bgc

Zach Gompert and Alex Buerkle
University of Wyoming

February 2012 – software v. 1.0, documentation v. 1.0

1 Overview

This manual provides documentation for the `bgc` software. The Bayesian genomic cline models that are implemented in this software are described and analyzed in Gompert & Buerkle (2011a) and Gompert *et al.* (2012a,b) and we expect that these paper will be studied thoroughly prior to using the `bgc` software. This document simply provides information on how to compile and run the software. We assume familiarity with the models, which is essential to make proper use of the software. Depending on the scale of the analysis, this software will be used to estimate thousands of parameters based on large sets of data. This is a reasonably large computational problem that requires some computational skills on the part of the user, including use of UNIX and the ability to write code to produce input files.

We refer to parameters using the same names and symbols as Gompert & Buerkle (2011a). We refer to several specific models that are selected using command-line arguments. The basic model, which was described in Gompert & Buerkle (2011a), assumes known genotypes. The model for genotype uncertainty is described by Gompert *et al.* (2012a). The model for linked loci is described by Gompert *et al.* (2012b). The sequence error model will be described in a forthcoming publication that also describes this software.

2 Obtaining the software

The `bgc` software can be downloaded from `https://sites.google.com/site/bgcsoftware/`. The software is distributed as C++ source code that can be compiled by the user on any Linux or UNIX platform. The software consists of a program that is run from the UNIX command-line and does not have a graphical interface.

The software depends on free and open-source software called the GNU Scientific Library (GSL), so the GSL needs to be installed on the user's system (`http://www.gnu.org/software/gsl/`). This means that the compiled binary of `bgc` requires that the GSL is installed in `/usr/local`, which is the default location. Compiled binaries for the GSL are available as part of many standard Linux distributions. Alternatively, the GSL can be compiled by the user. Users who are compiling `bgc` from the source code should install GSL in the standard location (`/usr/local`) or modify their compilation command to point to the proper location of the library (see below). The software also depends on free and open-source software called HDF5 (`http://www.hdfgroup.org/`). HDF5 is a data model, library, and file format that `bgc` uses for storing and managing MCMC samples.

We assume that users are familiar with moving the compiled binary into a directory in their UNIX `$PATH`, setting permissions for execution, etc., or can obtain assistance from other local users, books or the web.

2.1 Compiling the software

We have compiled the software on Mac OS X 10.6 and 10.7 and several linux distributions using GSL version 1.15 and HDF5 version 1.8.8. Assuming that the GSL and HDF5 have been installed, here are examples of how to compile the `bgc` software using the h5c++ compiler suite (provided with the HDF5 software).

- For Linux or other UNIX systems:
  ```
  h5c++ -Wall -O2 -o bgc bgc_main.C bgc_func_readdata.C bgc_func_initialize.C
  bgc_func_mcmc.C bgc_func_write.C bgc_func_linkage.C bgc_func_ngs.C bgc_func_hdf5.C
  mvrandist.c -lgsl -lgslcblas
  ```
- For Mac OS X:
  ```
  h5c++ -Wall -O2 -o bgc bgc_main.C bgc_func_readdata.C bgc_func_initialize.C
  bgc_func_mcmc.C bgc_func_write.C bgc_func_linkage.C bgc_func_ngs.C bgc_func_hdf5.C
  mvrandist.c -lgsl -lm
  ```
- For Linux or other UNIX systems with non-default locations for GSL:
  ```
  h5c++ -Wall -O2 -L/usr/local/gsl-1.15/lib -I/usr/local/gsl-1.15/include -o bgc
  bgc_main.C bgc_func_readdata.C bgc_func_initialize.C
  bgc_func_mcmc.C bgc_func_write.C bgc_func_linkage.C bgc_func_ngs.C bgc_func_hdf5.C
  mvrandist.c -lgsl -lgslcblas
  ```

3 Input file formats

3.1 Allele counts

The `bgc` software requires input text files that give the observed allele counts for each individual and locus in the parental and admixed populations. The exact format of each file differs a bit depending on whether the genotype-uncertainty model is used.

