Read Length Dominates Phylogenetic Placement Accuracy of Ancient DNA Reads

Benjamin Bettisworth[1], Nikolaos Psonis[2], Nikos Poulakakis[2,3,4], Pavlos Pavlidis[1], and Alexandros Stamatakis[1,5,6]

[1]Institute of Computer Science, Foundation for Research and Technology-Hellas (FORTH)
[2]Ancient DNA Lab, Institute of Molecular Biology and Biotechnology (IMBB), Foundation for Research and Technology – Hellas (FORTH)
[3]Natural History Museum of Crete, School of Sciences and Engineering, University of Crete
[4]Department of Biology, School of Sciences and Engineering, University of Crete
[5]Computational Molecular Evolution Group, Heidelberg Institute for Theoretical Studies
[6]Institute for Theoretical Informatics, Karlsruhe Institute of Technology

June 28, 2024

Abstract

One of the central problems facing researchers who analyze ancient DNA (aDNA) is identifying the species which corresponds to the recovered aDNA. Prior analysis of aDNA data normally uses sequence matching tools (such as BLAST) to identify reads obtained from aDNA. However, as the source of aDNA is often an previously unsampled taxon due to the taxon having gone extinct prior to the advent of modern sequencing technology, it is likely the case that there is no exact match in any database. As a consequence tools such as BLAST are of limited use in helping to place a read in a phylogenetic context, I.E. identifying the likely source of a read on a phylogenetic tree. Phylogenetic placement is a technique where a sequence or read is placed onto a specific branch phylogenetic tree. These tools offer a the potential for a much finer resolution when identifying reads. However, phylogenetic placement has primarily only been used to place reads obtained from extant sources. Phylogenetic placement's applicability to aDNA data is complicated by the characteristic pattern of degradation that aDNA undergoes. This characteristic damage is generally not accounted for by popular phylogenetic placement tools, and as a consequence some authors have cast doubt on the potential accuracy of such tools. To understand how the characteristic aDNA damage affects placement phylogenetic tools, implemented a statistical model of aDNA damage as a tool, which we call PyGargammel, that takes sequences applies damage characteristic of aDNA to them. We deploy PyGargammel, along with the existing phylogenetic placement assessment pipeline PEWO, to 7 empirical datasets. With this pipeline, we explore the parameter space of aDNA damage via a grid search in order to identify the factors of aDNA damage which are most impactful. We test 4 leading phlyogenetic placement tools: APPLES, EPA-NG, `pplacer`, and RAPPAS. We find that the frequency of DNA backbone nicks (and consequently read length) is the primary driver of

error for aDNA reads. Additionally, we find that other factors, such as the rate of A to G misincorporations, have a negligible effect on the overall accuracy of phylogenetic placement tools.

1 Introduction

The analysis of DNA obtained from archaeological sites as well as paleontological and sediment samples has changed our understanding of life in the past (Haile et al., 2009; Pedersen et al., 2015; Der Sarkissian et al., 2017). However, the characteristics of ancient DNA (aDNA) samples differ substantially from those of contemporary samples. Due to several degradation processes (including deamination and fragmentation), analyzing aDNA data exhibits more challenges than analyzing DNA obtained from contemporary sources. The DNA repair mechanisms which guarantee that a sample is of high quality become inactive when an organism dies, which allows the DNA to accumulate damage. This accumulated damage often yields poor quality samples.

The damage accumulation rate depends on time, temperature, acidity, and other environmental factors (Sawyer et al., 2012) which complicate the prediction of the *post-mortem* damage level for a given sample. Nonetheless, the damage itself can be described via a straight-forward model, originally proposed by Briggs *et al.* (Briggs et al., 2007). Of the possible damage types DNA accumulates as it ages, the Briggs model of aDNA damage (hereafter, simply the Briggs model) incorporates two major damage types: nicks to the DNA backbone, which shred the DNA into shorter fragments; and point errors caused by cytosine (C) deamination. When cytosine demanites it transforms into uracil (U) and is subsequently read as thymine (T) instead of cytosine (C) during DNA sequencing. This process is commonly known as C to T damage, which we denote as C $\rightarrow$ T. This damage is observed at the 5' end of the ancient sequence reads of double-stranded libraries, as well as at both the 5' and 3' ends of single-stranded libraries. There exists an analogous G to A damage (which we denote as G $\rightarrow$ A) that is only observed at the 3' read ends of double-stranded libraries. This is induced by the genomic library preparation in the wet lab, due to DNA complementarity.

Under the Briggs model, the deamination rate is controlled by the location of the respective base in relation to an overhang. An overhang is a section of the sequence where the complimentary strand has been separated. The bases of the primary strand have therefore been exposed to the environment without the stabilizing presence of the complimentary strand. These overhangs produce so-called single strand regions (in contrast to double strand regions), which exhibit substantially higher rates of deamination due to their unprotected exposure to environmental conditions. For example, Briggs *et. al* found deamination rates in the single strand region to be an order of magnitude higher than in the double strand region. As overhangs necessarily occur at strand ends, the vast majority of C to T damage (and consequently also G to A damage) occurs at the beginning and the end of a read. For an example read of 100 bases, the overhang only needs to be ≈ 3 bases long to exhibit more expected errors than the double stranded region. Many reads are substantially shorter, often only comprising 30 base pairs due to filtering steps after the sequencing process. Hence, due to the short length with respect to the overhang size, most damage occurs at read ends. This induces a characteristic "U" shaped curve of misincorporated point differences as shown in (Briggs et al., 2007; Sawyer et al., 2012).

One major challenge in aDNA data analysis is molecular identification, that is, assigning a DNA read to a taxon. In addition to the processes described above, aDNA samples are also likely to be contaminated by exogenous sources. Determining the organism a read originally belonged to allows researchers to make inferences based on the read context, for example by refining the range, both spatial and temporal, of extinct species (Haile et al., 2009) or uncovering admixture events in archaic hominins, for instance (Slatkin and Racimo, 2016). However, aDNA

degradation results in both, small DNA fragments, and erroneous point differences, yielding identification challenging.

Prior data analyses have attempted to assign aDNA reads via tools such as BLAST (Haile et al., 2009) or Kraken (Wood and Salzberg, 2014). These tools match a short (a)DNA query sequence or read against a (possibly large) database of known sequences. However, sequence similarity based methods only match against known sequences, or equivalently, sequences located at the tips of a corresponding phylogenetic tree. However, if the query sequence is sampled from a heretofore unknown, extinct, or hypothetical species, a similarity-based taxonomic assignment will be imprecise. These methods will indicate a single clade, and possibly a confidence associated with the assignment to that clade. Yet, if multiple clades contain a closely matching reference sequence, similarity-based methods will typically only assign the read to a single clade.

An alternative method for taxonomic identifications is to deploy phylogenetic placement, as implemented in EPA-NG, APPLES, `pplacer`, or RAPPAS (Matsen et al., 2010; Barbera et al., 2019; Linard et al., 2019; Balaban et al., 2020). Phylogenetic placement methods place query sequences into a phylogenetic reference tree. Phylogenetic placement is a technique which places unknown, sampled sequences, called query sequences, into a known reference tree inferred from known/named sequences, called reference sequences. By using a phylogeny, placement is able to match query sequences against hypothetical and unsampled species (i.e., an evolutionary history), as well as against known species. Therefore, placement-based taxonomic assignment is often more specific and accurate than a mere assignment to a potentially taxon-rich clade (Berger et al., 2011). This is because phylogenetic placement will assign the query sequence to a specific branch of the phylogenetic reference tree. For a comprehensive review of phylogenetic placement methods that also includes a plethora of references to empirical data analyses on contemporary DNA data please refer to (Czech et al., 2022).

