CarpeDeam: A *De Novo* Metagenome Assembler for Heavily Damaged Ancient Datasets

Louis Kraft[1*], Johannes Söding[4,5,6], Martin Steinegger[7,8,9], Annika Jochheim[4,5], Peter Wad Sackett[1], Antonio Fernandez-Guerra[†2,3], and Gabriel Renaud[†1,*]

[1]Department of Health Technology, Section for Bioinformatics, Technical University of Denmark, Kongens Lyngby, Denmark
[2]Centre for Ancient Environmental Genomics, Globe Institute, University of Copenhagen, Copenhagen, Denmark
[3]Lundbeck Foundation GeoGenetics Centre, Globe Institute, University of Copenhagen, Copenhagen, Denmark
[4]Quantitative and Computational Biology Group, Max Planck Institute for Multidisciplinary Sciences, Göttingen, Germany
[5]International Max-Planck Research School for Genome Sciences (IMPRS-GS)
[6]Campus Institute Data Science (CIDAS), University of Göttingen, Germany
[7]School of Biological Sciences, Seoul National University, Seoul, South Korea
[8]Artificial Intelligence Institute, Seoul National University, Seoul, South Korea
[9]Institute of Molecular Biology and Genetics, Seoul National University, Seoul, South Korea
[*]Correspondence: lokraf@dtu.dk, gabriel.reno@gmail.com
[†]These authors contributed equally to this work.

August 9, 2024

Abstract

De novo assembly of ancient metagenomic datasets is a challenging task. Ultra-short fragment size and characteristic postmortem damage patterns of sequenced ancient DNA molecules leave current tools ill-equipped for ideal assembly. We present `CarpeDeam`, a novel damage-aware *de novo* assembler designed specifically for ancient metagenomic samples. Utilizing maximum-likelihood frameworks that integrate sample-specific damage patterns, `CarpeDeam` recovers longer continuous sequences and more protein sequences from both simulated and empirical datasets compared to existing assemblers. As a pioneering ancient metagenome assembler, `CarpeDeam` opens the door for new opportunities in functional and taxonomic analyses of ancient microbial communities.

1 Keywords

Ancient DNA, Metagenomics, *de novo* Assembly, Microbes, Proteins

2 Background

DNA recovered from ancient organisms, termed ancient DNA (aDNA), has transformed evolutionary science. Besides the exploration of ancestral human populations, aDNA enables the reconstruction of past environments covering eukaryotic species diversity and microbial community dynamics [1, 2, 3, 4]. However, the computational analysis of aDNA is challenging [5, 6]. Two primary processes lead to the breakdown and chemical alteration of aDNA. Hydrolytic mechanisms cause the degradation of DNA molecules and deamination converts cytosines into uracil molecules. These are misinterpreted as putative base substitutions during DNA sequencing [7]. This phenomenon, known as "damage", occurs predominantly at the ends of aDNA fragments, and the rate of occurrence of damage at each fragment position can be quantified [8, 9, 10, 11]. As a result, aDNA fragments show elevated frequencies of C→T substitutions at the 5′ end and G→A substitutions

at the 3′ end, respectively [12]. Along with the short fragment size, damage profiles are commonly used for aDNA authentication [13, 14, 15, 16]. However, the ambiguity introduced by deaminated bases hampers downstream analysis such as read mapping or genome assembly [17, 18, 19].

Metagenomes represent the collective genetic material from various species within a sample, derived from environments such as the human gut or soil, including bacteria, archaea, viruses, and eukaryotes [20, 21]. The field has expanded with the advent of massive parallel sequencing, leading to a deeper understanding of microbial diversity and functional profiling on Earth [22, 23, 24]. However, analysing metagenomic datasets relies on dedicated tools as the data is inherently complex due to high species diversity and varying abundance profiles, and the choice of tools depends on the specific research goals [25]. There are two main approaches that follow different objectives when analysing metagenomic data: taxonomic classification and functional annotation [26].

Taxonomic classification categorizes genomic sequences to determine microbial composition, thus offering an overview of the taxonomic diversity [27]. Numerous tools have been developed, utilizing both read-based and assembly-based approaches. Read-based methods employ various strategies such as marker gene analysis, database alignment, or k-mer-based techniques [28, 27, 29, 30]. Assembly-based methods, which work with either assembled contigs or long reads, potentially increase classification accuracy. These approaches rely on tailored data structures or alignment techniques to efficiently process large numbers of sequences [31, 32, 33, 34]. Notably, taxonomic classification has been widely applied to ancient metagenomes [35, 36, 37, 38, 39, 3], with specialized tools developed to address the unique challenges posed by degraded, short aDNA fragments [40, 41].

While functional annotation of sequences can be achieved by mapping reads to protein databases, assembly-based methods often yield more accurate and comprehensive results. This is particularly the case in metagenomic studies where many unknown or evolutionary distant species are present [42]. In these scenarios, *de novo* assembly is commonly chosen, as it can uncover a wide range of elements, from protein-coding genes to entire genomes. The advantages of *de novo* assembly are especially noticeable when working with ancient samples. Distantly related species in ancient samples may not map properly to modern references. Additionally, short fragment lengths and typical aDNA damage patterns can further complicate alignment-based approaches [43].

Several studies have demonstrated the applicability of *de novo* assembly for searching aDNA samples for antibiotic-resistance genes or unknown metabolites, or for reconstructing whole genomes [44, 17, 45, 46, 47, 48, 49]. For instance, a study by Wan *et al.* [50] recently demonstrated the potential of mining proteomes from extinct species, termed the "extinctome", to identify novel antimicrobial genes. Considering that approximately 90% of prokaryotic genomes are protein-coding [51, 52, 53], *de novo* assembly of extinct species is the gateway for revealing unknown proteomes. While Klapper *et al.* [17] highlighted the fact that deeply sequenced datasets can be assembled using conventional assembly algorithms, the assembly process becomes more challenging when dealing with samples that exhibit high damage rates and low coverage.

To demonstrate the effects of varying fragment length and damage patterns, we simulated fragments of a simple metagenomic environment with different fragment sizes and deamination rates. As shown in Fig. **1(A)**, fragment length and damage drastically influence the outcome of ancient metagenome assembly. The damage patterns used for the simulations (Fig. **1(B)**) represent two levels of damage intensities as they can be found in studies profiling ancient microorganisms. Even at moderate damage levels with the first position of the 5′ end exhibiting a damage rate of ∼35%, we observed a significant drop in assembly performance. Several studies have documented rates of nucleotide misincorporation at the 5′ end, ranging from 15% to 60% [17, 36, 54, 55, 56, 3, 57]. Our findings highlight the need for optimized assembly algorithms to address the unique characteristics of ancient metagenomic datasets. While aDNA fragments cannot be computationally lengthened, damage could be mitigated via bespoke algorithms.

There are two major classes of *de novo* assembly algorithms: Overlap Layout Consensus (OLC) and de Bruijn graphs. OLC algorithms rely on the computation of full read overlaps, while de Bruijn graph algorithms construct contigs using sub-sequences of length k (k-mers) [60]. Although OLC algorithms are precise, they are computationally impractical for assembling large datasets produced by high-throughput sequencing technologies [61, 62]. Consequently, de Bruijn graph algorithms have become the prevalent algorithmic basis for most assemblers developed in the past decade. De Bruijn graph assemblers connect k-mers that overlap by at least k-1 nucleotides in a graph structure, thereby efficiently assembling short reads into contigs [63, 64, 58]. However, de Bruijn graph assemblers face a precision-sensitivity trade-off when assembling complex metagenomic data.

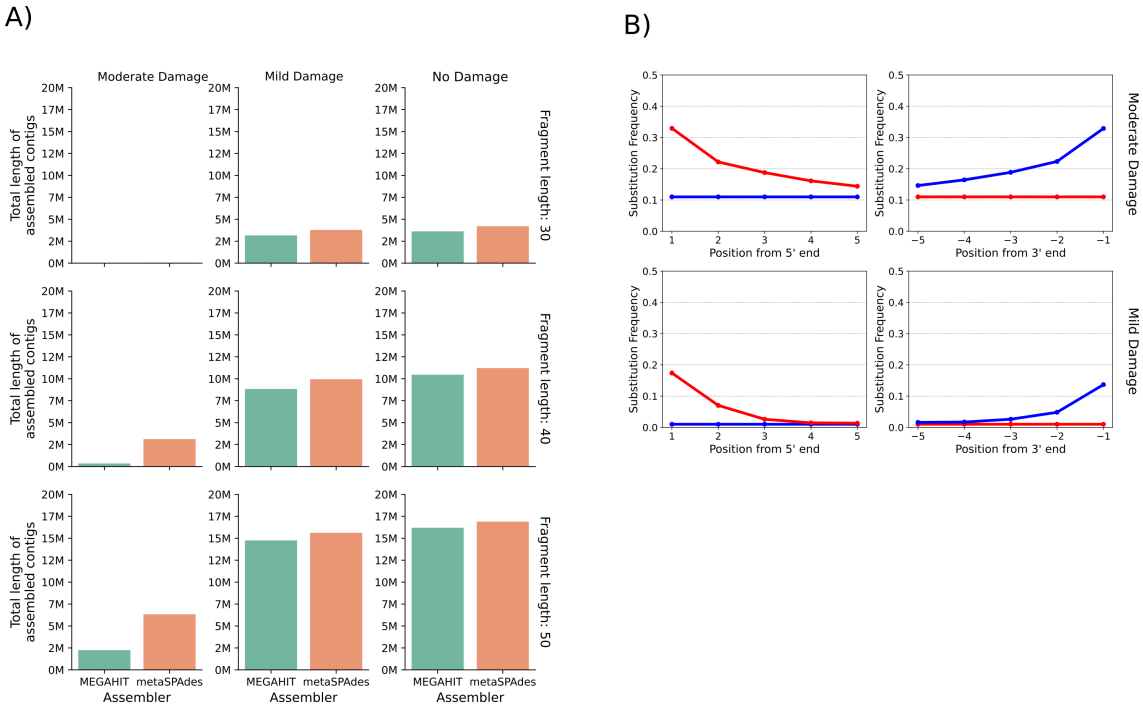

Figure 1: **A)** Impact of aDNA damage and fragment length on metagenomic assembly. The plots show the sum of all contigs larger than 500bp after assembling a simulated toy dataset with either `MEGAHIT` [58] or `metaSPAdes` [59]. Each bar plot refers to a different combination of fragment length and damage rate of the simulated data. While empirical data is inherently more complex in terms of variation in fragment lengths and damage patterns, our simplified dataset demonstrates how common assemblers are limited by aDNA damage. **B)** Damage patterns used for the simulation of aDNA fragments. We used two levels of deamination rates for the simulations: moderate damage and mild damage compared to the rates found in ancient microbial studies [17, 36, 54, 55, 56, 3, 57]. The blue traces represent C-to-T substitution rates, while the red traces indicate G-to-A substitution rates.

Longer k-mers provide higher specificity for individual species but suffer from reduced sensitivity in populations with high intra-species diversity. Conversely, shorter k-mers offer increased sensitivity but lack the specificity required to effectively distinguish between closely related species [65, 62]. Moreover, as k-mer length increases, sequencing errors are more likely to interfere with matching k-mers correctly, making read correction methods necessary to address this issue. Ancient metagenomic datasets, characterized by damaged and short fragments, amplify this issue by naturally increasing diversity and complexity and thereby stretching the limits of de Bruijn graph assemblers [6]. New assembly strategies are needed to overcome these issues.

The *de novo* metagenomic assembler `PenguiN` [62] uses a greedy-iterative overlap-based assembly method. Inspired by the protein level assembler `PLASS` [65], it clusters reads based on shared k-mers. Extension candidates are then selected from these clusters using a Bayesian model that leverages alignment length and sequence identity between the query and extension candidate in both protein and nucleotide space. The extension process continues iteratively, first clustering, then extending. Clustering reduces the computational complexity of finding overlaps from quadratic in the number of reads to quasi-linear. This overlap-centric approach avoids the precision-sensitivity trade-off seen for de Bruijn graph assemblers, resulting in improved recovery from highly diverse metagenomic datasets [65, 62].

Here we introduce `CarpeDeam`, a *de novo* assembler specifically designed for ancient metagenomic data, based upon the greedy-iterative workflow of `PenguiN` [62]. It addresses the aforementioned challenge of high damage levels. Each iteration comprises three phases: First, sequences are clustered based on shared k-mers analogues to PenguiN. `CarpeDeam` additionally refines clusters by using a filter that employs a purine-pyrimidine encoding [66]. This encoding, which we refer to as RYmer space, ensures that cluster assignments of sequences are robust to aDNA damage events. In the second phase, sequences with base substitutions that are likely to be due to damage in the aDNA fragments are corrected. The final phase involves the elongation of contigs through

an extension rule that takes into account aDNA-specific substitution patterns.

We demonstrate `CarpeDeam`'s improved performance over currently available tools by generating longer DNA contigs from challenging datasets characterized by low coverage and elevated damage levels. We show that our tool can recover unique genomic segments that other assemblers miss especially in coding regions. Crucially, our approach significantly outperforms existing methods in the recovery of RNA genes and protein sequences despite a slightly higher error rate. `CarpeDeam` has two modes: safe and unsafe. The first is the default mode and is more conservative, reducing the number of errors, while the latter increases sensitivity. We benchmark our new tool against commonly used *de novo* assemblers and compare their performances on both simulated ancient metagenomic datasets and empirical datasets.

3 Results

Workflow of `CarpeDeam`

`CarpeDeam` is an assembler based on the metagenome assembler `PenguiN`. It employs a similar greedy, iterative overlap strategy, yet it incorporates several critical adjustments tailored for assembling ancient metagenomic data. Whereas `PenguiN` utilizes six-frame translated reads to find overlaps in amino acid space, we omit this step as ancient DNA fragments are ultrashort, resulting in even shorter amino acid sequences. In the text, we use the term aDNA fragments to describe the trimmed and merged sequencing reads (see discussion in Lien *et al.* [67]).

Overall `CarpeDeam` relies on three iteratively repeated steps: During **PHASE 1** (see Fig. **2**), sequences are clustered based on shared k-mers via the `MMseqs2 linclust` algorithm [68]. In this process, each sequence within a cluster is required to align to the center sequence with a minimum sequence identity. The center sequence is defined as the longest sequence within the cluster and becomes the focus for subsequent correction or extension processes, depending on the stage. The remaining sequences in the cluster, termed member sequences, overlap with the center sequence and contribute to its damage correction. In the extension phases, they serve as extension candidates. While `PenguiN` applies a relatively high default sequence identity threshold (99%), the presence of deaminations in aDNA requires a reduction in this threshold as the sequence identity is naturally lower even for sequences of the same provenance. `CarpeDeam` filters clusters using a reduced sequence identity threshold of 90% while introducing the concept of RYmer sequence identity, which converts sequences to a reduced nucleotide alphabet of purines (adenine and guanine) and pyrimidines (cytosine and thymine) to account for deaminated bases. First, the clusters are generated with all member sequences sharing at least one k-mer with the center sequence and the sequence similarity of the full overlap is computed following `PenguiN`'s workflow. Additionally, we compute the sequence similarity in RYmer space, where an A or a G is encoded as R (one letter encoding for purine) and a C or T as Y (one letter encoding for pyrimidine), of the full overlap. For a detailed explanation of this concept, please refer to Section S6 of the Supplementary Material. Overall, the RYmer space sequence identity allows for mismatches due to deamination events.

In **PHASE 2** (see Fig. **2**), `CarpeDeam` corrects deaminated bases. Any base in the center sequence of a cluster that is covered by at least one other sequence from the cluster can be corrected. Employing the user-provided damage patterns, the method utilizes a maximum-likelihood estimation to infer the most probable base for each position. The likelihood model is explained in more detail in Methods Section 6.

During **PHASE 3** (refer to Fig. **2**), sequences are extended. `PenguiN` employs an extension rule based on a Bayesian model, which selects the most suitable extension for each cluster. In contrast, we split the extension process into two distinct phases. The initial phase exclusively targets non-extended sequences (i.e. aDNA fragments). The key aspect is that the damage pattern provided by the user is only valid for these sequences, enabling the application of our likelihood model. Consequently, the initial phase focuses on extending the center sequence using yet non-extended sequences. Within each iteration, the extension continues as long as our likelihood model supports a strong belief in the accuracy of the extension (see Methods Section 6). The subsequent phase involves merging contigs: sequences that have already been extended and corrected. For this, `CarpeDeam` applies `PenguiN`'s Bayesian model with adjustments to improve its applicability to ancient datasets (see Methods Section 6).

