Supplementary Information for:

FALCON: a method to infer metagenomic composition of ancient DNA

D. Pratas[1,2], A. J. Pinho[1,2], R. M. Silva[1,3],
J. M. O. S. Rodrigues[1,2], M. Hosseini[1],
T. Caetano[4], P. J. S. G. Ferreira[1,2]

IEETA, University of Aveiro, Portugal
DETI, University of Aveiro, Portugal
DMS & iBiMED, University of Aveiro, Portugal
DBio & CESAM, University of Aveiro, Portugal

Contents

Note 1: Normalized Relative Similarity (NRS)

The problem of quantifying the complexity (information, algorithmic entropy) [1, 2, 3, 4] contained in arbitrary sequences was deeply considered by Kolmogorov [5], who addressed three ways of measuring it: the combinatorial, the probabilistic and the algorithmic approaches [6].

The Kolmogorov complexity of a sequence x, represented by $K(x)$, is the size (l) of a shortest program that runs on a universal computer (or Turing machine [7]), prints x and halts. Fixing the machine, we can write

$$K(x) = \min_{p=x} l(p), \tag{1}$$

reading $p = x$ as "p prints x and halts".

In the conditional Kolmogorov complexity of x given y, denoted by $K(x|y)$, there is a second sequence y which is also made available to the program. In this case,

$$K(x|y) = \min_{p(y)=x} l(p), \tag{2}$$

reading $p(y) = x$ as "p, knowing y, prints x and stops". When y is the empty sequence λ, we have $K(x|\lambda) = K(x)$.

The conjoint Kolmogorov complexity of x and y, $K(x, y)$, is the size of a shortest program that prints (x, y) and stops:

$$K(x, y) = \min_{p=(x,y)} l(p). \tag{3}$$

Since the halting problem is undecidable, none of the previous definitions is computable. The chain rule [8]

$$K(x, y) = K(x) + K(y|x), \tag{4}$$

and the symmetry of $K(x, y) = K(y, x)$ led to the definition of the algorithmic mutual information, $I(x : y)$. Notice that we have included equalities assuming an asymptotic nature.

The Kolmogorov complexity has given rise to the information distance introduced by [9], normalized by [10] and algorithmically implemented by [11]. The Kolmogorov complexity has also been extended to the quantum field [12]. Moreover, other measures based on pure algorithmic measures have been proposed and made available online, namely the Online Algorithmic Complexity Calculator (`http://complexitycalculator.com/`) [13, 14].

The need for a computational measure able to assess the complexity of a string given exclusively other has led to a new concept, that of relative algorithmic entropy [15, 16, 11, 17, 18, 19]. We follow this view and propose a notion of algorithmic relative entropy and, further, algorithmic relative similarity. To compute it, we define and use a specific genomic relative compressor, namely because genomic sequences have specific properties and can only be efficiently modelled using specific compressors.

We define the relative algorithmic entropy of x given y, $K(x\|y)$, by the number of bits (or length) of a shortest program, Ξ, that computes x, using exclusively a static model of y, namely

$$K(x\|y) = \min_{\Xi(y)=x} l(\Xi), \tag{5}$$

and then halts.

This means that the program starts by learning y, building a model that represents y, within a certain finite time. Then, the program stops learning, i.e., the model is frozen. The efficiency of the model depends on how well it collects and organizes the information of y, so that questions about x can be quickly answered (without any prior knowledge about x). Every bit that is spent on representing x accounts for the relative algorithmic entropy.

The relative algorithmic entropy is non-computable, mostly because of the halting problem [20], therefore, we have to rely on an approximation, such as that provided by a relative compressor $C(x\|y)$.

Accordingly, we define the Relative Similarity as

$$\mathrm{RS}(x\|y) = |x| \log_2 |\Theta| - C(x\|y) \tag{6}$$

where $|x|$ is the number of symbols in x and Θ the cardinality of the alphabet. Finally, the Normalized Relative Similarity is defined as

$$\mathrm{NRS}(x\|y) = \frac{|x| \log_2 |\Theta| - C(x\|y)}{|x| \log_2 |\Theta|} = 1 - \frac{C(x\|y)}{|x| \log_2 |\Theta|}. \tag{7}$$

Note that, when x is equal to y, the NRS is approximately one and, when x has completely different nature from y, the NRS is approximately zero.

Relative compression bounds

We call $C(x\|y)$ a relative compressor because y and only y is used to represent x. In other words, this compressor cannot use self-similarities that might occur in x. A relative compressor needs to respect the following conditions:

1. $C(x\|x) \approx 0$;
2. $C(x\|\pi) \approx |x|$, where π is a random string;
3. $C(x|y) \leq C(x\|y)$;
4. $C(x^n\|y) \approx nC(x\|y)$, where x^n is the concatenation of all n copies of x;

where the expressions ignore constants (as usual when dealing with algorithmic complexity) that depend only on the underlying Turing machine and that become asymptotically irrelevant as $|x|$ grows.

Relative compressor computation

In order to compute $C(x\|y)$ we need to calculate the probabilities according to

$$C(x\|y) = \sum_{i=1}^{|x|} -\log_2 P(x_i|x_{i-k}^{i-1}, y), \tag{8}$$

where k is the context size of elements from x that can be used in order to search for regularities in y. For $i \leq k$ we assume a uniform distribution.

Both Ziv-Merhav dictionary-based models [15, 21, 18] and Markov models [22, 23, 24, 19] have been successfully used in diverse data type applications. Specifically, for DNA sequences, Markov models proved to have better efficiency [25].

Markov models (MMs), also known as context models, are statistical models. An MM of an information source assigns probability estimates to the symbols of an alphabet, Θ, according to a conditioning context computed over a finite and fixed number, k, of past outcomes (order-k MM) [26]. At element i, these conditioning outcomes are represented by $x_{i-k+1}^{i-1} = x_{i-k+1}, \ldots, x_{i-1}$. A non relative MM can store each outcome of the past in memory, while an MM working in relative mode can only store the outcomes seen in y. The number of conditioning states of a model in DNA sequences is 4^k. The cooperation between MMs of different orders has proven to have better efficiency for representing DNA sequences, instead of competition [27].

High order MM, typically with $k \geq 13$, for $|\Theta| = 4$, proved to be one of the most important models for DNA sequence representation [28], as well as to address other applications [29, 30, 31]. However, when substitutional mutations occur between two identical sequences, high order MM fall short to represent the data. This happens because, if, for example, we use an order-20 MM and we have a probability of one random substitution for each 20 bases, the probability that the same context is seen again is low. The DNA data between close species is frequently of this nature, because they share a common ancestral. Moreover, the majority of the editions in the DNA sequences are of substitutional nature.

Aware of these characteristics, we have recently proposed a preliminary approach to deal with substitutional mutations in DNA sequences [25, 32]. A substitutional tolerant Markov model (STMM), also known as extended finite-context model, is a probabilistic-algorithmic model. It assigns probabilities according to a conditioning context that considers the last symbol, from the sequence to occur, as the most probable symbol, given the occurrences stored in the memory, such as those from y, instead of the true occurring symbol.