Parental populations, genotypes known: The data for each locus begins with a line that gives the locus number (e.g., `locus 32`). The genetic region identification line should be followed by one line of data. The data line gives the count of each allele in the population. A separate file is required for each of the two parental populations. A short example with four loci and two alleles is given in Table 1.

Table 1: Sample input data file format for the parental population allele counts with genotypes known.

```
locus 27
1  29
locus 12
0  30
locus 71
1  29
locus 72
3  27
```

Admixed population(s), genotypes known: The data for each locus begins with a line that gives the locus number (e.g., `locus 32`). This line is followed by a line that gives the first population number (e.g., `pop 0`). This population line is followed by a data line for each individual in the population. Missing data is denoted with `-9`. The data line gives the count of each allele for each individual. Additional admixed populations are denoted with additional population lines. As a default `bgc` assumes all diploid loci, however, if `-d 0` is set, the ploidy of each locus should follow the locus number in this file (e.g., `locus 13 1` for haploid or `locus 13 2` for diploid). Only haploid or diploid loci are allowed. A short example with two populations, two loci, and two alleles is given in Table 2.

Parental populations, genotype uncertainty: The data for each locus begins with a line that gives the locus number (e.g., `locus 32`). The locus line is followed by a data line for each individual in the population. Each data line gives the number of sequences (i.e., reads) of each allele for each individual. If no reads are observed for an individual, the data line should simply contain zeros. A separate file is required for each of the two parental populations. A short example with two loci and two alleles is given in Table 3.

Admixed population(s), genotype uncertainty: The data for each locus begins with a line that gives the locus number (e.g., `locus 32`). If the locus-specific error model is implemented, the locus number should be followed by the sequence error probability (e.g., `locus 32 0.0001`; note, the genotype uncertainty model assumes all loci are diploid). This line is followed by a line that gives the first population number (e.g., `pop 0`). The population line is followed by a data line for each individual in the population. Each data line gives the number of sequences (i.e., reads) observed of each allele for each individual. If no reads were observed for an individual, the data line should simply contain zeros. Additional admixed populations are denoted with additional population lines. A short example with one population, two loci, and two alleles is given in Table 4. This example includes locus-specific error probabilities.

3.2 Other files

Genetic map file: This file is only used for the linkage model. The file gives the genomic location (chromosome and position) of each locus. Each row gives the data for one locus

Table 2: Sample input data file format for the parental population allele counts with genotypes known.

```
locus 27
pop 0
1  1
0  2
2  0
⋮  ⋮
0  2
0  2
1  1
pop 1
1  1
0  2
0  2
⋮  ⋮
0  2
1  1
0  2
locus 12
pop 0
1  1
⋮  ⋮
1  1
pop 1
1  1
⋮  ⋮
2  0
```

Table 3: Sample input data file format for the parental population allele counts with genotype uncertainty.

```
locus 27
1  3
0  0
2  1
⋮  ⋮
0  3
0  1
1  0
locus 12
0  0
0  0
0  2
⋮  ⋮
0  1
0  3
0  1
```

Table 4: Sample input data file format for admixed population(s) allele counts with genotype uncertainty.

```
locus 27
pop 0
0  0
0  4
0  1
⋮  ⋮
2  0
0  3
0  0
locus 12
pop 0
1  1
3  0
⋮  ⋮
1  6
1  0
```

Table 5: Sample input data file format for a genetic map, which is used with the linkage model.

```
0   1  0
1   1  0.1
2   1  0.2
3   1  0.3
4   1  0.4
5   1  0.5
6   1  0.6
7   1  0.7
8   1  0.8
9   1  0.9
10  1  1.0
0   2  0
1   2  0.1
2   2  0.2
3   2  0.3
4   2  0.4
5   2  0.5
6   2  0.6
7   2  0.7
8   2  0.8
9   2  0.9
10  2  1.0
```

and the loci must be in the same order as they are in the allele count files. Specifically, each line should give the locus number, chromosome number (start with 1 and number them consecutively), and the location (this can be in kb or Morgans, just adjust the maximum distance accordingly). A short example with two chromosomes is given in Table 5.

