rdmc: an open source R package implementing convergent adaptation models of Lee and Coop (2017)

Silas Tittes[*,1]
[*]Dept. of Evolution and Ecology and Center for Population Biology, University of California, Davis, CA, USA

ABSTRACT

The availability of whole genome sequencing data from multiple related populations creates opportunities to test sophisticated population genetic models of convergent adaptation. Recent work by Lee and Coop (2017) developed models to infer modes of convergent adaption at local genomic scales, providing a rich framework for assessing how selection has acted across multiple populations at the tested locus. Here I present, *rdmc*, an R package that builds on the existing software implementation of Lee and Coop (2017) that prioritizes ease of use, portability, and scalability. I demonstrate installation and comprehensive overview of the package's current utilities.

KEYWORDS R, Software, Convergent Adaptation, Population Genetics, Composite Likelihood

Introduction

Convergent adaptation occurs when natural selection independently orchestrates the evolution of the same set of trait or traits in multiple populations (Takuno *et al.* 2015; Tishkoff *et al.* 2007; Yeaman *et al.* 2016; Losos 2011). Efforts by Lee and Coop (2017) used coalescent theory to develop composite likelihood models to infer which among several competing modes of convergent adaptation best explains allele frequencies at a putatively selected region. These models provide rich statistical information about the inferred adaptive mutation, including its location along the region, the strength of selection, migration rate, age, and its initial allele frequency prior to selection.

To facilitate use of the convergent adaptation models of Lee and Coop (2017), I developed rdmc, an R package implementing their models that was designed to be easy to use, portable, and scalable. In this short manuscript, I provide an overview of the usage and installation of the package, concluding with opportunities for future improvements and expansion to the software.

Methods

Lee and Coop (2017) described three distinct modes of convergent adaption: independent mutations, where two or more populations independently gain the selected mutation; migration, where the mutation occurs once and subsequently migrates to multiple populations prior to fixation; and standing variation, where the mutation was present at low frequency in the ancestral population prior to divergence. The models are composite likelihood-based, where likelihood calculations are made over a gradient of user-chosen input parameters.

[1] Dept. of Evolution and Ecology, University of California, Davis, CA, USA E-mail: stittes@ucdavis.edu

Data requirements

Using *rdmc* requires allele frequency data from at least two populations that have putatively undergone convergent adaptation, as well as at least one population that did not. In addition the package requires an estimate of the per base recombination rate for the region or regions of interest, and an estimate of the effective population size. At the time of writing, all available models in *rdmc* assume a single effective population size. Depending on which modes of convergent adaptation are being investigated, users must also provide vectors of selection coefficients, migration rates, allele frequency ages, and initial allele frequencies prior to selection. The exhaustive list of required inputs and their definitions is given in table 1.

Installation and dependencies

Installation of *rdmc* requires the r package *devtools*. With devtools available, the packaged can be installed and made locally available with the following R commands:

```
devtools::install_github("silastittes/rdmc")
library(rdmc)
```

In addition to *devtools*, *rdmc* depends on several other packages. Namely, *MASS* (Venables and Ripley 2002), *dplyr* (Wickham *et al.* 2020), *tidyr* (Wickham and Henry 2020), *purrr* (Henry and Wickham 2019), *magrittr* (Bache and Wickham 2014), and *rlang* (Henry and Wickham 2020). All dependencies are automatically installed or updated when the installation command above is issued. I encourage users to update to the most recent version of R prior to issuing any of the above commands.

Specifying parameters and input data

For convenience, the original simulated example data generated by Lee and Coop (2017) are provided with the installation and can be loaded with:

```
#returns 'Allfreqs'
data(neutralAlleleFreqs_example)
#returns 'positions'
data(selectedRegionPositions_example)
```

```
#returns 'freqs_notRand'
data(selectedRegionAlleleFreqs_example)
```

The example data consists of 10,000 simulated basepairs from six populations, three of which (with indices 1,3,5) that independently mutated to the selected allele at position 0. Users should note that the simulated positions here take on values between zero and one, but that base pair positions along the chromosomes of empirical data should not be altered prior to fitting the models.