For a symbol $s \in \Theta$, the estimator of an STMM, working in relative mode, is given by

$$P(s|x'^{i-1}_{i-k}, y) = \frac{N(s|x'^{i-1}_{i-k}, y) + \alpha}{N(x'^{i-1}_{i-k}, y) + \alpha|\Theta|}, \tag{9}$$

where function N accounts for the memory counts regarding the model and x' is a copy of x, edited according to

$$x'_i = \underset{\forall s \in \Theta}{\operatorname{argmax}}\, P(s|x'^{i-1}_{i-k}, y). \tag{10}$$

For both MMs and STMMs, the parameter α allows balancing between the maximum likelihood estimator and a uniform distribution. For deeper orders, α should be generally lower than one.

When an STMM is cooperating with another model, it can be both probabilistic and algorithmic, since they can be switched on or off given its performance, according to a threshold, t, defined before the computation, that enables or disables the model, according to the number of times that the context has been seen [32].

The relative compressor, $C(x\|y)$, uses a soft blending of both MMs and STMMs [33]. Its computation is performed in two phases: training and estimation. In the first phase, the models accumulate statistical information regarding the y sequence. After processing the entire y sequence, the models are kept frozen and the second phase starts. At this point, the x sequence starts to be compressed using the models computed during the first phase. Here, the probability of the next symbol, x_{i+1}, is given by

$$P(x_{i+1}|y) = \sum_{m \in \mathcal{S}} P(x_{i+1}|x_{i-k}^{i}, y)\; w_{m,i} \tag{11}$$

where k is the order of the model and $P(x_{i+1}|x_{i-k}^{i}, y)$ is the probability assigned to the next symbol by a model of the set, and where $w_{m,i}$ denotes the corresponding weighting factors, with

$$w_{m,i} \propto (w_{m,i-1})^{\gamma} P(x_i|x_{i-k}^{i}, y). \tag{12}$$

The first term acts as a forgetting factor, with $\gamma \in [0, 1)$. The weights are constrained by

$$\sum_{m \in \mathcal{S}} w_{m,i} = 1. \tag{13}$$

The reason why we use this mechanism of cooperation instead of simply selecting the best model is discussed in the next section.

Reason to consider the information for choosing the best model

In genomic data compression, the combination of multiple models can be mainly addressed using two schemes: competition [34, 35, 28] and cooperation (or mixture) [27, 25].

In competition, the models compete to represent a certain number of contig bases, although there is the need to use extra information, called side information [36], to describe which model has been used.

In a mixture, instead of having competition, each model cooperates according to weights attributed. The weights are continuously adapted during compression, depending on the performance of each individual probabilistic model.

Independently of the approach, we have to consider the information for choosing the best model, because the information transits from the model prediction to the side information. To prove this, we can build a compressor using four models, from the class $\mathcal{M}$, as

$$M_1(x_i\|y) = A, \tag{14a}$$

$$M_2(x_i\|y) = C, \tag{14b}$$

$$M_3(x_i\|y) = G, \tag{14c}$$

$$M_4(x_i\|y) = T. \tag{14d}$$

These models give a static outcome and, for any x_i, one of them is correct. Therefore, if we do not consider side information, we have $C(x\|y) \approx 0$. When we consider side information we need to spend bits for describing the model that we used, and, since it is a direct attribution, the side information will contain the same information as to represent $C(x\|y)$. Accordingly, models that do not consider side information, such as simple alignment-based models, are overestimating the normalized relative similarity.

Note 2: Experience setup

In the following sections, we provide a description of the architecture, how to build the database, cleaning and trimming, parameters and computing environment for the inference of metagenomic composition, such as in hominid ancient whole genomes.

Notice that there are other frameworks for a general metagenomic identification purpose, such as those mentioned in the manuscript and, for example, CLARCK [37] (fast and accurate classification of metagenomic and genomic sequences using discriminative k-mers), CSSSCL [38] (a python package that uses combined sequence similarity scores for accurate taxonomic classification of long and short sequence reads), QIIME [39] (analysis of high-throughput community sequencing data). However, these are not suitable for ancient DNA, namely because they rely on 16S or in default parameter aligning methods. The reason why 16S analysis is not suitable, is mainly given by distinguishability problems and ambiguity on similar organisms, mostly because PCR amplification biases can confound standard metabarcoding analyses [40]. Regarding the default aligning parameters, quantifying somewhat dissimilar sequences by alignment methods is problematic, due to the need of fine-tuned thresholds, considering relaxed edit distances and consequent need of very high computational resources [41] (See **Supplementary Note 4**). Aware of this need, the PALEOMIX [42] and MALT [43] methods have been proposed using the BWA [44] and MEGAN [45] alignment algorithms, respectively, with custom parameters for metagenomic aDNA analysis.

Architecture

The pipeline is conducted according to Figure S1, where several databases, namely viral, bacterial, archaeal and fungi are downloaded from the NCBI and combined into a single multi-FASTA file.

For downloading, extracting reads by patterns, getting unique species sequences and building the database, we have used our home-made framework GOOSE (`https://github.com/pratas/goose`). GOOSE is a framework for DNA sequence analysis and manipulation. See subsection “Creating the Database” to get instructions on how to get and build the database.

Cleaning the reads is important, mostly to prevent reads with a short size, sequencing errors and low quality data, that may confuse statistical analysis. Moreover, it is necessary to trim the data. The sequencing of libraries containing molecules shorter than the read length may result in the production of reads that include adaptors or overlapping in paired end reads. The characteristics of aDNA catalyze the concerns on how to efficiently clean and trimming the reads. See Subsection “Cleaning and trimming the reads” to follow our protocol.