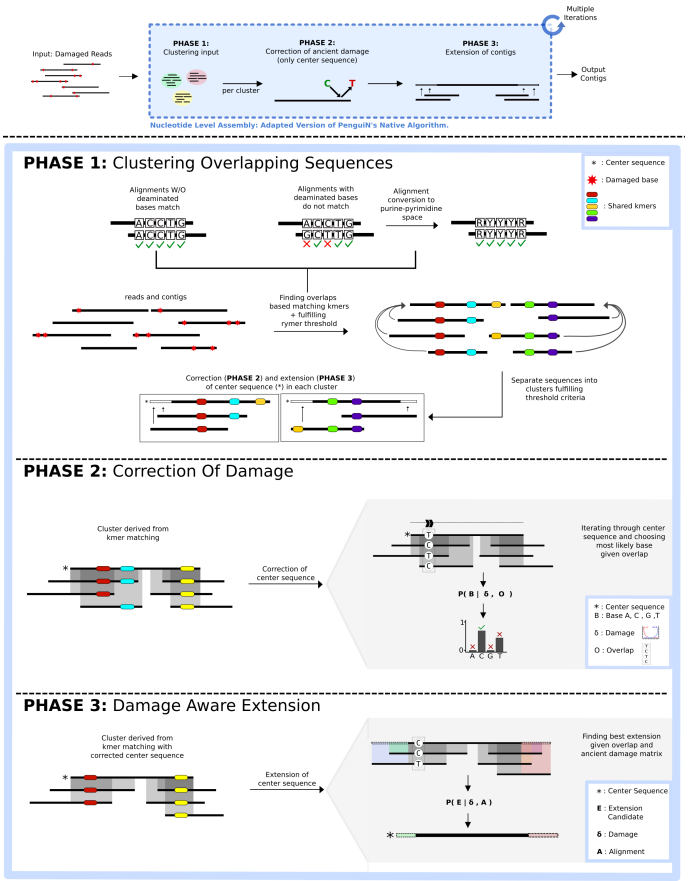

Figure 2: `CarpeDeam`'s main workflow: The input are aDNA sequences (FASTQ format) which have been trimmed and, for paired-end data, merged. During an iterative process, the fragments are corrected and extended to long contigs. In **PHASE 1** the fragments are grouped into clusters sharing at least one k-mer as well as an overlap sequence identity of 99% in RYmer space. **PHASE 2** corrects deaminated bases. In particular, the center sequence of each cluster (which is always the longest sequence) is assigned the most likely base per position given the evidence of overlapping sequences in the cluster and the user-provided damage patterns. In **PHASE 3**, the center sequence of each cluster is extended by the candidate sequence from the cluster that is most likely to be the correct extension. In fact, **PHASE 3** is divided into two steps. First, only aDNA fragments (non-extended sequences) are taken into account for extension, as the provided damage patterns are only valid for non-extended sequences. In the second step of **PHASE 3** exclusively contigs (sequences that already have been extended at least once) are used for the extension, applying a modified Bayesian extension model from the native `PenguiN` assembler.

We assessed the performance of `CarpeDeam` by generating 18 simulated datasets derived from three distinct ancient environments: gut microbiome [44], dental calculus [35], and bone material [69]. For each environment, six datasets were created with varying levels of coverage and damage. Furthermore, we assembled and evaluated two empirical datasets, performing taxonomic classification and protein similarity search on the contigs as evaluation criteria (see Methods Section 6 for further details).

According to our findings, `MEGAHIT` [58] and `metaSPAdes` [59] are the metagenomic assemblers of choice and have been applied in published studies for *de novo* assembly of ancient metagenomic datasets [44, 17, 45, 46, 47, 48, 49]. To evaluate `CarpeDeam`'s efficacy, we conducted an extensive benchmark comparison against the state-of-the-art assemblers `MEGAHIT` and `metaSPAdes` as well as the novel assembler `PenguiN` [62].

Overall, during our analysis we found that `CarpeDeam` can produce an elevated number of misassemblies. To offer users more flexibility in managing this trade-off, we introduced two operational modes for `CarpeDeam`: safe and unsafe. The safe mode (set as default) additionally implements a consensus-calling approach during the extension phase to reduce misassemblies. Users can opt for the unsafe mode, which disables only the consensus calling mechanism, potentially increasing

sensitivity at the cost of a higher rate of chimeric contigs. This dual-mode approach allows users to balance between assembly sensitivity and accuracy based on their specific research needs and dataset characteristics. A detailed description of the consensus approach can be found in section 6.

Simulated Data

Evaluation of Assembly Quality with `metaQUAST` and `Prokka`

We simulated 9 datasets of metagenomic paired-end Illumina short-reads with moderately high damage rates, varying in average coverage depth (3X, 5X and 10X). The simulation procedure is described in Section 6. The taxonomic profiles of our simulated datasets were derived from three publications [69, 35, 44] and represent different levels of complexity in terms of species abundance. For instance, the most complex dataset mirrors a gut microbiome consisting of 134 identified microbial species mirrored from Wibowo *et al.* [44] with abundances of individual species of maximum 10.77% and minimum 0.011%. To illustrate, at 10X average coverage, the least abundant species has a coverage of only 0.15X. While all species had the same simulated rate of damage and fragmentation patterns, we also created non-uniform simulations where different species have different rates of damage and fragmentation patterns (see Supplementary Material, Section S4).

We assembled the simulated datasets with `CarpeDeam`, `PenguiN`, `MEGAHIT` [58] and `metaSPAdes` [59].

All assemblers were run with their default parameters. For `CarpeDeam` we ran both the *safe* and the *unsafe* mode. We specified the flag `--only-assembler` for `metaSPAdes` because we observed the program getting stuck in the step that aims to correct sequencing errors. Skipping this step was advised in the issues section of the `metaSPAdes` GitHub repository (issue 306).

The reference genomes underlying our simulations were used to evaluate the assemblies of all four assemblers. We used the original references to compute the following alignment-based metrics: NA50, LA50, largest alignment, duplication ratio, misassemblies and mismatches per 100kb as generated by `metaQUAST` [70]. We set the sequence identity threshold at 90% for the `metaQUAST` [70] alignments as supposed in the `metaQUAST` documentation. Additionally, we mapped the aDNA fragments back to the contigs using `Bowtie2` [71] with `--very-sensitive-local` and reported the fraction of mapped fragments against the contigs via `SAMtools` [72].

For our benchmark, we set the minimum contig length to 500bp. In Fig. **3** we show four key metrics (NA50, largest alignment, covered genome fraction and number of misassemblies per contig) for all 9 datasets.

`CarpeDeam`, in both its safe and unsafe modes, and `MEGAHIT` demonstrated strong performance across the presented assembly metrics. In contrast, `PenguiN` and `metaSPAdes` struggled to effectively assemble genomes with moderately high levels of damage.

A notable difference was observed in the "largest alignment" metric (Fig. **3**, column 2), where `CarpeDeam`'s unsafe mode consistently produced substantially longer contigs compared to other assemblers. This advantage was particularly evident in the calculus and gut datasets. For most datasets even `CarpeDeam`'s safe mode yielded larger maximum alignments than `MEGAHIT`, while not achieving ultra-long alignments as the unsafe mode did. Only for the 3X gut dataset, `MEGAHIT` had a larger alignment than both `CarpeDeam` modes. `PenguiN` and `metaSPAdes` both had significantly shorter maximum alignments.

As expected, the increased sensitivity of `CarpeDeam`'s unsafe mode came at the cost of a higher misassembly rate, as shown by the "misassemblies per contig" metric (Fig. **3**, column 4). For the 3X gut dataset `CarpeDeam`'s unsafe mode generated more than twice as many misassemblies per contig than `MEGAHIT`, for the 5X gut dataset four times as many per contig than `MEGAHIT` and for the 10X gut datasets six times as many per contig than `MEGAHIT`.

Although `CarpeDeam`'s safe mode exhibited an elevated misassembly rate compared to `MEGAHIT`, `PenguiN`, and `metaSPAdes`, the difference was less pronounced. The most notable difference was observed in the 10X gut datasets, where the safe mode created about 2.5 times as many misassemblies per contig as `MEGAHIT`. Overall, the misassembly rate for all `CarpeDeam` modes remained

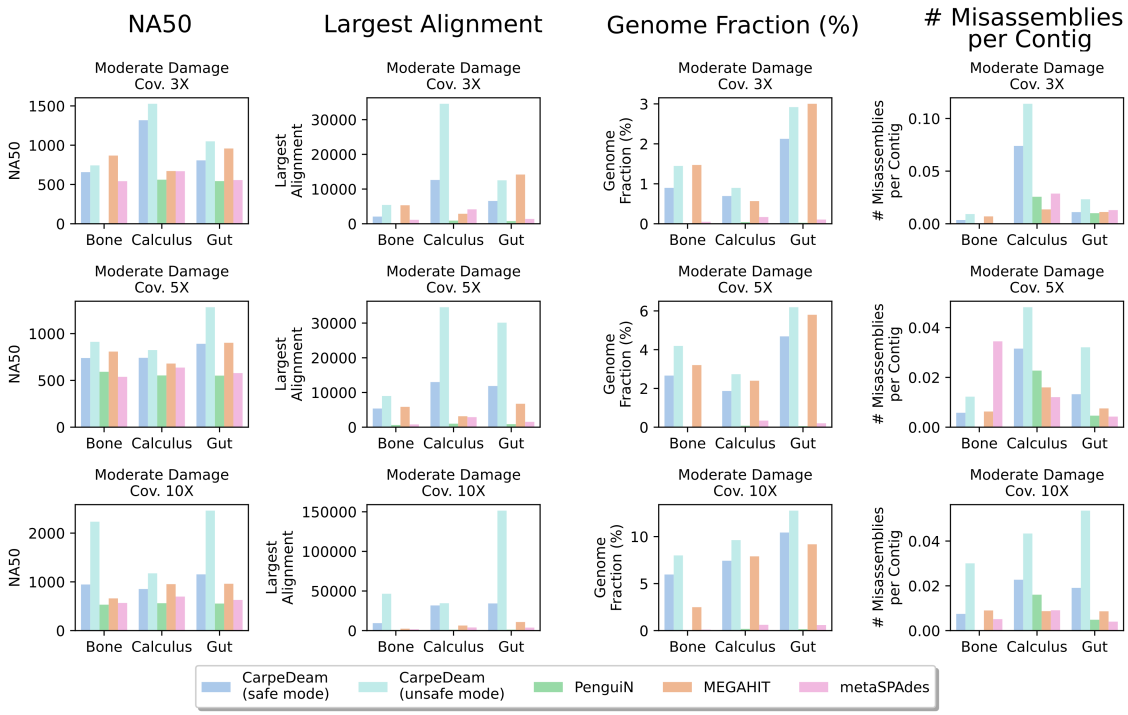

Figure 3: Performance evaluation of assemblers `CarpeDeam` (safe and unsafe modes), `MEGAHIT`, `metaSPAdes`, and `PenguiN` across various simulated datasets. The metrics shown are NA50 (column 1), largest alignment (column 2), genome fraction (column 3), and misassemblies per contig (column 4). Results are presented for three simulated environments (gut, dental calculus, and bone) with varying coverage levels. Each bar represents the performance of an assembler for a specific metric, coverage and environment.

within an acceptable range.

The fraction of recovered genomic content is shown in column 3 of Fig. **3** and `CarpeDeam` (both modes) score the highest in this category. Although the genome fraction recovered by `MEGAHIT` varied across datasets, it generally fell between the values obtained by `CarpeDeam`'s safe and unsafe modes. For the gut dataset the values of the recovered genomic fractions were for `CarpeDeam`'s safe mode, `CarpeDeam`'s unsafe mode and for `MEGAHIT` (in that order): 2.1%, 2.9% and 3.0% for the 3X dataset, 4.7%, 6.2% and 5.8% for the 5X dataset and 10.4%, 12.8% and 9.2% fot the 10X dataset. Both `metaSPAdes` and `PenguiN` recovered a substantially lower genome fraction.

Finally, the NA50 metric is presented in Fig. **3**, column 1. Generally, the N50 metric represents the length of the shortest contig such that half of the total assembled length is contained in contigs of this length or longer. While N50 is a commonly used metric to get a first impression of assembly quality, it does not distinguish between contigs that align with a reference genome and those that do not. The NA50 metric on the other hand considers only those contigs that have aligned to a reference genome. This adjustment helps to mitigate the influence of long chimeric or misassembled contigs that might artificially inflate the N50 value.

The top three performers in terms of NA50 values are `CarpeDeam` (in both safe and unsafe modes) and `MEGAHIT`. It is worth noting that `CarpeDeam`'s unsafe mode sometimes exhibits significantly higher NA50 values for certain datasets.

Table 1 presents additional metrics reported by `metaQUAST` [70] for evaluating the gut datasets, which represent the most complex of the three samples. For the additional results of the dental calculus and bone dataset, which are progressively less complex in that order, see Supplementary Material Table S2 and Supplementary Material Table S3. The table shown also includes the fraction of aDNA fragments mapped against the assembly. Across all three datasets, `CarpeDeam`'s unsafe mode consistently demonstrated a higher number of aDNA fragments mapping to its contigs than other assemblers. `CarpeDeam`'s safe mode also achieved significantly a higher number of mapped fragments in two out of three datasets relative to `MEGAHIT`. Notably, `MEGAHIT` performed very well for the 3X coverage gut dataset, with the highest values for both the "Genome fraction" and

"Largest Alignment" metrics.

The metric "mismatches per 100kb" is also presented in Table 1. MEGAHIT consistently exhibited the lowest number of mismatches per 100kb across all coverage levels. Interestingly, PenguiN displayed the highest mismatch values, exceeding 1000 mismatches per 100kb for all three coverages (3X, 5X, and 10X). This observation aligns with the fact that PenguiN does not perform base correction, as it was designed for modern viral metagenomes. In contrast, the de Bruijn graph assemblers MEGAHIT and metaSPAdes employ bubble merging algorithms to resolve bases in cases where multiple possibilities exist.

Despite CarpeDeam recovering significantly more than PenguiN, the higher mismatch rate per 100kb in CarpeDeam is indicative of its damage correction process, which is based on PenguiN. CarpeDeam's safe mode shows a lower mismatch rate per 100kb than its unsafe mode, yet is higher than MEGAHIT. For instance, for the 5x gut dataset in Table 1, the mismatch rates are 449 mm.p 100kb (CarpeDeam safe), 613 mm.p 100kb (CarpeDeam unsafe), 159 mm.p 100kb (MEGAHIT), 1147 mm.p 100kb (PenguiN), 379 mm.p.100kb (metaSPAdes). As indicated in Table 1, CarpeDeam exhibits a higher duplication rate. This impacts the mismatch rate per 100 kbp because the number of mismatches computed by metaQUAST is proportional to the duplication ratio. Consequently, mismatch rates can be elevated when there are numerous overlapping alignments, as each assembly alignment is analysed independently.

The tables reveal a general trend where CarpeDeam demonstrates higher sensitivity, producing a larger number of contigs. This is indicated by its elevated duplication ratio and higher mapped fraction, while achieving a genome fraction similar to MEGAHIT. Our observations align with the findings presented in the PenguiN [62] manuscript: PenguiN's approach recovers more strain-resolved viral genomes, while generally exhibiting a higher duplication rate. In contrast, MEGAHIT appears to prioritize precision, generating fewer contigs with a lower rate of misassemblies, resulting in capturing less of the genomic diversity present in the sample.

We show NA50 and LA50 in Table 1 as they can be viewed as more informative as their "non-aligned" counterparts N50 and L50. A higher NA50 is considered better as it describes the minimum alignment length to be considered, covering half of the alignment length of all contigs. For LA50, however, a lower value is considered better, as it describes the number of aligned blocks that need to be considered to cover half of the alignment length of all contigs. It must be noted that these metrics need to be evaluated in the context of the covered genome fraction. For instance, while PenguiN and metaSPAdes both have small LA50 values, the tools recovered a much smaller genome fraction than CarpeDeam and MEGAHIT.

To further evaluate the quality of assemblies beyond the genome fraction metric, we assessed the ability of each assembler to reconstruct long, non-misassembled contigs as classified by metaQUAST. While genome fraction provides valuable information, it does not account for the length of individual contigs mapping back to the reference genome (by default metaQUAST considers all alignments over 65bp). Consequently, a high number of short contigs may inflate the genome fraction without necessarily representing superior assembly quality. To address this limitation, we analyzed the number of non-misassembled contigs exceeding 2000 bp for each assembler across all datasets.

Fig. **4** illustrates this comparison, demonstrating CarpeDeam's capacity to generate longer accurate contigs across datasets. CarpeDeam (unsafe mode) consistently produced more long contigs for all datasets except one (3X coverage bone dataset, where MEGAHIT outperformed both CarpeDeam modes). CarpeDeam's safe mode and MEGAHIT exhibited alternating performance, with CarpeDeam producing more longer non-misassembled contigs in four cases and MEGAHIT in four cases. Notably, PenguiN and metaSPAdes generated significantly fewer long contigs compared to other assemblers. Although CarpeDeam produced a higher number of misassemblies, which would require careful filtering in downstream analyses, the increased quantity of non-misassembled contigs over 2000 bp suggests a notable improvement in assembly contiguity.

To complement our metaQUAST alignment analysis, we performed an additional examination of the assembly fractions (Fig. **5**). We used skani [73], a tool for calculating average nucleotide identity (ANI) and aligned fraction (AF), employing an approximate mapping method without base-level alignment. Contigs shorter than 1000bp were excluded to focus on long genomic segments.