A recent publication raised the question if phylogenetic placement, which was originally devised for analyzing contemporary DNA sequences, can adequately identify aDNA sequences due to their short read length and characteristic damage (Martiniano et al., 2022). However, a systematic investigation of the effects of aDNA damage on phylogenetic placement accuracy has yet to be conducted. In the following, we therefore systematically investigate the impact of aDNA damage on placement accuracy.

To this end, we have modified the existing phylogenetic placement benchmark pipeline PEWO (Linard et al., 2021) by including a step for injecting aDNA damage into the query sequences. Using this modified pipeline we conduct appropriate accuracy tests using the following placement tools: EPA-NG (Barbera et al., 2019), APPLES (Balaban et al., 2020), `pplacer` (Matsen et al., 2010), RAPPAS (Linard et al., 2019).

2 Methods

To evaluate the effects of aDNA damage on phylogenetic placement accuracy, we initially re-implemented the popular aDNA damage simulator `Gargammel` (Renaud et al., 2017), as an open source Python program which we call `PyGargammel` (available at `https://github.com/computations/pygargammel`). This re-implementation is required to better control the particular type and magnitude of aDNA damage (specifically the parameters of the Briggs Model (Briggs et al., 2007) that is implemented in `Gargammel` and `PyGargammel`), as this is difficult or impossible in `Gargammel`. For both simulators, the input is a FASTA file containing one or more sequences. `Gargammel` simulates reads, and therefore outputs the results as FASTQ files, which need to subsequently be assembled *and* aligned (output as BAM) in order to be suitable for use in PEWO. However, as we intend to isolate the effects of aDNA damage exclusively on placement

Dataset	Taxa	Sites	GC %	Source	Reference
ds01	141	16354	39 %	Aurochs	(Gurke et al., 2021)
ds02	77	932	26 %	Land Snails	(Psonis et al., 2022a)
ds03	36	1784	43 %	Hippos	(Psonis et al., 2022b)
ds04	30	16025	38 %	Elephant	(Psonis et al., 2020)
ds05	512	4686	17 %	Soil	(Mahé et al., 2017)
ds06	3748	3374	22 %	Ocean	(Sunagawa et al., 2015)
ds07	797	2763	25 %	Vaginal Swabs	(Srinivasan et al., 2012)

Table 1: Table of empirical datasets used for experiments. Source is a short description of either the environment the sample was collected from, or the taxonomic clade the samples represent. GC % is the average GC content for all sequences in the alignment.

results, our damage simulator implementation allows to skip these potentially confounding steps. To this end, `PyGargammel` outputs its results as sequences in FASTA format, with an option to preserve alignment with the original, undamaged, sequence. We discuss `PyGargammel` in more detail in Section 2.3.

To facilitate running multiple phylogenetic placement programs on alignments with simulated damage from PyGargammel, we used the placement accuracy assessment pipeline PEWO (Linard et al., 2021). However, executing PEWO with PyGargammel, required several modifications to PEWO which we detail in Section 2.4.

Finally, several damage simulation parameters needed to be chosen for this application. Instead of selecting single parameters or sampling the parameter space uniformly, we chose to perform a grid search over possible damage parameter configurations. These choices are detailed in Section 2.5.

2.1 Data

We investigated the effects of aDNA damage on placement accuracy using seven empirical datasets. The dataset properties, such as taxon number and site count, are provided in Table 1. We sought to include a wide range of datasets, including standard phylogenetic placement applications for environmental DNA (Srinivasan et al., 2012; Sunagawa et al., 2015; Mahé et al., 2017), but also datasets from prior aDNA studies. These datasets comprise for the most part larger animals, such as the Aurochs (Gurke et al., 2021), Land Snails (Psonis et al., 2022a), Hippos (Psonis et al., 2022b), and Elephants (Psonis et al., 2020). These datasets therefore represent a broad spectrum of aDNA identification tasks. We chose to use seven datasets in order to focus on parameter exploration, rather than expending a large number of resources on a shallow exploration of many datasets. Nonetheless, we have sought to include diverse enough datasets so that many typical aDNA data applications are well represented by our results.

In addition to 7 empirical datasets, we also investigated the impact of GC content on placement accuracy in the presence of aDNA damage using 9 simulated datasets. The description of how these datasets were generated, as well as the results from these datasets are described in the Supplemental Material.

2.2 The PEWO Pipeline

Instead of re-implementing an evaluation pipeline for analyzing phylogenetic placement accuracy, we appropriately modified the existing PEWO pipeline (Linard et al., 2021) to also support damage simulation. Therefore, we enhanced PEWO by directly integrating simulation of damaged

sequences via a damage simulator we call PyGargammel. The details of PyGargammel and how it simulates damage are found in Section 2.3.

Given a multiple sequence alignment (MSA) and an associated phylogeny, PEWO performs analysis by

Pruning Pruning the taxa contained in a randomly selected subtree, as well as removing and fragmenting the sequences contained in this subtree from the MSA,

Alignment Realigning the pruned and shortened sequences all at once with HMMER (Eddy, 2011),

Placement Running various placement programs to place the pruned sequences back on the reference tree,

Assessment and Computing and summarizing the placement error with regards to the ground truth placement.

PEWO is designed to investigate the accuracy of phlyogentic placement tools when given somewhat realistic data. To this end, it will fragment the pruned portion of the input MSA into multiple parts. However, the behavior of aDNA fragmentation is very different from how sequence fragmentation is implemented in PEWO. Therefore, we implement our own fragmenting method (detailed in Section 2.3), and disable the sequence fragmentation in PEWO.

2.3 Simulating Ancient DNA Damage

To simulate the characteristic *post-mortem* damage that DNA undergoes as it ages, we implemented the Briggs Model of DNA damage (Briggs et al., 2007) in a tool we call PyGargammel. The primary use case of PyGargammel is to take a FASTA file with sequences and insert aDNA typical damage. This model has four parameters which control the effects and magnitude of aDNA-like sequence damage. Using the same notation as Briggs et al. (2007), the parameters are:

- The probability of nicks $0 \leq \lambda \leq 1$,
- the parameter for the geometric distribution from which the overhang length is drawn $0 < \nu \leq 1$,
- the probability of deamination in single stranded regions $0 \leq \delta_{ss} \leq 1$,
- and the probability of deamination in double stranded regions $0 \leq \delta_{ds} \leq 1$.

The first mechanism in the model determines the probability of breaking (nicking) λ, for each base. These reads then have an overhang with a length that is drawn from geometric distribution with success probability ν. Which side of the reads (5' or 3') the overhang is located on depends on numerous factors which occur during sequencing. However, for the purpose of damage simulation, it is essential that the overhang is present only on the 5' or the 3' side of the read. Therefore, when simulating the overhang, we randomly generate an overhang on the 5' side for 50% of reads, and generate an overhang on the 3' side for the remaining 50% of the reads. Then, deamination damage is applied with a probability that depends on whether the base is in a single or double stranded region. If the base is in the overhang region, then the single stranded deamination rate δ_{ss} is used to determine the deamination probability. If the base is not located in an overhang region, then the double stranded deamination rate δ_{ds} applies.

