PaleoProPhyler, Supplementary Material

Ioannis Patramanis[1], Jazmin Ramos Madrigal[2], Enrico Cappellini[3], and Fernando Racimo[1,3]

[1]Section for Ecology and Evolution, Globe Institute, University of Copenhagen
[2]GeoGenetics Section, Globe Institute, University of Copenhagen
[3]Section for Evolutionary Hologenomics, Globe Institute, University of Copenhagen Center

November 10, 2022

Detailed Overview of Pipelines

In the next few pages you will find a more in depth, step by step, overview of what is running in each module of the pipelines.

Module 1:

Step by step scripts that run:

1. Python3 Script to fetch Ensembl Gene ID for each Protein - Organism Combination
2. Python3 Script to fetch Transcript ID (Ensembl-Canonical and Optional Alternative Isoforms) for each Gene ID
3. Python3 Script to fetch Protein Fasta Sequence for each Transcript ID
4. Python3 Script to fetch Protein Exon and Intron information for each Transcript ID (and create a table for them)
5. Python3 Script to fetch Gene location (corrected for Reference version) for each Transcript ID
6. Python3 Script to combine all FASTA Sequences into datasets (per protein and per organism)

Input of Module 1:

- a TXT file with a list of coded protein names, one per line e.g.:

  AMELX
  ENAM
  AMELY

- Alternatively this list can also contain the names of specific isoforms, or the word ALL to bring all known isoforms e.g.:

  AMELX::AMELX-201
  ENAM::ENAM-202
  AMELY::ALL

- a TXT file with a list of scientific organism names e.g.:

  homo sapiens
  pan troglodytes
  gorilla gorilla

- Alternatively this second list can also contain the names of specific reference versions e.g.:

  homo sapiens GRCh37

pan troglodytes CHIMP2.1.4
gorilla gorilla gorGor3.1

Module 2

This module has 3 alternative input file options:

- CRAM input:
  - Transformed into BAM file using the command `"samtools view -b"`
  - Follows BAM input path
- BAM input:
  - Headers are renamed, e.g. from 'Chr1' to just '1', using the commands `'samtools view'` and `'samtools reheader'`
  - Renamed BAM file is index with `'samtools index -b'`
  - BAM file is split into chromosome BAM files using `'samtools view -b'`
  - Chromosome BAM files are re-indexed using `'samtools index -b'`
  - BAM files are transformed to FASTA file using

    `"angsd -minQ 30 -minMapQ 30 -doFasta 2 -doCounts 1 -basesPerLine 60"`
- VCF input:
  - A reference genome in FASTA format needs to be provided and placed in the

    `'Dataset_Construction/Reference/'` folder
  - The provided reference genome is renamed to the same standard as the BAM files ('Chr1' to just '1')
  - The VCF file is renamed as well, using `'bcftools view'` , `'bgzip -c'` and `'tabix -C -p vcf'`
  - VCF file is converted to FASTA using a combination of `'samtools faidx'` and

    `'bcftools consensus --missing ? -s'`

All different inputs are now in the same format and will follow the same workflow:

1. Custom R script that uses Exon / Intron Locations to splice DNA FASTA
2. Blast Reference Protein onto spliced DNA FASTA with `'makeblastdb -dbtype nucl'` and

   `'tblastn -seg no -ungapped -comp_based_stats F -outfmt 5'`
3. Custom Python3 script to extract blasted / translated protein and output it in FASTA format.
4. Shell commands to merge together individual proteins into bigger datasets (per protein / per individual

Module 3:

Input

The main input of this module is a FASTA file containing both the ancient sequences to be analysed as well as the full reference data set.

- All proteins for all individuals, the format of fasta sequence labels should be: `>SampleName_ProteinName`
- User must also provide the names of the ancient samples in the analysis. Proteins that are not found in any of the ancient samples, will not be included in the analysis.