When fitting possible convergent adaptation models, several quantities are reused regardless of which modes of convergent adaptation are of interest. In efforts to minimize computation, all parameters and quantities that are common across models are stored in a single named list generated with the function *parameter_barge()* that can be used when fitting any of the possible models. The list of quantities is generated using:

```
param_list <-
  parameter_barge(
    Ne =  10000,
    rec = 0.005,
    neutral_freqs = allFreqs,
    selected_freqs = freqs_notRand,
    selected_pops = c(1, 3, 5),
    positions = positions,
    n_sites = 10,
    sample_sizes = rep(10, 6),
    num_bins = 1000,
    sels = c(1e-4, 1e-3, 0.01,
             seq(0.02, 0.14, by = 0.01),
             seq(0.15, 0.3, by = 0.05),
             seq(0.4, 0.6, by = 0.1)),
    times = c(0, 5, 25, 50, 100,
              500, 1000, 1e4, 1e6),
    gs = c(1/(2*10000), 10^-(4:1)),
    migs = c(10^-(seq(5, 1, by = -2)), 0.5, 1),
    sources = sel_pops,
    locus_name = "example_locus"
  )
```

where all the arguments are fully described in table 1. This command also determines the gradient of parameter values (namely the arguments, *sels*, *times*, *gs*, and *migs*) that will be used in the likelihood calculations. Depending on which modes of convergent adaptation are being studied, some of these arguments may not be used.

Naturally, features of the input data (the density and amount of variation in the allele frequencies, the effective population size, and the mutation and recombination rates), will impact the model results, and will determine the resolution we have to infer the model parameters. However, the number and density of points along the gradients of these parameters also effect the resolution one has to make inferences. As is described in more detail below, computation time and memory may also become concerns if these gradients are made too large.

Fitting the models

Once the parameter barge is constructed, all models can be fit using this list of quantities as the only input. All of the mode types (neutral, independent mutations, standing variation with and without a source population, migration, and mixed-modes) are coded as separate functions. The neutral, independent mutations, migration, and standing variation modes can be fit, respectively with:

```
neut_cle <- cle_neutral(param_list)
ind_cle <- cle_ind(param_list)
mig_cle <- cle_mig(param_list)
sv_cle <- cle_sv(param_list)
```

Table 1 Description of the arguments used with the function *parameter_barge()*.

Function argument	Description
neutral_freqs	Matrix of allele frequencies at putatively neutral sites with dimensions, number of populations x number of sites.
selected_freqs	Matrix of allele frequencies at putatively selected sites with dimensions, number of populations x number of sites.
selected_pops	Vector of indices for populations that experienced selection.
positions	Vector of genomic positions for selected region.
n_sites	Integer for the number of sites to propose as the selected site. Must be less than or equal to length(positions).
sample_sizes	Vector of sample sizes of length number of populations. (i.e. twice the number of diploid individuals sampled in each population).
num_bins	The number of bins in which to bin alleles a given distance from the proposed selected sites.
sets	A list of population indices, where each element in the list contains a vector of populations with a given mode of convergence. For example, if populations 2 and 6 share a mode and population 3 has another, sets = list(c(2,6), 3). Required for modeling multiple modes. Only required for fitting models with mixed modes. Must be used in conjunction with the "modes".
modes	Character vector of length sets defining mode for each set of selected populations ("ind", "sv", and/or "mig"). Only required for fitting models with mixed modes.
sels	Vector of proposed selection coefficients.
migs	Vector of proposed migration rates (proportion of individuals of migrant origin each generation). Cannot be 0.
times	Vector of proposed times in generations the variant is standing in populations before selection occurs and prior to migration from source population.
gs	Vector of proposed frequencies of the standing variant.
sources	Vector of population indices to propose as the source population of the beneficial allele. Used for both the migration and standing variant with source models. Note: the source must be one of the populations contained in selPops.
Ne	Effective population size (assumed equal for all populations).
rec	Per base recombination rate for the putatively selected region.
locus_name	String to name the locus. Helpful if multiple loci will be combined in subsequent analyses. Defaults to "locus".