Note that there are two distinct types of possible deamination damage, C $\rightarrow$ T and G $\rightarrow$ A. C $\rightarrow$ T damage is the direct result of a cytosine being deaminated (this is to say, losing a part of its chemical structure) which results in the base being misincorporated as a T. However, G $\rightarrow$ A damage is *not* a direct result of deamination. Instead, it is a consequence of the genomic library preparation for the sequencing process. More specifically, when the sample is prepared for sequencing, overhangs are repaired, which transforms the G $\rightarrow$ A damage into C $\rightarrow$ T damage on the complimentary strand. In this case, the overhang is repaired with an A instead of a G, as G is the complimentary base to a deaminated C. If the sequence is read from left to right, then *only* C $\rightarrow$ T damage can occur in the read if the overhang is on the left side, and *only* G $\rightarrow$ A damage occurs if the overhang is on the right side.

PyGargammel relies on this model and implements it as follows: It starts by reading a FASTA file and iterates over each sequence in the file by performing the following steps:

1. Iterate through bases and mark nicks with probability λ.
2. Break the sequence into reads at the nick points.
3. Compute overhang lengths on both ends with distribution $\text{Geom}(p = \nu)$.
4. Perform deaminations (C $\rightarrow$ T and G $\rightarrow$ A damage)
   - If the read has an overhang (with a possible length of zero) on the 5' end, and the base is in the overhang region, apply C $\rightarrow$ T damage with probability δ_{ss}. If the base is not in the overhang region, apply C $\rightarrow$ T damage with probability δ_{ds}.
   - Likewise, if the read has an overhang (with a possible length of zero) on the 3' end, and the base is in the overhang region, apply G $\rightarrow$ A damage with probability δ_{ss}. If the base is not in the overhang region, apply damage with probability δ_{ds}.
5. (Optional) Position the read such that it is aligned with its original position in the original sequence.
6. (Optional) Filter sequences that are too short if a minimum read length was supplied.
7. (Optional) If a minimum read count threshold was specified, repeat the process from step 1 accumulating additional reads until the threshold has been achieved.
8. (Optional) If a maximum number of reads was specified, and too many reads were generated, downsample uniformly at random to the provided threshold.

It is important to discuss how PyGargammel handles ambiguous characters and missing data, as empirical MSAs often have ambiguous characters which can affect the results of downstream analysis. As PyGargammel is designed to also operate on already aligned data, there might be gaps present in the sequences to be damaged. The resulting reads need to be still in alignment after processing, so gaps and ambiguous characters are retained as the respective sequence is being processed. This means that gap characters will appear in the resulting damaged reads, and gaps will also affect the damage distribution, as gap characters cannot undergo either C $\rightarrow$ T or G $\rightarrow$ A damage. However, PyGargammel will completely remove reads entirely consisting of either fully ambiguous or gap characters. In addition, when a minimum read length is specified, only non-gap sites are counted to determine read length.

Finally, if a reference alignment is provided, PyGargammel will reject reads which have no non-gap characters in common, a technique known as premasking. We believe this choice is justified, as such a reads would be difficult to align, and therefore difficult to classify by any method.

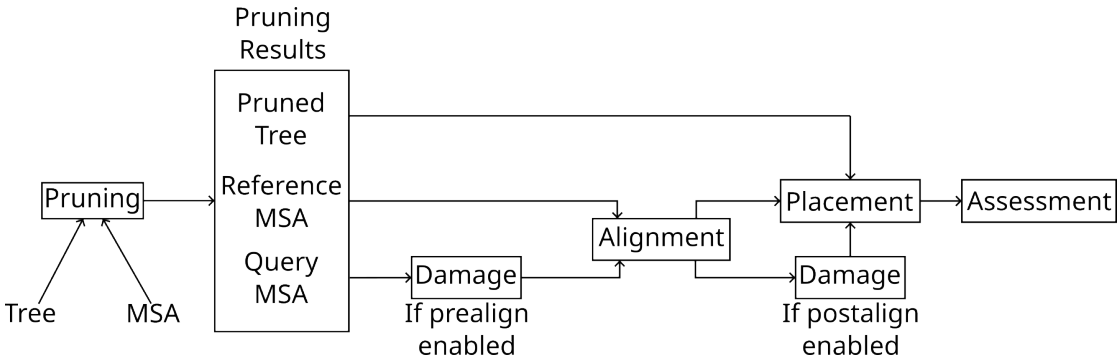

Figure 1: Diagram of the modified PEWO pipeline. The steps "Pruning", "Alignment", "Damage", "Placement", and "Assessment" are described in Section 2.3.

2.4 Modifications to the PEWO Pipeline

We modified PEWO by inserting PyGargammel both before and after the alignment step. Injecting aDNA sequence damage before alignment is more realistic as aligning reads is a necessary step prior to placement for empirical sequences. However, we also inject aDNA damage *after* the alignment stage, so as to isolate the effects of aDNA damage on placement accuracy, without any confounding alignment error. Unless otherwise stated, we report placement accuracy for damage injection *after* alignment. We did conduct some exploratory experiments with damage inserted before alignment, however a thorough investigation of the joint alignment and placement error is beyond the scope of this paper. Specifically, we wish to investigate the case when alignment error is negligible to determine the baseline effectiveness of placement algorithms.

Therefore, the operation order for our analysis pipeline is (unless otherwise stated).

Pruning Pruning the taxa associated with a random subtree, as well as removing the sequences from the MSA,

Alignment Realigning the pruned sequences all together with respect to the remaining reference MSA,

Damage Fragmenting and deaminating the aligned sequences,

Placement Running various placement programs on the pruned sequences,

Assessment and Computing and summarizing the placement error.

We have also prepared a graphical depiction of these steps which is presented in Figure 1. The full pipeline can be found on GitHub at `https://github.com/computations/PEWO`

2.5 Pipeline Parameters

In order to fully investigate aDNA damage impact on phylogenetic placement accuracy, we perform a grid search over the damage simulation parameter configurations. We based the endpoints of this grid search for each parameter on the estimated values from Briggs et al. (2007). The specific values that were tested are provided in Table 2.5. If we increase the number of investigated values per parameter equally, then the number of evaluated points grows with $\wr(n^4)$. This means that the number of points to be evaluated quickly becomes too large to

Parameter	Values
λ	0.0000, 0.0001, 0.0005, 0.0010, 0.0050, 0.0100, 0.0200, 0.0250, 0.030
ν	0.15, 0.30, 0.35, 0.40, 0.75, 1.00
δ_{ss}	0.00, 0.65
δ_{ds}	0.000, 0.015, 0.100

Table 2: Table containing the parameter values used in our experiments. End points for each parameter are based on the values estimated in Briggs et al. (2007).

be computed in reasonable time. We can be more efficient if we investigate the behavior and impact of a single parameter at a time while keeping the remaining parameters fixed. An initial parameter space exploration with a single dataset showed that the nick frequency is the most important factor (w.r.t. its impact on placement accuracy). Hence, we thoroughly investigate the effects of the nick frequency parameter, only. To this end, we sample λ values ranging from 0.0 to 0.03. The value of 0.03 slightly exceeds the value estimated in Briggs et al. (2007), which was found to be $\lambda = 0.024$ for Neanderthal samples. Overall, we assessed nine λ values, with a higher density near 0.0, as that is where placement accuracy is most sensitive to small changes in λ. In Table 3 we show both the theoretical and the actual median read length for each value of λ in order to better illustrate the impact of changes in λ on read length. Noticeably, the largest reductions in both, estimated, and actual median read length stem from values near 0.0. The values differ between the estimated values and the actual values due to finite data, as the maximum length of any read is limited by the original sequence length. Furthermore, we also exclude reads with less than 15 base pairs, which further skews the actual values.