Workflow

1. The initial FASTA is split into protein-specific FASTAs with a custom R script.
2. Each protein-specific dataset is aligned using Mafft:
3. Modern and Ancient samples are first separated.
4. Modern samples are aligned with

   `'mafft --ep 0 --op 0.5 --lop -0.5 --genafpair --maxiterate 20000 --bl 80 --fmodel'`
5. Ancient samples are merged and aligned with modern ones with the 'mafft-einsi –addlong' option.
6. Aligned dataset is trimmed of completely missing positions using 'trimal -noallgaps'
7. A custom R script generates a small table of statistics for each ancient sample in the dataset after correcting for I/L positions.
8. A custom R script concatenates protein-specific FASTAS into Concatenated FASTA. User has the option to filter proteins under certain coverage. A `'Partition_Helper'` file is also generated, which contains start and stop positions of each protein and can be utilised by NEXUS format phylogenetic software.
9. A custom R script is used to convert FASTA files to PHYLIP format.
10. Multithreaded version of PhyML is run on each separate protein-specific data set using:

```
mpirun -n threads phyml-mpi -i dataset.phy -d aa -b 100 -m JTT -a e -s BEST -v e -o tlr -f
m --rand_start --n_rand_starts 4 --r_seed (random_seed_number) --print_site_lnl --print_trace
--no_memory_check
```

11. Concatenated data set is converted to NEXUS format using 'seqmagick convert –output-format nexus – alphabet protein' , with minor bash command line fixes to ensure proper formatting.

12. Multithreaded version of MrBayes is run on the concatenated NEXUS file using the MrBayes commands:

```
prset aamodelpr = mixed;
mcmc nchains = (number of cores)/4 nruns=(number of cores)/8 ngen = 10000000 samplefreq=100
printfreq=100 diagnfreq=1000;
sumt relburnin = yes burninfrac = 0.25;
sump;
```

and adding the input of the `'Partition_Helper'` file and finally running the file by `"mpirun -np (number of cores/`

Palaeo proteomic hominid reference dataset

Choosing and preparing the list of proteins.

We selected 6 publications cataloging proteins identified in either teeth or bone tissue[6, 3, 2, 28, 14, 25]. From these publications we compiled a list of 1696 unique protein names, which are provided in this file,in the main repository: `\Reference_Protein_List.txt"` We modified this list and used it as an input for Module 1 of the pipeline. From these 1696 proteins, XXX could not be matched to an Ensembl Gene ID. Furthermore, XX Ensembl Gene IDs could not be matched onto an Ensembl Transcript ID. Finally XX Transcript IDs did not have a valid protein entry in Ensembl. This left a total of XXXX proteins that were successfully translated.

Choosing and preparing samples for translation.

Samples were chosen and used from the following publications:

- (1) 1KG high coverage[4].
- (2) Great ape genomes project[26].
- (3) Morphometric, Behavioral, and Genomic Evidence for a New Orangutan Species[24].
- (4) A high-coverage Neandertal genome from Vindija Cave in Croatia[27].
- (5) A high-coverage Neandertal genome from Chagyrskaya Cave[20].

Choosing individuals for the data set:

For publications 2,3 and 5 all available samples were used. For publication 1, only a maximum of 20 individuals from each population were used, resulting in a final X number of individuals. For publication 4 , the individual named 'Mezmaiskaya' was removed from the data set due to a high amount of predicted unique SAPs. We believe that due to the low coverage of the individual, a high number of variants might be miss-called.

Reference Genomes:

We chose to use the human reference genome as the basis for our translations, due to its higher level of annotation. For this purpose, all individuals were mapped onto either GRCh37[9] or GRCh38[29]. Individuals from datasets 1,4 and 5 were already mapped onto a human reference genome. For datasets 2 and 3, raw fastq files were downloaded and mapped onto the GRCh38 human reference. The mapping workflow is provided in the form of a snakemake python script, along with a conda environment containing all software necessary to run the script.

The re-mapped bam files are available at: XXXXXX

Final execution:

Both BAM files and VCF files were then used as input for Module 2, as exemplified by the Tutorial.

Masking: This file should contain two columns and as many rows as necessary. Each row should contain the names of two samples, first the name of a modern sample to be masked and second, separated by a whitespace, the name of an ancient sample. The missing positions of each ancient sample will be masked on top of the same positions of the modern sample, 'masking' it as an ancient sample.

Requirements for running the pipeline

The only true requirement for running any of the 3 modules, is having a Linux machine with Conda installed. All of the required software and packages are downloaded and installed through conda using the provided conda environments. Bellow you can find the full list of software and package used by the pipeline.

OS Requirements:

- Linux

List of Software used by the pipeline:

- Snakemake[22]
- Conda [1]
- Samtools [18]
- BCFtools [17]
- Blast [11]
- Angsd [16]
- Mafft [15, 19]
- Trimal [5]
- Mpirun [21]
- PhyML [12]
- MrBayes [13]
- Seqmagick https://github.com/fhcrc/seqmagick
- R [30]
- Python 3 [32]

R packages

- Bioconductor - ShortRead [23]
- Phyclust [8]
- Stringr [33]

Python packages

- Biopython [10]
- OS package [32]
- Sys package [32]
- Requests package [7]
- RE package [31]

Bibliography

[1] Anaconda software distribution, 2020.