Models of mixed modes, where specified populations are modeled under different modes, can be also implemented by modifying the parameter list object in-place. Specifically, mixed modes are constructed by adding the *sets* and *modes* arguments, which groups the population indices according the vector of modes, and specifies which modes are to be used. For example, to fit a model where populations with indices 1 and 3 adapted via standing variation, and population 5 gained the same mutation independently, and another mixed-mode model where populations 1 and 3 adapted via migration, and population 5 mutated independently:

```
#update to mixed-mode
param_list <- update_mode(barge = param_list,
                      sets = list(c(1, 3), 5),
                      modes = c("sv", "ind"))
multi_svind <- cle_multi(param_list)

#update to another mixed-mode
param_list <- update_mode(barge = param_list, sets =
     list(c(1, 3), 5), modes =  c("mig", "ind"))
multi_migind <- cle_multi(param_list)
```

Results and Discussion

Benchmarking

The computation time and memory usage of *rdmc* increases with the complexity of the model and size of the input data used (table 2). The composite likelihood calculations are made over the gradient of input parameters chosen when constructing the parameter barge (code shown above), hence, denser gradients will also have a considerable impact on time and memory. The size of the example data provided gives a realistic sense of memory and time usage for potential empirical data.

Table 2 Bencharking by rdmc model type. Execution time is reported in seconds. Benchmarking was conducted using the R package, *bench* (Hester 2020). Code was executed in an interactive job on the UC Davis Farm HPC (2.00GHz Intel Xeon CPU, 16GB RAM).

model	median time	memory	iterations	garbage collections
neutral	8.47	60.01MB	5	3
independent	11.96	200.9MB	5	4
migration	183.52	2.88GB	5	27
standing variation	553.22	8.65GB	5	103

While most modern laptops are capable of handling the required memory, many users will be interested in genome-wide analysis, where the mode of convergence for many separate regions are of interest. In these instances, cloud or high performance computing environments will be more appropriate. By making *rdmc* a portable and easy to install R package, it is simple to separate these regions as independent jobs to be run in parallel.

Extracting useful quantities and visualization

The functions above all return a data frame containing the complete set of variables generated by all of the possible models, where *NA*s are added as values to variables that are not used for a given mode type. Once the models of interest have finished, the common format of the returned data frames makes it trivial to combine all inferences into a single data frame, which simplifies creation of statistical and graphical summaries:

```
#rdmc loads %>% and full_join
#from mggrittr and dplyr
mergeby <- names(neut_cle)
all_mods <-
  full_join(ind_cle, mig_cle, by = mergeby) %>%
  full_join(., sv_cle, by = mergeby) %>%
  full_join(., multi_svind, by = mergeby) %>%
  full_join(., multi_migind, by = mergeby)
```

With a single data frame containing output from all tested models, many methods of visualization and summary statistics are available in the R ecosystem (R Core Team 2020). For example the maximum composite likelihood parameters across all models can be computed with:

```
#rdmc loads dplyr::group_by() and select()
all_mods %>%
  group_by(model) %>%
  filter(cle == max(cle)) %>%
  select(
    selected_sites, sels,
    cle, migs, model
    )

# A tibble: 5 x 5
# Groups:   model [5]
  selected_sites  sels   cle     migs model
           <dbl> <dbl> <dbl>    <dbl> <chr>
1          0.0017  0.03 3707. NA       independent
2          0.0017  0.03 3404.  0.00001 migration
3          0.0017  0.05 3732. NA       sv
4          0.0017  0.03 3745.  0.00001 sv-sv-ind
5          0.0017  0.03 3604.  0.00001 mig-mig-ind
```

Visualizing the composite likelihood values by genomic position (figure 1) can be done with:

```
neut <- unique(neut_cle$cle)
all_mods %>%
  group_by(selected_sites, model) %>%
  summarise(mcle = max(cle) - neut) %>%
  ggplot(aes(selected_sites, mcle, colour = model)) +
  geom_line() +
  geom_point() +
  xlab("Position") +
  ylab("Composite␣likelihood") +
  theme(legend.position = "n") +
  scale_color_brewer(palette = "Set1")
```

Lastly, one can visualize the likelihood surface with respect to specific parameter gradients, such as selection (figure 1):

```
all_mods %>%
  group_by(sels, model) %>%
  summarise(mcle = max(cle) - neut) %>%
  ggplot(aes(sels, mcle, colour = model)) +
  geom_line() +
  geom_point() +
  ylab("Composite␣likelihood") +
  xlab("Selection␣coefficient") +
  scale_color_brewer(palette = "Set1")
```

Concluding remarks and future developments

rdmc was made to facilitate the use of convergent adaptation models of Lee and Coop (2017). The package is easy to install, and requires only a few lines of code to generate and analyze the output. By making *rdmc* an R package, the code is highly portable and has relatively few, highly maintained dependencies, making it simpler to adopt to different computing systems. Because of its portability and ease of use, *rmdc* also simplifies downstream tasks which facilitates usage at large scales, such as modeling thousand of genomic regions simultaneously on high performance computing resources.