Beyond λ, we experimentally determined that the overhang parameter ν is the second most important parameter, again with respect to its impact on placement accuracy. As before, we base the endpoints of grid values on those listed in Briggs et al. (2007). Since overhangs are modeled as iteratively "growing" with probability ν of the process terminating at the current overhang length, the parameter value associated with no damage is 1.0. This is when the process terminates immediately, resulting in a zero length overhang. Correspondingly, the overhang lengths increase as the parameter value decreases. We tested six values with the most extreme damage being 0.15, which is slightly more extreme than the value estimated in Briggs et al. (2007).

The deamination rates δ_{ss} and δ_{ds} were both found to have a minor damage effect during initial experiments, therefore we only consider two values for δ_{ss} and three values for δ_{ds} in our grid. For δ_{ss} the upper endpoint follows, again, Briggs et al. (2007). On the other hand, the upper δ_{ds} value is a very extreme value of 0.1, which is far greater than the value found in Briggs et al. (2007). We included this extreme value because we intend to investigate the effects of high levels of C $\rightarrow$ T and G $\rightarrow$ A damage on placement accuracy, as this is the unique feature that yields aDNA damage being different from "just" very short reads. In particular, we explore placement accuracy for damage rates that are substantially higher than expected in order to investigate the maximal effect of misincorperation on placement accuracy.

Finally, we implemented two additional constraints on the simulated reads. The first is that a minimum of 10 reads per selected sequence from the reference MSA is generated. This ensures that we have a sufficient number of reads to compute proper summary statistics. The second is a minimum read length of 15 base pairs. This mimics representative analyses of aDNA samples, where reads under a certain length (typically 30 base pairs) are excluded in a prefiltering step.

Tool	Options and Heuristics
RAPPAS	`k` = 7, `omega` = 2.0, `reduction` = 0.99
EPA-NG	`h1`, `g` = 0.99999
`pplacer`	`max-strikes` = 6, `strike-box` = 3, `max-pitches` = 40
APPLES	`OLS`, `MLSE`

Table 3: Tool option and heuristics used for all datasets and parameter values.

2.6 Analysis

We ran PEWO with the following placement tools: RAPPAS(Linard et al., 2019), EPA-NG(Barbera et al., 2019), `pplacer`(Matsen et al., 2010), and APPLES(Balaban et al., 2020). We do not explore different placement program options (e.g., various heuristics) because initial experiments did not exhibit substantial placement accuracy differences as a function of these options. Therefore, in order to also economize on computational resources, we elected to use one set of placement program options for all runs. These options are described in Table 3.

Experiments were performed on a server with 40 x86 CPU[1] cores and 754 GiB of memory. The number of independent subtree prunings and accuracy evaluations (denoted as a "pruning" in PEWO) was set to 10 for all datasets. This is to say, that 10 separate experiments were conducted for each set of parameters on each reference MSA and the results were aggregated. PEWO was run with 40 cores.

To better visualize our experimental results, we fit a linear model augmented with cubic splines for each parameter against the expected node distance (eND). The expected node distance is computed as $\mathrm{LWR}(p) \times d(p)$ where, $\mathrm{LWR}(p)$ is the likelihood weight ratio of the placement, and $d(p)$ is the node distance between the estimated placement and the true placement. We use eND as it better accounts for uncertainty in the specific placement location. We added splines to the model in order to capture the decreasing marginal impact of the damage model parameters. The regression was performed using R, with the command `geom_smooth(method="lm", formula = y ~ splines::ns(x, k))` where k is the number of splines used for the regression on the specific parameter. We used $k := 3$ for the regression on λ and ν, $k := 2$ for the regression on δ_{ds}, and $k := 1$ for the regression on δ_{ss}. We vary the number of splines because the number of parameter values sampled varies between parameters.

3 Results

The regressions describing placement accuracy as a function of different simulation parameter values are displayed in Figure 4. Additionally, the R^2 for each regression is shown in Table 4. We found that the primary factor affecting placement accuracy is the nick frequency parameter λ. As seen in Figure 4 and in Table 4, the only parameter which consistently impacts placement accuracy (as measured by expected node distance) is λ. For all regressions, the variance explained due to variations in λ exceeded that of all other parameters by at least one order of magnitude.

The parameter δ_{ds} had the second most substantial impact on placement accuracy. Nonetheless, it is almost negligible compared to the impact of λ. For all datasets, the R^2 for the regression with δ_{ds} was at least 1 order of magnitude smaller than the R^2 for the regression with λ.

The remaining parameters had a negligible effect on placement accuracy. The R^2 for all models was at least two orders of magnitude smaller for all datasets. To further investigate this, we computed the error rate for generated reads. In Figure 5 we show the error rate as a

