STAR Protocols

CellPress
OPEN ACCESS

Protocol

Protocol for genetic load analysis in caribou using a modified genomic evolutionary rate profiling

Here, we present a protocol to analyze genetic load in caribou (*Rangifer tarandus*) using a modified version of the genomic evolutionary rate profiling (GERP) program. We describe steps for multi-species alignment including automation of input file production and generating evolutionary conservation scores. We detail procedures for streamlining the extraction of sites of interest from a variant calling format (VCF) file as known derived alleles using three outgroup species to enable the measurement of genetic load.

Publisher's note: Undertaking any experimental protocol requires adherence to local institutional guidelines for laboratory safety and ethics.

Rebecca S. Taylor, Micheline Manseau, Peng Liu, Paul J. Wilson

rebecca.taylor@ec.gc.ca

Highlights

Procedure for using an automated pipeline for whole-genome multi-species alignment

Instructions for running a modified genomic evolutionary rate profiling program

Guidance on interpretation of evolutionary conservation scores

Steps to use a custom script to extract derived alleles from a VCF file using results

Taylor et al., STAR Protocols 6, 103789
June 20, 2025
https://doi.org/10.1016/j.xpro.2025.103789

STAR Protocols

CellPress
OPEN ACCESS

Protocol

Protocol for genetic load analysis in caribou using a modified genomic evolutionary rate profiling

Rebecca S. Taylor,[1,3,4,*] Micheline Manseau,[1] Peng Liu,[1] and Paul J. Wilson[2]

[1]Landscape Science and Technology, Environment and Climate Change Canada, Colonel By Drive, Ottawa, ON K1S 5B6, Canada

[2]Biology Department, Trent University, East Bank Drive, Peterborough, ON K9L 1Z8, Canada

[3]Technical contact

[4]Lead contact

*Correspondence: rebecca.taylor@ec.gc.ca
https://doi.org/10.1016/j.xpro.2025.103789

SUMMARY

Here, we present a protocol to analyze genetic load in caribou (*Rangifer tarandus*) using a modified version of the genomic evolutionary rate profiling (GERP) program. We describe steps for multi-species alignment including automation of input file production and generating evolutionary conservation scores. We detail procedures for streamlining the extraction of sites of interest from a variant calling format (VCF) file as known derived alleles using three outgroup species to enable the measurement of genetic load.

For complete details on the use and execution of this protocol, please refer to Taylor et al.[1]

BEFORE YOU BEGIN

The protocol describes how genetic load was calculated in a study of caribou but can be applied to any species as long as appropriate taxa are used for the evolutionary rate profiling steps. We provide a script to automate the production of multi-species whole genome alignments, a modified version of the program GERP[2] which is more user friendly in non-model organisms, and an R script to extract derived alleles with high evolutionary constraint which can be used to calculate genetic load. For the latter step, we assume that an appropriately filtered variant calling format (VCF) file with high quality genotypes has been made from genomes of the species where genetic load is being measured. This protocol is designed for Linux and was tested on Linux CentOS 7 but may also work on MacOSX and WSL if the necessary software is installed. Our custom R script to extract derived SNPs from a VCF file requires the following R packages: vcfR, tidyverse, data.table, magrittr, and parallel.

KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Deposited data		
Reference genome sequences from multiple species	NCBI or other genome database	https://www.ncbi.nlm.nih.gov/data-hub/genome/?taxon=40674
Software and algorithms		
BBmap 38.86	Bushnell et al.[3]	https://github.com/BioInfoTools/BBMap
Java version 7 or above	Arnold et al.[4]	www.java.com
BWA-MEM 0.7.17	Li[5]	https://github.com/lh3/bwa
Samtools 1.5	Li et al.[6]	https://github.com/samtools/samtools

(*Continued on next page*)