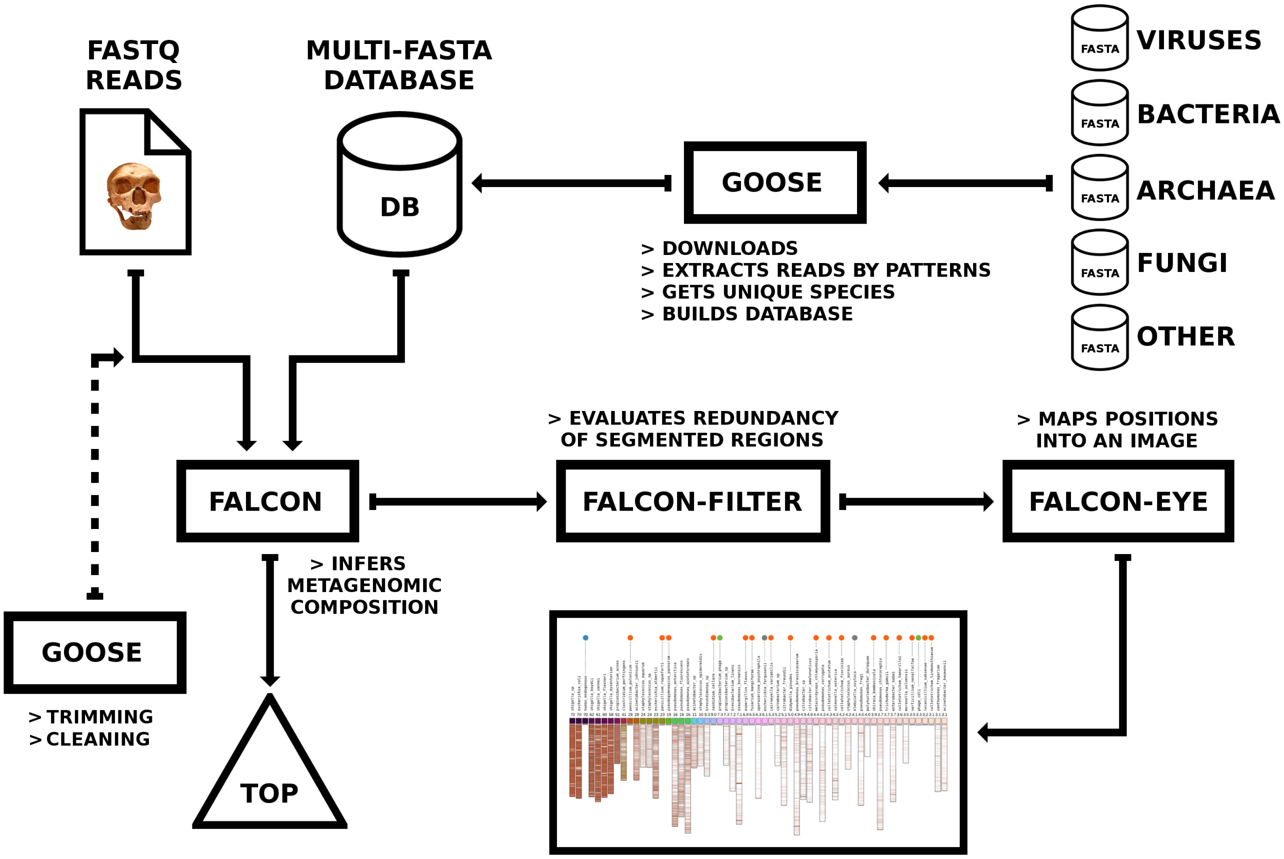

Figure S1: Setup for inference of metagenomic sample composition by FALCON. In the first phase, the database is built using GOOSE (`https://github.com/pratas/goose`), given the databases of several organism types (virus, bacteria, etc). In the second phase, FALCON runs by inferring the metagenomic sample composition. Then, FALCON-FILTER evaluates the similarity of segmented regions (outputting TOP) and, finally, FALCON-EYE outputs the results into an image. Cleaning and trimming is optional, although the authors strongly advise it.

Then, FALCON infers the composition using the FASTQ file containing the ancient genome and the combined database. After, FALCON creates a TOP file containing, sorted from higher to lower similarity, the information (name, size, ID) of the combined database containing higher similarity relatively to the ancient genome. Additionally, FALCON creates a (compressed) file containing the local relative similarity for each sequence that appears on TOP. Then, the FALCON-FILTER filters and segments the local relative similarity of each sequence, given a threshold, and writes the corresponding segmented coordinates in an output file. Finally, FALCON-EYE plots the coordinates, as well as the information from the TOP sequences, in a compact image. See **Supplementary Note 12** for more information on using FALCON.

Creating the Database

For building the database (DB), we have downloaded the entire NCBI database for viruses, bacteria, archaea and fungi, resulting in four datasets of several gigabytes (GB). For each dataset, we have extracted only the sequences labeled as “complete genomes”. The full experiment (install, download, build, compute) can be replicated using the following steps.

First, install FALCON using the following script:

```
#!/bin/bash
git clone https://github.com/pratas/falcon.git
cd falcon/src/
cmake .
make
cp FALCON ../../
cp FALCON-FILTER ../../
cp FALCON-EYE ../../
cd ../../
cp falcon/scripts/*.pl .
```

Then, install GOOSE, using the following script:

```
#!/bin/bash
git clone https://github.com/pratas/goose.git
cd goose/src/
make
cp goose-* ../../
cd ../../
```

Finally, run the following script to dowload and build the database (it will take a while):

```
#!/bin/bash
perl DownloadArchaea.pl
perl DownloadBacteria.pl
perl DownloadFungi.pl
perl DownloadViruses.pl
cat bacteria.fa \
| grep -v -e "ERROR" -e "eFetchResult" -e "DOCTYPE" -e "xml version" -e "Unable to obtain" \
| grep -v -x ">" > bacteria.fna
mv bacteria.fna bacteria.fa
cat viruses.fa bacteria.fa archaea.fa fungi.fa \
| tr ’ ’ ’_’ \
| ./goose-extractreadbypattern complete_genome \
| ./goose-getunique > DB.fa
```

The final output will be a FASTA file, named DB.fa (the database), which contains the reference genomes.

Cleaning and trimming the reads

For cleaning and trimming the reads we use our home-made software GOOSE (`https://github.com/pratas/goose`) (installation guide on previous section). For removing adaptors we use leeHom [46].

The following description provides a guide on how to filter and trim the ancient DNA reads, assuming that there is a Paired-End FASTQ file with name Org.fq.

Since the file is Paired-End there is the need to split it into two files, according to the strand direction (forward → Org.fw.fq and reverse → Org.rv.fq):

```
./goose-FastqSplit Org.fw.fq Org.rv.fq < Org.fq
```

If the files are not trimmed, it is possible to trim them using the respective sequencing adaptors [47] and leeHom [46] using the following:

```
./leeHom -f AGATCGGAAGAGCACACGTCTGAACTCCAG \
         -s GGAAGAGCGTCGTGTAGGGAAAGAGTGTAG \
         --ancientdna \
         -fq1 Org.fw.fq \
         -fq2 Org.rv.fq \
         -fqo out.fq
```

The output of leeHom merges the file, therefore, to continue there is the need to split the file again as previously done (use out.fq instead of Org.fq).

Then, for trimming by quality-scores, namely average quality-scores below 15 in a window of 5, to filter reads shorter than 35 bases and to merge back the files, proceed as:

```
# Forward:
cat Org.fw.fq \
| ./goose-FastqMinimumLocalQualityScoreForward -k 5 -w 15 -m 33 \
| ./goose-FastqMinimumReadSize 35 > Org.fw.filt.fq;
# Reverse:
cat Org.rv.fq \
| ./goose-FastqMinimumLocalQualityScoreReverse -k 5 -w 15 -m 33 \
| ./goose-FastqMinimumReadSize 35 > Org.rv.filt.fq;
# Merge:
cat Org.fw.filt.fq Org.rv.filt.fq > Org.filt.fq;
```

Finally, for excluding reads with more than a certain number of "N", use:

```
./goose-FastqExcludeN 5 < Org.filt.fq > final.fq
```

In this case we have removed reads with more than 5 "N".

For more details of sample preparation in ancient DNA see [48]. For another tool to trim Illumina NGS data (not built specifically to ancient DNA) see [49].

