metaDMG:

An Ancient DNA Damage Toolkit

Christian Stentoft Michelsen [1 †] ✉, **Mikkel Winther Pedersen** [2 †] ✉, **Antonio Fernandez-Guerra** [2], **Lei Zhao**[2], **Troels C. Petersen** [1], **Thorfinn Sand Korneliussen** [2]

[1]Niels Bohr Institute, University of Copenhagen; [2]Globe Institute, University of Copenhagen

✉ **For correspondence:** christianmichelsen@gmail.com (CM); mwpedersen@sund.ku.dk (MW)

[†]Authors contributed equally.

Present address: Niels Bohr Institute, University of Copenhagen, Blegdamsvej 17, 2100 Copenhagen, Denmark

Data availability: Data is available on Zenodo or the Github repository.

Funding: This work was supported by Carlsberg Foundation Young Researcher Fellowship awarded by the Carlsberg Foundation [CF19-0712], and the Lundbeck Foundation Centre for Disease Evolution: [R302-2018-2155 to L.Z]. The funders had no role in the decision to publish.

Competing interests: The author declare no competing interests.

Abstract

1. **Motivation** Under favourable conditions DNA molecules can survive for more than two million years (Kjaer et al in press). Such genetic remains can give unique insights to past assemblages, populations and evolution of species. However, DNA is degraded over time, and are therefore found in ultra low concentrations making it highly prone to contamination from modern DNA sources. Despite strict precautions implemented in the field (Llamas et al., 2017), DNA from modern sources does appear in the final output data. One authenticity criteria used in all ancient DNA studies are the high nucleotide mis-incorporation rates that can be observed as a result of chemical post-mortem DNA damage, in fact misincorporation patterns have become instrumental to authenticate ancient sequences. To date this has primarily been possible for single organisms (Jónsson et al., 2013)) and recently for assemblies (Borry et al., 2021), but these methods have not been designed, nor can they be computationally upscaled to calculate the thousands of taxonomic species that occur in just one metagenome.
2. **Methods** We present `metaDMG`, a novel framework that takes advantage of the information already contained in the alignment files to compute and statically evaluate post-mortem DNA damage, thus bypassing the need for classifying and splitting reads into individual organisms and realigning these to parse data to mapDamage2.0 (Jónsson et al., 2013). It

uses a Bayesian approach that combines a geometric damage profile with a beta-binomial model to fit the entire model to the misincorporations which drastically improve the damage estimates compared to previous methods.

3. **Results** Using a two-tier simulation setup, we find `metaDMG` to not only be a factor of 10 faster than previous methods but it is also more accurate and able to evaluate even complex metagenomes with tens of thousands of species. Even with very few number of reads, down to even below 1000 reads. BLABLA, more results here.
4. **Conclusion** `metaDMG` includes state-of-the-art statistical methods for computing nucleotide misincorporation and fragmentation patterns of even highly complex samples along with re-implementation of the current statistics used within the field such as PMDtools (Skoglund et al., 2014). This suite of programs is freely available and consists of computational parts implemented as multi threaded C++ programs as well as computationally optimized modern python libraries, and an interactive dashboard for displaying the results. `metaDMG` is furthermore flexible, compatible with custom databases, can output nucleotide misincorporation and fragmentation patterns at different taxonomic ranks as well as per reference ID.

keywords: ancient DNA, damage estimation, DNA damage, lowest common ancestor, metaDMG, metadamage, metagenomics, statistics.

1 | INTRODUCTION

Throughout the life of an organism, it contaminates its surrounding environment with cells or tissue and hence its DNA contained within. As the cell leaves its host, DNA repair mechanisms stops and the DNA is now subjected to chemical and mechanical degradation, resulting in fragmented molecules and chemical damages, characteristic for ancient DNA (Briggs et al., 2007; Dabney, Meyer, and Pääbo, 2013). Ancient DNA has been shown to be able to survive in the environment for thousands and even up to two million years (Kjaer et al in review), and have been widely used to study past organisms and organism composition (Cappellini et al., 2018). Particularly misincorporations of cytosines on thymines as a result of deamination has been found to independently authenticate ancient DNA origin (Dabney, Meyer, and Pääbo, 2013; Ginolhac et al.,

2011). Postmortem damage with regards to DNA is characterized by the four Briggs parameters (Briggs et al., 2007). A damaged dna fragment tend to be short, and is likely to be single stranded at the termini of the fragment. There is an high proportion of C→T substitutions at the single stranded part ϵ_{ss}, a somewhat higher C→ T at the double stranded part ϵ_{ds}. The length of the single stranded part (*overhang*) follows a geometric distribution λ, and finally there might be breaks at the backbone in the double stranded part ν. It is possible to estimate these four Briggs parameters Jónsson et al., 2013 but these four parameters are rarely used directly for asserting "ancientness", and researchers working with ancient DNA tend to simply use the empirical C→T on the first position of the fragment together with other supporting summary statistic of the experiment. This ancient DNA (aDNA) authenticity approach, were initially performed on single individual sources such as hair, bones, teeth and later on ancient environmental samples such as soil sediments CITE SOMETHING. While this is a relatively fast process for single individuals it becomes increasingly demanding, iterative and time consuming as the samples and the diversity within increases, as in the case for metagenomes from ancient soil, sediments, dental calculus, coprolites and other ancient environmental sources. It has therefore been practice to estimate damage for only the key taxa of interest in a metagenome, as a metagenomic sample easily includes thousands of different taxonomic entities, that would make a complete estimate an impossible task.

We have devised a novel test statistic in `metaDMG` which takes into account all relevant information in single scalar. For these reasons, we present here `metaDMG`, a tool that enables fast and accurate DNA damage estimation of whole metagenomes within hours. `metaDMG` is designed and upscales equally for the increasingly large datasets that are generated in the field of ancient environmental DNA, but can also with advantage be used to estimate DNA damage of single genomes and samples with low complexity, it can even compute an global damage estimate for a given sample. `metaDMG` is compatible with the NCBI taxonomy and can use ngsLCA to perform a naïve last common ancestor of the aligned reads to get precise damage estimates for the reads classified to different taxonomic nodes. In addition, it is also designed to be used with custom taxonomies and metagenomic assembled genomes.

After defining the method and notation used throughout this paper, we show through multiple sets of simulations that `metaDMG` not only improves on existing methods in the case of single-genome damage estimation but also work for metagenomic samples. Finally, we apply our method on a representative mix of nine different metagenomic samples to show the real life performance

of metaDMG.

2 | METHODS & MATERIALS

Perhaps the most basic bioinformatic analyses is the difference between two nucleotide sequences. This assumes that we have a haploid representation of our target organisms and larger differences can be interpreted as larger genetic differences. Obtaining a haploid representation is none trivial, firstly our target organism might not be haploid and we need to construct a consensus genome, secondly data from modern day sequencers are essentially a sampling with replacement process and we need to infer the relative location of each of the possible millions or even billions of short DNA fragments, this is the process which is called mapping or alignment. Thirdly, and the focus for this manuscript, is the quantification of the presence of postmortem damage (PMD) in DNA. PMD mainly manifests as an excess of cytosine to thymine substitutions at the termini of fragments that has been prepared for sequencing. A priori we can not directly observe these actual biochemical changes but we can align each fragment and consider the difference between reference and read as possible PMD, and it is even possible to use the excess of C to T at the single fragment level to separate modern from ancient (data with PMD) (Skoglund et al., 2014). Expanding from the single read all reads for a sequencing experiment and genome to tabulate the overall substitution or mismatch rates to obtain a stastistic of the damage (Borry et al., 2021) or even estimate the four Briggs parameters that is traditionally used to characterize the damage signal (Jónsson et al., 2013).

We have devised a general ancient DNA damage toolkit with a special emphasis in a metagenomic setting which implements and expands existing relevant methods but also expands with several state of the art novel methodologies. At the most basic level we have reimplemented the approach given in (Skoglund et al., 2014) which allows for the extracting and separation of highly damaged DNA reads. Secondly under the assumption of vast amounts of data we have defined a full multinomial regression model building on the method in (Cabanski et al., 2012), we show that this will give superior and stable results if it is possible to obtain high depth and coverage data. However in standard ancient DNA context it is generally not possible to obtain vast amounts of data and we propose two novel tests statistics that is especially suited for this scenario. To our knowledge there are no currently available methods that is geared towards damage analysis in a metagenomic setting and existing approaches are essentially based on remapping against the sin-

gle target organism and does not take into account any possible issues with regards to reads being well assigned or specified. Our solution called `metaDMG` (pronounced metadamage), estimates the damage patterns in metagenomic samples in a three step approach. First, the lowest common ancestor (LCA) for each read (mapped to a multi-species reference database) is computed and the the mismatch matrix for each leaf node (e.g. taxonomic ID or contig, depending on the database used) is computed based on the mapped reads. Second, `metaDMG` fits a damage model to each leaf node to compute the ancient damage estimates. Finally, the results are visualized in the `metaDMG` dashboard, which is a state of the art graphical user interface that allows for fast and user-friendly interaction with the results for further downstream analysis and visualization.

2.1 | Lowest Common Ancestor and Mismatch matrices

For environmental DNA (eDNA) studies we routinely apply a competitive alignment approach where we consider all possible alignments for a given read. Each read is mapped against a multi species reference databases (e.g. nucleotide or RefSeq from NCBI or custom downloaded). A single read might map to a highly conserved gene that is shared across higher taxonomic ranks such as class or even domains. This read will not provide relevant information due to the generality, whereas a read that maps solely to a single species or species from a genus would be indicative of the read being well classified. We seek to obtain the pattern or signal of damage which is done by the tabulation of the cycle specific mismatch rates between our reference and observed sequence for all well classified reads.

In details we compute the lowest common ancestor (lca) for all alignments for for each read, this is done using (Wang et al., 2022) and if a read is well classified or properly assigned based on a user defined threshold (specics, genus or family) we tabulate the mismatches for each cycle, if a read is not well assigned it is discarded. Pending on the run mode we allow for the construction of these mismatch tables on three different lavels. Either we obtain a basic single global mismatch matrix, which could be relevant in a standard single genome aDNA study and similar to the tabulation used in (Jónsson et al., 2013). Secondly we can obtain per reference counts or if a taxonomy database has been supplied we allow for the aggregation from leaf nodes to the internal taxonomic ranks towards the root.

To suit as many users as possible, `metaDMG` takes as input an alignment file (.bam, .sam, or .sam.gz), where Each read is hereafter allowed an equal chance to map against the multiple refer-

ences. One read can therefore attract multiple alignments, and we thus first seek to find the lowest common ancestor (LCA) among the alignments based on the tree structure from the databases and a user defined read-reference similarity interval (Wang et al., 2022). Note that `metaDMG` is not limited to the NCBI database and allow for custom databases as well.