Continued

REAGENT or RESOURCE	SOURCE	IDENTIFIER
BCFtools 1.19	Li[7]	https://samtools.github.io/bcftools/
Genomic evolutionary rate profiling (GERP)	Davydov et al.[2] This manuscript (https://doi.org/10.5281/zenodo.15065760)	https://github.com/BeckySTaylor/GERP/tree/main
R statistical software 4.2.2	R Core Team[8]	https://www.r-project.org/

STEP-BY-STEP METHOD DETAILS

Reference genomes of species used for evolutionary rate profiling

Timing: 2.0 h

1. To find sites which are evolutionarily conserved in caribou, compare the genomes of 48 cervid species.
   a. Download a tree which has appropriate branch lengths reflecting the evolutionary distance between species using TimeTree: http://www.timetree.org/.
   b. Copy and paste the list of the species being used and download the resulting tree in newick (.nwk) format (Figure 1).

   Note: The species must have available reference genomes.

2. Download the reference genome fasta file for each species.

   Note: Here is the link for available mammal genomes publicly available on the National Centre for Biotechnology Information (NCBI) where the genomes used for the caribou work were sourced: https://www.ncbi.nlm.nih.gov/data-hub/genome/?taxon=40674.

Alignment to the caribou reference genome and production of input alignment files

Timing: Using 32 cores, 1 h for indexing reference genome (if needed) and ~45 min per species included, so 37 h for 48 species. The memory requirement may vary with the number of species included but ran with 10 GB available for the 48 species

3. To identify sites which are evolutionarily conserved in caribou, align all species to our caribou reference genome and create a fasta file per scaffold containing the sequence for each of the species (a multi-fasta alignment) which is used as input for the GERP analysis.

   Note: We have created a script named 'Pipeline_input_alignment_GERP.sh' to run all of the steps with only five parameters needing to be input at the start, and is available on GitHub (https://github.com/BeckySTaylor/GERP/tree/main).

   a. Make a directory where you wish to run the script and put all of the downloaded reference genomes into that directory.
   b. Within the same directory, make a simple control file which contains the name of each genome file, followed by the species name after a tab space.

   Note: If you wish to do a test run to check everything works, given how long it takes with a full dataset, we recommend running the pipeline using only a few species first to ensure the correct outputs are made.