Parameters and computing environment

The amount of RAM required from computing hominid complete genomes is approximately 26 GB, while for smaller genomes, namely mitogenomes, it is substantially decreased (0.5 GB). For lower RAM use "`-c`" with lower values, such as "`-c 100`". Here, much less computational resources are used at the expense of a small payoff in precision. The general parameters used are: "`-v -n 8 -t 1000 -F -Z -m 20:200:1:5/20 -c 200`". The machine used 8 threads for the computation.

All experiments were carried out on an Ubuntu Linux v16.04 LTS, with gcc v5.3.1, 4 Intel Core i7-6700K 3.4 GHz CPUs (with hyperthreading) and 32 GB of RAM. [1].

[1] To replicate all the experiments, make sure the machine has 1.5 TB of free disk and 32 GB of RAM

Note 3: Evaluation on synthetic and ancient data

Similarity estimation methods (MUMMER, GREEN, FALCON)

The Figures corresponding to the evaluation of several methods on synthetic data (Fig. 1 of the paper) can be replicated using the following scripts:

- `https://raw.githubusercontent.com/pratas/falcon/master/simulation/runSubs.sh`;
- `https://raw.githubusercontent.com/pratas/falcon/master/simulation/runPerm.sh`;
- `https://raw.githubusercontent.com/pratas/falcon/master/simulation/runTime.sh`.

The data has been simulated using XS [50] and properly managed with GOOSE (`http://pratas.github.io/goose/`). For a comparison of tools for the simulation of genomic next-generation sequencing data, see [51]. We have included GREEN [52] for a control, namely because GREEN is also a relative compressor. We have estimated exhaustively its parameters for a better performance: "`-i -f 10 -k 15`". However, since GREEN is not prepared to handle large genomes (very high RAM for the case), we have only used it in these small synthetic sequences.

Comparison of CPU times between FALCON, BOWTIE and BWA

The following experience describes the comparison of CPU times between FALCON and two well known alignment/mapping methods (BOWTIE [53] and BWA [44]), in the case of mapping reads from the Neanderthal, according to a *Escherichia coli* reference. BOWTIE and BWA used two approaches: standard and estimated ancient parameters (AP). The latter is for efficiently dealing with ancestral data (BOWTIE: "`-v 2`", BWA: "`-L 16500 -N 0.01 -O 2`"). FALCON used the parameters "`-v -n 8 -t 400 -F -Z -m 20:200:1:5/20 -c 200`"). The following scripts are able to replicate the comparison and generate the plots:

- `https://raw.githubusercontent.com/pratas/falcon/master/ancient/runRCTimes.sh`;
- `https://raw.githubusercontent.com/pratas/falcon/master/ancient/runRCTimesPlots.sh`.

Note that we have used the complete Neanderthal FASTQ reads (631 GB), after filtering them according to **Supplementary Note 1**.

Note 4: Upper bounds on the Normalized Relative Similarity

In this note, we give upper bounds on the Normalized Relative Similarity (NRS) for two cases: substitutional mutated and block-permutated strings.

Mutated strings

Consider a string, x, on an alphabet, Θ, and denote by $|x|$ the length of x, i.e., the number of symbols of Θ in x. The notation $|\Theta|$ is also used to indicate the size of Θ. Logarithms are base two.

Consider a process that, uniformely and independently, randomly mutates the symbols of x with probability p. We denote the mutated string by x_p, with $|x_p| = |x|$.

We intend to give an upper bound on the Normalized Relative Similarity (NRS) between x and x_p, defined as

$$\mathrm{NRS}(x_p \| x) = 1 - \frac{C(x_p \| x)}{|x_p| \log |\Theta|},$$

where $C(x_p \| x)$ denotes the number of bits that the best possible compressor would require to represent x_p using exclusively information from x. A possible representation of x_p based on x can be viewed as a two-part code, where the first part indicates where the mutations occurred, encoded in a binary string l, and the second part lists the new symbols at the mutated positions. To describe l, on average we need $|l|H(p)$ bits, where

$$H(p) = -p \log p - (1-p) \log(1-p).$$

On the other hand, to describe the list of mutated symbols, on average we need $p|x| \log(|\Theta| - 1)$ bits. Therefore, most often,[2] and noting that $|l| = |x|$, we have

$$C(x_p \| x) \geq |x|(H(p) + p \log(|\Theta| - 1).$$

We may now establish an upper bound on the NRS,

$$\mathrm{NRS}(x_p \| x) \leq 1 - \frac{H(p) + p \log(|\Theta| - 1)}{\log |\Theta|}, \tag{15}$$

[2] There are sequences l that can be algorithmically represented in less than $k = |l|H(p)$ bits, i.e., those for which $K(l) < k$, where $K(l)$ denotes the Kolmogorov complexity of l. However, this happens only for a very tiny fraction of them. In fact, it can be shown that there are less than 2^k of those strings. Because the total number of possible length-n strings having k ones is $\binom{n}{k} \geq \left(\frac{n}{k}\right)^k$, then the fraction of those low-complexity strings is upper-bounded by $\left(\frac{2k}{n}\right)^k$. For example, for $n = 1000$ and $k = 100$ this fraction is less than 2×10^{-70}. Similar arguments also apply to the list of changed symbols.

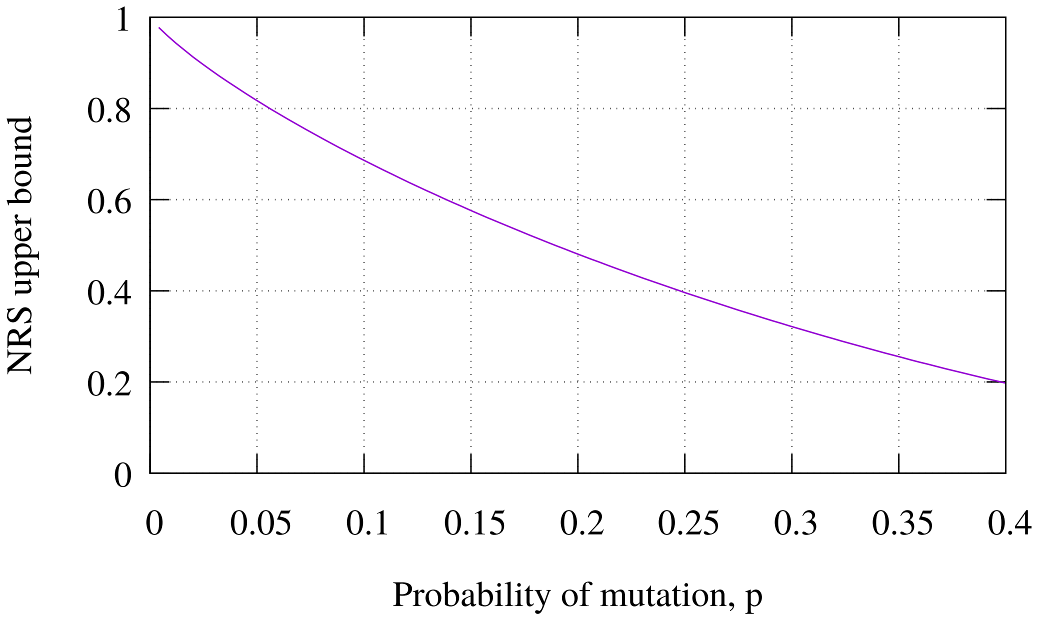

Figure S2: Curve of the NRS upper bound, for $|\Theta| = 4$.