[1] specifically a dual socket server with 2 Intel Xeon Gold 6148

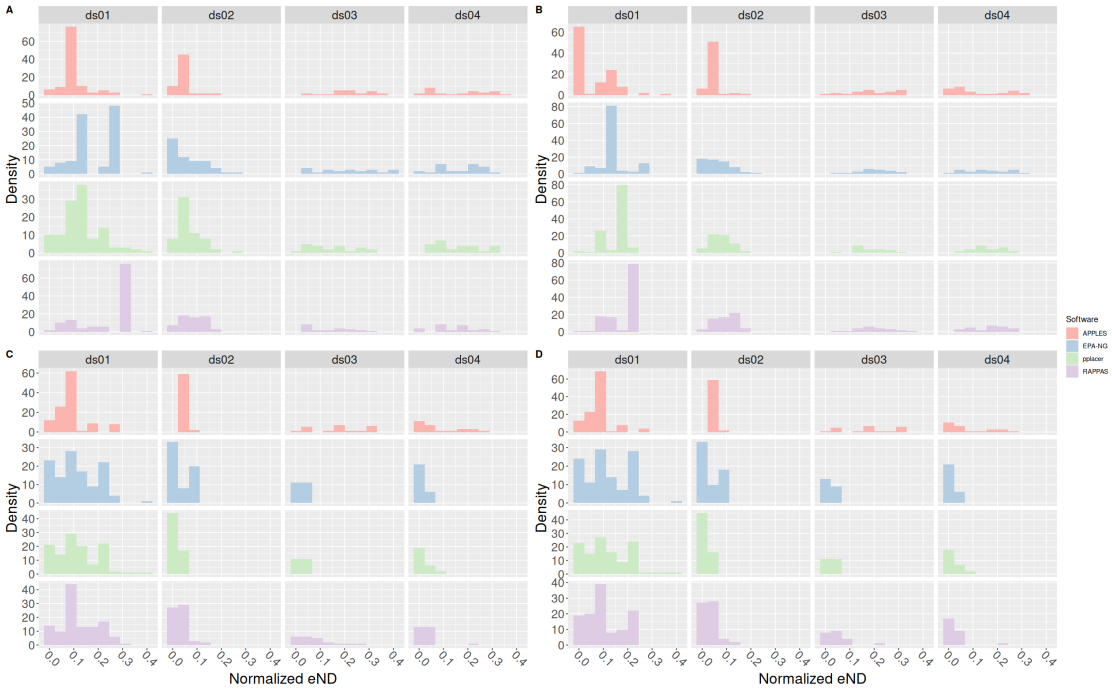

Figure 2: Histogram of normalized effective node distances for datasets ds01, ds02, ds03, and ds04. In addition, each subfigure has been faceted by the respective placement software being used. Top plots (A and B) have nicks enabled ($\lambda = 0.025$), and bottom plots (C and D) have nicks disabled ($\lambda = 0.0$). Plots on the left (A and C) have other forms of damage enabled ($\nu = 0.15, \delta_{ss} = 0.65, \delta_{ds} = 0.015$). Plots on the right (B and D) have other forms of damage disabled ($\nu = 1.0, \delta_{ss} = \delta_{ds} = 0.0$). All plots share the same scale for the x-axis.

histogram for datasets with $\lambda = 0.025$,$\nu = 0.15$, $\delta_{ss} = 0.65$ and $\delta_{ds} = 0.015$. Most reads exhibit a relatively small number of errors, with a median error rate of 4%.

To better illustrate the effect of a high nick rate on placement accuracy we plotted a histogram of eND normalized by tip count in Figures 2 and 3.

4 Discussion

Based on our analysis, the primary factor influencing aDNA placement accuracy is the nick frequency, or equivalently the read length. In Figures 2 and 3, for example, we can see a clear shift to the left between the top and bottom (that is, the plots with $\lambda = 0.025$ vs the plots with $\lambda = 0.0$). This shift to the left indicates an average improvement in placement accuracy. In contrast, we do not observe a similar shift between the left (other damage types enabled) and right (other damage types disabled) plots. This indicates, again, that the primary driver of placement accuracy is nick frequency, and that the effects of deamination errors are minor, confirming our repeated empirical observations that likelihood-based models as implemented in EPA-NG are generally robust with respect to noise and sequencing error.

One way to ameliorate the placement error from aDNA damage is to incorporate the aging process as an additional model process. Currently, none of the placement tools tested here do

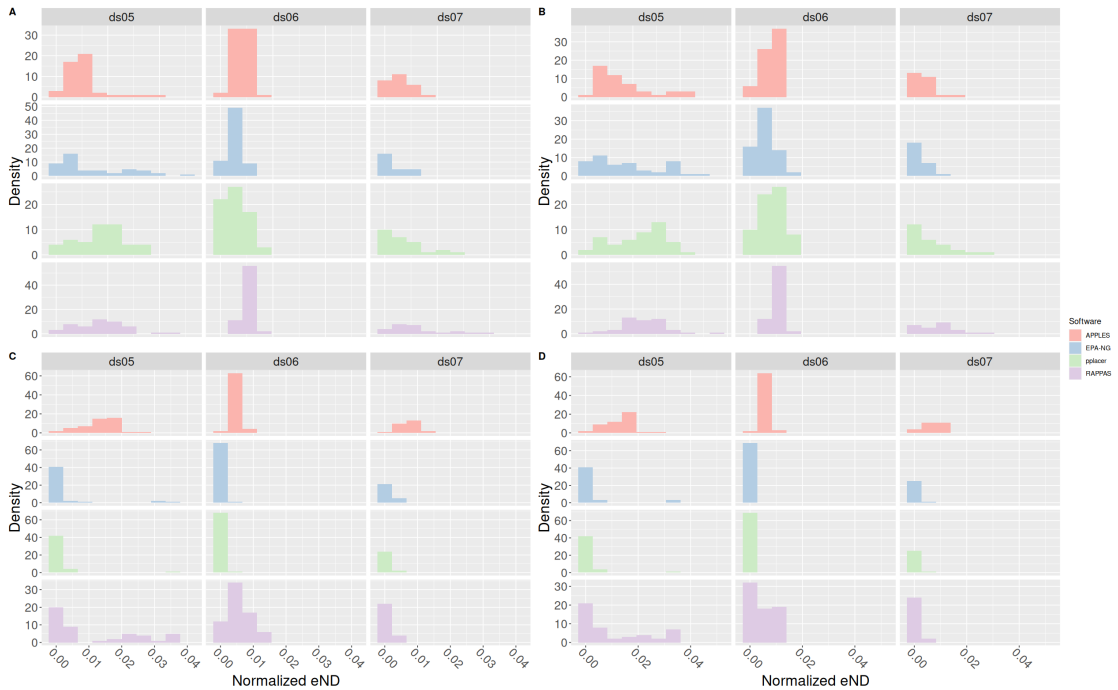

Figure 3: Histogram of normalized effective node distances for datasets ds05, ds06, and ds07. In addition, each subfigure has been faceted by the placement software used. Top plots (A and B) have nicks enabled ($\lambda = 0.025$), and bottom plots (C and D) have nicks disabled ($\lambda = 0.0$). Plots on the left (A and C) have other forms of damage enabled ($\nu = 0.15, \delta_{ss} = 0.65, \delta_{ds} = 0.015$). Plots on the right (B and D) have other forms of damage disabled ($\nu = 1.0, \delta_{ss} = \delta_{ds} = 0.0$). All plots share the same scale for the x-axis.

Dataset	R^2			
	λ	ν	δ_{ss}	δ_{ds}
ds01	0.0805	1.388×10^{-4}	9.199×10^{-6}	2.125×10^{-3}
ds02	0.1415	1.687×10^{-5}	1.896×10^{-5}	3.590×10^{-4}
ds03	0.1925	1.557×10^{-4}	2.973×10^{-5}	1.358×10^{-3}
ds04	0.1996	4.426×10^{-5}	1.896×10^{-5}	1.020×10^{-5}
ds05	0.1498	1.099×10^{-5}	3.374×10^{-6}	2.613×10^{-3}
ds06	0.2006	1.101×10^{-5}	3.147×10^{-5}	1.648×10^{-4}
ds07	0.1007	5.732×10^{-5}	1.132×10^{-4}	5.227×10^{-3}
All	0.0610	5.408×10^{-5}	2.230×10^{-7}	2.537×10^{-4}

Table 4: Table of R^2 values for the regressions in Figure 4. Additionally shown is the R^2 for the regression with all datasets included. Values were computed with `lm(e_nd ~ splines::ns(`p`, `k`))` where p is one of the parameters, and k is the number of splines used for that parameter's model. We used $k := 3$ for λ and ν, $k := 2$ for δ_{ds}, and $k := 1$ for δ_{ss}.