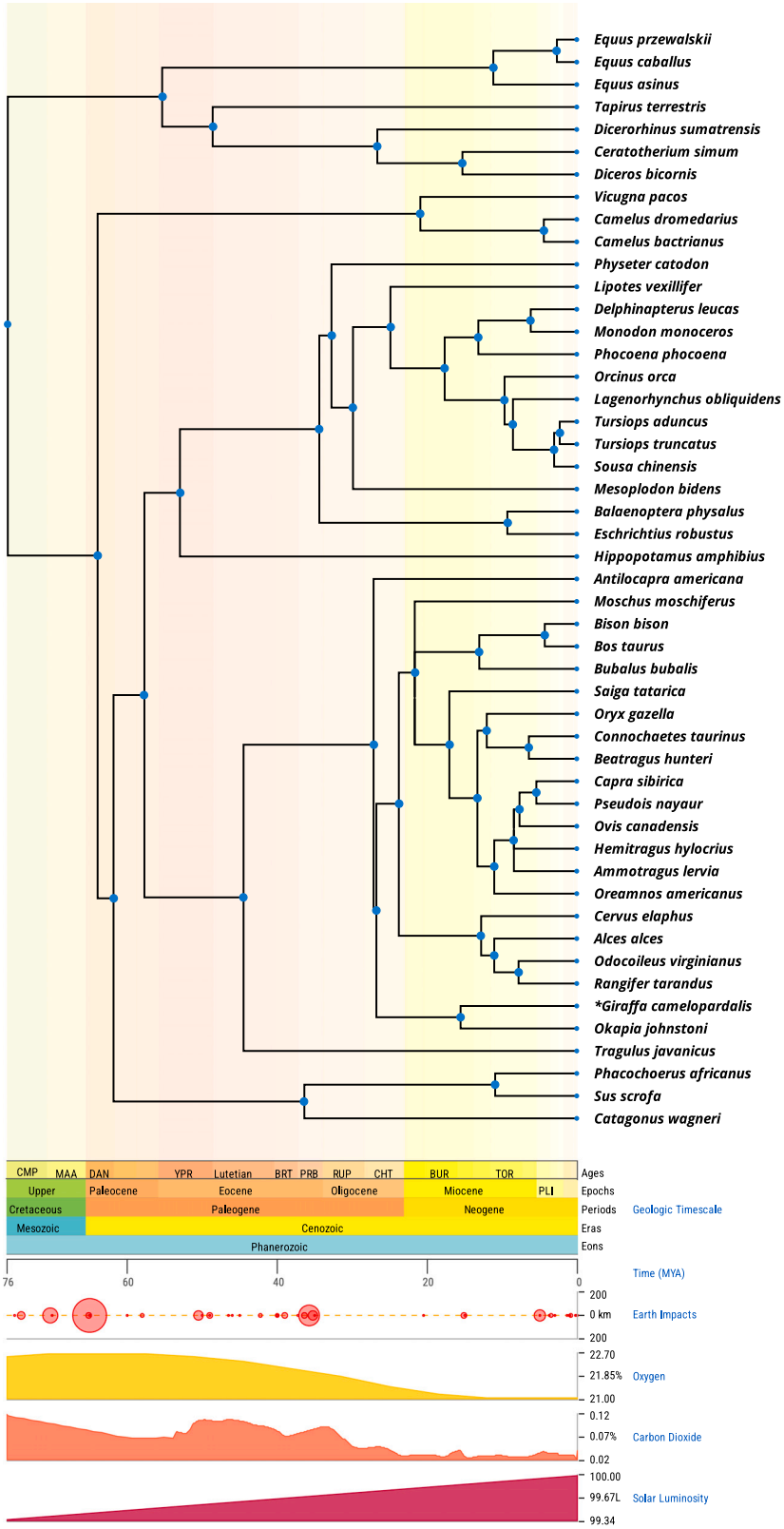

Figure 1. The TimeTree of the species used for the GERP analysis

```
# Example control file, ensure not to use a space in the species name

GCF_002863925.1_EquCab3.0_genomic.fasta Equus_caballus

GCF_002288925.2_Delphinapterus_leucas.fasta Delphinapterus_leucas

GCF_011762595.1_Tursiops_truncatus.fasta Tursiops_truncatus
```

4. Run the bash script provided on the GitHub page, for which we have provided explanations of the settings for each step below.
   a. At the top within the 'Analysis Options' section, input five settings.
      i. Change the path to the directory where the analysis will run and where the genomes and control file are stored.
      ii. Input the name of the control file.
      iii. Input the name of the reference genome fasta file that will be used (for your focal species).
      iv. Input the number of chromosomes (or scaffolds) for your reference genome that you wish to use.
      v. Input how many cores will be available for the analysis.

   Note: If you wish to use the caribou reference genome, which is the example included below, you can remove the '#' where indicated to download and unzip it. If you already have a bwa indexed version of the focal reference genome you can also put those files into the directory and put a '#' next to the 'bwa index' line to save run time.

```
> bash Pipeline_input_alignment_GERP.sh
```