4 Command line arguments

Filenames and MCMC parameters can be specified and adjusted using command line arguments. These command line arguments are defined in Table 6. Many of these arguments have default values, which are used when alternative values are not supplied. Default values are shown in square brackets.

Table 6: Command line arguments that are used by `bgc`.

`-a` Infile with genetic data for parental population 0.
`-b` Infile with genetic data for parental population 1.
`-h` Infile with genetic data for admixed population(s).

`-M` Infile with genetic map, only used for ICARrho model.

`-F` Prefix added to all outfiles.

`-O` Format to write MCMC samples: 0 = HDF5, 1 = ascii text, 2 = HDF5 and ascii text [default = 0].

`-x` Number of MCMC steps for the analysis [default = 1000].

`-n` Discard the first n MCMC samples as a burn-in [default = 0].

`-t` Thin MCMC samples by recording every nth value [default = 1].

`-p` Specifies which parameter samples to print: 0 = print log likelihood, alpha, beta, and hybrid index, 1 = also print precision parameters, 2 = also print eta and kappa [default = 0].

`-q` Boolean, calculate and print cline parameter quantiles [default = 0]. Cline parameter quantiles can be used to designate outlier loci.

`-i` Boolean, calculate and print interspecific-heterozygosity [default = 0]. This option should only be turned on if most loci exhibit fixed or nearly fixed differences between the parental species.

`-N` Boolean, use genotype-uncertainty model [default = 0]. This model should be used with next-generation sequence data.

`-E` Boolean, use sequence error model, only valid in conjunction with the genotype-uncertainty model [default = 0]. A single error probability can be specified using the command-line (e.g., `-E 0.001`). Use `-E 1` to provide locus-specific error probabilities, which are included in the infile for the admixed population(s).

`-m` Boolean, use ICARrho model for linked loci [default = 0]. This model requires a physical or genetic map and implements the linkage model. The model uses the weight function from Gompert *et al.* (Eqn. S1.3; 2012b) with $c1 = \frac{1}{4N_L}$, $c2 = 0.6$, and $c3 = 20$. Alternative weight models might be allowed in future versions of the software.

`-d` Boolean, all loci are diploid [default = 1]. Setting this option to 0 (false) is only valid for the known genotype model.

`-s` Boolean, sum-to-zero constraint on locus cline parameters [default = 1]. This constrains all γ (locus affect on cline center, α) and ζ (locus affect on cline rate, β) to sum-to-zero. Population affects are not similarly constrained.

`-o` Boolean, assume a constant population-level cline parameter variance for all loci [default = 0].

`-I` Select algorithm to initialize MCMC [default = 1]. 0 = use information from the data to initialize ancestry and hybrid index, 1 = do not use information from the data to initialize ancestry and hybrid index.

`-D` Maximum distance between loci, free recombination [default = 0.5]. This is only used for the linkage model.

`-T` If non-zero, use a truncated gamma prior for tau with this upper bound [default = 0]. Otherwise use a full gamma prior.

`-u` MCMC tuning parameter, maximum deviate from uniform for proposed hybrid index hybrid index [default = 0.1].

`-g` MCMC tuning parameter, standard deviation for Gaussian proposal of cline parameter gamma [default = 0.05].

`-z` MCMC tuning parameter, standard deviation for Gaussian proposal of cline parameter zeta [default = 0.05].

`-e` MCMC tuning parameter, standard deviation for Gaussian proposal of cline parameters eta and kappa [default = 0.02].

`-v` Display software version.