Regardless of runmode or weighing scheme used in the possible aggregation we obtain the nucleotide substitution frequencies across reads which provides us with the position dependent mismatch matrices, $\underline{\underline{M}}(x)$, with x denoting the position in the read, starting from 1. At a specific position, $M_{\text{ref}\to\text{obs}}(x)$ describes the number of nucleotides that was mapped to a reference base $\mathcal{B}_{\text{ref}}$ but observed to be $\mathcal{B}_{\text{obs}}$, where $\mathcal{B} \in \{A, C, G, T\}$. The number of C to T transitions, e.g., is denoted as $M_{C\to T}(x)$.

When calculating the mismatch matrix, two different approaches can be taken. Either all alignments of the read will be counted, which we will refer to as weight-type 0, or the counts will be normalized by the number of alignments of each read; weight-type 1 (default).

2.2 | Damage Estimation

The damage pattern observed in aDNA has several features which are well characterized. By modelling these, one can construct observables sensitive to aDNA signal. We model the damage patterns seen in ancient DNA by looking exclusively at the $C \to T$ transitions in the forward direction (5') and the $G \to A$ transitions in the reverse direction (3'). For each LCA, we denote the number of transitions $k(x)$ as:

$$k(x) = \begin{cases} M_{C\to T}(x) & \text{for } x > 0 \quad \text{(forward)} \\ M_{G\to A}(x) & \text{for } x < 0 \quad \text{(reverse)} \end{cases} \tag{1}$$

and the number of the reference counts $N(x)$:

$$N(x) = \begin{cases} \sum_{i\in\mathcal{B}} M_{C\to i}(x) & \text{for } x > 0 \quad \text{(forward)} \\ \sum_{i\in\mathcal{B}} M_{G\to i}(x) & \text{for } x < 0 \quad \text{(reverse)}, \end{cases} \tag{2}$$

where the sum is over all four bases. The damage frequency is thus $f(x) = k(x)/N(x)$. A natural choice of likelihood model would be the binomial distribution. However, we found that a binomial likelihood lacks the flexibility needed to deal with the large amount of variance (overdispersion) we found in the data due to bad references and misalignments.

To accommodate overdispersion, we instead apply a beta-binomial distribution, $\mathcal{P}_{\text{BetaBinomial}}$, which treats the probability, p, as a random variable following a beta distribution[1] with mean μ and concentration ϕ: $p \sim \text{Beta}(\mu, \phi)$. The beta-binomial distribution has the the following probability density function:

$$\mathcal{P}_{\text{BetaBinomial}}(k \mid N, \mu, \phi) = \binom{N}{k} \frac{B\left(k + \mu\phi,\ N - k + \phi(1-\mu)\right)}{B\left(\mu\phi,\ \phi(1-\mu)\right)}, \tag{3}$$

where B is defined as the beta function:

$$B(x, y) = \frac{\Gamma(x)\Gamma(y)}{\Gamma(x+y)}, \tag{4}$$

with $\Gamma(\cdot)$ being the gamma function (Cepeda-Cuervo and Cifuentes-Amado, 2017).

The close resemblance to a binomial model is most easily seen by comparing the mean and variance of a random variable k following a beta-binomial distribution, $k \sim \mathcal{P}_{\text{BetaBinomial}}(N, \mu, \phi)$:

$$\begin{aligned} \mathbb{E}[k] &= N\mu \\ \mathbb{V}[k] &= N\mu(1-\mu)\frac{\phi + N}{\phi + 1}. \end{aligned} \tag{5}$$

The expected value of k is similar to that of a binomial distribution and the variance of the beta-binomial distribution reduces to a binomial distribution as $\phi \to \infty$. The beta-binomial distribution can thus be seen as a generalization of the binomial distribution.

Note that both equation (3) and (5) relates to damage at a specific base position, i.e. for a single k and N. To estimate the overall damage in the entire read using the position dependent counts, $k(x)$ and $N(x)$, we model μ as position dependent, $\mu(x)$, and assume a position-independent concentration, ϕ. We model the damage frequency with a modified geometric sequence, i.e. exponential decreasing for discrete values of x:

$$\tilde{f}(x; A, q, c) = A(1-q)^{|x|-1} + c. \tag{6}$$

Here A is the amplitude of the damage and q is the relative decrease of damage pr. position. A background, c, was added to reflect the fact that the mismatch between the read and reference might be due to other factors than just ancient damage. As such, we allow for a non-zero amount of damage, even as $x \to \infty$. This is visualized in Fig. 1 along with a comparison between the classical binomial model and the beta-binomial model.

To estimate the fit parameters, A, q, c, and ϕ, we apply Bayesian inference to utilize domain specific knowledge in the form of priors. We assume weakly informative beta-priors[2] for both A, q,

[1] Note that we do not parameterize the beta distribution in terms of the common (α, β) parameterization, but instead using the more intuitive (μ, ϕ) parameterization. One can reparameterize $(\alpha, \beta) \to (\mu, \phi)$ using the following two equations: $\mu = \frac{\alpha}{\alpha+\beta}$ and $\phi = \alpha + \beta$ (Cepeda-Cuervo and Cifuentes-Amado, 2017).

[2] Parameterized as (μ, ϕ)

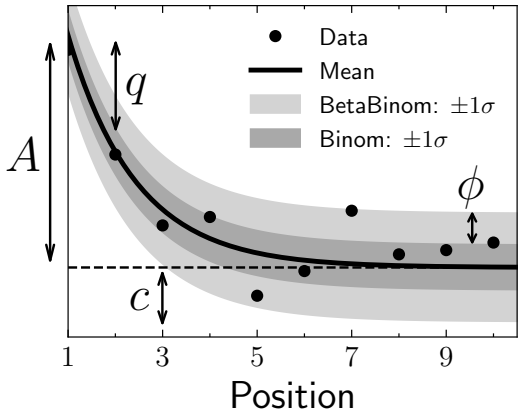


Figure 1. Illustration of the damage model. The figure shows data points as circles and the damage, $f(x)$, as a solid line. The amplitude of the damage is A, the offset is c, and the relative decrease in damage pr. position is given by q. The damage uncertainty for a binomial model is shown in dark grey and the uncertainty for a beta-binomial model in light grey. The additional uncertainty of the beta-binomial model, compared to the binomial model, is related to ϕ, see equation (5).

and c. In addition to this, we assume an exponential prior on ϕ with the requirement of $\phi > 2$ to avoid too much focus on 0-or-1 probabilities (McElreath, 2020). The final model is thus:

$$
\begin{aligned}
&[A \text{ prior}] && A \sim \text{Beta}(0.1, 10) \\
&[q \text{ prior}] && q \sim \text{Beta}(0.2, 5) \\
&[c \text{ prior}] && c \sim \text{Beta}(0.1, 10) \\
&[\phi \text{ prior}] && \phi \sim 2 + \text{Exponential}(1000) \\
&[\text{likelihood}] && k_i \sim \mathcal{P}_{\text{BetaBinomial}}\left(N_i, \tilde{f}\left(x_i; A, q, c\right), \phi\right),
\end{aligned}
\tag{7}
$$

where i is an index running over all positions.

We define the damage due to deamination, D, as the background-subtracted damage frequency at the first position: $D \equiv \tilde{f}(|x| = 1) - c$. As such, D is the damage related to ancientness. Using the properties of the beta-binomial distribution, eq. (5), we find the mean and variance of the damage, D:

$$
\begin{aligned}
\mathbb{E}\left[D\right] &\equiv \bar{D} = A \\
\mathbb{V}\left[D\right] &\equiv \sigma_D^2 = \frac{A(1-A)}{N}\frac{\phi + N}{(\phi + 1)}.
\end{aligned}
\tag{8}
$$

Since D estimates the overexpression of damage due to ancientness, not only the mean is relevant but also the certainty of $D > 0$. We quantify this through the significance $Z = \bar{D}/\sigma_D$

which is thus the number of standard deviations ("sigmas") away from zero. Assuming a Gaussian distribution of D, $Z > 2$ would indicate a probability of D being larger than zero, i.e. containing ancient damage, with more than 97.7% probability. These two values allows us to not only quantify the amount of ancient damage (ie. $\bar{D}$) but also the certainty of this damage (Z) without even having to run multiple models and comparing these.

We perform the Bayesian inference of the parameters models using Hamiltonian Monte Carlo (HMC) sampling which is a particular of Monte Carlo Markov Chain (MCMC) algorithm (Betancourt, 2018). Specifically, we use the NUTS implementation in NumPyro (Phan, Pradhan, and Jankowiak, 2019), a Python package which uses JAX (Bradbury et al., 2018) under the hood for automatic differentiation and JIT compilation. We treat each leaf node of the LCA as being independent and generate 1000 MCMC samples after an initial 500 samples as warm up.

Since running the full Bayesian model is computationally expensive, we also allow for a faster, approximate method by just fitting the maximum a posteriori probability (MAP) estimate. We use iMinuit (Dembinski et al., 2021) for the MAP optimization with Numba acceleration (Lam, Pitrou, and Seibert, 2015) for even faster run times. On a Macbook M1 Pro model from 2021, the timings for running the full Bayesian model is $1.41 \pm 0.04\,\mathrm{s/fit}$ and for the MAP it is $4.34 \pm 0.07\,\mathrm{ms/fit}$, showing more than a 2 order increase in performance (around 300x) for the approximate model. Both models allow for easy parallelisation to decrease the computation time.

2.3 | Visualisation

We provide an interactive dashboard to properly visualise the results from the modelling phase, see https://metadmg.onrender.com/ for an example. The dashboard allows for filtering, styling and variable selection, visualizing the mismatch matrix related to a specific leaf node, and exporting of both fit results and plots. By filtering, we include both filtering by sample, by specific cuts in the fit results (e.g. requiring D to be above a certain threshold), and even by taxonomic level (e.g. only looking tax IDs that are part of the Mammalia class). We greatly believe that a visual overview of the fit results increase understanding of the data at hand. The dashboard is implemented with Plotly plots and incorporated into a Dash dashboard (Plotly, 2015).

3 | SIMULATION STUDY

We conducted two sets of simulations, one to gauge the performance of the damage model itself and one to determine the performance of the full `metaDMG` pipeline, i.e. both LCA and damage model.

3.1 | Single-genome Simulations

The first set of simulations was performed by taking a single, representative genome and adding deamination and sequencing noise to it followed by a mapping step and finally damage estimation using `metaDMG`. The deamination was applied using NGSNGS (XXX, ref here) which is a recent implementation of the original Briggs model similar to Gargammel (Neukamm, Peltzer, and Nieselt, n.d.) but with better performance and more accurate deamination patterns. In this step we vary the simulated amount of damage added (in particular the single-stranded DNA deamination, δ_{SS} in the original Briggs model (Briggs et al., 2007)), the number of reads, and the fragment length distribution.

We chose five different, representative genomes, in each of these varying the three simulation parameters. These genomes where the homo sapiens, the betula, and three microbial organisms with respectively low, median, and high amount of GC-content. For each of these simulations, we performed 100 independent runs to measure the variability of the parameter estimations and quantify the robustness of the estimates. We simulated eight different sets of damage (approximately 0%, 1%, 2%, 5%, 10%, 15%, 20%, 30%), 13 sets of different number of reads (10, 25, 50, 100, 250, 500, 1.000, 2.500, 5.000, 10.000, 25.000, 50.000, 100.000), three sets of different fragment length distributions (samples from a lognormal distributions with mean 35, 60, and 90, each with a standard deviation of 10), and five different genomes, each simulation set repeated 100 times.