5. We recommend keeping the other parameters in the script unchanged but adjust if desired.
   a. Within the script, 'step 2.2' generates fastq files from the downloaded genomes using the reformat.sh script included with BBmap.
      i. qfake - quality value used for fasta to fastq reformatting.
      ii. fastareadlen - break fasta files into reads of at most this length.
      iii. qout - ASCII offset for output quality, 64 for Illumina.
      iv. addcolon - append ' 1:' and ' 2:' to read names, if not already present.
      v. trimreaddescription - trim the names of reads after the first whitespace.
      vi. int - determines whether input file is considered interleaved.
   b. Within the script, 'step 2.3' aligns to the reference genome and generates a BAM file, removes reads aligning to more than one genomic location as well as supplementary reads.
      i. t - number of threads to use.
      ii. B - mismatch penalty.
      iii. O - gap open penalties for deletions and insertions.
      iv. F will remove reads with a particular flag, in this case the flag 2048 will remove supplementary reads.
      v. b - ensure output is in BAM format.
      vi. q - remove reads below a chosen quality, removing very low-quality reads here should remove reads that are not uniquely mapped.
      vii. h - keep the header row in the output file.
   c. Within the script, 'step 2.4' converts each bam into a fasta file, splits into one fasta file per scaffold, and renames the header line of each resulting fasta file to contain the species name instead of the scaffold name, otherwise they are all the same across species.
      i. f - desired output format.
      ii. min-MQ - minimum mapping quality.
      iii. min-BQ - minimum base quality.
   d. Within the script, 'Step 3' concatenates each of the desired scaffolds and reformats into one multi-alignment fasta file per scaffold, containing all of the species one per line.

Note: Scaffold one is number 0, scaffold 2 is number 1 etc. The resulting files are used as input for the GERP analysis.

Run GERPcol function in GERP++

Timing: 40 min per scaffold

6. Calculate the transition/transversion ratio from the VCF file of the species where genetic load is being measured using BCFtools so we can use this parameter within GERPcol.

```
> bcftools stats Input_VCF.vcf.gz
```

Note: The default transition/transversion rate in GERPcol is 2. For our caribou genomes the rate is 2.06 (this value should be between 2.0 and 2.1 in a mammal species).[4]

7. Calculate the appropriate input parameter for the branch lengths which should be substitutions per site.

Note: As the newick file downloaded from TimeTree has the tree branch lengths in how many millions of years they have been separated (e.g., 4 for 4 million years), the numbers need to be scaled to ensure they are substitutions per site. Given that the average mammal mutation rate is 2.2×10^{-9} per base pair per year,[3] we can calculate the input parameter to be 0.0022 mutations per million years on average at a site. If working on a different taxonomic group, a different value should be used.

8. Run the GERPcol function on each scaffold to output the rejected substitution (RS) scores for evolutionary conservation at each site.

Note: It is best to exclude the focal species, here caribou, from the analysis as this can lead to biases. However, this leads to missing data in the alignment which made it difficult to interpret the output files using the original version of the program which didn't print which site the score pertains to. Our modified version of the code for the gerpcol function makes the outputs more user friendly and helps with extracting derived alleles of the evolutionarily conserved sites from the caribou genomes in VCF format, the important next step for getting a measure of genetic load in our re-sequenced individuals.

   a. Download the modified version of the program as a tar file (named 'gerp_modified.tar.gz') from: https://github.com/BeckySTaylor/GERP/tree/main.
   b. Run the GERPcol function for each scaffold.
      i. Specify the sister species used for the analysis, as named in the alignment file, as the first species listed after '-e'.
      ii. Specify the other two outgroup species, listed with a dash (–) in between. It will print the alleles at all positions for the other two specified species as well, but this is purely for the next step to extract derived alleles where we want to compare against three outgroup species (see below).

Note: Our modified script prints out the position for each score, as well as the allele for three specified sister species to enable the detection of derived alleles in the focal species, caribou (see below). Here we output the alleles for the white-tailed deer, the moose, and the red deer, the closest species to caribou in the multi-species alignment, used to determine the ancestral alleles for each site.

```
#Decompress the downloaded GERP tar file in desired location, and then compile

  >cd gerp

  >make clean && make

#The program is ready to run, here for scaffold 1, the '-a' parameter specifies #that the input alignment is in mfa format

  >./gerpcol -t TimeTree_species_website.nwk -s 0.0022 -f Scaffold1_GERP_formatted.mfa -a -e Odocoileus_virginianus-Alces_alces-Cervus_elaphus -v -r 2.06
```

Note: The function will output 6 columns of data (including a header row). The first is the position on the scaffold 'Pos', 'NeutralRate' values are the calculated neutral rate for each site which will vary depending on the number of species in your tree with missing data. The 'RS_score' values are the calculated RS scores. A value of −1 indicates that too many species in the alignment have missing data. Then the next three columns are the alleles for the three specific sister species which is to facilitate the next step to extract derived alleles (below). Table 1 shows output for six sites as an example.