5 Software output

One or several output files are produced by `bgc` (the specific files generated depend on the `-O` and `-p` arguments). The `-O` arguments determine whether `bgc` writes to plain text or HDF5 output file(s). If HDF5 format is specified, there will be single output file ('mcmcout.hdf5'). The program `estpost`, which is described in a separate section 6, is used to generate posterior summaries from 'mcmcout.hdf5' or to convert MCMC samples for individual parameters to plain text.

If ascii text format is specified each parameter or group of parameters will appear in a different file. All of these files simply contain the MCMC samples. These can be used to estimate various aspects of the posterior probability distribution for each parameter of interest (using for example `R`, R Development Core Team 2011). Be aware that some of these files might be quite large (i.e., many GB), and parsing these files using `Perl` or a similar computer language might be necessary prior to summarizing the results in a statistical package. Base names and file descriptions for the text files follow.

LnL_output.txt: Each line in this file contains the log likelihood for a single MCMC step. For this and all other files only post-burnin post-thinning MCMC samples are included.

alpha_output.txt: Each line begins with the locus number (in input order, but simply numbered 0 to N, where N is one less than the number of loci). This is followed by the cline parameter α for each population. Populations are separated by commas. Each line corresponds to a single MCMC step and additional lines give samples for subsequent MCMC steps.

beta_output.txt: This file contains MCMC samples for cline parameter β and follows the format described for alpha_output.txt.

eta_output.txt: This file contains MCMC samples for cline parameter population-effect η and follows the format described for alpha_output.txt.

kappa_output.txt: This file contains MCMC samples for cline parameter κ and follows the format described for alpha_output.txt.

hi_output.txt: This file contains MCMC samples of hybrid index (h). Each line corresponds to a single MCMC step and contains MCMC samples of hybrid index for each individual. The parameter values for each individual are separated by commas and appear in the order that individuals were listed in the input file.

tau_output.txt: This file contains MCMC samples of the cline precision parameter τ_α and τ_β. Each line gives the MCMC sample of τ_α followed by the MCMC sample of τ_β for a single MCMC step. Recall, that precision is $\tau = \frac{1}{\sigma^2}$.

q_gamma_output.txt: This file contains the quantile of each γ cline parameter in the estimated genome-wide distribution. Each line corresponds to a single MCMC step and gives the quantiles for each locus in order. Values are comma separated.

q_zeta_output.txt: This file contains the quantile of each ζ cline parameter in the estimated genome-wide distribution and follows the format described for q_gamma_output.txt.

heterozygosity_output.txt: The file contains MCMC samples of interspecific heterozygosity. Each line corresponds to a single MCMC step and contains MCMC samples of interspecific heterozygosity for each individual. The parameter values for each individual are separated by commas and appear in the order that individuals were listed in the input file.

rho_output.txt: The file contains MCMC samples of the genomic autocorrelation parameter ρ. Each line corresponds to a single MCMC step and contains the value of ρ for that step.

6 Working with HDF5 output

The software `estpost` is used to summarize MCMC output from `bgc` if it is written as HDF5. This software should be downloaded with the `bgc` software and can be compiled with the h5cc compiler suite (provided with the HDF5 software).

- For Linux or other UNIX systems:
```
h5cc -Wall -O3 -o estpost estpost_h5.c -lgsl -lgslcblas
```

- For Mac OS X:
```
h5cc -Wall -O3 -o estpost estpost_h5.c -lgsl -lm
```

The software `estpost` can do three things: (i) generate point estimates and credible intervals for parameters, (ii) generate histograms for posterior samples, and (iii) generate ascii text output of individual files. The HDFview software is available on-line (`http://www.hdfgroup.org/hdf-java-html/hdfview/`) and is also part of the standard HDF5 installation.

6.1 Command line arguments for `estpost`

Filenames and options can be specified and adjusted using command line arguments. These command line arguments are defined in Table 7. Some of these arguments have default values, which are used when alternative values are not supplied. Default values are shown in square brackets.