Figure S2 shows the value of the NRS upper bound defined in (15), for $|\Theta| = 4$.

Block-permutated strings

Consider now a partition of x into blocks of b consecutive symbols. For simplicity, we assume that the partition is sequential, starting from the left-hand side of the string and ending with a block possibly with less than b symbols. Therefore, there will be $\lceil |x|/b \rceil$ blocks in the partition. Let us represent by x_b the string built using a certain permutation of the blocks resulting from the partition of x just described. Note that $|x_b| = |x|$.

We intend to give an upper bound on the Normalized Relative Similarity (NRS) between x and x_b, defined as

$$\mathrm{NRS}(x_b \| x) = 1 - \frac{C(x_b \| x)}{|x_b| \log |\Theta|},$$

where $C(x_b \| x)$ denotes the number of bits that the best possible compressor would require to represent x_b using exclusively information from x. Note that, in general, $C(x_b \| x)$ requires at least to produce bits that represent a permutation of $n = \lceil |x|/b \rceil$ integers. Hence,

$$C(x_b \| x) \geq \log n!.$$

Using the well-known Stirling's approximation,

$$\log n! = n \log n - n \log e + \mathcal{O}(\log n),$$

we have

$$C(x_b \| x) \geq \left\lceil \frac{|x|}{b} \right\rceil \left(\log \left\lceil \frac{|x|}{b} \right\rceil - \log e \right) \geq \frac{|x|}{b} \left(\log \frac{|x|}{b} - \log e \right).$$

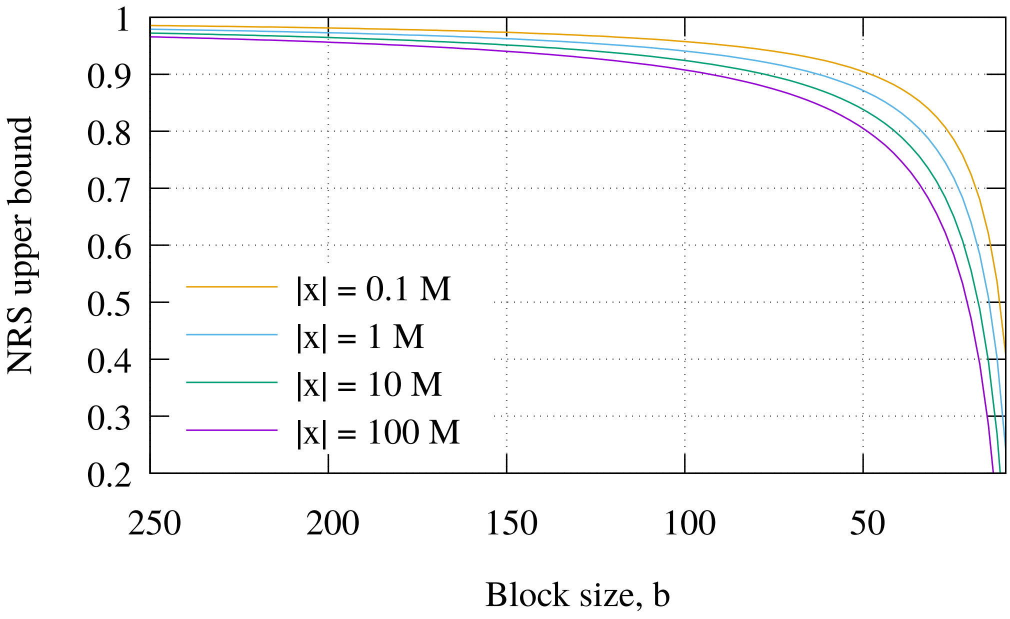

Figure S3: Curves of the NRS upper bound, for several values of $|x|$ and for $|\Theta| = 4$.

We may now establish an upper bound on the NRS,

$$\mathrm{NRS}(x_b \| x) \leq 1 - \frac{\frac{|x|}{b}\left(\log\frac{|x|}{b} - \log e\right)}{|x_b| \log |\Theta|},$$

and, since $|x_b| = |x|$,

$$\mathrm{NRS}(x_b \| x) \leq 1 - \frac{\log\frac{|x|}{b} - \log e}{b \log |\Theta|}. \tag{16}$$

Figure S3 shows the value of the NRS upper bound defined in (16), for several values of $|x|$ and for $|\Theta| = 4$.

Note 5: Database intra-similarity

Rough assumptions regarding the presence of organisms in a metagenomic composition must be avoided without proper analysis, mainly because the database contains intra-similarities that can interfere with the interpretation For this purpose, we use the tool GULL (`https://github.com/pratas/GULL`), which is able to measure the similarity between genomes contained in the database, more precisely, those genomes that have been previously identified as a composition by the FALCON tool.

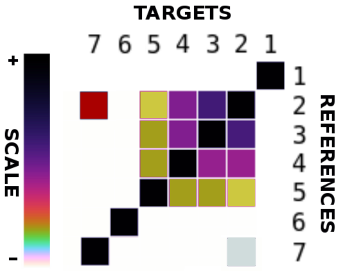

Figure S4: An hypothetical example of the GULL output showing the Normalized Relative Similarity (NRS) between seven sequences, where numbers stand for identification labels.

This analysis should be made to identify organisms that have high Normalized Relative Similarity (NRS) values according to other organisms (from the database), rather than being actually on the samples. For example, bacteria *Escherichia coli* and *Salmonella* spp. share high similarity [54], therefore, if an *Escherichia coli* is identified, it is very likely that a *Salmonella* is also identified. This doesn't mean the *Salmonella* is in the sample, but that it will appear by cross-similarity. Thus, using GULL as a complement to the FALCON tool is a major improvement for filtering misleading results. Additionally, we estimate the similarity within the database the same way as in FALCON, using the Normalized Relative Similarity (NRS), providing a way to compare final overall results.

Figure S4 shows an example of the GULL output after measuring the NRS between seven sequences. The numbers stand for identification labels (from 1 to 7). As shown, we are able to identify a cluster (from 2 to 5) indicating that these species share high similarity. Accordingly, the presence of sequence 3, 4 and 5 might be justified by cross-similarity, rather than being present in the samples.

The matrix is not symmetric (a particularity of the NRS). This enables, for example, to identify a high similarity when using reference sequence 2 and, as target sequence 7. These non-symmetric high similarities reveal a very important feature: a sequence that is very similar to some part or parts of a bigger sequence (a sequence containing a shorter sequence) that can indicate the presence of, for example, retroviruses.

Note 6: Composition analysis of the human, chimpanzee and gorilla reference genomes

In order to have a control while using FALCON, we have used several reference genomes and the same database, as described in **Supplementary Note 2**, to compute the metagenomic sample composition. The reference genomes are:

- human (GRC and CHM assemblies),
- chimpanzee,
- gorilla.

We have also included the respective non-assembled sequences (unplaced and unlocalized sequences).