We categorized the mapping results into six groups: the first group represents non-redundant base pairs mappable to original assemblies; the second group includes redundant base pairs attributable

Table 1: Assembly evaluation metrics for the simulated gut metagenomic environment.

Assembler	Coverage	Genome Fraction (%)	Largest Alignment	Fragments Mapped Fraction	NA50	LA50	# Misassemblies per contig	Duplication Ratio	# Mismatches per 100kb
`CarpeDeam` (safe mode)	3X	2.125	6,567	0.230	807	4,040	0.011	1.102	434.82
`CarpeDeam` (unsafe mode)	3X	2.919	12,520	**0.315**	**1,049**	3,812	0.023	1.124	574.37
`MEGAHIT`	3X	**3.002**	**14,186**	0.245	958	3,922	0.011	**1.009**	**154.65**
`PenguiN`	3X	0.025	769	0.003	543	**92**	**0.010**	1.066	1051.49
`metaSPAdes`	3X	0.105	1,381	0.007	556	347	0.013	1.005	476.22
`CarpeDeam` (safe mode)	5X	4.686	11,866	0.398	891	7,379	0.013	1.143	448.79
`CarpeDeam` (unsafe mode)	5X	**6.190**	**30,120**	**0.513**	**1,282**	5,283	0.032	1.194	612.86
`MEGAHIT`	5X	5.800	6,749	0.320	902	8,898	0.008	**1.009**	**158.62**
`PenguiN`	5X	0.054	875	0.005	551	**198**	0.005	1.073	1147.29
`metaSPAdes`	5X	0.201	1,538	0.014	578	607	**0.004**	1.008	378.63
`CarpeDeam` (safe mode)	10X	10.434	34,313	0.628	1,154	12,282	0.019	1.207	483.09
`CarpeDeam` (unsafe mode)	10X	**12.762**	**151,453**	**0.760**	**2,461**	5,446	0.054	1.394	717.630
`MEGAHIT`	10X	9.183	10,846	0.314	961	12,565	0.009	**1.010**	**153.76**
`PenguiN`	10X	0.152	1,177	0.015	555	**562**	0.005	1.103	1321.85
`metaSPAdes`	10X	0.591	3,793	0.049	629	1,462	**0.004**	1.015	321.18

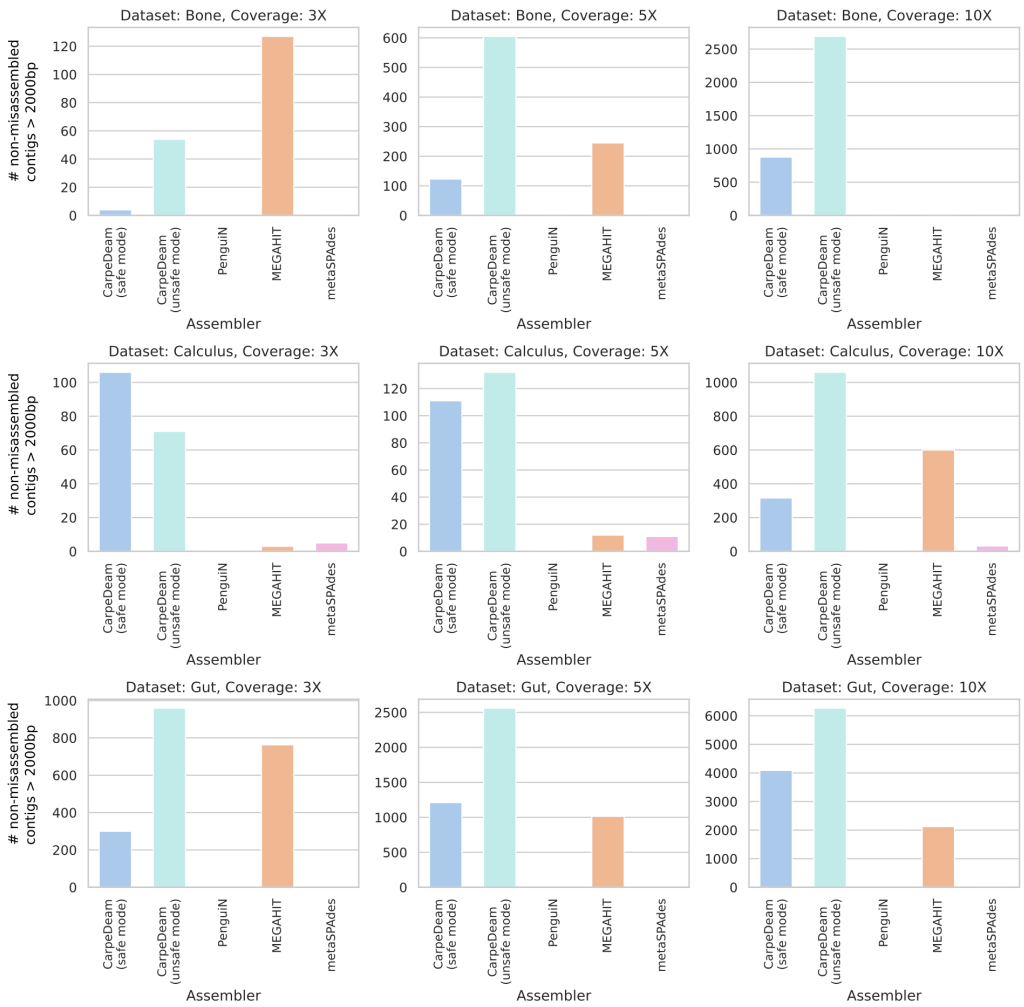

Figure 4: Comparison of non-misassembled contigs over 2000 bp produced by different assemblers across simulated metagenomic datasets. The bar plots show the number of non-misassembled contigs exceeding 2000 bp for `CarpeDeam` (safe and unsafe modes), `MEGAHIT`, `metaSPAdes`, and `PenguiN`. Results are presented for three simulated environments (bone, dental calculus, and gut) with varying coverage levels (3X, 5X, and 10X). Each row represents an environment (from top to bottom: bone, calculus, gut), while columns represent coverage levels (from left to right: 3X, 5X, 10X). Within each subplot, bars indicate the performance of individual assemblers for the specified environment and coverage combination.

to genetic diversity that are mappable to original assemblies; the third group denotes base pairs covering representative contigs of redundant segments that are mappable to original assemblies; the fourth group encompasses redundant base pairs mappable to original assemblies; the fifth group indicates non-redundant base pairs unmappable to original assemblies; and the sixth group refers to redundant base pairs unmappable to original assemblies.

Fig. 5, panel A, presents the results for the gut dataset across 10X, 5X, and 3X coverage levels. Our analysis revealed that Carpedeam (unsafe mode) exhibited the highest `mapped_non_dup_bp` fraction, particularly in the 10X coverage dataset. This advantage decreased at lower coverages (5X and 3X), with the difference between Carpedeam and MEGAHIT diminishing. Substantial fractions of `unmapped_non_dup_bp` sequences were observed across assemblies, most notably in Carpedeam (safe and unsafe) and MEGAHIT. PenguiN and metaSPAdes showed significant lower fractions across all mapping categories compared to other assemblers.

We further investigated the types of genomic features other than coding regions recovered by different assemblers (Fig. 5, panel B). For this analysis, we used `Prokka`'s annotation output. Our results showed that CarpeDeam demonstrated a strong ability to recover `rRNA`, `tRNA` and repeat region segments while `MEGAHIT` assembled significant less base pairs that could be classified by `Prokka`. It should be noted that only the results for `MEGAHIT` and `CarpeDeam` (safe and unsafe) are

presented here, as other assemblers recovered substantially less genomic content overall, making their inclusion in the plot impractical for comparative purposes. The results refer to the gut dataset.

Fig. 5, panel C, illustrates the recovery of `rRNA`s and `tRNA`s with sequence identities above and below 98%. For `tRNA`s in the 3X gut dataset, `CarpeDeam` (safe mode) and `MEGAHIT` reconstructed comparable numbers of high-identity ($ge98\%$) sequences, while `CarpeDeam` (unsafe) outperformed both, reconstructing more than three times as many. This difference became more pronounced in the 5X and 10X datasets, where both `CarpeDeam` modes assembled significantly more high-identity `tRNA`s than `MEGAHIT`. For `rRNA` reconstruction we could observe a distinct pattern: `MEGAHIT` recovered very few, whereas both `CarpeDeam` modes recovered several `rRNA`s. For instance, for the 10X gut dataset, `CarpeDeam`'s unsafe mode could recover more than 200 `rRNA`s of which more than 100 had a sequence identity of 98%. Notably, `CarpeDeam` (safe mode) recovered fewer `rRNA`s overall, but those recovered predominantly exhibited $ge98\%$ sequence identity. Conversely, `CarpeDeam` (unsafe mode) recovered a larger quantity of `rRNA`, albeit with a considerable fraction (¡50%) showing ¡98% sequence identity. Interestingly, some RNA sequences annotated by Prokka in the assemblies failed to match the reference annotation. While `CarpeDeam` (safe) and `MEGAHIT` exhibited similarly low rates of these "no hit" instances, `CarpeDeam` (unsafe) demonstrated the highest occurrence of unmatched annotations.

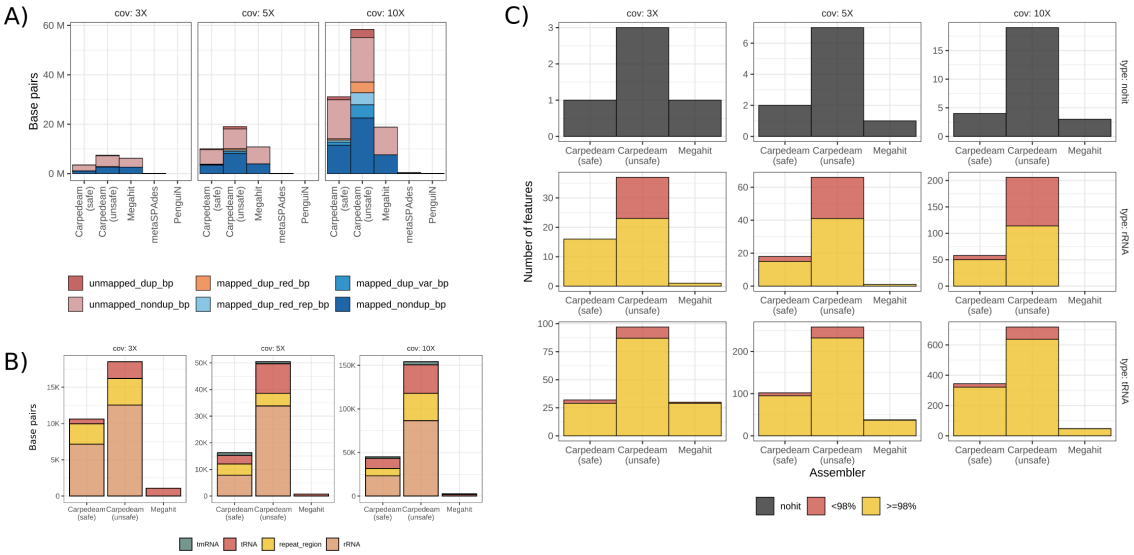

Figure 5: Analysis of mapped fractions of base pairs, genomic features, and RNA recovery across different assemblers and coverage levels. (A) Distribution of mapping categories (mapped non-duplicated, mapped duplicated variant, mapped duplicated redundant representative, mapped duplicated redundant, unmapped non-duplicated, and unmapped duplicated base pairs) for assemblies of the gut dataset at 10X, 5X, and 3X coverage levels. (B) Types of genomic features recovered in contigs based on Prokka annotations, for different assemblers (gut dataset). (C) Recovery of rRNAs and tRNAs with sequence identities above and below 98% for MEGAHIT and CarpeDeam (safe and unsafe modes) across different coverage levels (gut dataset).

Effectively reconstructing protein-coding sequences is essential for creating detailed microbial gene catalogs. These catalogs organize genes found in microbial communities and serve as references for standardized analysis across different samples and studies, making the accurate reconstruction of protein-coding sequences a crucial aspect of our evaluation [74]. Therefore, we assessed the performance of the assemblers in reconstructing proteins by predicting proteins from the reference genomes of our simulated metagenomes using `Prokka` [75] and searching for highly similar proteins with `MMseqs2 map` [76].

Fig. 6, panel **(A)** shows the number of predicted open reading frames (ORFs i.e. DNA translated into amino acid space) that exhibit significant similarity to predicted proteins from the reference genomes. We reported the number of unique matches in the reference. This approach allowed us to measure the assemblers' ability to reconstruct biologically meaningful protein-coding sequences.

In panel **(B)** of Fig. 6, we analysed the aligned fraction of amino acid residues in the assembled proteins. A higher fraction of mapped residues indicates that the assembler was better able to

recover reference proteins with high similarity.

`CarpeDeam`'s unsafe mode consistently outperformed other assemblers in assembling more proteins (Fig. 6, panel **(A)**). However, for the 3X bone dataset, `MEGAHIT` assembled more proteins than `CarpeDeam`'s unsafe mode. Apart from the 3X bone dataset, `MEGAHIT` assembled significantly more proteins than `CarpeDeam`'s safe mode for the bone 5X and gut 3X datasets. In contrast, `CarpeDeam`'s safe mode assembled significantly more proteins for all 10X datasets, as well as the 3X and 5X calculus datasets. For the gut 5X dataset, `MEGAHIT` and `CarpeDeam`'s safe mode assembled a very similar number of proteins. The other assemblers assembled significantly fewer proteins across all datasets.

Regarding the fraction of aligned amino acid residues panel **(B)**, the assemblers exhibited a more uniform performance compared to the pronounced differences observed in panel **(A)**. In the 5X and 10X datasets, `PenguiN`'s fraction of aligned amino acid residues was the highest. In the 3X bone dataset, `PenguiN` did not assemble any long contigs, resulting in an empty bar. While `PenguiN` recovered fewer overall proteins as seen in panel **(A)**, the proteins it did assemble had a high degree of sequence similarity to the references. For the other datasets, `CarpeDeam`'s unsafe mode had the highest alignment fraction, followed closely by `CarpeDeam`'s safe mode, `MEGAHIT` and `metaSPAdes`. The three assemblers achieved almost identical fractions of aligned amino acid residues, except for the 10X calculus and the 3X gut dataset where `MEGAHIT` performed slightly better.

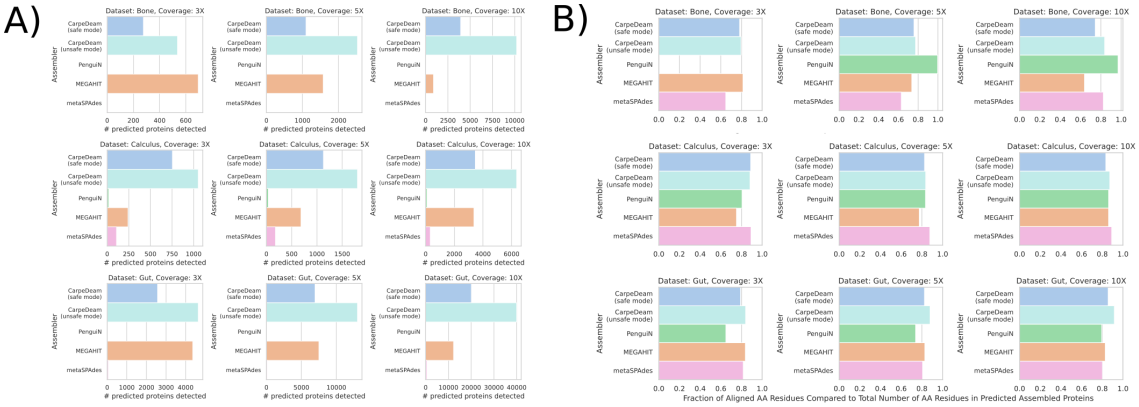

Figure 6: Evaluation predicted protein sequences. Results are presented for `CarpeDeam` (safe and unsafe modes), `MEGAHIT` , `metaSPAdes` , and `PenguiN` across three simulated environments (Bone, dental Calculus, and Gut) with varying coverage levels (3X, 5X, and 10X). Panel **(A)** shows the number of predicted ORFs with significant similarity to predicted reference proteins, filtered for alignments covering ≥90% of the reference protein and ≥90% sequence similarity. Panel **(B)** shows the fraction of aligned amino acid residues from the assembled proteins and the total number of amino acid residues in the assembled proteins. Aligned amino acid residues were counted if they belong to sequence that aligns to a reference protein with at least 90% similarity. Each row represents an environment (top to bottom: Bone, Calculus, Gut), while columns represent coverage levels (left to right: 3X, 5X, 10X).

As Table 1 of the metaQUAST results indicates that `CarpeDeam` tends to assemble a higher rate of duplicated contigs compared to other assemblers. Therefore, we conducted additional analysis to assess the impact of duplications on downstream processes. Of particular interest was the potential inflation of unique protein sequence counts in our protein analysis due to duplicated sequences. To address this concern, we performed three additional analyses that account for duplication rates.