Table 7: Command line arguments that are used by `estpost`.

`-i` Infile, MCMC results from bgc in HDF5 format.

`-o` Outfile [default = postout].

`-p` Name of parameter to summarize, possibilities include: 'LnL', 'alpha', 'beta', 'eta', 'eta-quantile', 'gamma-quantile', 'gamma-quantile-local', 'hi', 'interspecific-het', 'kappa', 'kappa-quantile', 'rho', 'tau-alpha', 'tau-beta', 'zeta-quantile', and 'zeta-quantile-local'.

`-c` Credible interval to calculate [default = 0.95].

`-b` Number of additional (beyond the burn-in passed to `bgc`) MCMC samples to discard for burn-in [default = 0]. This burn-in is based on the number of thinned samples.

`-h` Number of bins for posterior sample histogram [default = 20].

`-s` Which summary to perform: 0 = posterior estimates and credible intervals, 1 = histogram of posterior samples, 2 = convert to plain text.

`-w` Write parameter identification and headers to file, boolean [default = 1].

`-v` Display software version.

6.2 Output from `estpost`

The output from estpost depends on the summary requested.

Point estimates and CI (`-s 0`): This file contains an optional header row and parameter identification column (parameters are ordered and number as they were ordered in the input files but starting with 0). Each line gives the mean, median, and lower and upper bounds of the specified credible interval (this is an equal-tail probability interval).

Posterior histograms (`-s 1`): This file contains an optional parameter identification column (parameters are ordered and number as they were ordered in the input files but starting with 0). Each row contains paired (i.e., x1, y1, x2, y2, x3, y3 . . .) x (parameter value for the midpoint of a bin) and y (number of MCMC samples in the bin) coordinates to plot a posterior histogram for a parameter.

Convert to ascii text (`-s 2`): This file contains an optional parameter identification column (parameters are ordered and number as they were ordered in the input files but starting with 0). Each line contains the post-burnin MCMC samples for the designated parameter.

7 Example

We provide an example to illustrate the use of `bgc`. The data are in four input files: admixedin.txt (allele count file for admixed population), p0in.txt (allele count file for parental species 0), p1in.txt (allele count file for parental species 1), and map.txt (genetic map) that are included with the software distribution and in the supplementary material. The data set consists of 2010 loci (201 equally spaced loci arrayed along 10 chromosomes), 100 admixed individuals, and 50 individuals each from two parental species. We assumed relative fitness for admixed individuals was determined by a pair of interacting loci. Specifically, we assigned individuals homozygous for ancestry from the same parental species at both loci or species one ancestry at the first locus and species zero ancestry at the second locus a fitness of 1; we assigned all other individuals a fitness of 0.2. This fitness scheme models a Dobzhansky-Muller incompatibility (DMI) with a negative interaction between species zero ancestry at the first locus and species one ancestry at the second locus. We simulated an admixed population of 500 individuals that was foounded 20 generations in the past by hybridization between two species. Each subsequent generation 5% of the gametes in the population were migrants from parental species. We simulated data with variable allele frequencies and limited sequence coverage for each locus and individual. Specifically, we assumed that the number of sequences per individual and locus was distributed U(0, 8) and that the allele frequencies for each species and locus were independent and distributed U(0, 1). These data are similar to genotyping-by-sequencing data (Elshire *et al.* 2011; Parchman *et al.* 2012) from next-generation sequencers and do not yield known genotypes. We estimated model parameters using two independent chains.

We used the following command to analyze this data set (see the description of command line arguments and their defaults in Table 6 for more information). We ran two independent chains. This MCMC analysis required about 15 CPU hours on Macintosh computer with a 3.2 GHz Quad-Core Intel Xeon processor and 24 GB of RAM.

```
bgc -a p0in.txt -b p1in.txt -h admixedin.txt -M map.txt -O 0 -x 50000
-n 25000 -p 1 -q 1 -N 1 -m 1 -D 0.5 -t 5 -E 0.0001 -d 1 -s 1 -I 0
-u 0.04.
```

Note: You might need to adjust the tuning parameters to achieve efficient mixing. Observe how the chains behave by analyzing the output, and verify good mixing before using the results from these analyses in any way.