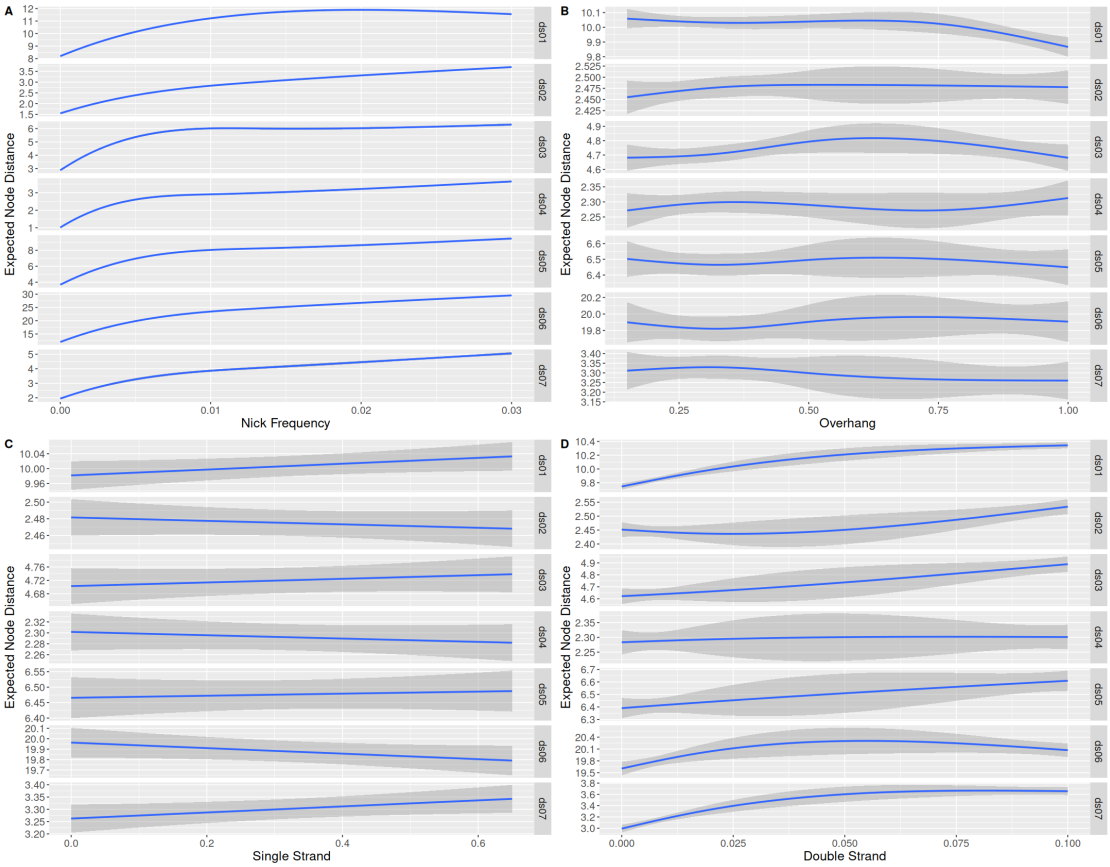

Figure 4: Plot of regressions against parameters for each dataset. Please note that the y scale varies for each dataset. Each regression is against the expected node distance (eND). The subfigures **A**, **B**, **C**, and **D** plot λ, ν, δ_{ss}m and δ_{ds} against eND, respectively. Plot generated using `ggplot2` with `geom_smooth(method="lm", sd=TRUE, formula = y ~ splines::ns(x,` k`))`, where k varies with the dataset. $k = 3$ for plots **A** and **B**, $k = 2$ for plot **D**, and $k = 1$ for plot **C**.

λ	Median Read Length Estimated	Actual
0.0000	∞	3409
0.0001	6931	3388
0.0005	1385	1309
0.001	692	736
0.005	138	165
0.01	68	91
0.02	34	53
0.025	27	45
0.03	22	40

Table 5: Table of estimated and actual read lengths based on the Briggs model. The median is computed for each dataset. Discrepancy between the estimated median and the actual median is due to: the finite length of the data; and the minimum read length specified as part of the pipeline.

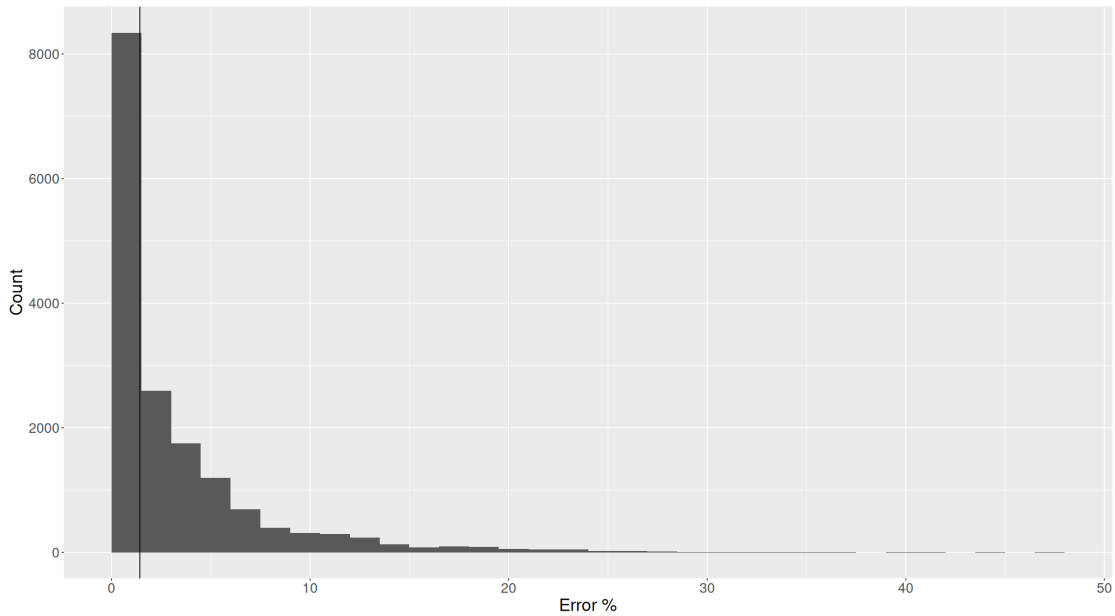

Figure 5: Histogram of percent of error sites per read for reads generated under the model parameters $\lambda = 0.025$, $\nu = 0.15$, $\delta_{ss} = 0.65$ and $\delta_{ds} = 0.015$. An error site is a site which has either experienced C $\rightarrow$ T or G $\rightarrow$ A damage. A vertical line is drawn at the median, where 1.4% of sites are errors. The max percent error for all reads is 47%

this. However, the largest contributor to placement error when analysing aDNA data is the amount of information available in the sequence to be placed. More simply, read length is the primary driver of error for aDNA reads. When reads become shorter, the probability of correctly placing the reads decreases (e.g. Fig 4). The reason for this becomes clear if we examine the distribution of read lengths as shown in Table 3. Both the estimated and actual median read lengths rapidly decrease as λ increases. This reflects the increase in placement error shown in Figure 4**A** with increasing nick frequency.