In addition to this, we also create 1000 repetitions of the non-damaged simulations for Homo Sapiens to be able to gauge the risk of finding false positives. Finally, to show that the damage estimates that `metaDMG` provides are independent of the contig size, we artificially create three different genomes by sampling 1.000, 10.000 or 100.000 different basepairs from a uniform categorical distribution of $\{A, C, G, T\}$.

To be able to compare our estimates to a known value, we generate 1.000.000 reads from NGSNGS without any added sequencing noise for each of other sets of simulation parameters.

The difference in damage frequency at position 1 and 15 is then the value to compare to:

$$D_{\text{known}} = \frac{f(x=1) - f(x=15)}{2} + \frac{f(x=-1) - f(x=-15)}{2}, \tag{9}$$

where we take the average of the C to T damage frequency difference and the G to A damage frequency difference.

The fastq files were simulated with NGSNGS using the above mentioned simulation parameters, all with the same quality scores profiles as used in ART (Huang et al., 2012), based on the Illumina HiSeq 2500 (150 bp). The mapping was performed using Bowtie-2 with the `--no-unal` flag (Langmead and Salzberg, 2012).

3.2 | Metagenomic Simulations

While the previously mentioned simulation study is perfectly aimed at quantifying the performance of the damage model in the case of single-reference genomics it does lack the complexity related to metagenomic samples. Therefore, we also conduct a more advanced simulation study to determine the accuracy of the full `metaDMG` pipeline.

The previously mentioned simulation study quantifies the damage model's performance for single-reference genomics, but it does not address the complexity of metagenomic samples. Therefore, we also conducted a more advanced simulation study to determine the accuracy of the full `metaDMG` pipeline. Based on an ancient metagenome, we created a synthetic dataset that reproduces the composition, fragment length distribution, and damage patterns for each genome. We selected X metagenomes (Supp table XXX) covering several environmental conditions and ages. First, we mapped the reads of each metagenome with bowtie2 against a database that contained the GTDB r202 (Parks et al., 2018) species cluster reference sequences, all organelles from NCBI RefSeq (NCBI Resource Coordinators, 2018), and the reference sequences from CheckV (Nayfach et al., 2021). We used bam-filter 1.0.11 with the flag `--read-length-freqs` to get the mapped read length distribution for each genome and their abundance. The genomes with an observed-to-expected coverage ratio greater than 0.75 were kept. The filtered BAM files were processed by `metaDMG` to obtain the misincorporation matrices. The abundance tables, fragment length distribution, and misincorporation matrices were used in aMGSIM-smk v0.0.1 (Fernandez-Guerra, 2022), a Snakemake workflow (Mölder et al., 2021) that facilitates the generation of many synthetic ancient metagenomes. The data used and generated by the workflow can be obtained from Figshare link

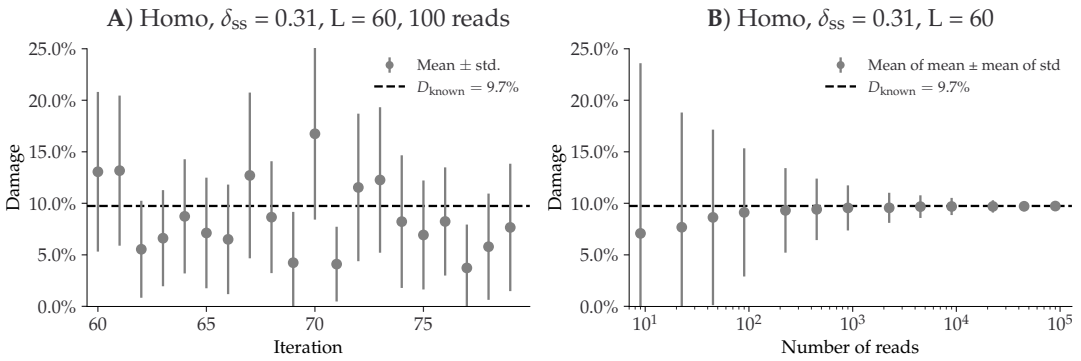


Figure 2. Overview of the single-genome simulations based on the homo sapiens genome with the Briggs parameter $\delta_{ss} = 0.065$ and a fragment length distribution with mean 60. **A**) This plot shows the estimated damage (D) of 10 simulations with 100 simulated reads. The grey points shows the mean damage (with its standard deviation as errorbars). The known damage (D_{known}) is shown as a dashed line, see eq. (9). **B**) This plot shows the average damage as a function of the number of reads. The grey points show the average of the individual means (with the average of the standard deviations as errors).

(XXX). We then performed taxonomic profiling using the same parameters used for the synthetic reads generated by aMGSIM-smk.

4 | RESULTS

The accuracy of all methods in `metaDMG` was tested in various simulation scenarios. In general we find that `metaDMG` yields accurate, precise damage estimates even in extreme low-coverage data.

4.1 | Single-genome Simulations

The results of the single-genome simulations can be seen in Figure 2. The left part of the figure shows `metaDMG` damage estimates based on the homo sapiens genome with the Briggs parameter $\delta_{ss} = 0.31$ and a fragment length distribution with mean 60, each of the 10 simulations generated with 100 simulated reads for 10 representative simulations. When the damage estimates are low, the distribution of D is highly skewed (restricted to positive values) leading to errorbars sometimes going into negative damage, which of course represents un-physical values. The right hand side of the figure visualizes the average amount of damage across a varying number of reads. This shows that the damage estimates converge to the known value with more data, and that one needs more than 100 reads to even get stricly positive damage estimates (when including uncertainties).

Across more than 5 different species, 3 different fragment length distributions, and 3 different

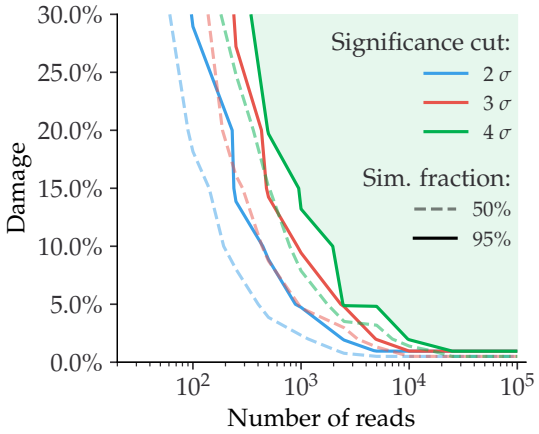


Figure 3. Given a specific significance cut, the solid contour line shows the relationship between the amount of damage and the number of reads required to be able to correctly infer damage in 95% of the species. The dashed line shows the similar value for a simulation fraction of 50%. The green part of the figure shows the "good" region of number of reads and estimated damage, given than one wants to be more than 95% certain of correctly identifying damage with more than 4σ confidence.

contig length distributions, each with 100 simulations for 104 different sets of simulation parameters, the only difference we note in the damage estimates is between species with low, median, and high GC-levels. In general, species with higher GC-levels exhibit lower variations in their damage estimates compared to species with lower GC-levels, leading to high-GC species requiring fewer reads to establish damage estimates.

Based on the single-genome simulations, we can compute the relationship between the amount of damage in a species and the number of reads required to correctly infer that the given species is damaged, see Figure 3. If we want to find damage with a significance of more than 2 (solid blue line) in a sample with around 5% expected damage, it requires about 1000 reads to be 95% certain that we will find results this good. Said in other words: given 100 different samples, each with 1000 reads and around 5% damage, one would expect to find damage (with a $Z > 2$) in 95 of the total 100 samples, on average. If we loose the requirement such that it is okay to only find it in every second sample, it would be enough with only around 250 reads in each sample (dashed blue line).

Finally, to quantify the risk of incorrectly assigning damage to a non-damaged species, we created 1000 independent simulations for a varying number of reads, where none of them had any artificial ancient damage applied, only sequencing noise. Figure 4 shows the damage (D) as a function of the significance (Z) for the case of 1000 simulated reads. Even though the estimated damage is larger than zero, the damage is non-significant since the significance is less than one. When looking

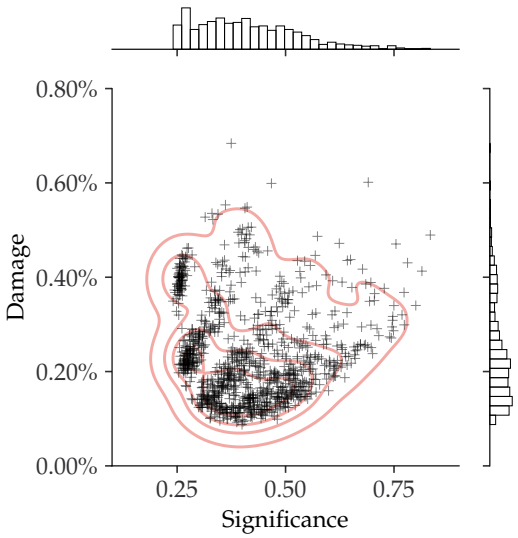


Figure 4. This figure shows the inferred damage estimates of 1000 independent simulations, each with 1000 reads and no artificial ancient damage applied, with the inferred damage shown on the y-axis and the significance on the x-axis. Each simulation is shown as a single cross and the red lines show the kernel density estimate (KDE) of the damage estimates. The marginal distributions are shown as histograms next to the scatter plot.

at all the figures across the different number of reads, see Figure `bayesian_zero_damage_plots.pdf`, we note that a loose cut requiring that $D > 1\%$ and $Z > 2$ would filter out all of non-damaged points.

4.2 | Metagenomic Simulations

With the full metagenomic simulation pipeline we can further probe the performance of `metaDMG`. By looking at the six different samples at different steps in the pipeline we are able to show that `metaDMG` provides relevant, accurate damage estimates. First of all, we run `metaDMG` on the six samples after fragmentation with FragSim. Since no deamination has yet been added at this step in the pipeline, this is also a test of the risk of getting false positives. The results can be seen in Figure 5 where we see the damage estimates for both the species that we simulate to be ancient and the species that we do not add deamination to. We see that the damage estimates are quite similar, as expected, and that our previously established loose cut of $D > 1\%$ and $Z > 2$ still filters out all of non-damaged points.