We evaluated mixing and convergence to the posterior distribution by examining sample history plots in `R` (R Development Core Team 2011). We first converted MCMC samples for a subset of parameters from HDF5 to ASCII text using `estpost`. For example, to convert the log-likelihood and α_0 samples to ASCII text we used the following commands:

```
estpost -i mcmcout.hdf5 -p LnL -o ln1 -s 2 -w 0
```

```
estpost -i mcmcout.hdf5 -p alpha -o a1 -s 2 -w 0.
```

Figure 1 contains plots of the log-likelihood and α_0 as a function of MCMC step, which were generated using the files `LnL` and `alpha` and the `R` functions `read.table` and `plot`. These plots suggest reasonable mixing and the need for only a short burn-in, although longer chains might be required for accurate estimates of credible intervals.

Next, we calculated posterior point estimates and 95% credible intervals for the $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ parameters and cline parameter quantiles using `estpost`:

```
estpost -i mcmcout.hdf5 -p alpha -o a.out -s 0 -c 0.95 -w 0
```

```
estpost -i mcmcout.hdf5 -p beta -o b.out -s 0 -c 0.95 -w 0.
```

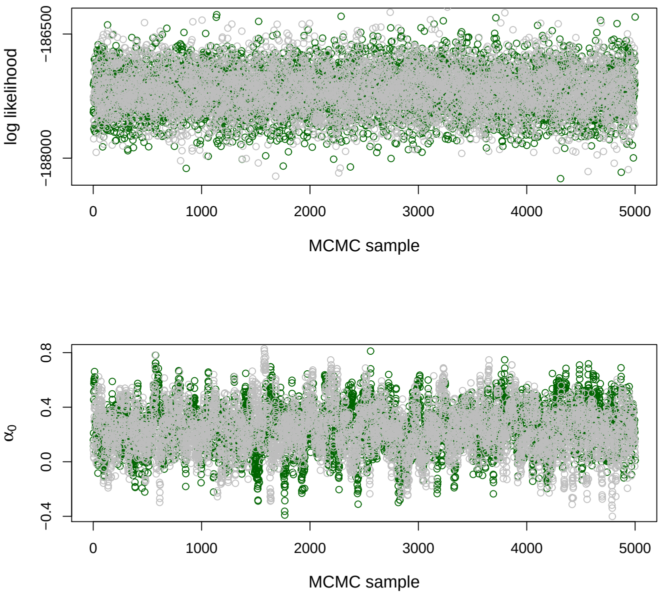

Figure 1: Scatter-plots showing the log likelihood and α_0 as a function MCMC sample. These are the thinned MCMC samples (every 5th iteration retained) following a 25000 iteration burn-in. The combined effective sample size for the two α_0 chains was 604.

```
estpost -i mcmcout.hdf5 -p gamma-quantile -o qa.out -s 0 -c 0.95 -w 0
estpost -i mcmcout.hdf5 -p zeta-quantile -o qb.out -s 0 -c 0.95 -w 0.
```

The parameter arguments `gamma-quantile` and `zeta-quantile` refer to the locus effects for α (γ) and β (ζ; Gompert & Buerkle 2011a). We use the credbile intervals to identify loci with excess ancestry and the quantile estimates to identify outlier loci (Lexer *et al.* 2007; Gompert & Buerkle 2011a; Gompert *et al.* 2012a). The points estimates and credible intervals for the cline parameters and locus effect quantiles can easily be read into `R` or similar software to generate plots. We plot α and β parameters in marker order (which happens to correspond to the order of these simulated loci along chromosomes). See Figs. 2 and 3 for the results. Note using `-w 1` provides headers to make the files more human-readable, but a bit slower to read into `R`.