In contrast, the total amount of deamination events (that is, C $\rightarrow$ T and G $\rightarrow$ A damage) is not as important for placement accuracy. As we have shown in Figure 5, the rate of site errors is relatively low compared to the length of the read, even for higher damage rates. This is expected, as only two bases (C and G) are available to being misincorporated. On a typical "random" sequence, only 50% of sites will be eligible to be damaged. Furthermore, in practice GC content is often below Of course, the so-called GC-content varies among species and genomic regions. Hence we would expect that sequences will be more difficult to place correctly as a function of their GC-content. However, we investigated the effect of increasing amounts of GC-content on placement accuracy in the presence of aDNA damage using simulated data, and found no major differences between high and and low GC-content datasets. Details of this analysis are described in the Supplemental Material.

While we based our investigation around parameters found in Briggs et al. (2007), which is based on reads obtained from Neanderthal samples, the parameter values tested also cover other data sources, such as molluscs (Pedersen et al., 2015). Additionally, further investigation into some of the parameters is unlikely to clarify the issue any further, especially for the nick frequency λ. At the maximum value of λ in this study, many of the reads generated were filtered by the already permissive (w.r.t. to current practice in aDNA studies) minimum read length cutoff at 15 sites. Further increases in λ would only marginally decrease the realized read lengths, as most reads would be filtered by the minimum read length.

All of this together indicates that phylogenetic placement is an appropriate tool for analysing aDNA sequence. While the frequency of nicks strongly affects the probability of accurately placing reads, this is predominantly due to length of the reads to be placed, and not to other damage types. Importantly, *all* methods to identify aDNA reads will experience difficulties with extremely short reads, as the amount of information contained within a read is directly proportional to its length. Furthermore, placement seems to be robust to other forms of aDNA damage, even when reads are short (see Figures 2 and 3.)

There are some limitations to this work. First, we did not examine the case where C $\rightarrow$ T damage occurs in both overhangs instead of G $\rightarrow$ A in the second overhang, as is the case in single stranded libraries. However, we do not expect this to have any effect on the results presented here, as it is likely the number of misincorporations that matters, and not the type. Additionally, we did not thoroughly examine the effects of alignment on placement accuracy, even though we might expect alignment to strongly affect results. In particular, we used the multiple sequence aligner HMMER (Eddy, 2011), which aligns all sequences simultaneously. Another method, which we did not explore, is to conduct the simpler step of single sequence alignment, where only one read is aligned to the reference MSA at a time. In contrast to multiple sequence alignment, single sequence alignment does not change the reference alignment, and so it might be more robust to mis-aligning common errors like C $\rightarrow$ T or G $\rightarrow$ A damage.

Finally, our work indicates that a dedicated aDNA damage-aware phylogenetic placement model is unlikely to substantially improve results.

Data Availability

The modified PEWO pipeline is availible at `github.com/computations/PEWO`. Datasets, result files, plotting scripts, and utility scripts used can be found

Acknowledgment

This work was funded by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement No. 882500) and the European Union (EU) under Grant Agreement No 101087081 (Comp-Bio- div-GR).

References

Metin Balaban, Shahab Sarmashghi, and Siavash Mirarab. Apples: Scalable distance-based phylogenetic placement with or without alignments. *Systematic Biology*, 69:566–578, 5 2020. doi: 10.1093/sysbio/syz063. URL `http://dx.doi.org/10.1093/sysbio/syz063`.

Pierre Barbera, Alexey M Kozlov, Lucas Czech, Benoit Morel, Diego Darriba, Tomáš Flouri, and Alexandros Stamatakis. Epa-ng: Massively parallel evolutionary placement of genetic sequences. *Systematic Biology*, 68:365–369, 3 2019. doi: 10.1093/sysbio/syy054. URL `http://dx.doi.org/10.1093/sysbio/syy054`.

Simon A. Berger, Denis Krompass, and Alexandros Stamatakis. Performance, Accuracy, and Web Server for Evolutionary Placement of Short Sequence Reads under Maximum Likelihood. *Systematic Biology*, 60(3):291–302, 03 2011. ISSN 1063-5157. doi: 10.1093/sysbio/syr010. URL `https://doi.org/10.1093/sysbio/syr010`.

Adrian W. Briggs, Udo Stenzel, Philip L. F. Johnson, Richard E. Green, Janet Kelso, Kay Prüfer, Matthias Meyer, Johannes Krause, Michael T. Ronan, Michael Lachmann, and Svante Pääbo. Patterns of damage in genomic dna sequences from a neandertal. *Proceedings of the National Academy of Sciences*, 104:14616–14621, 9 2007. doi: 10.1073/pnas.0704665104. URL `http://dx.doi.org/10.1073/pnas.0704665104`.

Lucas Czech, Alexandros Stamatakis, Micah Dunthorn, and Pierre Barbera. Metagenomic analysis using phylogenetic placement—a review of the first decade. *Frontiers in Bioinformatics*, 2, 5 2022. doi: 10.3389/fbinf.2022.871393. URL `http://dx.doi.org/10.3389/fbinf.2022.871393`.

Clio Der Sarkissian, Vianney Pichereau, Catherine Dupont, Peter C. Ilsøe, Mickael Perrigault, Paul Butler, Laurent Chauvaud, Jón Eiríksson, James Scourse, Christine Paillard, and Ludovic Orlando. Ancient dna analysis identifies marine mollusc shells as new metagenomic archives of the past. *Molecular Ecology Resources*, 17:835–853, 9 2017. doi: 10.1111/1755-0998.12679. URL `http://dx.doi.org/10.1111/1755-0998.12679`.

Sean R. Eddy. Accelerated profile hmm searches. *PLoS Computational Biology*, 7:e1002195, 10 2011. doi: 10.1371/journal.pcbi.1002195. URL `http://dx.doi.org/10.1371/journal.pcbi.1002195`.

Marie Gurke, Amalia Vidal-Gorosquieta, Johanna L. A. Pajimans, Karolina W cek, Axel Barlow, Gloria González-Fortes, Stefanie Hartmann, Aurora Grandal-d'Anglade, and Michael Hofreiter. Insight into the introduction of domestic cattle and the process of neolithization to the spanish region galicia by genetic evidence. *PLOS ONE*, 16:e0249537, 4 2021. doi: 10.1371/journal.pone.0249537. URL `http://dx.doi.org/10.1371/journal.pone.0249537`.

James Haile, Duane G. Froese, Ross D. E. MacPhee, Richard G. Roberts, Lee J. Arnold, Alberto V. Reyes, Morten Rasmussen, Rasmus Nielsen, Barry W. Brook, Simon Robinson, Martina Demuro, M. Thomas P. Gilbert, Kasper Munch, Jeremy J. Austin, Alan Cooper, Ian Barnes, Per Möller, and Eske Willerslev. Ancient dna reveals late survival of mammoth and horse in interior alaska. *Proceedings of the National Academy of Sciences*, 106:22352–22357, 12 2009. doi: 10.1073/pnas.0912510106. URL `http://dx.doi.org/10.1073/pnas.0912510106`.

Benjamin Linard, Krister Swenson, and Fabio Pardi. Rapid alignment-free phylogenetic identification of metagenomic sequences. *Bioinformatics*, 35:3303–3312, 9 2019. doi: 10.1093/bioinformatics/btz068. URL `http://dx.doi.org/10.1093/bioinformatics/btz068`.