As it can be seen in Figure S5 and S6, the sequences with the highest similarity detected are the human endogenous retroviruses. Retrovirus are known to be one of the major factors in chromosomal rearrangements [55]. The degrees of similarity observed seem to be in accordance to the split between human, chimp and gorila ancestors [56, 57, 58, 59].

Regarding other sequences with some degree of similarity, with the exception of the baboon endogenous retrovirus and RD114 retrovirus, the number of sequences is low and with low values reporting similarity. Once more, this shows that FALCON does not overestimate, reporting only consistent results. For the sequences found with lower levels of similarity, the cause is probably given by the non-assembled sequences (unplaced and unlocalized) that have also been considered.

Relatively to the experience setup, the Human GRC and CHM top results, with the characteristics mentioned in Parameters and computing environment (**Supplementary Note 2**), have been computed in 46 and 42 minutes, respectively. On the other hand, the Chimpanzee and Gorilla top results, with the characteristics mentioned in Parameters and computing environment (**Supplementary Note 2**), have been computed in 47 and 43 minutes, respectively.

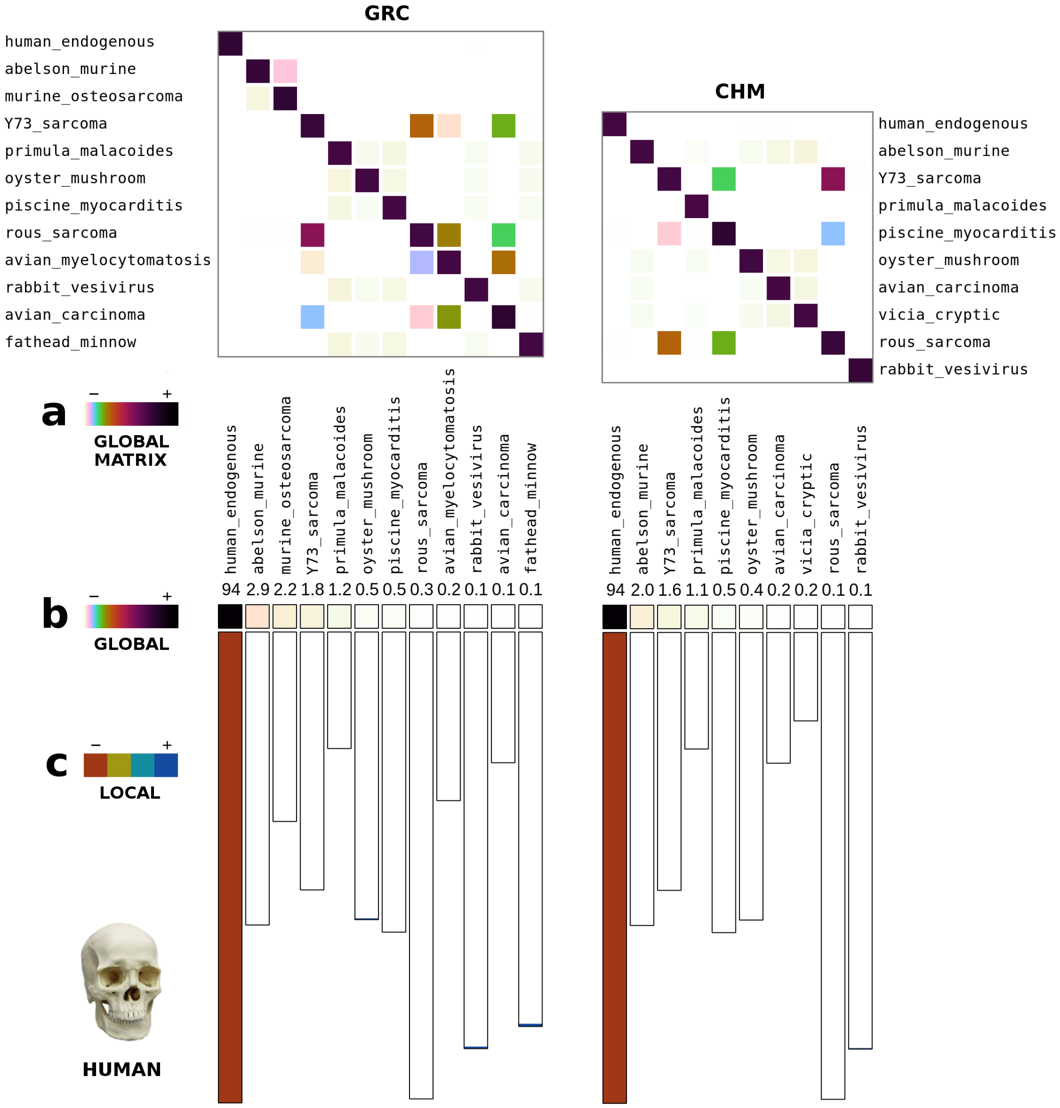

Figure S5: Metagenomic sample composition of Human reference genome GRCv38 and CHMv1.1 inferred by FALCON. (a) Relative similarity between the complete genomes of the database that shared some degree of similarity with the Human references. (b) Relative similarity between Human whole genome, using the full genome (including non-assembled non-localized and mtDNA sequences), and multiple NCBI databases. (c) Local similarity of filtered regions with relative similarity above 0.5. The Figure can be replicated using the script: `https://raw.githubusercontent.com/pratas/falcon/master/human/runHumanGRC.sh` for GRC and `https://raw.githubusercontent.com/pratas/falcon/master/human/runHumanCHM.sh` for CHM reference assemblies.