In comparison we can look at Figure 6 which shows the same plot, but after the deamination (deamSim) and sequencing errors (ART) has been added. Here we see a clear difference between

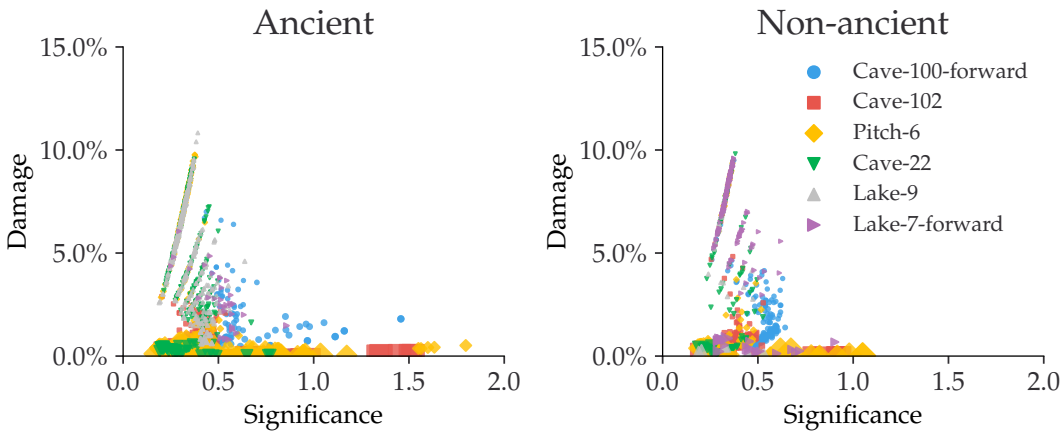


Figure 5. Estimated amount of damage as a function of significance using the fragSim data. The left figure shows the damage of the species that we simulated to be ancient (however with no deamination added yet) and the right figure shows the same for the species that are not going to have deamination added.

the ancient and the non-ancient ones, as expected. The non-ancient species would still not pass the loose cut, however, we note that a large number of the ancient samples would. By looking at Figure 6 we see that not all of the samples show similar amount of damage. These observations are summarised in Table 1 where we see that Cave-100-forward, Cave-102, Pitch-6 all have more than 60% of their ancient species labelled as damaged according to the loose cut, Cave-22 (18%) and Lake-7-forward (12%) a bit lower, while Lake-9 (0.5%) does not show any clear signs of damage. However, once we condition on the requirement of having more than 100 reads, the fraction of ancient species correctly identified as ancient increases to more than 90% for most the samples.

To better understand the damage estimates, we can look a them individually. Figure 7 shows the Stenotrophomonas maltophilia species from the Pitch-6 sample. We see that none of the fragmentation-only files were estimated to have damage and that most of the deamination and final files including sequencing errors have damage – at a simulation size of 1 million, the significance of both are $Z \approx 1.9$, so this one of the few fits with more than 100 reads that does not pass the loose cut. Furthermore, we notice that the error bars decrease with simulation size, as expected.

The rest of the metagenomic simulation results are shown in Figure XXX.

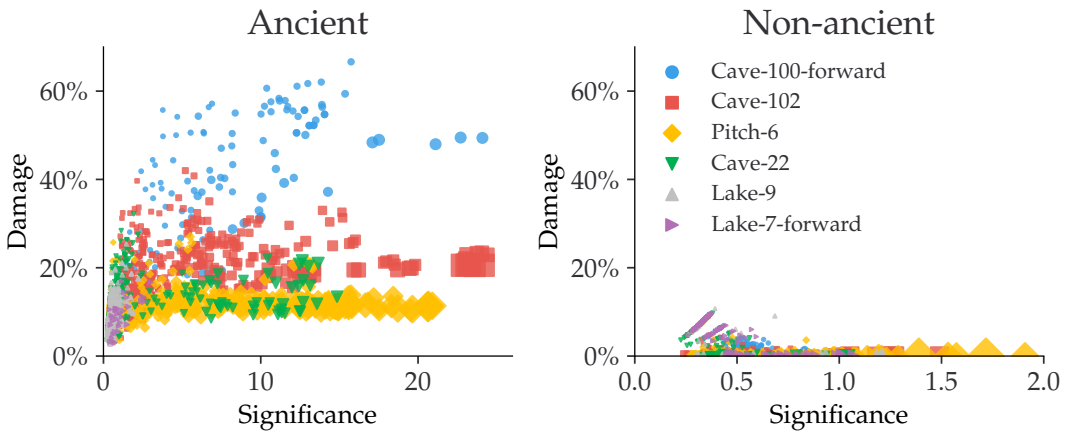


Figure 6. Estimated amount of damage as a function of significance using the ART data. The left figure shows the damage of the species that we simulated to be ancient and the right figure shows the same for the species that have not had deamination added.

Table 1. Number of ancient species for each of the six simulated samples. The first column is the total number of species, the second column is the total number of species that would pass the loose cut of $D > 1\%$ and $Z > 2$, the third column is the number of species with more than 100 reads, and the final column is the number of species with more than 100 reads that also do pass the cut.

Sample	Total	Pass		+100 Reads	+100 Reads and Pass	
Cave-100-forward	135	107	79.3%	88	87	98.9%
Cave-102	500	326	65.2%	309	285	92.2%
Pitch-6	415	260	62.7%	274	260	94.9%
Cave-22	393	71	18.1%	73	69	94.5%
Lake-9	410	2	0.5%	8	0	0%
Lake-7-forward	32	4	12.5%	6	4	66.7%

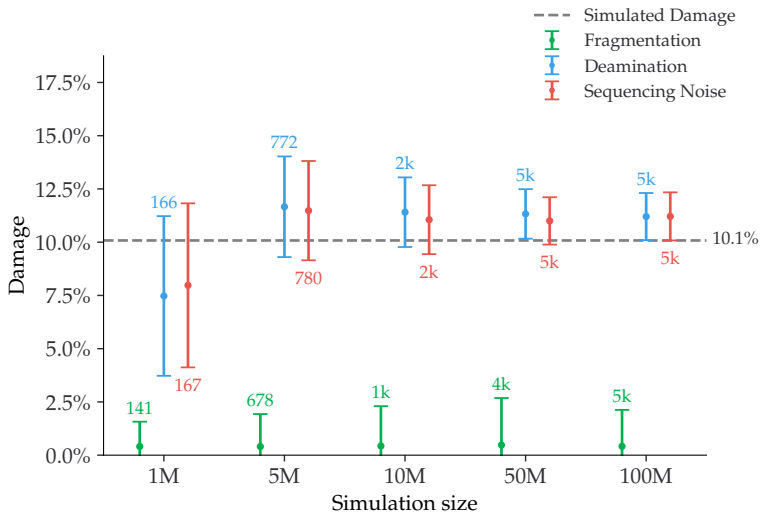


Figure 7. Damage estimates of the Stenotrophomonas maltophilia species from the Pitch-6 sample. Damage is shown as a function of simulation size, with the fragmentation files in green, the deamination files in blue and the final files including sequencing errors in red. All errors are 1σ error bars (standard deviation). The number of reads for each fit is shown as text and since this was a species simulated to have ancient damage, the simulated amount of damage is shown as a dashed grey line.

4.3 | Real Data

The results from running the full `metaDMG` pipeline on real data can be seen in Figure 8. The figures shows Blablabla, real life data here. We find that the loose cut ($D > 1\%$, $Z > 2$) accepts only one of the fits from the control test Library-0, which would not have been accepted by more conservative cut ($D > 2\%$, $Z > 3$, more than 100 reads).

4.4 | Bayesian vs. MAP

Due to increased computational burden of running the full Bayesian model compared to faster, approximate MAP model, in samples with several thousand species, the MAP model is often the most realistic model to use due to time constrains. In this case, it is of course important to know that the damage estimates are indeed trustworthy. Figure 9 compares the estimated damage between the Bayesian model and the MAP model and the estimated significances for species with $D > 1\%$, $Z > 2$ and more than 100 reads. The figure shows that the vast majority of species map 1:1 between the Bayesian and the MAP model. One should note, though, that the few species with the highest mismatch, all are based on forward-only fits, i.e. with no information from the reverse strand, which thus leads to less data to base the fits on. For the comparison with no cuts, see

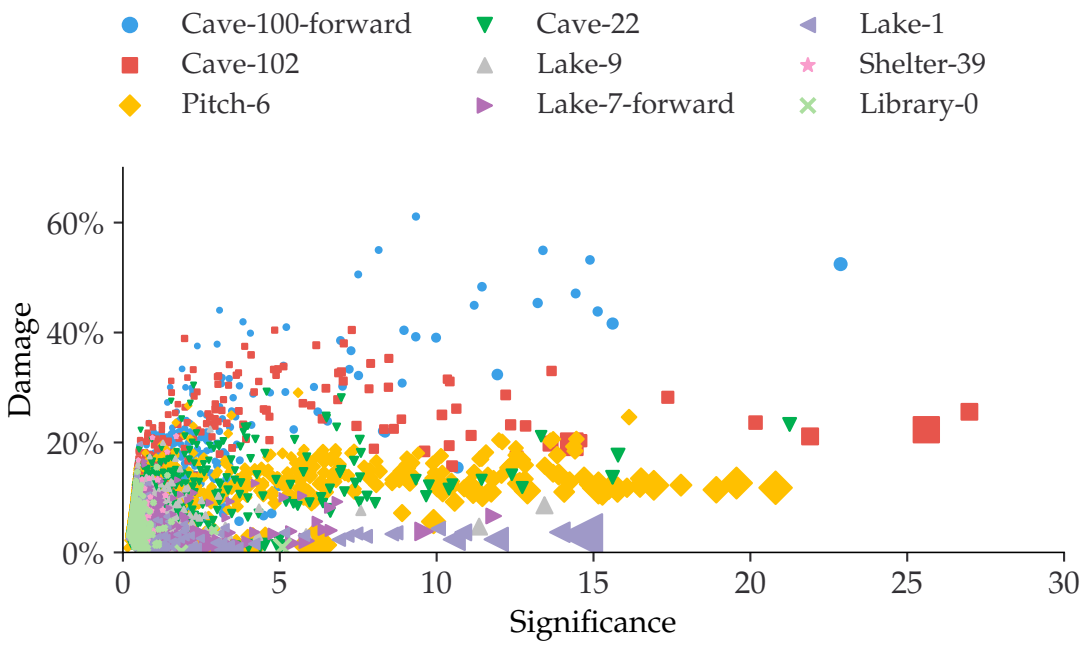


Figure 8. Estimated amount of damage as a function of significance using the real data.

Figure 1 in appendix.

4.5 | Existing Methods

We have also compared `metaDMG` to existing methods such as PyDamage (Borry et al., 2021). Since PyDamage does not include the LCA step, this comparison is based on the non-LCA mode (local-mode) of `metaDMG`. This mode iterates through the different assigned species for all mapped reads and estimates the damage for each. In general, we find that `metaDMG` is more conservative, accurate and precise in its damage estimates.

On example of this is can be found in Figure 10, which shows both the `metaDMG` and PyDamage results of the 100 Homo Sapiens single-genome simulations with 100 reads and 15% added artificial damage (and a fragment length distribution with mean 60).

