passed by reference back to R and not serialized to disk (though this can introduce memory constraints).
- It is now also standard practice to register the C routines explicitly.
- `SystemRequirements` field was not provided in the package `DESCRIPTION`. By adding this to list `libgsl`, it is possible to have the package successfully install on the `rhub` (Csárdi and Salmon 2019) testing platforms.

Note that these tighter integration would also prevent the C libraries from running as purely stand-alone implementation. In any event, meeting all the automated checks of a formal package is not an expectation of the compendium approach, which merely borrows some of the package tooling.

A dynamic document using RMarkdown (Xie, Allaire, and Grolemund 2018) for reproducing the present manuscript can be found in the `rescience` sub-directory which I have now added to the GitHub repository housing the original R package, https://github.com/cboettig/fluctuationDomains, git tag `rescience-submission`, now archived with DOI https://doi.org/xxxx.

Conclusions

In conclusion, I have been able to successfully replicate the original results. As the original results were not performed with a fixed random seed, replicated results are statistically comparable but not bit-wise identical. In a scientific sense this is perhaps more useful than replication with a fixed seed. My compendium approach did not install entirely out of the box, as an autoconf configure script first required updating. Ironically, the original version of the code published on a co-author's personal website and linked from the manuscript both still existed and did install out-of-the-box. Reproducing the results

of the paper required some careful reading of parameters listed in the paper and manually knowing what functions to call, as this paper did not include a complete "dynamic document" (Xie 2014, 2015; Xie, Allaire, and Grolemund 2018) approach discussed in Marwick, Boettiger, and Mullen (2018) and also used in this replication paper. I think this experience underscored the critical role dependency management has in frustrating reproducibility: this replication was relatively simple in part due to the low number of dependencies and the technical simplicity of the code. The other take-away for me is how easy it is to leave out small but important steps needed to reproduce a figure when not using the dynamic document approach. Despite having R scripts called "figure1.R" and "figure2.R", these scripts could not simply be run by themselves to create the figures.

References

Boettiger, Carl. 2015. "An Introduction to Docker for Reproducible Research." *ACM SIGOPS Operating Systems Review* 49 (1): 71–79. https://doi.org/10.1145/2723872.2723882.

Boettiger, Carl, Jonathan Dushoff, and Joshua S. Weitz. 2010. "Fluctuation Domains in Adaptive Evolution." *Theoretical Population Biology* 77 (1): 6–13. https://doi.org/10.1016/j.tpb.2009.10.003.

Boettiger, Carl, and Dirk Eddelbuettel. 2017. "An Introduction to Rocker: Docker Containers for R." *The R Journal* 9 (2): 527. https://doi.org/10.32614/RJ-2017-065.

Csárdi, Gábor, and Maëlle Salmon. 2019. *Rhub: Connect to 'R-Hub'.* https://CRAN.R-project.org/package=rhub.

Dieckmann, Ulf, and Richard Law. 1996. "The Dynamical Theory of Coevolution: A Derivation from Stochastic Ecological Processes." *Journal of Mathematical Biology* 34 (5-6): 579–612. https://doi.org/10.1007/BF02409751.

Gentleman, Robert, and Duncan Temple Lang. 2007. "Statistical Analyses and Reproducible Research." *Journal of Computational and Graphical Statistics* 16 (1): 1–23. https://doi.org/10.1198/106186007X178663.

Gillespie, Daniel. 1977. "Exact Stochastic Simulation of Coupled Chemical Reactions." *Journal of Physical Chemistry* 81.25: 2340–61.

Marwick, Ben, Carl Boettiger, and Lincoln Mullen. 2018. "Packaging Data Analytical Work Reproducibly Using R (and Friends)." *The American Statistician* 72 (1): 80–88. https://doi.org/10.1080/00031305.2017.1375986.

Ushey, Kevin. 2019. *Renv: Project Environments.* https://CRAN.R-project.org/package=renv.

Ushey, Kevin, Jonathan McPherson, Joe Cheng, Aron Atkins, and JJ Allaire. 2018. *Packrat: A Dependency Management System for Projects and Their R Package Dependencies.* https://CRAN.R-project.org/package=packrat.

Xie, Yihui. 2014. "Knitr: A Comprehensive Tool for Reproducible Research in R." In *Implementing Reproducible Computational Research*, edited by Victoria Stodden, Friedrich Leisch, and Roger D. Peng. Chapman; Hall/CRC. http://www.crcpress.com/product/isbn/9781466561595.

———. 2015. *Dynamic Documents with R and Knitr.* 2nd ed. Boca Raton, Florida: Chapman; Hall/CRC. https://yihui.org/knitr/.

Xie, Yihui, J. J. Allaire, and Garrett Grolemund. 2018. *R Markdown: The Definitive Guide.* Boca Raton, Florida: Chapman; Hall/CRC. https://bookdown.org/yihui/rmarkdown.