First, we clustered predicted proteins with the `linclust` [68] algorithm at 100% sequence identity and 80% coverage, then searched the cluster representatives against the predicted proteins from the reference using the `MMseqs2 map` [76] module.

Second, to account for potential duplicates in the reference itself, we clustered the predicted proteins from the reference at 100% sequence identity and 80% coverage. We then searched these clustered reference proteins against the predicted proteins from the contigs using `MMseqs2 map`.

Finally, we employed `miniprot` [77] to search the predicted proteins from the reference directly against the DNA contigs. This method bypasses potential undercalling by our protein predictor and focuses on protein-to-DNA alignments directly, providing a complementary perspective to the

second analysis. We applied a filter to only obtain hits with at least 95% sequence identity and 95% coverage of the reference proteins.

In all cases we reported the number of unique proteins from the reference.

Overall, these additional analyses still favor `CarpeDeam`, although `MEGAHIT` recovers similar numbers of proteins especially in the 3X datasets. The detailed results of these analyses are visualized in the Supplementary Material, Section S3.

Assembly of Empirical Datasets

Next, we assembled two empirical metagenomic samples with varying levels of damage rates, using the datasets GDN001.B0101 and EMN001.A0101 (referred to as GDN001 and EMN001 in the following) from Fellows Yates *et al.* [36]. These datasets are ancient metagenomic samples from oral microbiomes of Neanderthals and Homo sapiens from different collection sites (see Table 2). The study provides detailed information about taxonomies and damage patterns in the oral microbiome. EMN001 has a maximum C to T damage rate of 0.244, while GDN001 has a maximum damage rate of 0.434 as reported in the supplements of Fellows Yates *et al.* [36]. For the damage pattern input parameter of `CarpeDeam`, we used the average damage rate across all taxonomies reported (see Section 6 and Fig. **9**).

Dataset	Microbiome Origin	Age	Collection Site	Library Prep	# Fragments	Length	N50	GC(%)
GDN001	Neanderthal	70,000 BP	Veneto, Italy	PE	14,803,019	30-141	49	57.58
EMN001	Homo sapiens	18,700 BP	Burgos, Spain	PE	11,182,824	30-101	78	62.54

Table 2: aDNA fragments statistics and sample metadata for the empirical ancient oral microbiome datasets GDN001 and EMN001 from Fellows Yates *et al.* [36]. BP = years before present. Library Prep indicates whether the sequencing data was paired-end. The table includes the total number of fragments, minimum and maximum aDNA fragments lengths, N50 aDNA fragments length, and GC content percentage.

We merged the paired-end reads with leeHom [78] and assembled them with `CarpeDeam`, `MEGAHIT`, `metaSPAdes`, and `PenguiN`.

Given the absence of ancient reference genomes for the empirical datasets, our evaluation focused on the proteome space. In the context of ancient prokaryotic samples, the ideal scenario would be to identify ancient proteins that exhibit strong similarity to existing sequences while remaining previously undiscovered. To maximize the detection of similar proteins from our assembled contigs, we employed a sensitive screening approach. We used the `--easy-search` module of `MMseqs2` [76] with `--search-type 4`, which extracts all possible open reading frames (ORFs) from the contigs and searches the resulting amino acid sequences against a provided database. For this analysis, we queried the UniRef100 database [79], which encompasses over 400 million reference protein sequences.

The search results were subsequently filtered using stringent criteria: sequence identity ($\geq$35%), E-value ($\leq$E-12), and alignment length ($\geq$100 residues). Figure **7** presents the number of unique hits in the UniRef100 database, considering only the best hit for each query based on E-value. This approach allows us to assess the potential unknown protein content in our aDNA assemblies while maintaining high confidence in the reported matches.

In the GDN001 dataset, which exhibits a higher damage rate according to Fellows Yates *et al.* [36], `CarpeDeam` (in both safe and unsafe modes) yielded more unique protein hits compared to other assemblers. Notably, the difference between `CarpeDeam`'s safe and unsafe modes was not substantial. `MEGAHIT` identified approximately 20% fewer hits for the GDN001 dataset, while `PenguiN` and `metaSPAdes` produced considerably fewer hits (Figure **7**, Panel **(A)**).

The EMN001 dataset, which exhibits lower damage rates, yielded more protein hits across all assemblers compared to GDN001 (Figure **7**, Panel **(B)**). This observation aligns with the intuitive expectation that less damaged DNA would lead to more successful protein identifications.

CarpeDeam consistently outperformed other assemblers in terms of hit count, with a marginal advantage observed in the safe mode (12,572 hits) compared to the unsafe mode (12,026 hits). In contrast to the GDN001 results, metaSPAdes identified the third-highest number of hits for the EMN001 dataset, surpassing MEGAHIT. While a detailed analysis of these identified proteins would be valuable for further insights, such an investigation extends beyond the scope of the current study.

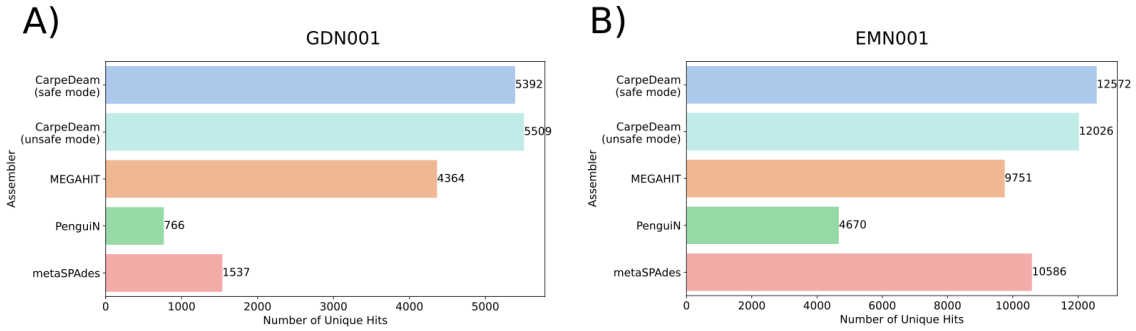

Figure 7: Comparison of unique hits in the UniProt100 Protein Database for full-length open reading frames (ORFs) predicted from contigs generated by CarpeDeam, MEGAHIT, PenguiN, and metaSPAdes for two ancient metagenomic datasets, GDN001 and EMN001. ORFs were predicted using Prodigal with the "meta" setting, and the translated protein sequences were searched against the UniRef100 database using MMseqs2. Unique hits were filtered by E-value $\leq$ 1e-5, sequence identity $\geq$ 35% and alignment length $\geq$ 100 residues. **(A)** Results for GDN001 dataset. **(B)** Results for EMN001 dataset.

Another important use case for assembled contigs is taxonomic classification. This approach offers several advantages over classifying individual short reads, especially in the context of aDNA analysis. Contig-based classification improves accuracy by leveraging longer sequences, which is particularly beneficial for ultra-short, degraded aDNA fragments. It also helps mitigate misclassifications caused by conserved regions or horizontal gene transfer events [32, 80, 34].

Considering these advantages, we assessed the assemblers' performance in recovering proteins informative for taxonomic assignment. We extracted all open reading frames (ORFs) from the assembled contigs using Prodigal [81]. The resulting amino acid sequences were then queried against a proteome database using the --easy-taxonomy module of MMseqs2 [34]. MMseqs taxonomy downloads proteomes that are the cluster representatives of species according to the Genome Taxonomy Database (GTDB)(for a detailed description of the workflow see Section 6).

It is important to note that this approach primarily focuses on detecting bacteria closely related to modern species, specifically those represented in the GTDB, which is derived from RefSeq and GenBank genomes. Consequently, this method may not necessarily identify completely unknown species. The discovery of novel species presents significant challenges, as potential misassemblies would require careful downstream analysis to ensure accurate identification and exclusion of misassembled sequences.

Given these considerations, we adopted a conservative approach with stringent criteria for taxonomic assignment. Matches were filtered to meet the following thresholds: a minimum of 95% sequence identity in amino acid space, at least 70% target protein coverage, and an E-value ($\leq$E-6). This conservative strategy aims to minimize false positives while maintaining high confidence in the reported taxonomic assignments.

Finally, we quantified the number of distinct species identified by each assembler based on the proteome alignment (Fig. 8, Panel **(A)**). This metric provides a comparative measure of the assemblers' performance in recovering taxonomically informative sequences from our empirical aDNA samples.

Figure 8, Panel **(A)**, row 1 presents the taxonomic assignment results for the GDN001 dataset. A substantial number of species were identified by all assemblers, with MEGAHIT and CarpeDeam (unsafe mode) sharing the largest common set. Notably, both MEGAHIT and CarpeDeam identified a considerable number of unique taxa, with MEGAHIT slightly outperforming CarpeDeam (234 vs. 218 unique taxa, respectively; first Venn diagram).

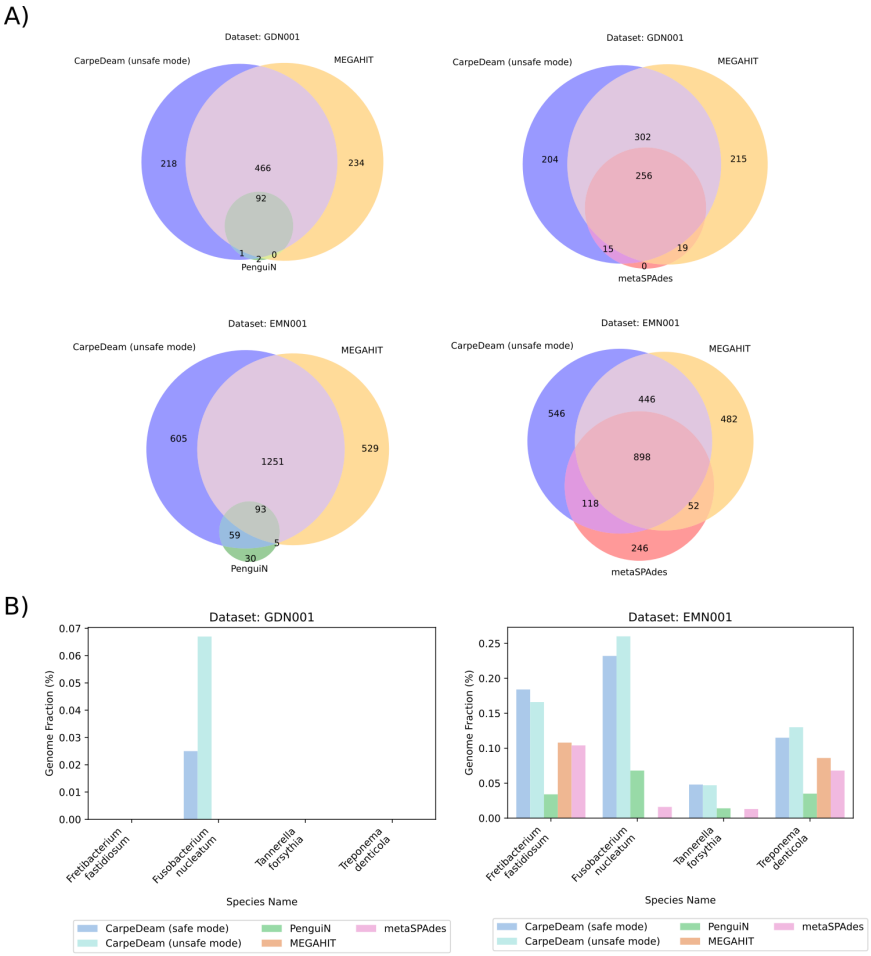

Figure 8: **(A)** Taxonomy assignment on species level. We compared the number of taxonomic hits from searching translated contigs against the Genome Taxonomy Database. The Venn diagrams showing the unique and shared taxonomic assignments from contigs assembled by four different metagenomic assemblers (`CarpeDeam`, `MEGAHIT`, `metaSPAdes` , `PenguiN`) and searched against the Genome Taxonomy Database using `MMseqs2` taxonomy [34]. Row 1 shows results for the GDN001 ancient oral microbiome dataset from a Neanderthal. Row 2 shows results for the EMN001 dataset from a Homo sapiens. Numbers indicate the number of unique taxonomies assigned to contigs from each assembler or the overlapping taxonomies assigned by multiple assemblers. **(B)** Assembly Performance of highly damaged Taxa identified by Fellows Yates *et al.* [36]. Recovered genome fractions reported by `metaQUAST` [70] for highly damaged bacterial species across two datasets.

Row 2 of Panel **(A)** displays results for the EMN001 dataset, revealing a similar pattern to GDN001. However, consistent with its lower damage rate, the assemblies of the EMN001 dataset yielded identifications for a larger number of taxa overall. `CarpeDeam` demonstrated a marginal advantage in unique taxa identification compared to `MEGAHIT` (605 vs. 529, respectively; third Venn diagram).

A key observation from this analysis is that no single assembler consistently outperforms the others. While the different algorithms naturally share a significant portion of their results, each assembler reveals unique findings.

To further evaluate the assemblers' performance on highly damaged taxa, we focused on four species reported by Fellows Yates *et al.* [36] to exhibit damage patterns of up to 60% that were identified in both samples: *Fretibacterium fastidiosum*, *Fusobacterium nucleatum*, *Tannerella forsythia*, and *Treponema denticola*. We used the reference genomes of these taxa as input for `metaQUAST` [70] and reported the recovered genome fraction using a contig length cutoff of 500 bp.

Analysis of the GDN001 dataset revealed that `CarpeDeam` was the only assembler capable of recovering any fraction of a reference genome. While no assembler successfully recovered sequences from *F. fastidiosum* or *T. denticola*, `CarpeDeam` in unsafe mode recovered approximately 0.07% of

the *F. nucleatum* genome, whereas safe mode recovered around 0.025%.

The EMN001 dataset yielded markedly different results, with most assemblers recovering small fractions of the reference genomes. Notably, `MEGAHIT` did not assemble any contigs longer than 500 bp that mapped to either *F. nucleatum* or *T. forsythia*. `CarpeDeam` outperformed other assemblers, recovering the largest fraction for all four target species. Interestingly, `CarpeDeam`'s safe and unsafe modes exhibited comparable performance, recovering similar fractions of the reference genomes.

4 Discussion

The reconstruction of full-length genes or entire genomes from ancient metagenomes poses significant challenges for common *de novo* assemblers. While *de novo* assembly of metagenomic samples is already complex for modern samples, it becomes exceptionally challenging with aDNA due to its degraded nature, characterized by extremely short fragment sizes and deaminated nucleotides (Fig. **1**,B).

While modern metagenomic assemblers like `MEGAHIT` [58] and `metaSPAdes` [59] are widely used, their De Bruijn Graph implementations face a precision-sensitivity trade-off when dealing with highly complex datasets [65, 62]. As graph complexity increases, these tools must employ simplification strategies, potentially compromising assembly completeness or accuracy, especially in regions of high genomic diversity or low coverage.

`PenguiN` assembler [62] addresses this issue through a greedy iterative overlap approach, leveraging whole read information for improved performance in variant-rich datasets. Ancient metagenomic datasets, particularly those from poorly preserved samples, present additional challenges for both de Bruijn graph and overlap-based assembly strategies. Base substitutions in aDNA fragments artificially increase the genomic diversity in metagenomic samples. While deep sequencing can facilitate assembly [17], it may be impractical due to sample limitations or cost constraints [6].

Our tool, `CarpeDeam`, builds upon `PenguiN`'s whole-read approach, incorporating a maximum-likelihood framework that utilizes sample-specific damage pattern matrices for contig correction and extension. It represents a significant advancement in aDNA assembly as a pioneering sample-specific, damage-aware assembler. Our experiments on both simulated and empirical data demonstrate `CarpeDeam`'s ability to improve the assembly of long, continuous sequences. While the reconstruction of full-length genomes remains exceptionally challenging, `CarpeDeam` demonstrated its capability in recovering more and different protein content than other assemblers. Additionally, we showed that `CarpeDeam` can reconstruct other genomic features such as `tRNA`s and `rRNA`s at high sequence identity, enabling research endeavors that leverage ancient RNA (aRNA) for purposes such as tissue-specific RNA analysis [82].

`CarpeDeam`'s improvement is particularly valuable as it facilitates more comprehensive functional and taxonomic composition analyses of metagenomic samples, enabling the study of their relationships with environmental parameters [65].

For instance, Wan *et al.* [50] demonstrated the potential of ancient proteomes as a source of novel antimicrobials. They employed deep learning techniques to systematically analyze proteomes of extinct organisms, successfully identifying potential antibiotics effective against highly pathogenic bacteria. This study demonstrates the practical applications of aDNA research beyond theoretical investigations.

A slightly different approach was taken by Klapper *et al.* [17], covering the entire process from sequencing to the discovery of novel bioactive compounds. In their study, deeply sequenced ancient metagenomic samples were assembled to create metagenome-assembled genomes (MAGs). They identified biosynthetic gene clusters (BGCs) and experimentally validated previously unknown molecules.