To compare the computational performance, we use the Pitch-6 sample which has 11.433 unique taxa. When using only a single core, PyDamage took 1105 s to compute all fits, while `metaDMG` took 88 s, a factor of 12.6x faster. Out of the 88 s, `metaDMG` spent 53 s on the actual fits, the rest was for loading and reading the alignment file and computing the mismatch matrices. This makes `metaDMG` more than 20x faster than PyDamage for the fit computation. For the rest of the timings, see Table 2. PyDamage requires the alignment file to be sorted by chromosome position and be supplied with an index file, allowing it to iterate fast through the alignment file, at the expense of computa-

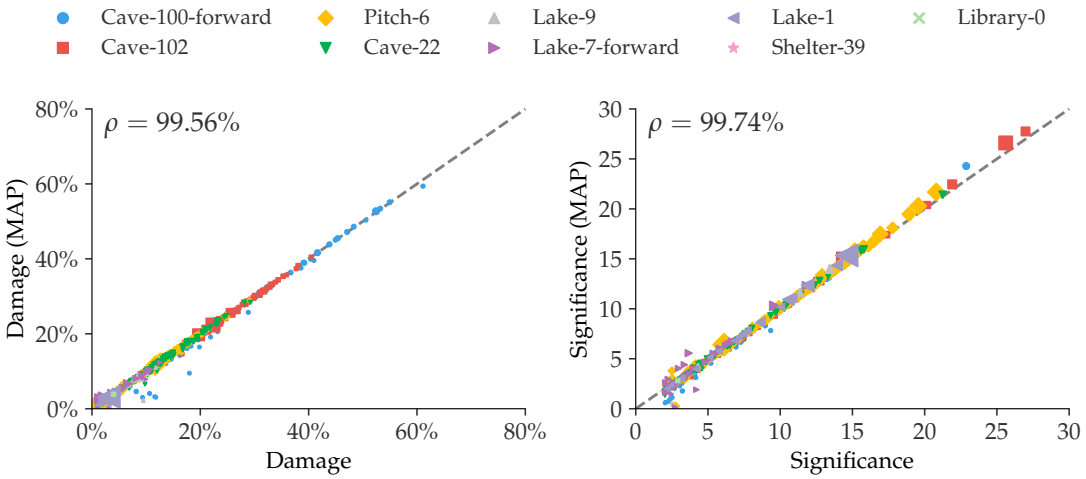


Figure 9. Comparison between the full Bayesian model and the fast, approximate, MAP model for the estimated damage and significance. The figure shows data after a loose cut of $D > 1\%$, $Z > 2$ and more than 100 reads. The dashed, grey line shows the 1:1 ratio and the correlation, ρ, is shown in the upper right corner.

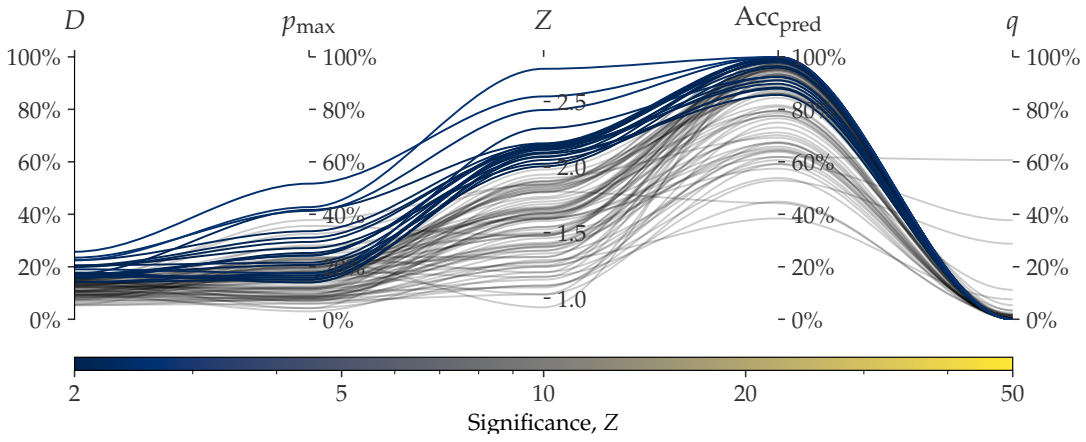


Figure 10. Parallel Coordinates plot comparing `metaDMG` and PyDamage for the Homo Sapiens single-genome simulation with 100 reads and 15% added artificial damage. The different axis shows the five different variables: metaDMG-damage (D, by `metaDMG`), PyDamage-damage ($p_{\max}$, by PyDamage), significance (Z, by `metaDMG`), predicted accuracy (Acc_{pred}, by PyDamage), and the p-value (q, by PyDamage). Each of the 100 simulations are plotted as single lines showing the values of the different dimensions. Simulations with $D > 1\%$ and $Z > 2$, i.e. damaged according to the loose `metaDMG` cut, are shown in color proportional to their significance. Non-damaged simulations are shown in semi-transparent black lines.

Table 2. Computational performance of PyDamage and `metaDMG`. The table contains the times it takes to run either PyDamage or `metaDMG` on the full Pitch-6 sample containing 11.433 species. The timings are shown for both single-processing case (1 core) and multi-processing (2 and 4 cores). The timings were performed on a Macbook M1 Pro model from 2021. "12.6x" means that `metaDMG` was 12.6 times faster than PyDamage for that particular test.

Pitch-6	Pydamage		metaDMG			
Cores	Total	Fits	Total		Fits	
1	1105 s	1102 s	88 s	12.6x	53 s	20.8x
2	592 s	590 s	66 s	9.0x	25 s	23.6x
4	398 s	397 s	54 s	7.4x	14s	28.4x

tional load before running the actual damage estimation. `metaDMG` on the other hand requires the reads to be sorted by name to minimize the time it takes to run the LCA, which however, is not tested in this comparison.

5 | DISCUSSION

Preliminary work indicates that the computational performance of the models can be even further optimized by using Julia (Bezanson et al., 2017), which shows around $7\times$ optimization for the Bayesian model ($\sim 0.2\,\mathrm{s/fit}$) and $4\times$ for the MAP model ($\sim 1.1\,\mathrm{ms/fit}$).

5.1 | Acknowledgment

Acknowledgements here

5.2 | Data availability

Source code is hosted at GitHub: https://github.com/metaDMG-dev. Sequencing data can be found at: https://somewhere.com XXX.

REFERENCES

Betancourt, Michael (2018). "A Conceptual Introduction to Hamiltonian Monte Carlo". In: *arXiv:1701.02434 [stat]*. arXiv: 1701.02434.

Bezanson, Jeff et al. (2017). "Julia: A fresh approach to numerical computing". In: *SIAM review* 59.1. Publisher: SIAM, pp. 65–98. URL: https://julialang.org/.

Borry, Maxime et al. (2021). "PyDamage: automated ancient damage identification and estimation for contigs in ancient DNA de novo assembly". en. In: *PeerJ* 9. Publisher: PeerJ Inc., e11845. ISSN: 2167-8359. DOI: 10.7717/peerj.11845. URL: https://peerj.com/articles/11845 (visited on 2022).

Bradbury, James et al. (2018). *JAX: composable transformations of Python NumPy programs*. URL: http://github.com/google/jax.

Briggs, Adrian W. et al. (2007). "Patterns of damage in genomic DNA sequences from a Neandertal". en. In: *Proceedings of the National Academy of Sciences* 104.37. Publisher: National Academy of Sciences Section: Biological Sciences, pp. 14616–14621. ISSN: 0027-8424, 1091-6490. DOI: 10.1073/pnas.0704665104. URL: https://www.pnas.org/content/104/37/14616.

Cabanski, Christopher R. et al. (2012). "ReQON: a Bioconductor package for recalibrating quality scores from next-generation sequencing data". In: *BMC Bioinformatics* 13.1, p. 221. ISSN: 1471-2105. DOI: 10.1186/1471-2105-13-221. URL: https://doi.org/10.1186/1471-2105-13-221 (visited on 2022).

Cappellini, Enrico et al. (2018). "Ancient Biomolecules and Evolutionary Inference". In: *Annual Review of Biochemistry* 87.1. _eprint: https://doi.org/10.1146/annurev-biochem-062917-012002, pp. 1029–1060. DOI: 10.1146/annurev-biochem-062917-012002. URL: https://doi.org/10.1146/annurev-biochem-062917-012002 (visited on 2022).

Cepeda-Cuervo, Edilberto and MARíA VICTORIA Cifuentes-Amado (2017). "Double Generalized Beta-Binomial and Negative Binomial Regression Models". en. In: *Revista Colombiana de Estadística* 40.1. Publisher: Universidad Nacional de Colombia., pp. 141–163. ISSN: 0120-1751. DOI: 10.15446/rce.v40n1.61779.

Dabney, Jesse, Matthias Meyer, and Svante Pääbo (2013). "Ancient DNA Damage". In: *Cold Spring Harbor Perspectives in Biology* 5.7, a012567. ISSN: 1943-0264. DOI: 10.1101/cshperspect.a012567. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3685887/ (visited on 2022).

Dembinski, Hans et al. (2021). *scikit-hep/iminuit: v2.8.2*. DOI: 10.5281/ZENODO.3949207. (Visited on 2021).

Fernandez-Guerra, Antonio (2022). *genomewalker/aMGSIM-smk: v0.0.1*. DOI: 10.5281/zenodo.7298422. URL: https://doi.org/10.5281/zenodo.7298422.

Ginolhac, Aurelien et al. (2011). "mapDamage: testing for damage patterns in ancient DNA sequences". In: *Bioinformatics* 27.15, pp. 2153–2155. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btr347. URL: https://doi.org/10.1093/bioinformatics/btr347 (visited on 2022).

Huang, Weichun et al. (2012). "ART: a next-generation sequencing read simulator". eng. In: *Bioinformatics (Oxford, England)* 28.4, pp. 593–594. ISSN: 1367-4811. DOI: 10.1093/bioinformatics/btr708.

Jónsson, Hákon et al. (2013). "mapDamage2.0: fast approximate Bayesian estimates of ancient DNA damage parameters". en. In: *Bioinformatics* 29.13, pp. 1682–1684. ISSN: 1367-4803, 1460-2059. DOI: 10.1093/bioinformatics/btt193.

Lam, Siu Kwan, Antoine Pitrou, and Stanley Seibert (2015). "Numba: a LLVM-based Python JIT compiler". In: *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*. LLVM '15. New York, NY, USA: Association for Computing Machinery, pp. 1–6. ISBN: 978-1-4503-4005-2. DOI: 10.1145/2833157.2833162. URL: https://github.com/numba/numba.

Langmead, Ben and Steven L. Salzberg (2012). "Fast gapped-read alignment with Bowtie 2". en. In: *Nature Methods* 9.4. Number: 4 Publisher: Nature Publishing Group, pp. 357–359. ISSN: 1548-7105. DOI: 10.1038/nmeth.1923. URL: https://www.nature.com/articles/nmeth.1923 (visited on 2022).

Llamas, Bastien et al. (2017). "From the field to the laboratory: Controlling DNA contamination in human ancient DNA research in the high-throughput sequencing era". en. In: *STAR: Science & Technology of Archaeological Research* 3.1, pp. 1–14. ISSN: 2054-8923. DOI: 10.1080/20548923.2016.1258824. URL: https://www.tandfonline.com/doi/full/10.1080/20548923.2016.1258824 (visited on 2022).

McElreath, Richard (2020). *Statistical rethinking: a Bayesian course with examples in R and Stan*. 2nd ed. CRC texts in statistical science. Boca Raton: Taylor and Francis, CRC Press. ISBN: 978-0-367-13991-9.