[2] Yahya Acil, Ali E Mobasseri, Patrick H Warnke, Hendrik Terheyden, Jörg Wiltfang, and Ingo Springer. Detection of mature collagen in human dental enamel. *Calcified tissue international*, 76(2):121–126, 2005.

[3] Rodrigo DAM Alves, Jeroen AA Demmers, Karel Bezstarosti, Bram CJ van der Eerden, Jan AN Verhaar, Marco Eijken, and Johannes PTM van Leeuwen. Unraveling the human bone microenvironment beyond the classical extracellular matrix proteins: a human bone protein library. *Journal of proteome research*, 10(10):4725–4733, 2011.

[4] M Byrska-Bishop, US Evani, X Zhao, AO Basile, HJ Abel, AA Regier, A Corvelo, WE Clarke, R Musunuri, K Nagulapalli, et al. High coverage whole genome sequencing of the expanded 1000 genomes project cohort including 602 trios. biorxiv. 2021. *Publisher Full Text*.

[5] Salvador Capella-Gutiérrez, José M Silla-Martínez, and Toni Gabaldón. trimal: a tool for automated alignment trimming in large-scale phylogenetic analyses. *Bioinformatics*, 25(15):1972–1973, 2009.

[6] Gina A Castiblanco, Dorothea Rutishauser, Leopold L Ilag, Stefania Martignon, Jaime E Castellanos, and Wilson Mejía. Identification of proteins from human permanent erupted enamel. *European journal of oral sciences*, 123(6):390–395, 2015.

[7] Rakesh Vidya Chandra and Bala Subrahmanyam Varanasi. *Python requests essentials*. Packt Publishing Ltd, 2015.

[8] Wei-Chen Chen. *Overlapping codon model, phylogenetic clustering, and alternative partial expectation conditional maximization algorithm*. PhD thesis, Iowa State University, 2011.

[9] Deanna M Church, Valerie A Schneider, Tina Graves, Katherine Auger, Fiona Cunningham, Nathan Bouk, Hsiu-Chuan Chen, Richa Agarwala, William M McLaren, Graham RS Ritchie, et al. Modernizing reference genome assemblies. *PLoS biology*, 9(7):e1001091, 2011.

[10] Peter JA Cock, Tiago Antao, Jeffrey T Chang, Brad A Chapman, Cymon J Cox, Andrew Dalke, Iddo Friedberg, Thomas Hamelryck, Frank Kauff, Bartek Wilczynski, et al. Biopython: freely available python tools for computational molecular biology and bioinformatics. *Bioinformatics*, 25(11):1422–1423, 2009.

[11] National Center for Biotechnology Information (US) and Christiam Camacho. *BLAST (r) Command Line Applications User Manual*. National Center for Biotechnology Information (US), 2008.

[12] Stéphane Guindon, Jean-François Dufayard, Vincent Lefort, Maria Anisimova, Wim Hordijk, and Olivier Gascuel. New algorithms and methods to estimate maximum-likelihood phylogenies: assessing the performance of phyml 3.0. *Systematic biology*, 59(3):307–321, 2010.

[13] John P Huelsenbeck and Fredrik Ronquist. Mrbayes: Bayesian inference of phylogenetic trees. *Bioinformatics*, 17(8):754–755, 2001.

[14] Michal Jágr, Adam Eckhardt, Statis Pataridis, and Ivan Mikšík. Comprehensive proteomic analysis of human dentin. *European journal of oral sciences*, 120(4):259–268, 2012.

[15] Kazutaka Katoh and Daron M Standley. Mafft multiple sequence alignment software version 7: improvements in performance and usability. *Molecular biology and evolution*, 30(4):772–780, 2013.

[16] Thorfinn S. Korneliussen, Anders Albrechtsen, and Rasmus Nielsen. ANGSD: Analysis of next generation sequencing data. *BMC Bioinformatics*, 15(1):356, November 2014.

[17] Heng Li. A statistical framework for snp calling, mutation discovery, association mapping and population genetical parameter estimation from sequencing data. *Bioinformatics*, 27(21):2987–2993, 2011.