By improving the reconstruction of protein sequences from degraded genetic material, `CarpeDeam` may contribute to expanding the scope and efficiency of mining the extinctome for bioactive compounds.

Additionally, it is important to note that while protein-level assemblers such as `PLASS` [65] exist, the highly degraded nature of aDNA, resulting in ultra-short aDNA fragments, presents significant challenges for these tools. The extraction of informative ORFs from very short ancient sequences

becomes problematic, making the assembly ineffective while also not coming up for the correction of deaminated bases.

Our experiments also showed that CarpeDeam generates a slightly higher rate of chimeric contigs, a trade-off for increased sensitivity, which is why we implemented two operational modes: safe and unsafe. Both modes employ a filter leveraging the "RYmer" space (detailed in Supplementary Material, Section S6) to assess sequence identity under a purine-pyrimidine encoding and thereby reduce misassemblies. The safe mode additionally utilizes a consensus-calling approach during the extension phase. Users can opt for the unsafe mode, which disables only the consensus calling mechanism, potentially increasing sensitivity at the cost of a higher rate of misassembled or chimeric contigs. Despite utilizing a uniform damage pattern in CarpeDeam's likelihood model, which may not fully capture the heterogeneous damage patterns present in metagenomic datasets, CarpeDeam has demonstrated its capability in assembling empirical datasets with non-uniform damage patterns (Fig. **9**).

Investigation of the higher rate of misassemblies observed with CarpeDeam can be found in the Supplementary Material, Section S4. First, we analyzed the synthetic datasets with metaQUAST, showing the same key metrics as above (Fig. S3). Then, further analysis revealed that identical sequences can occur either within a single genome or between two or more species' genomes, with considerable variation in frequency across prokaryotic taxa [83, 84] (Supplementary Material Fig. S4 and Fig. S5).

When CarpeDeam encounters two identical sequences as extension candidates, it fails to determine the optimal choice. This scenario contributes to the observed increase in misassembled contigs, particularly in the unsafe mode. While the implementation of the safe mode option has demonstrated improvement in reducing misassemblies, it does not completely eliminate the issue. These observations highlight the trade-off between sensitivity and accuracy in metagenomic assembly, especially in the context of aDNA and diverse prokaryotic communities. The availability of safe and unsafe modes allows users to adjust the assembly approach based on their specific research goals and the nature of their datasets.

It should be noted that MEGAHIT, despite not being specifically designed for aDNA assembly, does frequently choose correct bases in deaminated regions through its bubble merging process. This approach selects the most supported path through the De Bruijn graph. In scenarios with sufficient coverage, MEGAHIT appears to successfully identify the correct base (C over T, or G over A) in cases of deamination-related ambiguities.

Furthermore, our taxonomic classification results revealed that both CarpeDeam and MEGAHIT identified a substantial number of unique taxa not detected by other assemblers. This complementary performance suggests that the optimal approach for ancient metagenomic *de novo assembly* may involve the integration or co-implementation of both algorithms. By combining MEGAHIT's conservative assembly strategy with CarpeDeam's increased sensitivity, a more accurate reconstruction of ancient metagenomes could potentially be achieved.

5 Conclusion

CarpeDeam represents a significant advance in aDNA assembly as the first sample-specific, damage-aware assembler. Our experiments on both simulated and empirical data demonstrate CarpeDeam's ability to improve the assembly of long, continuous sequences from degraded aDNA. By incorporating a maximum-likelihood framework that utilizes sample-specific damage pattern matrices, CarpeDeam outperforms existing assemblers in recovering genes from metagenomic samples. The tool's dual operational modes - safe and unsafe - offer users flexibility in balancing sensitivity and accuracy based on their research goals. While challenges remain in reconstructing full-length genomes from highly degraded samples, CarpeDeam's improved gene recovery has potential applications in functional and taxonomic composition analyses, as well as in mining the extinctome for bioactive compounds. As the first damage-aware metagenome assembler, CarpeDeam lays the groundwork for future innovations in this field. Looking ahead, combining CarpeDeam's novel approach with established de Bruijn graph methods could potentially lead to even more robust and accurate aDNA assembly techniques, further advancing our ability to reconstruct and analyze ancient microbial communities.

6 Methods

Test Data

To assess the assembly performance of ancient metagenomic samples, we simulated three distinct metagenomic environments. Each environment varied in complexity, including different abundance profiles of species. The taxonomic composition was modeled on previously published studies from Granehall *et al.* Wibowo *et al.* and Der Sarkissian *et al.* [35, 44, 69]. We simulated the data with three levels of average coverage (3X, 5X and 10X) and moderately high damage. As a result, we generated three datasets for each metagenomic environment. An overview of the characteristics of the simulated datasets is shown in Table 3.

Source	# Species	Average Coverage	Damage	# fragments	GC(%)
Gut	59	3X	Moderate	25,583,992	42.45
Gut	59	5X	Moderate	42,640,040	42.45
Gut	59	10X	Moderate	85,280,026	42.45
Dental Calculus	116	3X	Moderate	8,169,302	41.57
Dental Calculus	116	5X	Moderate	13,615,491	41.57
Dental Calculus	116	10X	Moderate	27,230,820	41.58
Bone	34	3X	Moderate	13,481,616	59.81
Bone	34	5X	Moderate	22,469,427	59.81
Bone	34	10X	Moderate	44,938,829	59.81

Table 3: Overview of simulated ancient metagenomic datasets from three different environments: Gut, Dental Calculus, and Bone. The table shows the number of species, average sequencing coverage, level of damage, number of sequencing fragments, and GC content percentage rounded to two significant figures for each simulated dataset.

The damage patterns were derived from Gunther *et al.* [85] (labeled as "moderate" damage). We used gargammel [86] to simulate paired-end reads for the bacterial communities using the aforementioned damage patterns. The fragment length distribution stems from the Vi33.19 [87] sample (see Supplementary Material Fig. S1). Adapters were trimmed and the paired-end reads were merged with leeHom with the `--ancientdna option` [78]. We then aligned the simulated reads against our references with `Bowtie2` [71] and used the resulting BAM file as input for `DamageProfiler` [9]. The estimated damage patterns were then used as input for `CarpeDeam`, which uses the damage pattern for its maximum likelihood framework. Thereby we avoided any bias to give `CarpeDeam` an advantage as the process simulates how damage patterns are derived for in a realistic scenario.

The two empirical datasets we benchmarked the assemblers against were GDN001.B0101 and EMN001.A0101, obtained from Fellows Yates *et al.*. These datasets were suitable for our analysis as they were not deeply sequenced, meaning that the coverage of the genomes was relatively low. Fellows Yates *et al.* also provided extensive analysis of the species found in these datasets, which made them highly suitable for our analysis, as we could utilize the presented damage patterns for the input of `CarpeDeam`. The damage patterns observed in these datasets were similar to those used in our simulations, yet they varied across species, allowing us to test the robustness of `CarpeDeam`'s likelihood model. An overview of the damage patterns for identified species can be seen in Fig. **9**. Each dot belongs to the substitution frequency of a species. Overlapping dots result in different colour shades. We used the average damage frequency per position for the input of `CarpeDeam`, which is marked as a black line in the plot. Accession IDs and additional information about the empirical datasets are listed in the data repository (see Section 7).

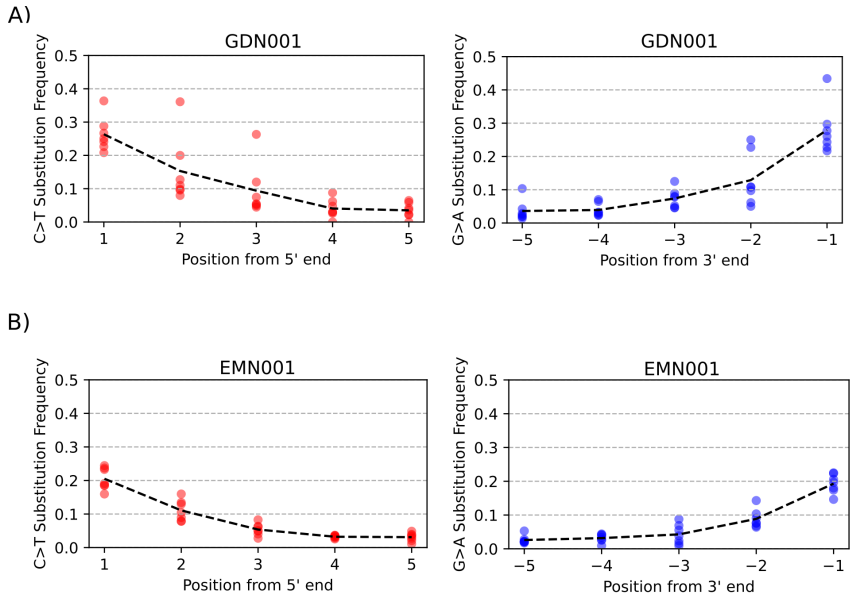

Figure 9: Overview of damage patterns in selected species from the empirical datasets GDN001.B0101 and EMN001.A0101. The damage patterns belong to the species *Fusobacterium nucleatum*, *Porphyromonas gingivalis*, *Pseudopropionibacterium propionicum*, *Streptococcus gordonii*, *Tannerella forsythia*, *Treponema denticola*,*Treponema socranskii*. Each dot belongs to the substitution frequency of a single species. The black line indicates the average substitution rate at a position. The damage matrix with the average values was used as input for `CarpeDeam` assemblies of these datasets.

Benchmark Workflow

Evaluation of Assembling Simulated Datasets

The assembly performance was evaluated using metrics reported by `metaQUAST` [70]. We provided the reference genomes used for the simulations, setting the minimum sequence identity for alignments to 90% as suggested in the `metaQUAST` documentation. Informative metrics are reported in Fig. **3** and Table 1, with additional tables for simulated datasets of dental calculus and bone in the Supplementary Material (Tables S2 and S3). For the alignment evaluation, `metaQUAST` uses the pairwise sequence aligner `minimap2` tool [58].

To complement the genome fraction metric and provide a more comprehensive assessment of assembly quality, we evaluated the ability of each assembler to reconstruct longer, correct contigs. Specifically, we quantified the number of correct contigs of length 2000 bp or longer for each assembler across all datasets. This analysis addresses a limitation of the genome fraction metric, which does not account for the length distribution of contigs mapping to the reference genome (Fig. **4**).

We further assessed the assemblers' performance in reconstructing protein content. This analysis involved annotating the produced contigs and searching for similar proteins using `Prokka` [75], which employs `Prodigal` [81] for open reading frame prediction and searches against three core databases.

After open reading frame prediction with `Prokka` [75] of both the assembled contigs and the reference genomes, we utilized the `MMseqs2 map` [76] module in `--cov-mode 1` with the `--min-seq-id` parameter set to 0.9 to find very similar sequence matches with at least 90% sequence identity and covers the target by at least 90%. We then reported the number of unique matches in the reference (Fig. **6**, panel **(A)**).

In panel **(B)** of Fig. **6**, we analyzed the fraction of amino acid residues in the assembled proteins that aligned to the predicted reference proteins. Again, we used `Prokka` [75] to predict proteins from the assembled contigs. We then searched these predicted proteins against the reference proteins using the `MMseqs2 map` module. We set the minimum coverage parameter `-c` to 0 and the `--min-seq-id` parameter to 0.9, to report any alignments with at least 90% sequence identity. From these high-confidence alignments, we counted the number of amino acid residues for each

assembled protein. We then calculated the fraction of mapped amino acid residues compared to the total number of residues in the assembled proteins. This fraction of aligned amino acid residues is shown in panel **(B)** of the figure.

Taxonomy Assignment of Contigs and Protein Similarity Search of Empirical Datasets

Since we did not have the ground truth for the empirical datasets (only selected identified species from Fellows Yates *et al.* [36]) we employed different evaluation methods. First, we used the `MMseqs2` taxonomy module [34] for taxonomic classification. This module translates contigs in all six reading frames and searches the amino acid sequences against a target database. For our target, we used proteomes that are representatives for species according to the Genome Taxonomy Database (GTDB) [88]. These proteomes are the cluster representatives for each species, and this database was downloaded using the `databases` module of `MMseqs2` [76]. The GTDB is a phylogenetically consistent, genome-based taxonomy that uses a large number of quality-controlled genomes to infer evolutionary relationships among bacteria and archaea.

We analysed the number of unique species assigned to the translated contigs, considering all species where at least one translated contig covered a single-copy protein by at least 90%. We then examined the number of shared taxa and unique taxa per assembler (see Fig. **8**).

Additionally, we analyzed the capability of the assemblers to assemble protein fractions that can be used for identifying evolutionary related proteins. We predicted open reading frames (ORFs) from the contigs generated by all assemblers using `Prodigal` [81] with the "meta" setting, considering only full-length ORFs for further analysis. We then used MMseqs2's easysearch module [76] to search the amino acid sequences against the UniProt100 database [89], filtering the results by a minimum sequence identity of 0.35, an E-value smaller than 1e-5, and a minimum alignment length of 100 residues. The number of unique protein hits was reported in the bar plots in Fig. **7**.

Deamination Correction in Assembly Process

The assembly workflow incorporates a correction stage for deaminated bases at the start of each iteration. First, sequences are clustered based on shared k-mers, following `PenguiN`. The choice of k-mer size impacts clustering accuracy, with larger k values generally producing clusters that more reliably group sequences of common origin. Given the typically short length of aDNA fragments, a default k-mer size of 20 is set. The clusters are characterized by:

- A center sequence, which is either a contig (an extended sequence not found in the raw input data) or an aDNA fragment. It is the longest sequence in the cluster.
- All remaining members of the cluster are aDNA fragments (non-extended fragments).
- All members of a cluster fulfill threshold criteria such as sequence identity and E-value (as in `PenguiN`) and the newly introduced RYmer sequence identity, which exclusively counts mismatches between purines and pyrimidines, thereby tolerating mismatches due to deamination, i.e. C→T or G→A substitutions.

The goal is to correct the center sequence of each cluster, using alignments from member sequences and user-provided damage patterns that estimate base substitutions in the input data.

Consider a cluster containing a center sequence C and at a given position in this sequence, we seek to find the most likely base b_C given the data. The data consists of a set of aDNA fragments $\mathcal{R}$ aligning at that position. The fragments in $\mathcal{R}$ share at least one k-mer with C. More prosaically, we seek $P[b_C \mid \mathcal{R}]$ which can be obtained by finding b_C which maximizes the expression $P[\mathcal{R} \mid b_C]$ assuming a uniform prior for all 4 DNA nucleotides. The independence assumption allows us to say:

$$P[\mathcal{R} \mid b_C] = \prod_{R \in \mathcal{R}} P[R \mid b_C]$$

For a base b_R in R at that position, each factor is defined as:

$$P[b_R \mid b_C] = \begin{cases} \delta(b_C \to b_R)(1-\epsilon) & \text{if } b_R = b_C \\ \frac{\epsilon}{3} + \delta(b_C \to b_R)(1-\epsilon) & \text{if } b_R \neq b_C \end{cases}$$

Here ϵ represents the sequencing error rate and is set at $\frac{1}{1000}$ for all bases as `PenguiN` does not store the quality scores at this stage. The term $\delta(b_C \to b_R)$ represents the rate of substitution from C to R due to aDNA damage as provided by the user for this particular position in the fragment. To illustrate, if the rate of C to T at the 5' end is 30%, then $\delta('C' \to' C') = 0.7$ and $\delta('C' \to' T') = 0.3$. In the case of the bases not matching, we consider the possibility of either sequencing error or damage. If the bases match, then no damage has occurred and the base was fragment correctly. We neglect the possibility of a deamination followed by a sequencing error which would result in the same base as it is dwarfed by the other term.

Assembly with aDNA fragments

Selecting the extension candidate

Following the initial correction of the center sequence in each cluster, the workflow proceeds with a damage-aware extension process. This involves extending the center sequences of each cluster using sequences that are part of the cluster that share at least one k-mer with the center sequence and meet additional criteria such as minimum sequence identity, E-value, and RYmer sequence identity. In this phase, only member sequences that are fragments rather than assembled contigs are considered, as the provided damage patterns are only applicable to sequences not yet merged as contigs. The selection of the most suitable sequence for extension is based on the overlap length and the number of matches within this overlap. However, as deamination events occur in aDNA, potential mismatches between the center sequence and the extension candidates are introduced. Therefore, the identification of the optimal sequence for extension is challenging.

Assuming that all bases in the center sequence C are correctly identified, we can calculate the likelihood of observing a specific overlap between C and an aDNA fragment R_i, conditional on the damage pattern δ. This approach accounts for potential deamination events, such as observing a base 'G' in the center sequence and 'A' in the overlapping fragment, without classifying them as highly improbable, depending on the damage pattern.

The question is which R_i is the best candidate for extension. More specifically, we seek R_i which maximizes $P[R_i \mid C]$. Let us define b_C as the base in the center sequence and b_{R_i} in the fragment. We have a nuisance parameter b_δ which is the base post damage.