Regarding the cline parameter α, we found 248 loci with excess ancestry (65% percent of the loci with excess ancestry occur on the two chromosomes involved in the DMI; Fig. 2). We identified fewer α outlier loci than loci with excess ancestry (122 using the median and eight using the bounds of the 95% ETPI as the posterior estimate). But a greater proportion of outlier loci occurred on the two chromosomes involved in a DMI (82% or 100%). We identified only three β outlier loci and did not detect excess ancestry with respect β (Fig. 3). Non-zero estimates of β require ancestry-based linkage disequilibrium and are expected when gene flow and selection are high or with population structure in the hybrid zone.

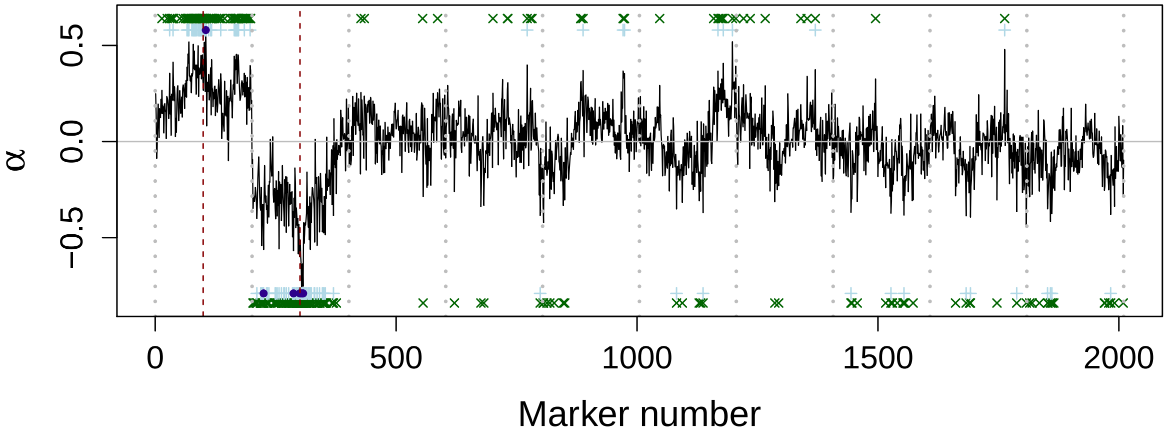

Figure 2: Plot of point estimates (black lines) of cline parameter $\boldsymbol{\alpha}$ along all markers in the simulated genome (201 equally spaced loci on each of 10 chromosomes) from one representative replicate. Loci with excess ancestry are marked with × symbols (i.e., estimates of α_i that deviate significantly from the genome-wide average (specifically, $P[\alpha_i > 0] \geq 0.95$ or $P[\alpha_i < 0] \geq 0.95$). Outlier loci with exceptional introgression are marked with + symbols (with $q_n = 0.05$ and using the median as the posterior estimate of the α_i quantile). Solid circles identify outlier loci using the upper (for $\alpha_i < 0$) or lower (for $\alpha_i > 0$) bounds of the 95% ETPI as the posterior estimate of the α_i quantile. Dashed vertical lines mark the location of a pair of interacting loci that cause a Dobzhansky-Muller incompatibility and dotted vertical lines denote chromosome boundaries.