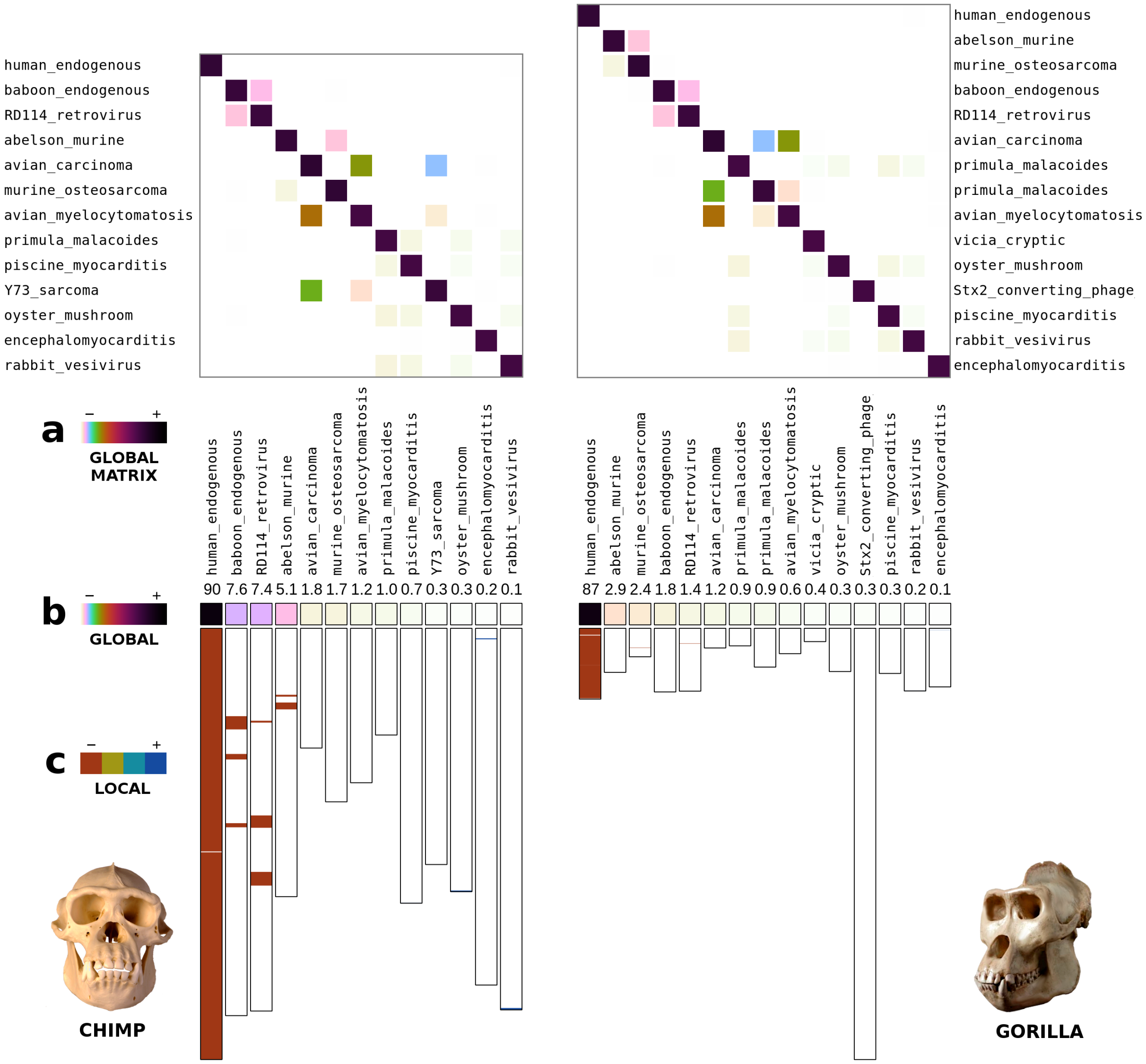

Figure S6: Metagenomic sample composition of Chimpanzee (v3.0) and Gorilla (v3.1) reference genomes inferred by FALCON. (a) Relative similarity between the complete genomes of the database that shared some degree of similarity with the chimpanzee and gorilla references. (b) Relative similarity between chimpanzee and gorilla whole genomes, using the full genome (including, when available, non-assembled non-localized and mtDNA sequences), and multiple NCBI databases. (c) Local similarity of filtered regions with relative similarity above 0.5. The Figure can be replicated using the script: `https://raw.githubusercontent.com/pratas/falcon/master/chimp/runChimp.sh` for chimpanzee and `https://raw.githubusercontent.com/pratas/falcon/master/gorilla/runGorilla.sh` for gorilla.

Note 7: Composition analysis of the Neanderthal

We have inferred the metagenomic sample composition of a Neanderthal [60] using only the unmapped reads. These reads supposly contain data where the PE did not overlaped, low quality data and exogeneuos DNA. The data was already trimmed and cleaned. We have converted the data from BAM to FASTQ, using SAMtools [61],

```
./samtools bam2fq Neanderthal.bam > Neanderthal.fq
```

and from FASTQ to FASTA using GOOSE (`https://github.com/pratas/goose`). The reads have been downloaded from: `http://cdna.eva.mpg.de/neandertal/altai/AltaiNeandertal/bam/unmapped_qualfail/`, while the database has been built according to **Supplementary Note 2** (Creating the Database). For a script to run the full experience, see `https://raw.githubusercontent.com/pratas/falcon/master/ancient/runNeanderthal.sh`.

The Neanderthal results, with the characteristics mentioned in parameters and computing environment (**Supplementary Note 2**), have been computed in 22 hours (desktop computer without SSD). The Neanderthal computation used half a TB (approximately 486 GB) of data.

NRS rankings

Tables S1, S2, S3 and S4 display the information of the genomes with similarity relatively to the Neanderthal, namely the ID, percentage (%) of similarity, genome size, GI (genome identifier) and name.

ID	Size	% Sim	GI	Name
1	4558287	70.615	1008930592	Shigella sp. PAMC 28760
2	4668621	70.577	1069460419	Escherichia coli strain 210221272
3	9472	70.085	548558394	Human endogenous retrovirus K113 complete genome
4	4574246	62.303	844762407	Shigella boydii strain ATCC 9210
5	4878853	61.327	992379426	Shigella sonnei strain FDAARGOS 90
6	4607196	60.882	344915202	Shigella flexneri 2a str. 301 chromosome
7	4369232	58.307	82775382	Shigella dysenteriae Sd197 chromosome
8	2469997	51.879	983475690	Propionibacterium acnes strain PA 12 1 R1
9	3530409	41.531	1002985342	Clostridium perfringens strain JP838
10	28192	29.810	1030034275	Penicillium polonicum mitochondrion
11	3509795	28.036	972005651	Acinetobacter johnsonii XBB1

Table S1: Information of the genomes with similarity relatively to the Neanderthal (1-11).