The likelihood for a single nucleotide b_{R_i} of fragment R_i is given by:

$$P(b_{R_i} \mid b_C) = \sum_{b_\delta \in \{A,C,G,T\}} \delta(b_C \to b_\delta) \left(\begin{cases} (1-\epsilon) & \text{if } b_\delta = b_{R_i} \\ \frac{\epsilon}{3} & \text{if } b_\delta \neq b_{R_i} \end{cases} \right)$$

again, ϵ is $\frac{1}{1000}$. $\delta(b_C \to b_\delta)$ is the rate of nucleotide substitution due to aDNA damage.

However, not every aDNA fragment overlaps with the same amount of bases with C. Only computing the overlapping portions would put longer overlaps at a disadvantage. Let us say that the longest overlap among the candidates for extension is l_{max} bp long. If $l_{R_i} < l_{max}$ counting the $l_{max} - l_{R_i}$ missing bases as mismatches would be unfair to shorter overlaps as even random sequences will be 25% similar. We therefore count the probability of the missing bases as $K^{(l_{max} - l_{R_i})}$. K is a constant between 0 and 1, representing the probability of a random match between two bases. The value of K is set by default at $\frac{1}{4}$.

Stopping criterion

After identifying the extension candidate with the highest likelihood, this candidate is merged with the center sequence. This process continues iteratively, incorporating aDNA fragments that meet the sequence identity criteria, until no further suitable candidates remain. The extension

process is terminated if no candidates satisfy the sequence identity threshold, or if the likelihood is significantly lower than the likelihood for a random match, given that it satisfies the minimum sequence identity threshold in the overlap (e.g. 90%). The likelihood of an alignment given the damage pattern is calculated, as well as the likelihood of observing a random match. The ratio of these two likelihoods determines the termination of the extension and can be set by the user. The minimum ratio used as threshold for accepting an extension was established through empirical testing to optimize the balance between sensitivity and precision. In Fig. S6 in the Supplementary Material the influence of different damage levels on the odds ratio threshold is visually modeled. The likelihood of observing the alignment by chance is calculated by multiplying the probability of a random match. Since every member sequence aligns to the center sequence with a user-defined sequence identity threshold, we use these values for a random match. For instance, if the threshold is set at 90% and the longest overlap is 20 bases, then the random model will have a likelihood of 0.9^{20}.

This likelihood ratio approach allows for a relatively low sequence identity threshold (e.g., 90%) in the first phase which includes clustering sequences based on shared k-mers and calculating the overlap alignment, facilitating the inclusion of highly deaminated sequences. At the same time, it ensures an early termination of the extension process to guard against erroneous extensions.

Damage-aware Rule for Merging Contigs

The extension process is divided into two distinct phases: an initial extension phase that exclusively considers non-extended aDNA fragments (damage patterns are only valid for unmodified sequences), followed by a second extension phase dedicated to merging contigs in the same cluster.

This second phase of extension focuses on merging contigs, allowing for a higher sequence identity threshold (98-99%) due to the correction of contigs in the previous phase. This threshold aligns with the default values in `PenguiN`. However, the potential presence of residual uncorrected bases, i.e. due to very low coverage, requires a modification of the extension strategy. However, we could not use the same strategy as described in Section 6, because firstly, the damage matrix becomes inapplicable to previously extended contigs due to the loss of positional information. Secondly, the significant variation in overlap length when merging contigs demands different penalizing strategies for shorter overlaps. For example, the contig merging step may encounter extension candidates with overlap lengths ranging from a few hundred to several thousand base pairs.

We modified `PenguiN`'s original extension rule (see Supplementary Note 1 in Jochheim *et al.* [62]) to score residual uncorrected bases. The original Bayesian approach in `PenguiN`'s algorithm models the extension process using a beta-binomial distribution, considering alignment length and mismatch count within overlaps. The extension of contigs stops when there are no longer any contigs matching above the identity threshold.

In the original rule, a match is assigned a value of 1 and a mismatch a value of 0. However, we have modified the value returned for mismatches between either C and T or G and A. The idea being that an extension candidate might have a mismatch due to an uncorrected base. Therefore, the sequence similarity of the overlap will be lower than it would be if the mismatch had been correctly corrected.

Given an observed mismatch of either C and T or G and A, we established a scoring function $f(s, \phi(L), \delta)$, that is:

$$f(s, L, \delta) = \delta \left(W \cdot s \cdot + (1 - W) \cdot \frac{\phi(L) - f_{min}}{f_{max} - f_{min}} \right) \tag{1}$$

where:

- L is the length of the alignment
- $s = \frac{m}{L}$ is the sequence similarity, with m being the number of matches in the alignment
- $\phi(L)$ is a scaling factor that accounts for the overlap length, which is based on a power law described by Sheinman *et al.* [83].
- δ is an approximation for the substitution rate. As the positional context is lost, we set it as the average damage rate for a particular base pair found in the original damage matrix.

Here, W is a weighting constant that balances the influence of the sequence similarity and the length-dependent factor. The value of W is set to 0.5. The function $\phi(L)$ is motivated by the work of Sheinman *et al.* [83], describing the relative frequency of identical sequences found among bacterial genomes depending on their length. We utilize the power law to evaluate an overlap being genuine by scoring the overlap length:

$$\phi(L) = \frac{1.4 \times 10^{-9}}{L^3} \tag{2}$$

For practical implementation, we establish boundary conditions for the scaling factor $\phi(L)$:

$$f_{min} = \phi(10) \quad \text{and} \quad f_{max} = \phi(10000) \tag{3}$$

These constraints indicate that an overlap of 10 base pairs provides minimal confidence that two sequences belong to the same genome, while an overlap of 10000 base pairs or more provides maximum confidence.

Thereby, we aim to account for residual damaged bases in the contig merging step, ensuring that extension candidates with short overlaps but very high sequence identity, and candidates with long overlaps but lower sequence identity due to residual damages, are not disproportionately favoured.

Misassembly Prevention by Consensus Calling: Introducing two modes of `CarpeDeam`

`CarpeDeam`'s algorithm relies on clustering sequences based on shared k-mers, utilizing the algorithm from the `MMseqs2` [76] k-mers matcher module. As described by Jochheim *et al.* [62], this approach enables the use of whole sequence overlaps, providing an advantage over de Bruijn graph assemblers that use fixed-size k-mers, losing the context of the aDNA fragment information.

For modern samples with low sequencing error rates ($\leq$1% [90]), `PenguiN` can apply a high default sequence similarity threshold of 99% between the center sequence and any member sequence, ensuring highly specific clusters. However, aDNA samples require a more lenient approach due to deaminated bases appearing as mismatches during clustering. To mitigate this issue while still excluding unsuitable candidates for extension, we implemented an additional threshold: the RYmer sequence identity. This method examines overlaps in Purine-Pyrimidine space, allowing mismatches between C and T as well as G and A, but restricting other mismatches. We also considered modifying the clustering algorithm to cluster sequences based on RYmers instead of k-mers. This would enable clustering impervious to damage. However, the extremely short nature of aDNA fragments prevented us from increasing the RYmer size sufficiently to achieve the specificity required for effective clustering and extension. We demonstrated this by plotting the fraction of unique k-mers and RYmers of different sizes found in 1000 different bacterial genomes. The results can be found in the Supplementary Material.

In practice, we applied an RYmer sequence identity threshold of 99.9% and a sequence identity threshold of 90%, permitting a certain level of Purine-Purine or Pyrimidine-Pyrimidine mismatches while minimizing other mismatches. Despite these measures, we observed a higher rate of misassemblies, particularly emerging in early extension phases where sequences are still very short ($\leq$200bp), as these sequences lack sufficient specificity for clustering while allowing for deamination-induced mismatches. Investigation of the higher rate of misassemblies observed with `CarpeDeam` can be found in the Supplementary Material.

To address these challenges, we developed a consensus-based approach that independently evaluates the left and right extending regions of all candidate sequences. These extending regions are the areas not covered by the center sequence but present in overlapping extension candidates. For each side (left and right), we construct a consensus sequence by considering all extension candidates. The consensus is built base-by-base, requiring at least five extension candidates to cover each position. This method results in two consensus sequences—one for the left extension and one for the right extension. The new center sequence is then formed by concatenating the left consensus, the original center sequence, and the right consensus. A schematic representation of this approach is shown in Fig. **10**. This approach is referred to as "safe mode" in the manuscript and is used as the default mode in `CarpeDeam`. It can be changed to "unsafe mode" by setting the parameter flag `--unsafe` to 1.

By aligning each member sequence to the consensus sequence and applying the updated overlap information to both the fragment extension and contig merging steps, we successfully reduced the level of misassemblies. However, this improvement came at the cost of a slight decrease in sensitivity. This trade-off resulted in the development of two distinct modes for `CarpeDeam`, offering users flexibility in balancing assembly accuracy and completeness based on their specific research requirements.

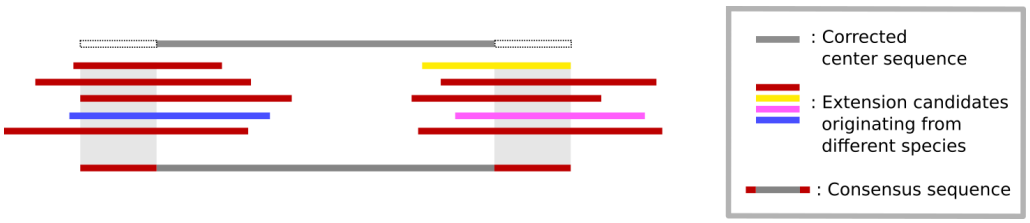

Figure 10: Consensus calling in the extending regions in a sequence cluster. `CarpeDeam` derives a consensus sequence from regions covered by at least five member sequences. Users can disable this "safe mode" to increase sensitivity, potentially at the cost of higher chimeric content ("unsafe mode"). The top sequence in grey is the center sequence: The longest sequence in the cluster that is to be extended.

Software limitations

The `CarpeDeam` assembler employs a greedy iterative approach, which may not be as time-efficient as the de Bruijn graph methods used by `MEGAHIT` or metSPAdes (see runtime and memory usage in Supplementary Material, Section S7). However, `CarpeDeam`'s enhanced sensitivity allows for more extensive genetic information recovery from ancient samples, compensating for this potential drawback. A key feature of `CarpeDeam` is its integration of damage matrices, which is a novel approach in aDNA assembly. Currently, `CarpeDeam` applies a global model of damage, which, while useful, does not account for the variability in damage across different sequences within a sample. Future refinements of `CarpeDeam` may incorporate species-specific damage profiles.

7 Declarations

Data Availability

The simulated reads for the analyzed datasets in this publication, along with additional files, are accessible on Zenodo under the DOI: 10.5281/zenodo.13208898.

Code Availability

The software is publicly available at https://github.com/LouisPwr/CarpeDeam. All scripts and benchmark data for assembly and analysis presented in this manuscript are available at https://github.com/LouisPwr/CarpeDeamAnalysis.

Software Versions

`PenguiN` 4-687d7 (GitHub clone from Jan. 2024), `MEGAHIT` v1.2.9, `metaSPAdes` v3.15.5, `metaQUAST` v5.0.2, `Bowtie2` 2.3.5.1-6, `SAMtools` 1.19, `seqkit` 2.6.1, `prodigal` V2.6.3, `MMseqs2` Release 15-6f452, `prokka` 1.14.6, `skani` 0.2.2, `miniprot` 0.13-r248

Competing Interests

The authors declare no competing interests.

Acknowledgements

We would like to acknowledge that this research and the PhD scholarships of Louis Kraft were funded by the Novo Nordisk Data Science Investigator grant number NNF20OC0062491. We thank the Technical University of Denmark's Department of Health Technology for additional funding. Additionally, we would like to thank Joshua Rubin and Nicola Vogel for their thoughtful comments

on our manuscript. We would like to acknowledge the computer resources provided by the BMBF-funded de.NBI Cloud within the German Network for Bioinformatics Infrastructure (de.NBI). L.K. acknowledges the use of OpenAI/ChatGPT and Anthropic/Claude to proofread the manuscript.

Author Contributions

G.R. conceived the project. L.K. and G.R. designed and developed the method. L.K. implemented the method and tested it. L.K. conducted all tests. L.K. and A.F.G simulated datasets. A.F.G. provided expertise, computational resources and benchmark workflows and conducted additional analyses. J.S. supported the method development. M.S. and A.J. contributed software expertise and guidance in addressing data-centric issues. P.W.S provided IT infrastructure expertise. L.K. and G.R. wrote the manuscript. All authors approved the final manuscript.

Funding

Funding for this research was provided by a Novo Nordisk Data Science Investigator grant number NNF20OC0062491 (GR). This funding source provided the salaries for LK. Additional funding for computational resources was provided by the Department for Health Technology at DTU. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

References

[1] A. Margaryan, D. J. Lawson, M. Sikora, F. Racimo, S. Rasmussen, I. Moltke, L. M. Cassidy, E. Jørsboe, A. Ingason, M. W. Pedersen, T. Korneliussen, H. Wilhelmson, M. M. Buś, P. de Barros Damgaard, R. Martiniano, G. Renaud, C. Bhérer, J. V. Moreno-Mayar, A. K. Fotakis, M. Allen, R. Allmäe, M. Molak, E. Cappellini, G. Scorrano, H. McColl, A. Buzhilova, A. Fox, A. Albrechtsen, B. Schütz, B. Skar, C. Arcini, C. Falys, C. H. Jonson, D. Błaszczyk, D. Pezhemsky, G. Turner-Walker, H. GestsdÓttir, I. Lundstrøm, I. Gustin, I. Mainland, I. Potekhina, I. M. Muntoni, J. Cheng, J. Stenderup, J. Ma, J. Gibson, J. Peets, J. Gustafsson, K. H. Iversen, L. Simpson, L. Strand, L. Loe, M. Sikora, M. Florek, M. Vretemark, M. Redknap, M. Bajka, T. Pushkina, M. Søvsø, N. Grigoreva, T. Christensen, O. Kastholm, O. Uldum, P. Favia, P. Holck, S. Sten, S. V. Arge, S. EllingvÅg, V. Moiseyev, W. Bogdanowicz, Y. Magnusson, L. Orlando, P. Pentz, M. D. Jessen, A. Pedersen, M. Collard, D. G. Bradley, M. L. Jørkov, J. Arneborg, N. Lynnerup, N. Price, M. T. P. Gilbert, M. E. Allentoft, J. Bill, S. M. Sindbæk, L. Hedeager, K. Kristiansen, R. Nielsen, T. Werge, and E. Willerslev, "Population genomics of the viking world," Nature, no. 7825, pp. 390–396, 2020.

[2] K. H. Kjær, M. Winther Pedersen, B. De Sanctis, B. De Cahsan, T. S. Korneliussen, C. S. Michelsen, K. K. Sand, S. Jelavić, A. H. Ruter, A. M. A. Schmidt, K. K. Kjeldsen, A. S. Tesakov, I. Snowball, J. C. Gosse, I. G. Alsos, Y. Wang, C. Dockter, M. Rasmussen, M. E. Jørgensen, B. Skadhauge, A. Prohaska, J. Å. Kristensen, M. Bjerager, M. E. Allentoft, E. Coissac, A. Rouillard, A. Simakova, A. Fernandez-Guerra, C. Bowler, M. Macias-Fauria, L. Vinner, J. J. Welch, A. J. Hidy, M. Sikora, M. J. Collins, R. Durbin, N. K. Larsen, and E. Willerslev, "A 2-million-year-old ecosystem in greenland uncovered by environmental DNA," Nature, vol. 612, pp. 283–291, 12 2022.

[3] A. Fernandez-Guerra, G. Borrel, T. O. Delmont, B. Elberling, A. M. Eren, S. Gribaldo, A. Jochheim, R. A. Henriksen, K.-U. Hinrichs, T. S. Korneliussen, M. Krupovic, N. K. Larsen, R. Laso-Pérez, M. W. Pedersen, V. K. Pedersen, K. K. Sand, M. Sikora, M. Steinegger, I. Veseli, L. Wörmer, L. Zhao, M. Žure, K. Kjær, and E. Willerslev, "A 2-million-year-old microbial and viral communities from the kap københavn formation in north greenland," Cold Spring Harbor Laboratory, 6 2023.

[4] N. A. Vogel, J. D. Rubin, M. Swartz, J. Vlieghe, P. W. Sackett, A. G. Pedersen, M. W. Pedersen, and G. Renaud, "euka: Robust tetrapodic and arthropodic taxa detection from modern and ancient environmental ¡scp¿dna¡/scp¿ using pangenomic reference graphs," Methods in Ecology and Evolution, vol. 14, p. 2717–2727, 9 2023.

[5] K. Prüfer, U. Stenzel, M. Hofreiter, S. Pääbo, J. Kelso, and R. E. Green, "Computational challenges in the analysis of ancient dna," Genome Biology, vol. 11, no. 5, p. R47, 2010.