Several elaborations to the currently available utilities in the *rdmc* package could be addded. Since the methods developed

in Lee and Coop (2017), additional models have been developed, including ones that can use putatively selected deletion variation, strong selection, concurrent sweeps, and variation in population size among populations (Oziolor *et al.* 2019). Lee and Coop (2017) also introduced parametric bootstrapping to evaluate support for alternative modes. While not currently incorporated into *rdmc*, future development of the package would include functions to perform bootstrapping. However, for the same reasons mentioned above, *rmc* should facilitate creation and computation of bootstrap replicates in parallel.

Web resources

The source of the package and workflow outlined above are available at https://github.com/silastittes/rdmc. The package is released under GNU General Public License (v3.0). All of the presented analyses were computed on a personal laptop (x86_64, linux-gnu) using R version 3.6.3 (2020-02-29).

The original code associated with Lee and Coop (2017) is available https://github.com/kristinmlee/dmc.

Acknowledgments

I would like to thank Jeff Ross-Ibarra for early reviews and encouragement to write this manuscript, to Kristin Lee, Sivan Yair, and Graham Coop for developing the methods and code that form the foundation of the *rdmc* package, and for giving me their blessing in pursuing the development of the package. Lastly, I thank Felix Andrews for helpfully suggesting the function name, *parameter_barge()*.

References

Bache, S. M. and H. Wickham, 2014 *magrittr: A Forward-Pipe Operator for R*. R package version 1.5.

Henry, L. and H. Wickham, 2019 *purrr: Functional Programming Tools*. R package version 0.3.3.

Henry, L. and H. Wickham, 2020 *rlang: Functions for Base Types and Core R and 'Tidyverse' Features*. R package version 0.4.5.

Hester, J., 2020 *bench: High Precision Timing of R Expressions*. R package version 1.1.1.

Lee, K. M. and G. Coop, 2017 Distinguishing among modes of convergent adaptation using population genomic data. Genetics **207**: 1591–1619.

Losos, J. B., 2011 Convergence, adaptation, and constraint. Evolution: International Journal of Organic Evolution **65**: 1827–1840.

Oziolor, E. M., N. M. Reid, S. Yair, K. M. Lee, S. G. VerPloeg, *et al.*, 2019 Adaptive introgression enables evolutionary rescue from extreme environmental pollution. Science **364**: 455–457.

R Core Team, 2020 *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Takuno, S., P. Ralph, K. Swarts, R. J. Elshire, J. C. Glaubitz, *et al.*, 2015 Independent molecular basis of convergent highland adaptation in maize. Genetics **200**: 1297–1312.

Tishkoff, S. A., F. A. Reed, A. Ranciaro, B. F. Voight, C. C. Babbitt, *et al.*, 2007 Convergent adaptation of human lactase persistence in africa and europe. Nature genetics **39**: 31–40.

Venables, W. N. and B. D. Ripley, 2002 *Modern Applied Statistics with S*. Springer, New York, fourth edition, ISBN 0-387-95457-0.

Wickham, H., 2016 *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York.

Wickham, H., R. François, L. Henry, and K. Müller, 2020 *dplyr: A Grammar of Data Manipulation*. R package version 0.8.5.

Wickham, H. and L. Henry, 2020 *tidyr: Tidy Messy Data*. R package version 1.0.2.

Yeaman, S., K. A. Hodgins, K. E. Lotterhos, H. Suren, S. Nadeau, *et al.*, 2016 Convergent local adaptation to climate in distantly related conifers. Science **353**: 1431–1433.