⚠ CRITICAL: It is important to note that the range of possible scores from a GERP analysis varies depending on your input species and phylogenetic tree and so the score ranges are different between studies and not directly comparable. The larger the phylogenetic tree (with a longer evolutionary distance between taxa), the higher the possible range of scores. For this study, the maximum RS score is 2.48.

Calculate genetic load

Timing: 15 min per scaffold

9. Split the VCF file by scaffold as we are running the analysis for each separately. This is done using BCFtools, here for scaffold 1.
   a. r - indicates which scaffold you wish to retain in the output.
   b. O - indicates the desired output type, in this case the z will indicate a compressed VCF file as output.
   c. o - indicates the desired name of the output VCF file.

```
> bcftools view -r Scaffold_1 -Oz -o Input_VCF_Scaffold1.vcf.gz Input_VCF.vcf.gz
```

10. Make a control file for the R script.
    a. Put the path to and name of the VCF file on the first line.
    b. Put the path to and name of the corresponding '.rates' file on the second line.
    c. Put the minimum GERP score that you would like to include in the output (and so, for example, setting to 0 will output all positive scores) on line three.
    d. Put the output filename on line 4 (without the file extension).
    e. Put the number of threads that will be used on line five.

```
# Example control file, ensure not to use empty lines

/path/to/file.vcf

/path/to/file.rates
```

```
0

Out_name

8
```

Note: For the caribou analysis, we compared the derived alleles at different score ranges to see if there was evidence of purging the most putatively deleterious alleles (i.e. SNPs at those sites with the highest evolutionary conservation scores). We therefore pulled out all derived alleles, all of those with a positive RS score, and all of those with a score over 2 which is at the top end of the range in this dataset.

11. To get a measure of genetic load in our re-sequenced caribou genomes (filtered into the commonly used VCF format), pull out those derived alleles with RS scores of interest.
    a. Run the custom R script, (named 'Derived_alleles_extract.R') and available for download here: https://github.com/BeckySTaylor/GERP/tree/main, specifying the name of the control file.

Note: The script pulls out alleles when they are not found in any of the three sister species and thus are likely to be derived in caribou.

```
> Rscript Derived_alleles_extract.R control_file.txt
```

Note: The beginning part of the R script reads in the VCF file, and removes any non-standard alleles coded with a * in the file as well as ensuring consistency in how the data is formatted. It then reads in the specified 'rates' file as output from GERPcol and corrects the position numbering as it starts at 0 in the file. The script will ensure that it only outputs sites not found in the three outgroup species included in the '.rates' file as we want derived alleles in the species of interest only. The script outputs one file per individual in CSV format listing the position on the scaffold, the 'NeutralRate' and 'RS_score' values, the alleles for the three outgroup species, and the alleles for the caribou from the VCF file (ref and tar), as well as the sample ID (Table 2).

EXPECTED OUTCOMES

The goal of the analysis, for being able to measure genetic load, is to find sites with high conservation scores. The assumption is that a mutation at a site which is highly conserved over the phylogenetic tree (and thus millions of years of evolution) is likely to be deleterious. Finding how many sites which have mutations at these highly conserved locations can give a measure of genetic load.[1] The resulting CSV file gives the derived alleles, based on three outgroup species, at sites which have specified evolutionary conservation scores which can be used to calculate measures of genetic load. For example, for each individual you can add the number of derived alleles in different score ranges, or similarly calculate the average RS score in different score ranges (Figure 2). The CSV files can easily be imported into R and manipulated as desired.