[18] Heng Li, Bob Handsaker, Alec Wysoker, Tim Fennell, Jue Ruan, Nils Homer, Gabor Marth, Goncalo Abecasis, and Richard Durbin. The sequence alignment/map format and samtools. *Bioinformatics*, 25(16):2078–2079, 2009.

[19] HaiXia Long, ManZhi Li, and HaiYan Fu. Determination of optimal parameters of mafft program based on balibase3. 0 database. *SpringerPlus*, 5(1):1–9, 2016.

[20] Fabrizio Mafessoni, Steffi Grote, Cesare de Filippo, Viviane Slon, Kseniya A Kolobova, Bence Viola, Sergey V Markin, Manjusha Chintalapati, Stephane Peyrégne, Laurits Skov, et al. A high-coverage neandertal genome from chagyrskaya cave. *Proceedings of the National Academy of Sciences*, 117(26):15132–15136, 2020.

[21] Message Passing Interface Forum. *MPI: A Message-Passing Interface Standard Version 4.0*, June 2021.

[22] Felix Mölder, Kim Philipp Jablonski, Brice Letcher, Michael B Hall, Christopher H Tomkins-Tinch, Vanessa Sochat, Jan Forster, Soohyun Lee, Sven O Twardziok, Alexander Kanitz, et al. Sustainable data analysis with snakemake. *F1000Research*, 10, 2021.

[23] Martin Morgan, Simon Anders, Michael Lawrence, Patrick Aboyoun, Hervé Pages, and Robert Gentleman. Shortread: a bioconductor package for input, quality assessment and exploration of high-throughput sequence data. *Bioinformatics*, 25(19):2607–2608, 2009.

[24] Alexander Nater, Maja P Mattle-Greminger, Anton Nurcahyo, Matthew G Nowak, Marc De Manuel, Tariq Desai, Colin Groves, Marc Pybus, Tugce Bilgin Sonay, Christian Roos, et al. Morphometric, behavioral, and genomic evidence for a new orangutan species. *Current Biology*, 27(22):3487–3498, 2017.

[25] Eun-Sung Park, Hye-Sim Cho, Tae-Geon Kwon, Sin-Nam Jang, Sang-Han Lee, Chang-Hyeon An, Hong-In Shin, Jae-Young Kim, and Je-Yoel Cho. Proteomics analysis of human dentin reveals distinct protein expression profiles. *Journal of proteome research*, 8(3):1338–1346, 2009.

[26] Javier Prado-Martinez, Peter H Sudmant, Jeffrey M Kidd, Heng Li, Joanna L Kelley, Belen Lorente-Galdos, Krishna R Veeramah, August E Woerner, Timothy D O'connor, Gabriel Santpere, et al. Great ape genetic diversity and population history. *Nature*, 499(7459):471–475, 2013.

[27] Kay Prüfer, Cesare De Filippo, Steffi Grote, Fabrizio Mafessoni, Petra Korlević, Mateja Hajdinjak, Benjamin Vernot, Laurits Skov, Pinghsun Hsieh, Stéphane Peyrégne, et al. A high-coverage neandertal genome from vindija cave in croatia. *Science*, 358(6363):655–658, 2017.

[28] Cristiane R Salmon, Ana Paula O Giorgetti, Adriana Franco Paes Leme, Romênia R Domingues, Enilson Antonio Sallum, Marcelo C Alves, Tamara N Kolli, Brian L Foster, and Francisco H Nociti Jr. Global proteome profiling of dental cementum under experimentally-induced apposition. *Journal of proteomics*, 141:12–23, 2016.

[29] Valerie A Schneider, Tina Graves-Lindsay, Kerstin Howe, Nathan Bouk, Hsiu-Chuan Chen, Paul A Kitts, Terence D Murphy, Kim D Pruitt, Françoise Thibaud-Nissen, Derek Albracht, et al. Evaluation of grch38 and de novo haploid genome assemblies demonstrates the enduring quality of the reference assembly. *Genome research*, 27(5):849–864, 2017.

[30] R Core Team et al. R: A language and environment for statistical computing. 2013.

[31] Guido Van Rossum. *The Python Library Reference, release 3.8.2*. Python Software Foundation, 2020.

[32] Guido Van Rossum and Fred L. Drake. *Python 3 Reference Manual*. CreateSpace, Scotts Valley, CA, 2009.

[33] Hadley Wickham. *stringr: Simple, Consistent Wrappers for Common String Operations*, 2022. http://stringr.tidyverse.org, https://github.com/tidyverse/stringr.