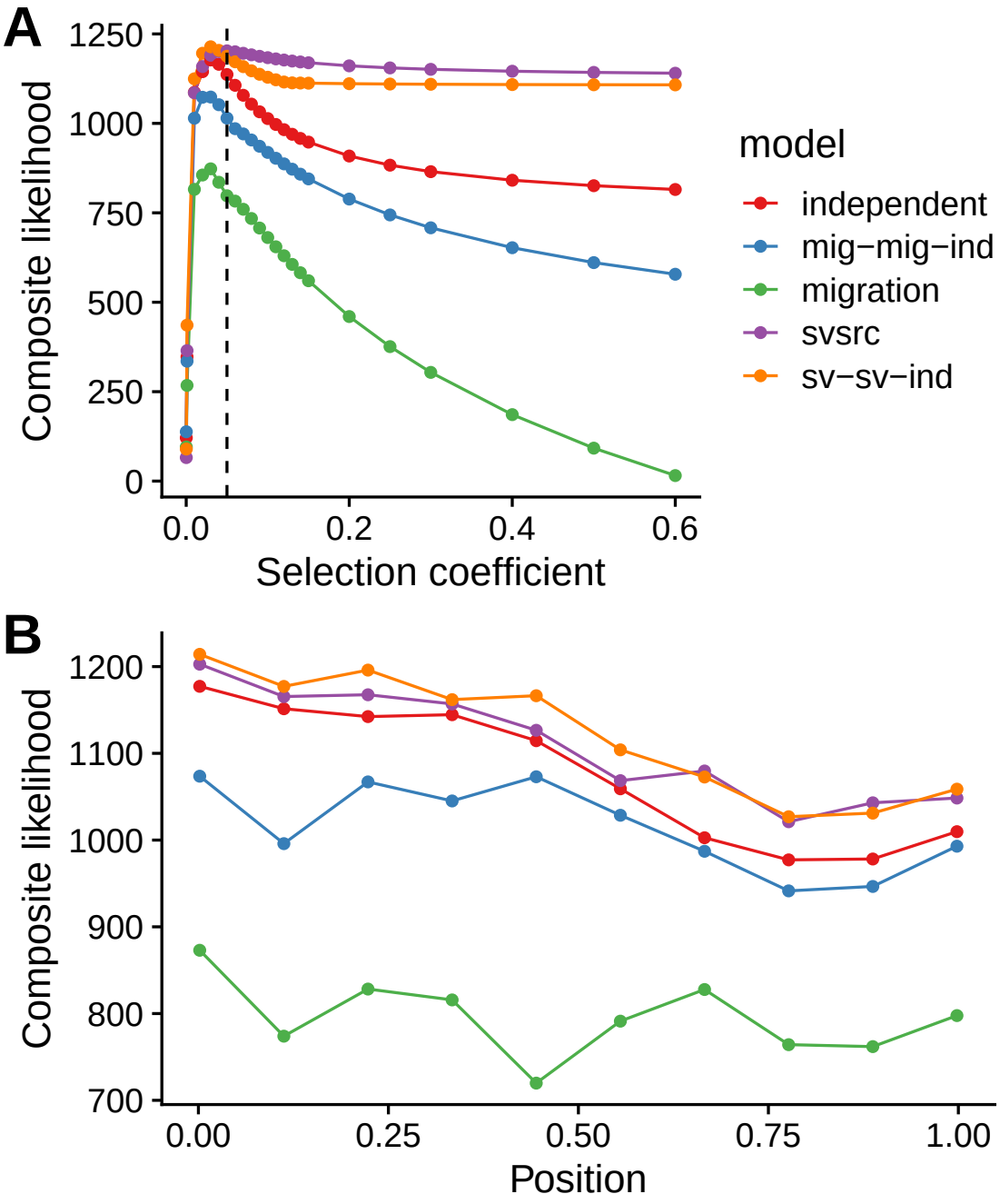

Figure 1 Visualizing rdmc results for several modes. Composite likelihood scores over grid of selection coefficients (A). Dotted line indicates true selection coefficient ($s = 0.05$) the data was modeled under. Composite likelihood score at each of the 10 proposed sites of selection for each model (B). The true selected site was modeled at position 0. The data was simulated as independent mutations in the three selected populations. Visualizations were made using the R package, *ggplot2* Wickham (2016).