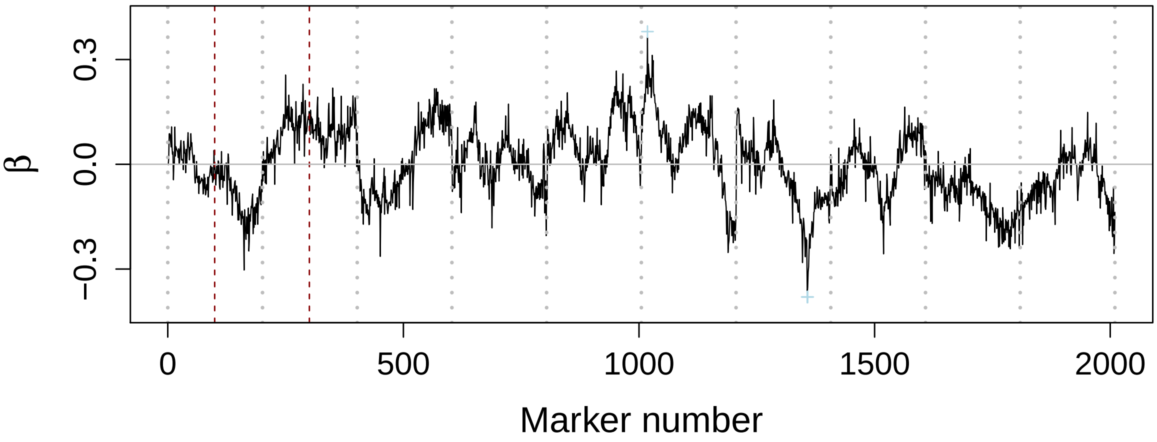

Figure 3: Plot of point estimates (black lines) of cline parameter $\boldsymbol{\beta}$ along all markers in the simulated genome (201 equally spaced loci on each of 10 chromosomes) from one representative replicate. Three introgression outlier loci are marked with + symbols (with $q_n = 0.05$ and using the median as the posterior estimate of the β_i quantile). We did not detect excess ancestry. Dashed vertical lines give the location of a pair of interacting loci that cause a Dobzhansky-Muller incompatibility and dotted vertical lines denote chromosome boundaries.

8 Questions and additional information

We will do our best to respond to requests for additional information and features in the software. Should the need arise, we will post answers to Frequently Asked Questions on the `bgc` web-site: `https://sites.google.com/site/bgcsoftware/`

8.1 Terms of use

The `bgc` software is available for use under the GNU Public License (GPL). This means it is free to use. You are encouraged to download the source code, review it, and improve it. If you use parts of the code in software of your own, you are required to give your users the same rights that were granted to you under the GPL.

Note that under the GPL, the software is distributed without warranty.

We request that you cite Gompert & Buerkle (2011b), Gompert *et al.* (2012a), or Gompert *et al.* (2012b) if you use the software in analyses that appear in print.

References

Elshire RJ, Glaubitz JC, Sun Q, *et al.* (2011) A robust, simple genotyping-by-sequencing (GBS) approach for high diversity species. *PLoS One*, **6**, e19379.

Gompert Z, Buerkle CA (2011a) Bayesian estimation of genomic clines. *Molecular Ecology*, **20**, 2111–2127.

Gompert Z, Buerkle CA (2011b) A hierarchical Bayesian model for next-generation population genomics. *Genetics*, **187**, 903–917.

Gompert Z, Lucas LK, Nice CC, Fordyce JA, Forister ML, A BC (2012a) Genomic regions with a history of divergent selection affect fitness of hybrids between two butterfly species. *Evolution*, **XX**, doi: 10.1111/j.1558–5646.2012.01587.x.

Gompert Z, Parchman TL, Buerkle CA (2012b) Genomics of isolation in hybrids. *Philosophical Transactions of the Royal Society B-Biological Sciences*, **367**, 439–450.

Lexer C, Buerkle CA, Joseph JA, Heinze B, Fay MF (2007) Admixture in European *Populus* hybrid zones makes feasible the mapping of loci that contribute to reproductive isolation and trait differences. *Heredity*, **98**, 74–84.

Parchman TL, Gompert Z, Schilkey F, Benkman CW, Buerkle CA (2012) Genome-wide association genetics of cone serotiny in lodgepole pine. *Molecular ecology*, **XX**, doi: 10.1111/j.1365–294X.2012.05513.x.

R Development Core Team (2011) *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, ISBN 3-900051-07-0.