Mölder, Felix et al. (2021). *Sustainable data analysis with Snakemake*. en. Tech. rep. 10:33. Type: article. F1000Research. DOI: 10.12688/f1000research.29032.2. URL: https://f1000research.com/articles/10-33 (visited on 2022).

Nayfach, Stephen et al. (2021). "CheckV assesses the quality and completeness of metagenome-assembled viral genomes". en. In: *Nature Biotechnology* 39.5. Number: 5 Publisher: Nature Publishing Group, pp. 578–585. ISSN: 1546-1696. DOI: 10.1038/s41587-020-00774-7. URL: https://www.nature.com/articles/s41587-020-00774-7 (visited on 2022).

NCBI Resource Coordinators (2018). "Database resources of the National Center for Biotechnology Information". In: *Nucleic Acids Research* 46.D1, pp. D8–D13. ISSN: 0305-1048. DOI: 10.1093/nar/gkx1095. URL: https://doi.org/10.1093/nar/gkx1095 (visited on 2022).

Neukamm, Judith, Alexander Peltzer, and Kay Nieselt (n.d.). "DamageProfiler: Fast damage pattern calculation for ancient DNA". en. In: (), p. 10.

Parks, Donovan H. et al. (2018). "A standardized bacterial taxonomy based on genome phylogeny substantially revises the tree of life". en. In: *Nature Biotechnology* 36.10. Number: 10 Publisher: Nature Publishing Group, pp. 996–1004. ISSN: 1546-1696. DOI: 10.1038/nbt.4229. URL: https://www.nature.com/articles/nbt.4229 (visited on 2022).

Phan, Du, Neeraj Pradhan, and Martin Jankowiak (2019). "Composable Effects for Flexible and Accelerated Probabilistic Programming in NumPyro". In: *arXiv:1912.11554 [cs, stat]*. arXiv: 1912.11554.

Plotly, Technologies (2015). *Collaborative data science*. Place: Montreal, QC Publisher: Plotly Technologies Inc. URL: https://plot.ly.

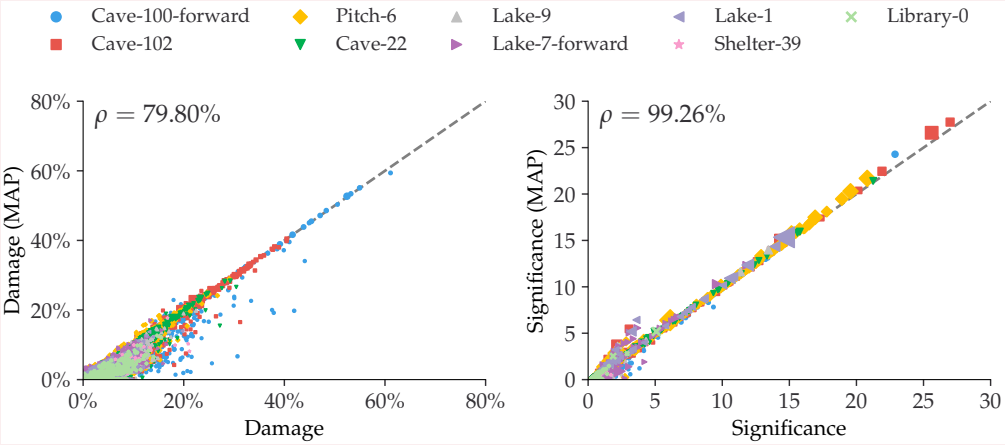
Skoglund, Pontus et al. (2014). "Separating endogenous ancient DNA from modern day contamination in a Siberian Neandertal". In: *Proceedings of the National Academy of Sciences* 111.6. Publisher: Proceedings of the National Academy of Sciences, pp. 2229–2234. DOI: 10.1073/pnas.1318934111. URL: https://www.pnas.org/doi/10.1073/pnas.1318934111 (visited on 2022).

Wang, Yucheng et al. (2022). "ngsLCA—A toolkit for fast and flexible lowest common ancestor inference and taxonomic profiling of metagenomic data". en. In: *Methods in Ecology and Evolution* n/a.n/a (). _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/2041-210X.14006. ISSN: 2041-210X. DOI: 10.1111/2041-210X.14006. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/2041-210X.14006 (visited on 2022).

Appendix 1

A | EXAMPLE FIGURE

This is an example of including a figure in the appendix.



Appendix 1—figure 1. Comparison between the full Bayesian model and the fast, approximate, MAP model for the estimated damage and significance. The dashed, grey line shows the 1:1 ratio.

B | EXAMPLE TABLE

This is an example of including a table in the appendix.

Appendix 2—table 1. An example table.

Speed (mph)	Driver	Car	Engine	Date
407.447	Craig Breedlove	Spirit of America	GE J47	8/5/63
413.199	Tom Green	Wingfoot Express	WE J46	10/2/64
434.22	Art Arfons	Green Monster	GE J79	10/5/64
468.719	Craig Breedlove	Spirit of America	GE J79	10/13/64
526.277	Craig Breedlove	Spirit of America	GE J79	10/15/65
536.712	Art Arfons	Green Monster	GE J79	10/27/65
555.127	Craig Breedlove	Spirit of America, Sonic 1	GE J79	11/2/65
576.553	Art Arfons	Green Monster	GE J79	11/7/65
600.601	Craig Breedlove	Spirit of America, Sonic 1	GE J79	11/15/65
622.407	Gary Gabelich	Blue Flame	Rocket	10/23/70
633.468	Richard Noble	Thrust 2	RR RG 146	10/4/83
763.035	Andy Green	Thrust SSC	RR Spey	10/15/97

C | LIKELIHOOD CALCULATION

C.1 | Full multinomial logistic Regression models

Postmortem damages will have impacts on the NGS (next generation sequencing) reads. A common phenomenon is the calling error rates increases from nucleotide C to T due to the cytosine deamination process. Unawareness of this will lead to inaccurate inferences. Evidences show that the magnitude of such changes are related to the positions the site is within a read (the fraction of the ancient DNA). Here we present 3 slightly different ways to unveil the relationship between the calling error rates and the mismatching reference/read pairs as well as the site positions within a read. The methods are based on the multinomial logistic regressions.

Data Description

We perform the regression based on the summary statistic of the mismatch matrix which is a table which contains the counts of reads of different reference/read categories (in total 16) and positions on the forward/reversed strand (15 positions on each direction). Table 1 and Table 2 give an example of the data format we use for the inference.

Read Counts								
Ref.	A				C			
Read	A	C	G	T	A	C	G	T
1	12794053	8325	28769	16073	10404	8045811	8020	2092619
2	13480290	6812	21107	12102	9151	8260185	6531	1145605
3	12760253	6131	18859	10327	7772	8385423	5899	914709
4	12995572	5240	17671	8940	7880	8345892	5252	767237
5	12930102	4601	17021	8188	8374	8474964	5161	703283
6	12879355	4684	16435	7536	8726	8571141	4811	643607
7	12684349	4557	15298	7394	8835	8727254	4762	586674
8	12585563	4454	15497	7236	8898	8888173	5058	527691
9	12468622	4309	14704	6942	8948	9076851	4673	481170
10	12491183	4437	14567	6912	9103	9237982	4702	443329
11	12430899	4296	14083	6515	9313	9364121	4609	404431
12	12419506	4226	13985	6503	9342	9357468	4367	371475
13	12469412	4147	13851	6375	9586	9386737	4588	345390
14	12549936	4045	13650	6246	9673	9324488	4628	322294
15	12566555	4174	13499	6213	9735	9305820	4518	301360
-1	11599167	8800	16164	14851	90888	9613102	10843	19810
-2	11985637	8769	14044	12040	28799	9561124	7184	18424
-3	12941743	7805	13861	12001	24988	9400151	6368	15466
-4	12808985	7141	12885	9889	23067	9509723	5421	14901
-5	12869585	6954	12100	9428	22349	9464831	5789	13987
-6	12784911	6440	12080	8735	20556	9566794	6544	14021
-7	12878349	5946	12311	8225	19480	9566359	6478	16419
-8	12719722	9521	12156	8131	19226	9725468	6709	23434
-9	12652860	5634	11940	7671	18035	9762224	6321	31667
-10	12566817	5448	11850	7178	17353	9701382	6306	37831
-11	12702498	5309	12092	7568	16121	9526031	6035	43215
-12	12731940	5207	11933	6856	15637	9533858	5557	47650
-13	12697647	4989	12199	7153	15072	9508117	5434	51614
-14	12689924	4944	11891	6816	15050	9525285	5237	55598
-15	12660634	4746	11753	6732	14815	9561359	5184	59633

Appendix 3—table 1. The read counts per position given the reference nucleotides are A or C of an ancient human data. The negative position indices are the position on the reversed strand. In the manuscript, the elements (the values of a specific nucleotide read counts per position given the reference nucleotide is A or C) in this table are denoted as $o_{\mathrm{A}\to i,p}$ or $o_{\mathrm{C}\to i,p}$.

Read Counts								
Ref.	G				T			
Read	A	C	G	T	A	C	G	T
1	16389	8976	9639767	86584	11733	15878	8351	11718463
2	17614	6483	9510149	26655	10761	13958	7011	11974947
3	15164	5949	9488917	23374	9509	13767	6046	12839015
4	14844	5186	9566468	21960	8170	12509	5585	12721790
5	14005	5612	9497118	20468	7186	11991	5233	12795244
6	13671	6195	9622572	19096	6948	11683	4790	12686645
7	16648	6394	9609855	18594	6203	12122	4780	12794172
8	23659	6405	9768666	17341	6131	11847	4758	12626614
9	31680	6139	9785449	17034	5998	12040	4469	12579260
10	38484	5982	9700857	16235	5487	11546	4175	12513653
11	44665	5722	9536341	15284	5651	12044	4176	12646627
12	48949	5371	9547134	14569	5449	11663	4060	12684645
13	53076	5234	9543953	14090	5262	11785	4046	12631297
14	57343	5186	9551477	13855	5257	11768	4006	12624840
15	61236	5137	9583481	13667	5122	11733	3947	12612416
-1	2078554	7947	8096447	11847	15732	28461	8551	12890628
-2	1138478	6656	8232666	10760	12299	20759	6999	13446882
-3	921712	5970	8399013	8643	10514	18226	6564	12718084
-4	775038	5720	8319235	8416	9415	17800	5388	12977322
-5	710955	5499	8462058	8926	8526	17088	4911	12886576
-6	647761	5052	8545455	9193	7640	16351	4879	12852322
-7	593854	4872	8693834	9318	7600	15523	5048	12664576
-8	535542	7828	8889921	9399	7163	18704	4718	12510123
-9	486549	4696	9075263	9522	7109	14547	4611	12409220
-10	448895	4622	9226758	9432	6816	14567	4668	12438344
-11	409027	4654	9352528	9544	6575	14019	4611	12388650
-12	376069	4637	9344701	9419	6511	13874	4486	12390148
-13	350609	4655	9384853	9885	6197	13877	4327	12432024
-14	326760	4595	9337266	9889	5986	13928	4403	12490990
-15	305014	4541	9310617	10065	5919	13442	4232	12529684

Appendix 3—table 2. The read counts per position given the reference nucleotides are G or T of the same human data as in Table 1. The negative position indices are the position on the reversed strand. In the manuscript, the elements (the values of a specific nucleotide read counts per position given the reference nucleotide is G or T) in this table are denoted as $o_{\mathrm{G}\to i,p}$ or $o_{\mathrm{T}\to i,p}$.