[6] C. Der Sarkissian, I. M. Velsko, A. K. Fotakis, Å. J. Vågene, A. Hübner, and J. A. Fellows Yates, "Ancient metagenomic studies: Considerations for the wider scientific community," mSystems, vol. 6, 12 2021.

[7] A. W. Briggs, U. Stenzel, P. L. F. Johnson, R. E. Green, J. Kelso, K. Prüfer, M. Meyer, J. Krause, M. T. Ronan, M. Lachmann, and S. Pääbo, "Patterns of damage in genomic dna sequences from a neandertal," Proceedings of the National Academy of Sciences, vol. 104, p. 14616–14621, 9 2007.

[8] C. Michelsen, M. W. Pedersen, A. Fernandez-Guerra, L. Zhao, T. C. Petersen, and T. S. Korneliussen, "metadmg – a fast and accurate ancient dna damage toolkit for metagenomic data," Cold Spring Harbor Laboratory, 12 2022.

[9] J. Neukamm, A. Peltzer, and K. Nieselt, "Damageprofiler: fast damage pattern calculation for ancient dna," Bioinformatics, vol. 37, p. 3652–3653, 4 2021.

[10] H. Jónsson, A. Ginolhac, M. Schubert, P. L. F. Johnson, and L. Orlando, "mapdamage2.0: fast approximate bayesian estimates of ancient dna damage parameters," Bioinformatics, vol. 29, p. 1682–1684, 4 2013.

[11] L. Zhao, R. A. Heriksen, A. D. Ramsøe, R. Nielsen, and T. S. Korneliussen, "Revisiting the briggs ancient dna damage model: a fast regression method to estimate postmortem damage," Cold Spring Harbor Laboratory, Nov. 2023.

[12] J. Dabney, M. Meyer, and S. Paabo, "Ancient dna damage," Cold Spring Harbor Perspectives in Biology, vol. 5, p. a012567–a012567, 5 2013.

[13] S. Pääbo, H. Poinar, D. Serre, V. Jaenicke-Després, J. Hebler, N. Rohland, M. Kuch, J. Krause, L. Vigilant, and M. Hofreiter, "Genetic analyses from ancient dna," Annual Review of Genetics, vol. 38, p. 645–679, 12 2004.

[14] E. Willerslev and A. Cooper, "Review paper. ancient dna," Proceedings of the Royal Society B: Biological Sciences, vol. 272, p. 3–16, 12 2004.

[15] S. Peyrégne and B. M. Peter, "Authentict: a model of ancient dna damage to estimate the proportion of present-day dna contamination," Genome Biology, vol. 21, 9 2020.

[16] G. Renaud, V. Slon, A. T. Duggan, and J. Kelso, "Schmutzi: estimation of contamination and endogenous mitochondrial consensus calling for ancient dna," Genome Biology, vol. 16, 10 2015.

[17] M. Klapper, A. Hübner, A. Ibrahim, I. Wasmuth, M. Borry, V. G. Haensch, S. Zhang, W. K. Al-Jammal, H. Suma, J. A. Fellows Yates, J. Frangenberg, I. M. Velsko, S. Chowdhury, R. Herbst, E. V. Bratovanov, H.-M. Dahse, T. Horch, C. Hertweck, M. R. González Morales, L. G. Straus, I. Vilotijevic, C. Warinner, and P. Stallforth, "Natural products from reconstructed bacterial genomes of the middle and upper paleolithic," Science, vol. 380, p. 619–624, 5 2023.

[18] A. Oliva, R. Tobler, A. Cooper, B. Llamas, and Y. Souilmi, "Systematic benchmark of ancient dna read mapping," Briefings in Bioinformatics, vol. 22, 4 2021.

[19] I. M. Velsko, L. A. F. Frantz, A. Herbig, G. Larson, and C. Warinner, "Selection of appropriate metagenome taxonomic classifiers for ancient microbiome research," mSystems, vol. 3, Aug. 2018.

[20] J. C. Wooley, A. Godzik, and I. Friedberg, "A primer on metagenomics," PLoS Computational Biology, vol. 6, p. e1000667, 2 2010.

[21] J. Handelsman, "Metagenomics: Application of genomics to uncultured microorganisms," Microbiology and Molecular Biology Reviews, vol. 68, p. 669–685, Dec. 2004.

[22] S. Nayfach, S. Roux, R. Seshadri, D. Udwary, N. Varghese, F. Schulz, D. Wu, D. Paez-Espino, I.-M. Chen, M. Huntemann, K. Palaniappan, J. Ladau, S. Mukherjee, T. B. K. Reddy, T. Nielsen, E. Kirton, J. P. Faria, J. N. Edirisinghe, C. S. Henry, S. P. Jungbluth, D. Chivian, P. Dehal, E. M. Wood-Charlson, A. P. Arkin, S. G. Tringe, A. Visel, H. Abreu, S. G. Acinas, E. Allen, M. A. Allen, L. V. Alteio, G. Andersen, A. M. Anesio, G. Attwood, V. Avila-Magaña, Y. Badis, J. Bailey, B. Baker, P. Baldrian, H. A. Barton, D. A. C. Beck, E. D. Becraft, H. R. Beller, J. M. Beman, R. Bernier-Latmani, T. D. Berry, A. Bertagnolli, S. Bertilsson, J. M. Bhatnagar, J. T. Bird, J. L. Blanchard, S. E. Blumer-Schuette, B. Bohannan, M. A. Borton, A. Brady, S. H. Brawley, J. Brodie, S. Brown, J. R. Brum, A. Brune, D. A. Bryant, A. Buchan, D. H. Buckley, J. Buongiorno, H. Cadillo-Quiroz, S. M. Caffrey, A. N. Campbell, B. Campbell, S. Carr, J. Carroll, S. C. Cary, A. M. Cates, R. A. Cattolico, R. Cavicchioli, L. Chistoserdova, M. L. Coleman, P. Constant, J. M. Conway, W. P. Mac Cormack, S. Crowe, B. Crump, C. Currie, R. Daly, K. M. DeAngelis, V. Denef, S. E. Denman, A. Desta, H. Dionisi, J. Dodsworth, N. Dombrowski, T. Donohue, M. Dopson, T. Driscoll, P. Dunfield, C. L. Dupont, K. A. Dynarski, V. Edgcomb, E. A. Edwards, M. S. Elshahed, I. Figueroa, B. Flood, N. Fortney, C. S. Fortunato, C. Francis, C. M. M. Gachon, S. L. Garcia, M. C. Gazitua, T. Gentry, L. Gerwick, J. Gharechahi, P. Girguis, J. Gladden, M. Gradoville, S. E. Grasby, K. Gravuer, C. L. Grettenberger, R. J. Gruninger, J. Guo, M. Y. Habteselassie, S. J. Hallam, R. Hatzenpichler, B. Hausmann, T. C. Hazen, B. Hedlund, C. Henny, L. Herfort, M. Hernandez, O. S. Hershey, M. Hess, E. B. Hollister, L. A. Hug, D. Hunt, J. Jansson, J. Jarett, V. V. Kadnikov, C. Kelly, R. Kelly, W. Kelly, C. A. Kerfeld, J. Kimbrel, J. L. Klassen, K. T. Konstantinidis, L. L. Lee, W.-J. Li, A. J. Loder, A. Loy, M. Lozada, B. MacGregor, C. Magnabosco, A. Maria da Silva, R. M. McKay, K. McMahon, C. S. McSweeney, M. Medina, L. Meredith, J. Mizzi, T. Mock, L. Momper, M. A. Moran, C. Morgan-Lang, D. Moser, G. Muyzer, D. Myrold, M. Nash, C. L. Nesbø, A. P. Neumann, R. B. Neumann, D. Noguera, T. Northen, J. Norton, B. Nowinski, K. Nüsslein, M. A. O'Malley, R. S. Oliveira, V. Maia de Oliveira, T. Onstott, J. Osvatic, Y. Ouyang, M. Pachiadaki, J. Parnell, L. P.

Partida-Martinez, K. G. Peay, D. Pelletier, X. Peng, M. Pester, J. Pett-Ridge, S. Peura, P. Pjevac, A. M. Plominsky, A. Poehlein, P. B. Pope, N. Ravin, M. C. Redmond, R. Reiss, V. Rich, C. Rinke, J. L. M. Rodrigues, W. Rodriguez-Reillo, K. Rossmassler, J. Sackett, G. H. Salekdeh, S. Saleska, M. Scarborough, D. Schachtman, C. W. Schadt, M. Schrenk, A. Sczyrba, A. Sengupta, J. C. Setubal, A. Shade, C. Sharp, D. H. Sherman, O. V. Shubenkova, I. N. Sierra-Garcia, R. Simister, H. Simon, S. Sjöling, J. Slonczewski, R. S. Correa de Souza, J. R. Spear, J. C. Stegen, R. Stepanauskas, F. Stewart, G. Suen, M. Sullivan, D. Sumner, B. K. Swan, W. Swingley, J. Tarn, G. T. Taylor, H. Teeling, M. Tekere, A. Teske, T. Thomas, C. Thrash, J. Tiedje, C. S. Ting, B. Tully, G. Tyson, O. Ulloa, D. L. Valentine, M. W. Van Goethem, J. VanderGheynst, T. J. Verbeke, J. Vollmers, A. Vuillemin, N. B. Waldo, D. A. Walsh, B. C. Weimer, T. Whitman, P. van der Wielen, M. Wilkins, T. J. Williams, B. Woodcroft, J. Woolet, K. Wrighton, J. Ye, E. B. Young, N. H. Youssef, F. B. Yu, T. I. Zemskaya, R. Ziels, T. Woyke, N. J. Mouncey, N. N. Ivanova, N. C. Kyrpides, and E. A. Eloe-Fadrosh, "A genomic catalog of earth's microbiomes," Nature Biotechnology, vol. 39, p. 499–509, Nov. 2020.

[23] L. P. Coelho, R. Alves, A. R. del Rio, P. N. Myers, C. P. Cantalapiedra, J. Giner-Lamia, T. S. Schmidt, D. R. Mende, A. Orakov, I. Letunic, F. Hildebrand, T. Van Rossum, S. K. Forslund, S. Khedkar, O. M. Maistrenko, S. Pan, L. Jia, P. Ferretti, S. Sunagawa, X.-M. Zhao, H. B. Nielsen, J. Huerta-Cepas, and P. Bork, "Towards the biogeography of prokaryotic genes," Nature, vol. 601, pp. 252–256, Dec. 2021.

[24] L. A. Hug, "The ever-changing tree of life," Nature Microbiology, Aug. 2024.

[25] A. M. Eren, E. Kiefl, A. Shaiber, I. Veseli, S. E. Miller, M. S. Schechter, I. Fink, J. N. Pan, M. Yousef, E. C. Fogarty, F. Trigodet, A. R. Watson, O. C. Esen, R. M. Moore, Q. Clayssen, M. D. Lee, V. Kivenson, E. D. Graham, B. D. Merrill, A. Karkman, D. Blankenberg, J. M. Eppley, A. Sjödin, J. J. Scott, X. Vázquez-Campos, L. J. McKay, E. A. McDaniel, S. L. R. Stevens, R. E. Anderson, J. Fuessel, A. Fernandez-Guerra, L. Maignien, T. O. Delmont, and A. D. Willis, "Community-led, integrated, reproducible multi-omics with anvi'o," Nature Microbiology, vol. 6, p. 3–6, Dec. 2020.

[26] C. Quince, A. W. Walker, J. T. Simpson, N. J. Loman, and N. Segata, "Shotgun metagenomics, from sampling to analysis," Nature Biotechnology, vol. 35, p. 833–844, 9 2017.

[27] S. H. Ye, K. J. Siddle, D. J. Park, and P. C. Sabeti, "Benchmarking metagenomics tools for taxonomic classification," Cell, vol. 178, p. 779–794, 8 2019.

[28] A. Blanco-Míguez, F. Beghini, F. Cumbo, L. J. McIver, K. N. Thompson, M. Zolfo, P. Manghi, L. Dubois, K. D. Huang, A. M. Thomas, W. A. Nickols, G. Piccinno, E. Piperni, M. Punčochář, M. Valles-Colomer, A. Tett, F. Giordano, R. Davies, J. Wolf, S. E. Berry, T. D. Spector, E. A. Franzosa, E. Pasolli, F. Asnicar, C. Huttenhower, and N. Segata, "Extending and improving metagenomic taxonomic profiling with uncharacterized species using metaphlan 4," Nature Biotechnology, vol. 41, p. 1633–1644, 2 2023.

[29] D. E. Wood and S. L. Salzberg, "Kraken: ultrafast metagenomic sequence classification using exact alignments," Genome Biology, vol. 15, 3 2014.

[30] J. Lu, F. P. Breitwieser, P. Thielen, and S. L. Salzberg, "Bracken: estimating species abundance in metagenomics data," PeerJ Computer Science, vol. 3, p. e104, 1 2017.

[31] D. Kim, L. Song, F. P. Breitwieser, and S. L. Salzberg, "Centrifuge: rapid and sensitive classification of metagenomic sequences," Genome Research, vol. 26, p. 1721–1729, Oct. 2016.

[32] D. H. Huson, B. Albrecht, C. Bağcı, I. Bessarab, A. Górska, D. Jolic, and R. B. H. Williams, "Megan-lr: new algorithms allow accurate binning and easy interactive exploration of metagenomic long reads and contigs," Biology Direct, vol. 13, Jan. 2018.

[33] P. Menzel, K. L. Ng, and A. Krogh, "Fast and sensitive taxonomic classification for metagenomics with kaiju," Nature Communications, vol. 7, Apr. 2016.

[34] M. Mirdita, M. Steinegger, F. Breitwieser, J. Söding, and E. Levy Karin, "Fast and sensitive taxonomic assignment to metagenomic contigs," Bioinformatics, vol. 37, p. 3029–3031, 3 2021.

[35] L. Granehäll, K. D. Huang, A. Tett, P. Manghi, A. Paladin, N. O'Sullivan, O. Rota-Stabelli, N. Segata, A. Zink, and F. Maixner, "Metagenomic analysis of ancient dental calculus reveals unexplored diversity of oral archaeal methanobrevibacter," Microbiome, vol. 9, 9 2021.

[36] J. A. Fellows Yates, I. M. Velsko, F. Aron, C. Posth, C. A. Hofman, R. M. Austin, C. E. Parker, A. E. Mann, K. Nägele, K. W. Arthur, J. W. Arthur, C. C. Bauer, I. Crevecoeur, C. Cupillard, M. C. Curtis, L. Dalén, M. Díaz-Zorita Bonilla, J. C. Díez Fernández-Lomana, D. G. Drucker, E. Escribano Escrivá, M. Francken, V. E. Gibbon, M. R. González Morales, A. Grande Mateu, K. Harvati, A. G. Henry, L. Humphrey, M. Menéndez, D. Mihailović, M. Peresani, S. Rodríguez Moroder, M. Roksandic, H. Rougier, S. Sázelová, J. T. Stock, L. G. Straus, J. Svoboda, B. Teßmann, M. J. Walker, R. C. Power, C. M. Lewis, K. Sankaranarayanan, K. Guschanski, R. W. Wrangham, F. E. Dewhirst, D. C. Salazar-García, J. Krause, A. Herbig, and C. Warinner, "The evolution and changing ecology of the african hominid oral microbiome," Proceedings of the National Academy of Sciences, vol. 118, 5 2021.

[37] Z. Patterson Ross, J. Klunk, G. Fornaciari, V. Giuffra, S. Duchêne, A. T. Duggan, D. Poinar, M. W. Douglas, J.-S. Eden, E. C. Holmes, and H. N. Poinar, "The paradox of hbv evolution as revealed from a 16th century mummy," PLOS Pathogens, vol. 14, p. e1006750, 1 2018.

[38] C. P. Andam, C. J. Worby, Q. Chang, and M. G. Campana, "Microbial genomics of ancient plagues and outbreaks," Trends in Microbiology, vol. 24, p. 978–990, 12 2016.

[39] M. L. Carpenter, J. D. Buenrostro, C. Valdiosera, H. Schroeder, M. E. Allentoft, M. Sikora, M. Rasmussen, S. Gravel, S. Guillén, G. Nekhrizov, K. Leshtakov, D. Dimitrova, N. Theodossiev, D. Pettener, D. Luiselli, K. Sandoval, A. Moreno-Estrada, Y. Li, J. Wang, M. T. P. Gilbert, E. Willerslev, W. J. Greenleaf, and C. D. Bustamante, "Pulling out the 1," The American Journal of Human Genetics, vol. 93, p. 852–864, 11 2013.

[40] R. Hübler, F. M. Key, C. Warinner, K. I. Bos, J. Krause, and A. Herbig, "Hops: automated detection and authentication of pathogen dna in archaeological remains," Genome Biology, vol. 20, 12 2019.

[41] Z. Pochon, N. Bergfeldt, E. Kırdök, M. Vicente, T. Naidoo, T. van der Valk, N. E. Altınışık, M. Krzewińska, L. Dalén, A. Götherström, C. Mirabello, P. Unneberg, and N. Oskolkov, "ameta: an accurate and memory-efficient ancient metagenomic profiling workflow," Genome Biology, vol. 24, 10 2023.