Table 1. Example of a few lines of output from the modified GERPcol function from scaffold 34

Pos	NeutralRate	RS_score	Odocoileus_virginianus	Alces_alces	Cervus_elaphus
48417	0.0805	0.0805	G	G	G
48418	0.471	0.471	A	A	A
48419	0.471	−0.941	T	T	T
48420	0.471	0.471	T	T	T
48421	0.471	0.471	T	T	T
48422	0.471	0.471	T	T	T

Table 2. Example output from the derived alleles with a RS score over 2

Pos	NeutralRate	RS_score	Odocoileus_virginianus	Alces_alces	Cervus_elaphus	ref	tar	Sam
191655	2.36	2.36	C	C	C	C	T	28575
191989	2.28	2.28	C	C	C	C	T	28575
301301	2.26	2.26	G	G	G	G	C	28575
306839	2.4	2.4	C	C	C	C	A	28575

LIMITATIONS

This method detects sites in the genome with high evolutionary conservation scores across millions of years, and we thus assume that derived SNPs at these sites are deleterious. However, we do not know the function of these sites and so the derived alleles are only putatively deleterious. This method uses large computing resources and produces hundreds of large files and so does require access to computational power such as from cloud computing or high-performance computers. It is also likely better to use another independent method to measure genetic load to check for a consistent pattern, for example using a genome annotation method if a high-quality annotation, preferably done using RNA sequencing data, is available.

TROUBLESHOOTING

Problem 1

Reference genome of focal species is highly fragmented with many small scaffolds (this refers to step 4).

Potential solution

Our bash script automates the process and so creating the input files for GERP is simple, although the run time may be affected with many scaffolds. The GERP analysis does have to be run once per scaffold, however.

Problem 2

The bash script fails due to not finding the needed commands (this refers to step 4).

Potential solution

It is important to download all programs listed in the table if not already done so, and it is also important to ensure they are accessible from the specified path which is input into the bash script.

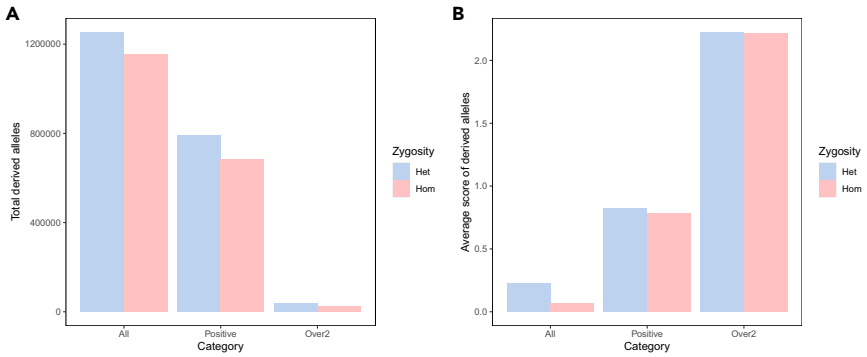

Figure 2. Individual genetic load

Results from one caribou (ID 28575) showing the total numbers of all derived alleles, alleles with a positive score, and those with a score over two (A), as well as the average scores of all derived alleles, alleles with a positive score, and those with a score over two (B).

Problem 3

You already have a multispecies alignment you wish to use as input into GERPcol to run step 8.

Potential solution

It is fine to start our protocol at step 6 if you already have an alignment you wish to use. However, this does need to be split into one alignment per scaffold/chromosome as the output lists the position but not the scaffold information.

Problem 4

Error running modified GERPcol script during step 8 due to dependencies 'not found'.