The terminology used here might not be standard. The term full regression here is to distinguish itself from the folded regression discussed later, which simply means inferring the coefficients of forward strand and reversed strand separately. Full regression includes both unconditional regression and conditional regression. The unconditional regression's objective is to infer the probability of observing a read of nucleotide j and its reference is i at position p, i.e., $\mathrm{P}(\mathrm{Obs}:i\to j|\mathrm{Pos}:p)$; while the unconditional regression's target is to estimate the probability of observing a read of nucleotide j given its reference is i at position p, i.e., $\mathrm{P}(\mathrm{Obs}:j|\mathrm{Ref}:i,\mathrm{Pos}:p)$. Their relationship is as follows:

$$\mathrm{P}(\mathrm{Obs}:j|\mathrm{Ref}:i,\mathrm{Pos}:p)=\frac{\mathrm{P}(\mathrm{Obs}:i\to j|\mathrm{Pos}:p)}{\sum_j \mathrm{P}(\mathrm{Obs}:i\to j|\mathrm{Pos}:p)}.$$

So in fact, unconditional regression can give us more detailed inferred results (extra information the nucleotide composition per position of the reference, which may be related to the prepared libraries).

Unconditional Regression likelihood

$$\begin{aligned} l_1 &= \sum_p \sum_{i,j\in\{\mathrm{A,C,G,T}\}} o_{i\to j,p}\log \mathrm{P}_{i,j|p} \\ &= \sum_p \left[o_p \log \mathrm{P}_{\mathrm{TT}|p} + \sum_{(i,j)\neq \mathrm{TT}} o_{i\to j,p}\log\frac{\mathrm{P}_{ij|p}}{\mathrm{P}_{\mathrm{TT}|p}} \right], \end{aligned} \tag{10}$$

where $\mathrm{P}_{ij|p}=\mathrm{P}(\mathrm{Obs}:i\to j|\mathrm{Pos}:p)$, and $o_p=\sum_{i,j\in\{\mathrm{A,C,G,T}\}} o_{i\to j,p}$.

$$\log\frac{\mathrm{P}_{ij|p}}{\mathrm{P}_{\mathrm{TT}|p}}=\sum_{n=0}^{\mathrm{order}}\alpha_{i,j,p,n}p^n \tag{11}$$

$$\begin{aligned} l_1 &= \sum_p \left\{ -o_p \log\left[1+\sum_{(i,j)\neq\mathrm{TT}}\exp\left(\sum_{n=0}^{\mathrm{order}}\alpha_{i,j,p,n}p^n\right)\right] + \sum_{(i,j)\neq\mathrm{TT}} o_{i\to j,p}\sum_{n=0}^{\mathrm{order}}\alpha_{i,j,p,n}p^n \right\} \\ &= l_{1,5'}+l_{1,3'}. \end{aligned} \tag{12}$$

The number of inferred parameters for the full conditional regression is $30\,(\text{order}+1)$.

$$\frac{\partial l_1}{\partial \alpha_{i,j,p,n}} = -o_p \frac{p^n \exp\left(\sum_{n=0}^{\text{order}} \alpha_{i,j,p,n} p^n\right)}{1 + \sum_{(i,j)\neq \text{TT}} \exp\left(\sum_{n=0}^{\text{order}} \alpha_{i,j,p,n} p^n\right)} + o_{i\to j,p} p^n. \tag{13}$$

Conditional Regression likelihood

$$\begin{aligned} l_2 &= \sum_{i\in\{\text{A,C,G,T}\}} \sum_p \sum_{j\in\{\text{A,C,G,T}\}} o_{i\to j,p} \log \text{P}_{j|p,i} \\ &= \sum_{i\in\{\text{A,C,G,T}\}} \sum_p \left[o_{i,p} \log \text{P}_{\text{T}|p,i} + \sum_{j\neq \text{T}} o_{i\to j,p} \log \frac{\text{P}_{j|p,i}}{\text{P}_{\text{T}|p,i}} \right], \end{aligned} \tag{14}$$

where $\text{P}_{j|p,i} = \text{P}(\text{Obs}: j|\text{Ref}: i, \text{Pos}: p)$, and $o_{i,p} = \sum_{j\in\{\text{A,C,G,T}\}} o_{i\to j,p}$.

$$\log \frac{\text{P}_{j|p,i}}{\text{P}_{\text{T}|p,i}} = \sum_{n=0}^{\text{order}} \beta_{i,j,p,n} p^n \tag{15}$$

$$\begin{aligned} l_2 &= \sum_{i\in\{\text{A,C,G,T}\}} \sum_p \left\{ -o_{i,p} \log\left[1 + \sum_{j\neq \text{T}} \exp\left(\sum_{n=0}^{\text{order}} \beta_{i,j,p,n} p^n\right)\right] + \sum_{j\neq \text{T}} o_{i\to j,p} \sum_{n=0}^{\text{order}} \beta_{i,j,p,n} p^n \right\} \\ &= l_{2,\text{A},5'} + l_{2,\text{C},5'} + l_{2,\text{G},5'} + l_{2,\text{T},5'} + l_{2,\text{A},3'} + l_{2,\text{C},3'} + l_{2,\text{G},3'} + l_{2,\text{T},3'}. \end{aligned} \tag{16}$$

The number of inferred parameters for the full unconditional regression is $24\,(\text{order}+1)$.

$$\frac{\partial l_2}{\partial \beta_{i,j,p,n}} = -o_{i,p} \frac{p^n \exp\left(\sum_{n=0}^{\text{order}} \beta_{i,j,p,n} p^n\right)}{1 + \sum_{j\neq \text{T}} \exp\left(\sum_{n=0}^{\text{order}} \beta_{i,j,p,n} p^n\right)} + o_{i\to j,p} p^n. \tag{17}$$

Folded Regression

The folded regressions use the same log-likelihood function as the full regression (i.e., Equation) but are conducted based on the assumptions that,

$$\alpha_{i,j,p,n} = \alpha_{c(i),c(j),-p,n}, \tag{18}$$

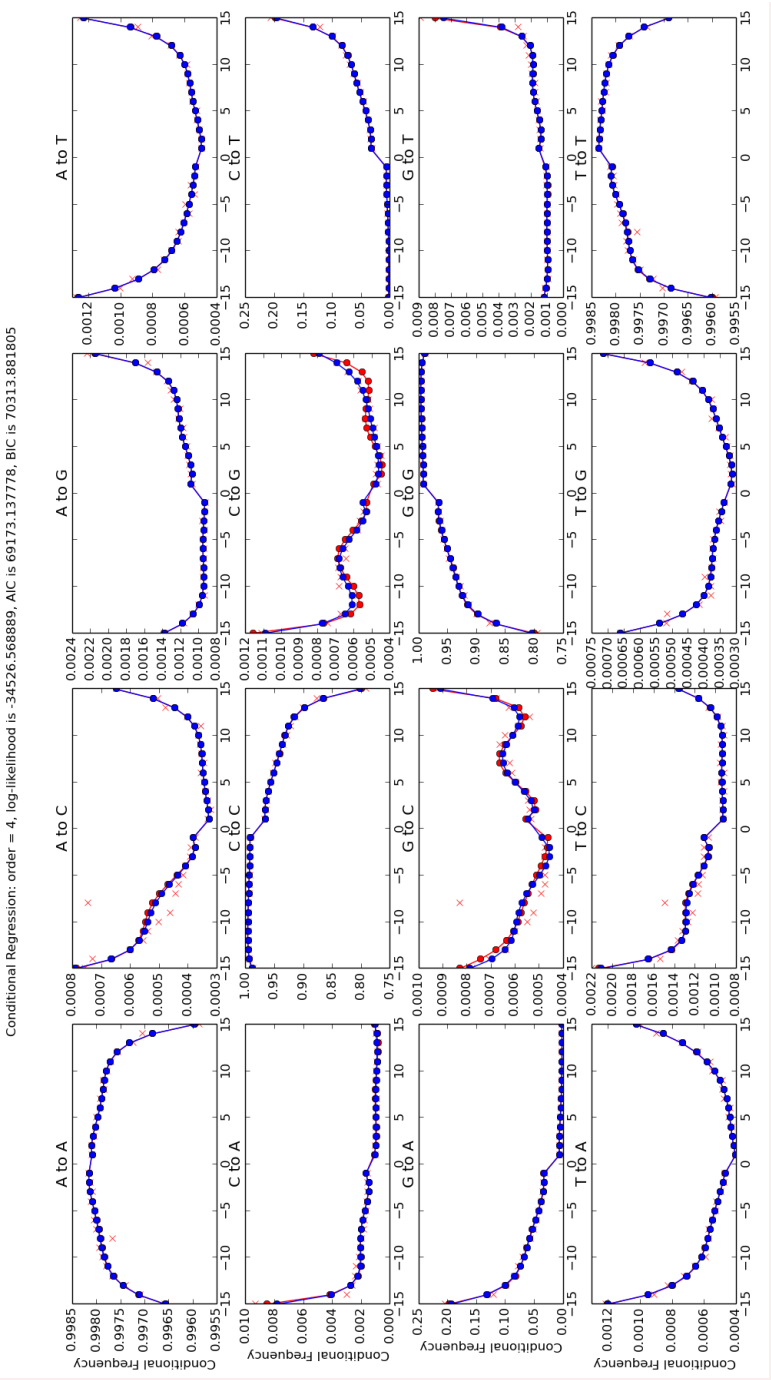
$$\beta_{i,j,p,n} = \beta_{c(i),c(j),-p,n}, \tag{19}$$

where $c(i)$ means the complimentary nucleotide of the nucleotide i, e.g., $c(\text{A}) = \text{T}$ and $c(\text{G}) = \text{C}$. Our data (both taxon and human data) and some models studies seem to support this assumption.