[42] S. Liu, C. D. Moon, N. Zheng, S. Huws, S. Zhao, and J. Wang, "Opportunities and challenges of using metagenomic data to bring uncultured microbes into cultivation," Microbiome, vol. 10, May 2022.

[43] M. Hofreiter, J. L. A. Paijmans, H. Goodchild, C. F. Speller, A. Barlow, G. G. Fortes, J. A. Thomas, A. Ludwig, and M. J. Collins, "The future of ancient dna: Technical advances and conceptual shifts," BioEssays, vol. 37, p. 284–293, Nov. 2014.

[44] M. C. Wibowo, Z. Yang, M. Borry, A. Hübner, K. D. Huang, B. T. Tierney, S. Zimmerman, F. Barajas-Olmos, C. Contreras-Cubas, H. García-Ortiz, A. Martínez-Hernández, J. M. Luber, P. Kirstahler, T. Blohm, F. E. Smiley, R. Arnold, S. A. Ballal, S. J. Pamp, J. Russ, F. Maixner, O. Rota-Stabelli, N. Segata, K. Reinhard, L. Orozco, C. Warinner, M. Snow, S. LeBlanc, and A. D. Kostic, "Reconstruction of ancient microbial genomes from the human gut," Nature, vol. 594, p. 234–239, 5 2021.

[45] M. A. Spyrou, K. I. Bos, A. Herbig, and J. Krause, "Ancient pathogen genomics as an emerging tool for infectious disease research," Nature Reviews Genetics, vol. 20, p. 323–340, 4 2019.

[46] P. Rozwalak, J. Barylski, Y. Wijesekara, B. E. Dutilh, and A. Zielezinski, "Ultraconserved bacteriophage genome sequence identified in 1300-year-old human palaeofaeces," Nature Communications, vol. 15, 1 2024.

[47] J. C. Brealey, H. G. Leitão, T. van der Valk, W. Xu, K. Bougiouri, L. Dalén, and K. Guschanski, "Dental calculus as a tool to study the evolution of the mammalian oral microbiome," Molecular Biology and Evolution, vol. 37, p. 3003–3022, 5 2020.

[48] H. P. Hodgins, P. Chen, B. Lobb, X. Wei, B. J. M. Tremblay, M. J. Mansfield, V. C. Y. Lee, P.-G. Lee, J. Coffin, A. T. Duggan, A. E. Dolphin, G. Renaud, M. Dong, and A. C. Doxey, "Ancient clostridium dna and variants of tetanus neurotoxins associated with human archaeological remains," Nature Communications, vol. 14, 9 2023.

[49] L. G. E. Wilkins, C. L. Ettinger, G. Jospin, and J. A. Eisen, "Metagenome-assembled genomes provide new insight into the microbial diversity of two thermal pools in kamchatka, russia," Scientific Reports, vol. 9, Feb. 2019.

[50] F. Wan, M. D. T. Torres, J. Peng, and C. de la Fuente-Nunez, "Deep-learning-enabled antibiotic discovery through molecular de-extinction," Nature Biomedical Engineering, 6 2024.

[51] S. L. Salzberg, "Next-generation genome annotation: we still struggle to get it right," Genome Biology, vol. 20, 5 2019.

[52] V. L. Chan, "Bacterial genomes and infectious diseases," Pediatric Research, vol. 54, p. 1–7, 7 2003.

[53] T. T. Binnewies, Y. Motro, P. F. Hallin, O. Lund, D. Dunn, T. La, D. J. Hampson, M. Bellgard, T. M. Wassenaar, and D. W. Ussery, "Ten years of bacterial genome sequencing: comparative-genomics-based discoveries," Functional & Integrative Genomics, vol. 6, p. 165–185, 5 2006.

[54] S. Rasmussen, M. E. Allentoft, K. Nielsen, L. Orlando, M. Sikora, K.-G. Sjögren, A. G. Pedersen, M. Schubert, A. Van Dam, C. M. O. Kapel, H. B. Nielsen, S. Brunak, P. Avetisyan, A. Epimakhov, M. V. Khalyapin, A. Gnuni, A. Kriiska, I. Lasak, M. Metspalu, V. Moiseyev, A. Gromov, D. Pokutta, L. Saag, L. Varul, L. Yepiskoposyan, T. Sicheritz-Pontén, R. A. Foley, M. M. Lahr, R. Nielsen, K. Kristiansen, and E. Willerslev, "Early divergent strains of yersinia pestis in eurasia 5,000 years ago," Cell, vol. 163, p. 571–582, 10 2015.

[55] V. J. Schuenemann, K. Bos, S. DeWitte, S. Schmedes, J. Jamieson, A. Mittnik, S. Forrest, B. K. Coombes, J. W. Wood, D. J. D. Earn, W. White, J. Krause, and H. N. Poinar, "Targeted enrichment of ancient pathogens yielding the ppcp1 plasmid ofyersinia pestisfrom victims of the black death," Proceedings of the National Academy of Sciences, vol. 108, 8 2011.

[56] S. Rampelli, S. Turroni, C. Mallol, C. Hernandez, B. Galván, A. Sistiaga, E. Biagi, A. Astolfi, P. Brigidi, S. Benazzi, C. M. Lewis, C. Warinner, C. A. Hofman, S. L. Schnorr, and M. Candela, "Components of a neanderthal gut microbiome recovered from fecal sediments from el salt," Communications Biology, vol. 4, 2 2021.

[57] M. Sikora, E. Canteri, A. Fernandez-Guerra, N. Oskolkov, R. Ågren, L. Hansson, E. K. Irving-Pease, B. Mühlemann, S. H. Nielsen, G. Scorrano, M. E. Allentoft, F. V. Seersholm, H. Schroeder, C. Gaunitz, J. Stenderup, L. Vinner, T. C. Jones, B. Nystedt, J. Parkhill, L. Fugger, F. Racimo, K. Kristiansen, A. K. N. Iversen, and E. Willerslev, "The landscape of ancient human pathogens in eurasia from the stone age to historical times," Cold Spring Harbor Laboratory, 10 2023.

[58] D. Li, C.-M. Liu, R. Luo, K. Sadakane, and T.-W. Lam, "Megahit: an ultra-fast single-node solution for large and complex metagenomics assembly via succinct de bruijn graph," Bioinformatics, vol. 31, p. 1674–1676, 1 2015.

[59] S. Nurk, D. Meleshko, A. Korobeynikov, and P. A. Pevzner, "metaspades: a new versatile metagenomic assembler," Genome Research, vol. 27, p. 824–834, 3 2017.

[60] Z. Li, Y. Chen, D. Mu, J. Yuan, Y. Shi, H. Zhang, J. Gan, N. Li, X. Hu, B. Liu, B. Yang, and W. Fan, "Comparison of the two major classes of assembly algorithms: overlap-layout-consensus and de-bruijn-graph," Briefings in Functional Genomics, vol. 11, p. 25–37, 12 2011.

[61] B. Haider, T.-H. Ahn, B. Bushnell, J. Chai, A. Copeland, and C. Pan, "Omega: an overlap-graphde novoassembler for metagenomics," Bioinformatics, vol. 30, p. 2717–2722, June 2014.

[62] A. Jochheim, F. A. Jochheim, A. Kolodyazhnaya, É. Morice, M. Steinegger, and J. Söding, "Strain-resolved de-novo metagenomic assembly of viral genomes and microbial 16s rrnas," Cold Spring Harbor Laboratory, 3 2024.

[63] V. Raghavan, L. Kraft, F. Mesny, and L. Rigerte, "A simple guide to de novo transcriptome assembly and annotation," Briefings in Bioinformatics, vol. 23, 1 2022.

[64] A. Bankevich, S. Nurk, D. Antipov, A. A. Gurevich, M. Dvorkin, A. S. Kulikov, V. M. Lesin, S. I. Nikolenko, S. Pham, A. D. Prjibelski, A. V. Pyshkin, A. V. Sirotkin, N. Vyahhi, G. Tesler, M. A. Alekseyev, and P. A. Pevzner, "Spades: A new genome assembly algorithm and its applications to single-cell sequencing," Journal of Computational Biology, vol. 19, p. 455–477, 5 2012.

[65] M. Steinegger, M. Mirdita, and J. Söding, "Protein-level assembly increases protein sequence recovery from metagenomic samples manyfold," Nature Methods, vol. 16, p. 603–606, 6 2019.

[66] S. Whelan, J. E. Allen, B. P. Blackburne, and D. Talavera, "Modelomatic: Fast and automated model selection between ry, nucleotide, amino acid, and codon substitution models," Systematic Biology, vol. 64, p. 42–55, 9 2014.

[67] A. Lien, L. P. Legori, L. Kraft, P. W. Sackett, and G. Renaud, "Benchmarking software tools for trimming adapters and merging next-generation sequencing data for ancient dna," Frontiers in Bioinformatics, vol. 3, Dec. 2023.

[68] M. Steinegger and J. Söding, "Clustering huge protein sequence sets in linear time," Nature Communications, vol. 9, 6 2018.

[69] C. Der Sarkissian, L. Ermini, H. Jónsson, A. N. Alekseev, E. Crubezy, B. Shapiro, and L. Orlando, "Shotgun microbial profiling of fossil remains," Molecular Ecology, vol. 23, p. 1780–1798, 3 2014.

[70] A. Mikheenko, V. Saveliev, and A. Gurevich, "Metaquast: evaluation of metagenome assemblies," Bioinformatics, vol. 32, p. 1088–1090, 11 2015.

[71] B. Langmead and S. L. Salzberg, "Fast gapped-read alignment with bowtie 2," Nature Methods, vol. 9, p. 357–359, 3 2012.

[72] H. Li, B. Handsaker, A. Wysoker, T. Fennell, J. Ruan, N. Homer, G. Marth, G. Abecasis, and R. Durbin, "The sequence alignment/map format and samtools," Bioinformatics, vol. 25, p. 2078–2079, 6 2009.

[73] J. Shaw and Y. W. Yu, "Fast and robust metagenomic sequence comparison through sparse chaining with skani," Nature Methods, vol. 20, p. 1661–1665, Sept. 2023.

[74] S. Commichaux, N. Shah, J. Ghurye, A. Stoppel, J. A. Goodheart, G. G. Luque, M. P. Cummings, and M. Pop, "A critical assessment of gene catalogs for metagenomic analysis," Bioinformatics, vol. 37, p. 2848–2857, Apr. 2021.

[75] T. Seemann, "Prokka: rapid prokaryotic genome annotation," Bioinformatics, vol. 30, p. 2068–2069, 3 2014.

[76] M. Steinegger and J. Söding, "Mmseqs2 enables sensitive protein sequence searching for the analysis of massive data sets," Nature Biotechnology, vol. 35, p. 1026–1028, 10 2017.

[77] H. Li, "Protein-to-genome alignment with miniprot," Bioinformatics, vol. 39, Jan. 2023.

[78] G. Renaud, U. Stenzel, and J. Kelso, "leehom: adaptor trimming and merging for illumina sequencing reads," Nucleic Acids Research, vol. 42, p. e141–e141, 8 2014.

[79] B. E. Suzek, Y. Wang, H. Huang, P. B. McGarvey, and C. H. Wu, "Uniref clusters: a comprehensive and scalable alternative for improving sequence similarity searches," Bioinformatics, vol. 31, p. 926–932, 11 2014.

[80] F. A. B. von Meijenfeldt, K. Arkhipova, D. D. Cambuy, F. H. Coutinho, and B. E. Dutilh, "Robust taxonomic classification of uncharted microbial sequences and bins with cat and bat," Genome Biology, vol. 20, Oct. 2019.

[81] D. Hyatt, G.-L. Chen, P. F. LoCascio, M. L. Land, F. W. Larimer, and L. J. Hauser, "Prodigal: prokaryotic gene recognition and translation initiation site identification," BMC Bioinformatics, vol. 11, 3 2010.

[82] O. Smith, G. Dunshea, M.-H. S. Sinding, S. Fedorov, M. Germonpre, H. Bocherens, and M. T. P. Gilbert, "Ancient rna from late pleistocene permafrost and historical canids shows tissue-specific transcriptome survival," PLOS Biology, vol. 17, p. e3000166, July 2019.

[83] M. Sheinman, K. Arkhipova, P. F. Arndt, B. E. Dutilh, R. Hermsen, and F. Massip, "Identical sequences found in distant genomes reveal frequent horizontal transfer across the bacterial domain," eLife, vol. 10, 6 2021.

[84] M. G. Durrant, M. M. Li, B. A. Siranosian, S. B. Montgomery, and A. S. Bhatt, "A bioinformatic analysis of integrative mobile genetic elements highlights their role in bacterial adaptation," Cell Host & Microbe, vol. 27, pp. 140–153.e9, 1 2020.

[85] T. Günther, C. Valdiosera, H. Malmström, I. Ureña, R. Rodriguez-Varela, Ó. O. Sverrisdöttir, E. A. Daskalaki, P. Skoglund, T. Naidoo, E. M. Svensson, J. M. Bermúdez de Castro, E. Carbonell, M. Dunn, J. Storå, E. Iriarte, J. L. Arsuaga, J.-M. Carretero, A. Götherström, and M. Jakobsson, "Ancient genomes link early farmers from atapuerca in spain to modern-day basques," Proceedings of the National Academy of Sciences, vol. 112, p. 11917–11922, 9 2015.

[86] G. Renaud, K. Hanghøj, E. Willerslev, and L. Orlando, "gargammel: a sequence simulator for ancient dna," Bioinformatics, vol. 33, p. 577–579, 11 2016.

[87] M.-T. Gansauge and M. Meyer, "Selective enrichment of damaged dna molecules for ancient genome sequencing," Genome Research, vol. 24, p. 1543–1549, 7 2014.

[88] D. H. Parks, M. Chuvochina, D. W. Waite, C. Rinke, A. Skarshewski, P.-A. Chaumeil, and P. Hugenholtz, "A standardized bacterial taxonomy based on genome phylogeny substantially revises the tree of life," Nature Biotechnology, vol. 36, p. 996–1004, 8 2018.

[89] A. Bateman, M.-J. Martin, S. Orchard, M. Magrane, R. Agivetova, S. Ahmad, E. Alpi, E. H. Bowler-Barnett, R. Britto, B. Bursteinas, H. Bye-A-Jee, R. Coetzee, A. Cukura, A. Da Silva, P. Denny, T. Dogan, T. Ebenezer, J. Fan, L. G. Castro, P. Garmiri, G. Georghiou, L. Gonzales, E. Hatton-Ellis, A. Hussein, A. Ignatchenko, G. Insana, R. Ishtiaq, P. Jokinen, V. Joshi, D. Jyothi, A. Lock, R. Lopez, A. Luciani, J. Luo, Y. Lussi, A. MacDougall, F. Madeira, M. Mahmoudy, M. Menchi, A. Mishra, K. Moulang, A. Nightingale, C. S. Oliveira, S. Pundir, G. Qi, S. Raj, D. Rice, M. R. Lopez, R. Saidi, J. Sampson, T. Sawford, E. Speretta, E. Turner, N. Tyagi, P. Vasudev, V. Volynkin, K. Warner, X. Watkins, R. Zaru, H. Zellner, A. Bridge, S. Poux, N. Redaschi, L. Aimo, G. Argoud-Puy, A. Auchincloss, K. Axelsen, P. Bansal, D. Baratin, M.-C. Blatter, J. Bolleman, E. Boutet, L. Breuza, C. Casals-Casas, E. de Castro, K. C. Echioukh, E. Coudert, B. Cuche, M. Doche, D. Dornevil, A. Estreicher, M. L. Famiglietti, M. Feuermann, E. Gasteiger, S. Gehant, V. Gerritsen, A. Gos, N. Gruaz-Gumowski, U. Hinz, C. Hulo, N. Hyka-Nouspikel, F. Jungo, G. Keller, A. Kerhornou, V. Lara, P. Le Mercier, D. Lieberherr, T. Lombardot, X. Martin, P. Masson, A. Morgat, T. B. Neto, S. Paesano, I. Pedruzzi, S. Pilbout, L. Pourcel, M. Pozzato, M. Pruess, C. Rivoire, C. Sigrist, K. Sonesson, A. Stutz, S. Sundaram, M. Tognolli, L. Verbregue, C. H. Wu, C. N. Arighi, L. Arminski, C. Chen, Y. Chen, J. S. Garavelli, H. Huang, K. Laiho, P. McGarvey, D. A. Natale, K. Ross, C. R. Vinayaka, Q. Wang, Y. Wang, L.-S. Yeh, J. Zhang, P. Ruch, and D. Teodoro, "Uniprot: the universal protein knowledgebase in 2021," Nucleic Acids Research, vol. 49, p. D480–D489, 11 2020.

[90] N. Stoler and A. Nekrutenko, "Sequencing error profiles of illumina sequencing instruments," NAR Genomics and Bioinformatics, vol. 3, 1 2021.