Potential solution

The GERPcol function is written in C++ and our modified version requires some dependencies to be available. These are usually already installed on Linux systems but if this error occurs ensure dependencies are downloaded: libm.so.6: version 'GLIBC_2.29', libc.so.6: version 'GLIBC_2.34', libstdc++.so.6: version 'GLIBCXX_3.4.20', libstdc++.so.6: version 'CXXABI_1.3.9', libstdc++.so.6: version 'GLIBCXX_3.4.29', and libstdc++.so.6: version 'GLIBCXX_3.4.21'. It will list missing dependencies which need to be installed as an error message when you try to run the script. If the dependencies are there but still do not work, it is possible that the version being used is the issue. If so, a solution is to recompile the code with the commands 'make clean && make', and it is necessary to recompile the program on every new computer you run it on.

Problem 5

R script doesn't read in VCF file correctly for step 11.

Potential solution

We optimized this script using our VCF file which was created using GATK4. We assume it will work from VCF files produced by other programs given that it is a standard file format, but this has not been tested.

RESOURCE AVAILABILITY

Lead contact

Further information and requests for resources should be directed to and will be fulfilled by the lead contact, Rebecca S. Taylor (rebecca.taylor@ec.gc.ca).

Technical contact

Technical questions on executing this protocol should be directed to and will be answered by the technical contact, Rebecca S. Taylor (rebecca.taylor@ec.gc.ca).

Materials availability

This study did not generate new unique reagents.

Data and code availability

All scripts and code are available at: https://github.com/BeckySTaylor/GERP/tree/main (https://doi.org/10.5281/zenodo.15065760).

ACKNOWLEDGMENTS

We are thankful to the Shared Hierarchical Academic Research Computing Network (SHARCNET: www.sharcnet.ca), Compute Canada, and Amazon Cloud Computing for high-performance computing services. We are also thankful to Broderick Crosby for help with writing the bash script. Funding for this research was provided by the Genomic Applications Partnership Program of Genome Canada, Environment and Climate Change Canada, and the Government of Canada's Genomics Research and Development Initiative (GRDI). We are also thankful to three anonymous reviewers for their comments which have helped us to improve the pipeline and manuscript.

AUTHOR CONTRIBUTIONS

R.S.T. developed the pipeline and wrote the manuscript. M.M. secured funding and edited the manuscript. P.L. wrote the scripts for the GERP analysis and ancestral allele detection and edited the manuscript. P.J.W. secured funding and edited the manuscript.

DECLARATION OF INTERESTS

The authors declare no competing interests.

REFERENCES

1. Taylor, R.S., Manseau, M., Keobouasone, S., Liu, P., Mastromonaco, G., Solmundson, K., Kelly, A., Larter, N.C., Gamberg, M., Schwantje, H., et al. (2024). High genetic load without purging in caribou, a diverse species at risk. Curr. Biol. *34*, 1234–1246.e7.

2. Davydov, E.V., Goode, D.L., Sirota, M., Cooper, G.M., Sidow, A., and Batzoglou, S. (2010). Identifying a high fraction of the human genome to be under selective constraint using GERP++. PLoS Comput. Biol. *6*, e1001025.

3. Bushnell, B., Rood, J., and Singer, E. (2017). BBMerge—Accurate paired shotgun read merging via overlap. PLoS One *12*, e0185056.

4. Arnold, K., Gosling, J., and Holmes, D. (2005). The Java Programming Language (Addison Wesley Professional).

5. Li, H. (2013). Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. Preprint at arXiv. https://doi.org/10.48550/arXiv.1303.3997.

6. Li, H., Handsaker, B., Wysoker, A., Fennell, T., Ruan, J., Homer, N., Marth, G., Abecasis, G., and Durbin, R.; 1000 Genome Project Data Processing Subgroup (2009). The sequence alignment/map format and samtools. Bioinformatics *25*, 2078–2079.

7. Li, H. (2011). A statistical framework for SNP calling, mutation discovery, association mapping and population genetical parameter estimation from sequence data. Bioinformatics *27*, 2987–2993.

8. R Core Team (2021). R: A Language and Environment for Statistical Computing (Vienna, Austria: R Foundation for Statistical Computing). https://www.R-project.org/.