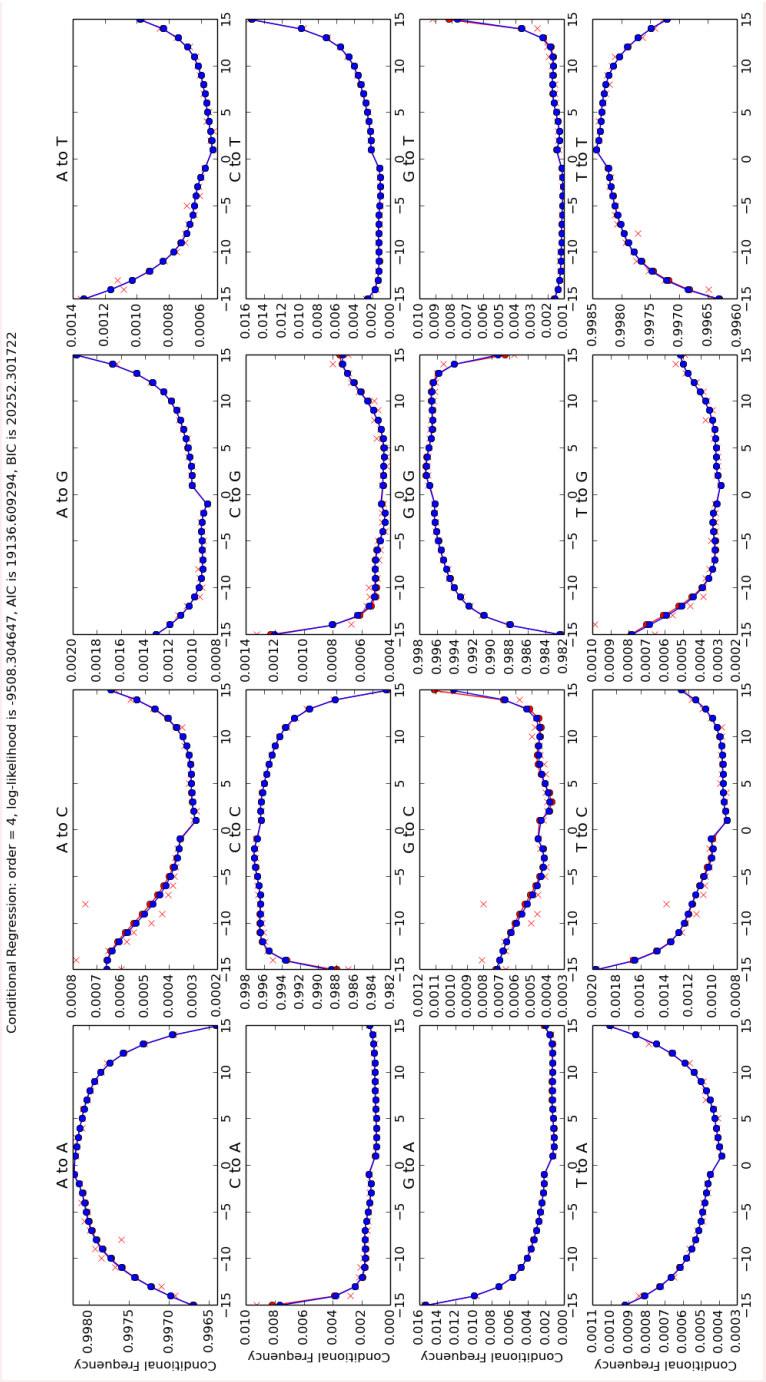
By doing the folded regression, we halve the number of inferred parameters. Hence The number of inferred parameters for the folded unconditional regression is $15\,(\text{order}+1)$, and that of folded conditional regression is $12\,(\text{order}+1)$.

Results for multinomial logistic regression

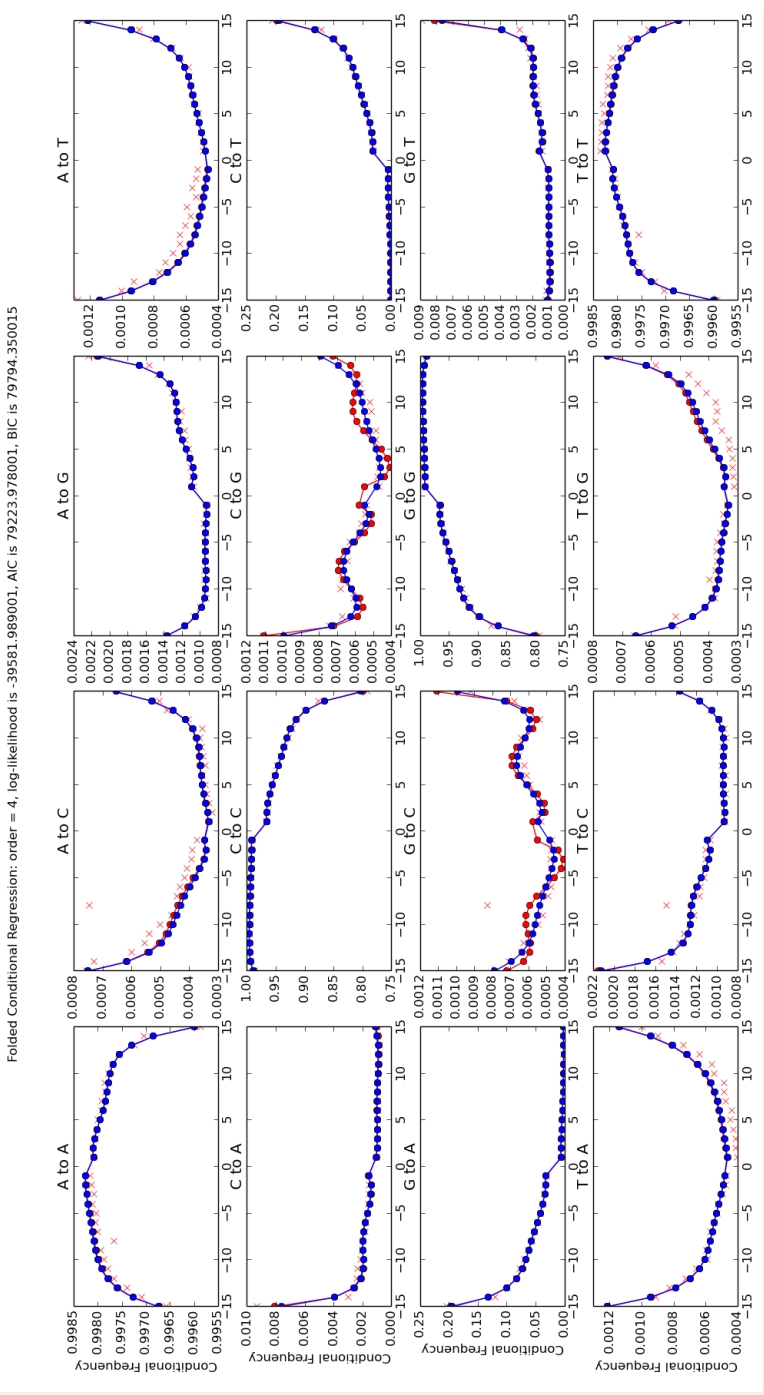
Currently, the optimization of the likelihood functions are based on the C++ library of gsl and use the function *gsl_multimin_fminimizer_nmsimplex2*. with the initial searching point is set to be the results of logistic regression. We here present here 4 figures pertaining to showcase the performance of our model. The regression methods are based on the summary statistic of the counts of mismatches and the optimization is therefore in the scale of miliseconds. Fig. 1 and Fig. 2 are the conditional regression results of the ancient and control human data correspondingly. And Fig. 3 and Fig. 4 are the folded conditional regression results of the same data as above. Our codes can also do the unconditional regression, but I have not generated the results for now.



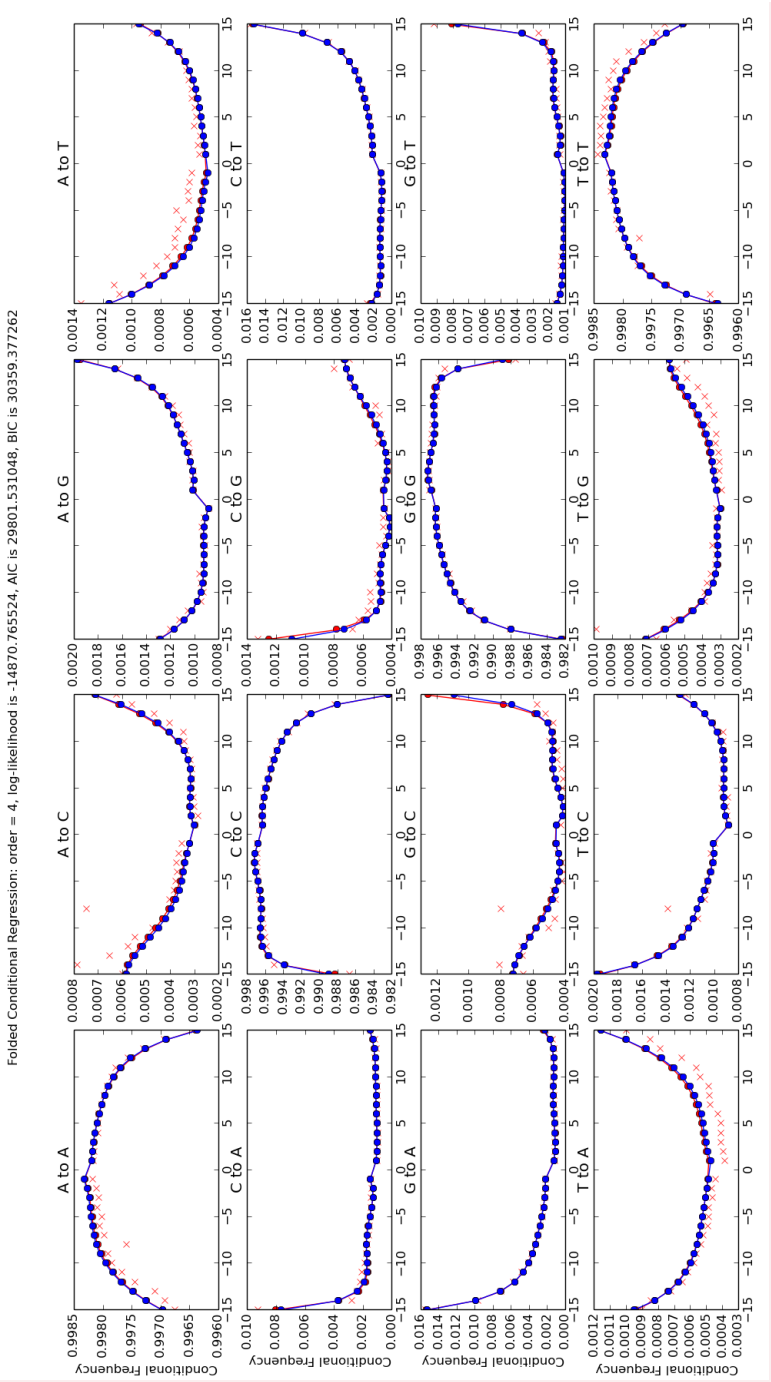
Appendix 3—figure 1. Conditional regression results with the order 4 of the ancient human data. Each panel of figure represents a specific reference/read pair and plots its frequency across different positions. The positions from left to right are −1 to −15 and 15 to 1.



Appendix 3—figure 2. Conditional regression results with the order 4 of the control human data. Each panel of figure represents a specific reference/read pair and plots its frequency across different positions. The positions from left to right are −1 to −15 and 15 to 1.



Appendix 3—figure 3. Folded conditional regression results with the order 4 of the ancient human data. Each panel of figure represents a specific reference/read pair and plots its frequency across different positions. The positions from left to right are −1 to −15 and 15 to 1.



Appendix 3—figure 4. Folded conditional regression results with the order 4 of the control human data. Each panel of figure represents a specific reference/read pair and plots its frequency across different positions. The positions from left to right are −1 to −15 and 15 to 1.