Benjamin Linard, Nikolai Romashchenko, Fabio Pardi, and Eric Rivals. Pewo: a collection of workflows to benchmark phylogenetic placement. *Bioinformatics*, 36:5264–5266, 1 2021. doi: 10.1093/bioinformatics/btaa657. URL `http://dx.doi.org/10.1093/bioinformatics/btaa657`.

Frédéric Mahé, Colomban de Vargas, David Bass, Lucas Czech, Alexandros Stamatakis, Enrique Lara, David Singer, Jordan Mayor, John Bunge, Sarah Sernaker, Tobias Siemensmeyer, Isabelle Trautmann, Sarah Romac, Cédric Berney, Alexey Kozlov, Edward A. D. Mitchell, Christophe V. W. Seppey, Elianne Egge, Guillaume Lentendu, Rainer Wirth, Gabriel Trueba, and Micah Dunthorn. Parasites dominate hyperdiverse soil protist communities in neotropical rainforests. *Nature Ecology \& Evolution*, 1, 3 2017. doi: 10.1038/s41559-017-0091. URL `http://dx.doi.org/10.1038/s41559-017-0091`.

Rui Martiniano, Bianca De Sanctis, Pille Hallast, and Richard Durbin. Placing ancient dna sequences into reference phylogenies. *Molecular Biology and Evolution*, 39, 2 2022. doi: 10.1093/molbev/msac017. URL `http://dx.doi.org/10.1093/molbev/msac017`.

Frederick A Matsen, Robin B Kodner, and E Virginia Armbrust. pplacer: linear time maximum-likelihood and bayesian phylogenetic placement of sequences onto a fixed reference tree. *BMC Bioinformatics*, 11, 12 2010. doi: 10.1186/1471-2105-11-538. URL `http://dx.doi.org/10.1186/1471-2105-11-538`.

Mikkel Winther Pedersen, Søren Overballe-Petersen, Luca Ermini, Clio Der Sarkissian, James Haile, Micaela Hellstrom, Johan Spens, Philip Francis Thomsen, Kristine Bohmann, Enrico Cappellini, Ida Bærholm Schnell, Nathan A. Wales, Christian Carøe, Paula F. Campos, Astrid M. Z. Schmidt, M. Thomas P. Gilbert, Anders J. Hansen, Ludovic Orlando, and Eske Willerslev. Ancient and modern environmental dna. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 370:20130383, 1 2015. doi: 10.1098/rstb.2013.0383. URL `http://dx.doi.org/10.1098/rstb.2013.0383`.

Nikolaos Psonis, Carlos Neto de Carvalho, Silvério Figueiredo, Eugenia Tabakaki, Despoina Vassou, Nikos Poulakakis, and Dimitris Kafetzopoulos. Molecular identification and geographic origin of a post-medieval elephant finding from southwestern portugal using high-throughput sequencing. *Scientific Reports*, 10, 11 2020. doi: 10.1038/s41598-020-75323-y. URL `http://dx.doi.org/10.1038/s41598-020-75323-y`.

Nikolaos Psonis, Katerina Vardinoyannis, and Nikos Poulakakis. High-throughput degraded dna sequencing of subfossil shells of a critically endangered stenoendemic land snail in the aegean. *Molecular Phylogenetics and Evolution*, 175:107561, 10 2022a. doi: 10.1016/j.ympev.2022.107561. URL http://dx.doi.org/10.1016/j.ympev.2022.107561.

Nikolaos Psonis, Despoina Vassou, Loucas Nicolaou, Socrates Roussiakis, George Iliopoulos, Nikos Poulakakis, and Spyros Sfenthourakis. Mitochondrial sequences of the extinct cypriot pygmy hippopotamus confirm its phylogenetic placement. *Zoological Journal of the Linnean Society*, 196:979–989, 11 2022b. doi: 10.1093/zoolinnean/zlab089. URL http://dx.doi.org/10.1093/zoolinnean/zlab089.

Gabriel Renaud, Kristian Hanghøj, Eske Willerslev, and Ludovic Orlando. gargammel: a sequence simulator for ancient dna. *Bioinformatics*, 33:577–579, 2 2017. doi: 10.1093/bioinformatics/btw670. URL http://dx.doi.org/10.1093/bioinformatics/btw670.

Susanna Sawyer, Johannes Krause, Katerina Guschanski, Vincent Savolainen, and Svante Pääbo. Temporal patterns of nucleotide misincorporations and dna fragmentation in ancient dna. *PLoS ONE*, 7:e34131, 3 2012. doi: 10.1371/journal.pone.0034131. URL http://dx.doi.org/10.1371/journal.pone.0034131.

Montgomery Slatkin and Fernando Racimo. Ancient dna and human history. *Proceedings of the National Academy of Sciences*, 113:6380–6387, 6 2016. doi: 10.1073/pnas.1524306113. URL http://dx.doi.org/10.1073/pnas.1524306113.

Sujatha Srinivasan, Noah G. Hoffman, Martin T. Morgan, Frederick A. Matsen, Tina L. Fiedler, Robert W. Hall, Frederick J. Ross, Connor O. McCoy, Roger Bumgarner, Jeanne M. Marrazzo, and David N. Fredricks. Bacterial communities in women with bacterial vaginosis: High resolution phylogenetic analyses reveal relationships of microbiota to clinical criteria. *PLoS ONE*, 7:e37818, 6 2012. doi: 10.1371/journal.pone.0037818. URL http://dx.doi.org/10.1371/journal.pone.0037818.

Shinichi Sunagawa, Luis Pedro Coelho, Samuel Chaffron, Jens Roat Kultima, Karine Labadie, Guillem Salazar, Bardya Djahanschiri, Georg Zeller, Daniel R. Mende, Adriana Alberti, Francisco M. Cornejo-Castillo, Paul I. Costea, Corinne Cruaud, Francesco d'Ovidio, Stefan Engelen, Isabel Ferrera, Josep M. Gasol, Lionel Guidi, Falk Hildebrand, Florian Kokoszka, Cyrille Lepoivre, Gipsi Lima-Mendez, Julie Poulain, Bonnie T. Poulos, Marta Royo-Llonch, Hugo Sarmento, Sara Vieira-Silva, Céline Dimier, Marc Picheral, Sarah Searson, Stefanie Kandels-Lewis, Chris Bowler, Colomban de Vargas, Gabriel Gorsky, Nigel Grimsley, Pascal Hingamp, Daniele Iudicone, Olivier Jaillon, Fabrice Not, Hiroyuki Ogata, Stephane Pesant, Sabrina Speich, Lars Stemmann, Matthew B. Sullivan, Jean Weissenbach, Patrick Wincker, Eric Karsenti, Jeroen Raes, Silvia G. Acinas, Peer Bork, Emmanuel Boss, Chris Bowler, Michael Follows, Lee Karp-Boss, Uros Krzic, Emmanuel G. Reynaud, Christian Sardet, Mike Sieracki, Didier Velayoudon, and None None. Structure and function of the global ocean microbiome. *Science*, 348, 5 2015. doi: 10.1126/science.1261359. URL http://dx.doi.org/10.1126/science.1261359.

Derrick E Wood and Steven L Salzberg. Kraken: ultrafast metagenomic sequence classification using exact alignments. *Genome Biology*, 15, 3 2014. doi: 10.1186/gb-2014-15-3-r46. URL http://dx.doi.org/10.1186/gb-2014-15-3-r46.