ID	Size	% Sim	GI	Name
12	2693398	24.097	1043351086	Staphylococcus equorum strain KM1031
13	2721344	24.034	1039416963	Staphylococcus sp. AntiMn-1
14	4657167	23.242	985817611	Escherichia albertii DNA
15	29908	23.162	849123577	Penicillium roqueforti strain UASWS P1 mitochondrion
16	26918	19.665	849123696	Pseudogymnoascus pannorum strain NN050741 mitochondrion
17	6441449	16.943	1033149375	Pseudomonas antarctica strain PAMC 27494
18	5841722	16.904	1071444414	Pseudomonas fluorescens strain Pt14
19	6859618	16.459	1003709406	Pseudomonas azotoformans strain S4
20	2738313	11.641	1056253829	Acinetobacter sp. NCu2D-2
21	2616530	10.183	57865352	Staphylococcus epidermidis RP62A
22	3263775	9.298	1054792290	Brevundimonas sp. GW460-12-10-14-LB2
23	23743	9.046	1037241235	Zasmidium cellare mitochondrion
24	29347	7.346	1070106186	Propionibacterium phage Enoki
25	2571505	7.311	1061378163	Propionibacterium sp. oral taxon 193 strain F0672
26	4209935	7.157	1068600609	Brevibacterium linens strain SMQ-1335
27	6301761	7.084	1033049601	Pseudomonas koreensis strain D26
28	31602	6.777	813559934	Aspergillus flavus strain TCM2014 mitochondrion
29	30585	6.548	992331403	Fusarium mangiferae mitochondrion
30	4674191	6.323	1055571898	Sporosarcina psychrophila strain DSM 6497
31	55150	6.144	218561636	Escherichia fergusonii ATCC 35469 plasmid pEFER
32	27186	5.284	1011058795	Cairneyella variabilis isolate VPRI 42388 mitochondrion
33	2605518	5.238	971851441	Carnobacterium sp. CP1
34	5096586	5.136	1061913597	Citrobacter freundii strain SL151
35	55973	4.983	1002162993	Didymella pinodes strain 165/T mitochondrion
36	6976764	4.948	983452107	Pseudomonas brassicacearum strain LBUM300
37	4727434	4.902	992381540	Citrobacter sp. FDAARGOS 156
38	4904610	4.901	992391073	Citrobacter amalonaticus strain FDAARGOS 122
39	25615	4.447	558603437	Metacordyceps chlamydosporia strain 170 mitochondrion
40	6124177	4.324	1059918478	Pseudomonas corrugata strain RM1-1-4
41	30892	4.261	836643807	Colletotrichum acutatum strain KC05 mitochondrion
42	4755700	4.254	62178570	Salmonella enterica subsp. enterica serovar Choleraesuis str. SC-B67
43	30020	4.207	1025806322	Colletotrichum fioriniae mitochondrion
44	2821452	4.171	387149188	Staphylococcus aureus 04-02981
45	3897	4.119	1033027607	Klebsiella oxytoca plasmid pKO JKo3 4 DNA
46	5101809	3.983	984941068	Pseudomonas fragi strain P121
47	2026292	3.952	809277835	Polynucleobacter duraquae strain MWH-MoK4
48	39030	3.878	813423937	Shiraia bambusicola mitochondrion
49	6646309	3.742	798809582	Pseudomonas chlororaphis strain PCL1606
50	29303	3.715	1031917273	Trichoderma gamsii strain KUC1747 mitochondrion
51	4880257	3.655	1068614800	Enterobacter kobei strain DSM 13645
52	30823	3.616	1011056737	Colletotrichum tamarilloi mitochondrion
53	2434688	3.496	1011424644	Moraxella osloensis strain CCUG 350
54	26139	3.490	992330782	Verticillium nonalfalfae mitochondrion
55	47021	3.265	148609382	Phage cdtI DNA
56	25919	3.229	948299288	Lecanicillium saksenae mitochondrion
57	36957	3.119	589144606	Colletotrichum lindemuthianum mitochondrion
58	4232635	3.086	1057114188	Xanthomonas fragariae isolate Fap21

Table S2: Information of the genomes with similarity relatively to the Neanderthal (12-58).

ID	Size	% Sim	GI	Name
59	4178958	3.054	1068582773	Acinetobacter baumannii DU202
60	4724316	3.040	1068589113	Enterobacter hormaechei subsp. oharae strain DSM 16687
61	4661849	2.880	1068585475	Enterobacter xiangfangensis strain LMG27195
62	4551186	2.794	1062553183	Enterobacter sp. HK169
63	4182028	2.783	1059928630	Pantoea agglomerans strain C410P1
64	24565	2.712	1016472806	Acremonium fuci strain 3a34 culture-collection VKM:F-4473 mitochondrion
65	2146611	2.710	289166909	Streptococcus mitis B6
66	2583617	2.684	992332033	Staphylococcus saprophyticus strain FDAARGOS 168
67	2220	2.684	1063656990	Common bean-associated gemycircularvirus isolate 53 2
68	2622031	2.635	1057408040	Turicibacter sp. H121
69	26821	2.612	1043378333	Cladophialophora bantiana mitochondrion
70	103844	2.572	817530248	Fusarium culmorum strain CBS 139512 mitochondrion
71	4748414	2.560	1068593796	Enterobacter cloacae complex ’Hoffmann cluster IV’ strain DSM 16690
72	5280349	2.449	336246508	Enterobacter aerogenes KCTC 2190 chromosome
73	4803917	2.377	976150947	Leclercia adecarboxylata strain USDA-ARS-USMARC-60222
74	4616371	2.312	1032822074	Lelliottia amnigena strain ZB04
75	22376	2.275	761546238	Acremonium implicatum mitochondrion
76	3372992	2.236	1011513961	Limnohabitans sp. 63ED37-2
77	36554	2.232	992330294	Colletotrichum lupini strain CBS 119142 culture-collection CBS:119142 mitochondrion
78	5002	2.180	23334621	Casphalia extranea densovirus
79	2159490	2.172	145588189	Polynucleobacter asymbioticus QLW-P1DMWA-1
80	4902106	2.068	1043354620	Kosakonia sacchari strain BO-1
81	6181873	2.027	1002825811	Pseudomonas putida KT2440 chromosome
82	4902027	1.988	1045495283	Enterobacter sacchari SP1
83	42895	1.958	1037241508	Ilyonectria destructans isolate 2007/P/476 mitochondrion
84	2549338	1.930	1016088174	Staphylococcus haemolyticus strain S167
85	6555569	1.814	1069642234	Pseudomonas syringae pv. actinidiae ICMP 18884
86	4322479	1.786	1011381500	Arthrobacter alpinus strain ERGS4:06
87	22240	1.769	1070102990	Lactococcus phage M5938
88	6397126	1.748	28867243	[Pseudomonas syringae] pv. tomato str. DC3000 chromosome
89	2191044	1.730	984925192	Streptococcus salivarius strain JF
90	53439	1.730	885000563	Diaporthe longicolla mitochondrion
91	5006	1.619	601447520	Danaus plexippus plexippus iteravirus isolate Granby
92	4702949	1.403	1011934043	Klebsiella sp. G5
93	1818293	1.395	970549833	Streptococcus infantarius strain ICDDRB-NRC-S5
94	5398272	1.352	1051485383	Raoultella ornithinolytica strain A14
95	3862530	1.310	375133618	Acinetobacter pittii PHEA-2 chromosome
96	5303342	1.280	1035732180	Kosakonia oryzae strain Ola 51
97	2199877	1.145	984937715	Aerococcus viridans strain CCUG4311
98	5312744	1.129	1063296287	Klebsiella pneumoniae strain UCLAOXA232KP Pt0
99	2251	1.083	807743884	Mongoose feces-associated gemycircularvirus d strain 478d
100	32745	1.067	90995396	Candida parapsilosis mitochondrion
101	5521203	1.019	1051750068	Klebsiella variicola strain DSM 15968
102	2729848	1.013	1060094849	Propionibacterium avidum strain DPC 6544
103	4294639	0.946	974636588	Cronobacter malonaticus LMG 23826
104	2236	0.945	608055536	Dragonfly-associated circular virus 2 isolate FL2-5X-2010
105	3614992	0.917	257067223	Brachybacterium faecium DSM 4810 chromosome

Table S3: Information of the genomes with similarity relatively to the Neanderthal (59-105).

ID	Size	% Sim	GI	Name
106	5314161	0.855	1008930624	Klebsiella quasipneumoniae strain ATCC 700603
107	2173607	0.778	1049349156	Neisseria meningitidis strain M22772
108	3332539	0.764	1015863798	Psychrobacter alimentarius strain PAMC 27889
109	5178466	0.737	78045556	Xanthomonas campestris pv. vesicatoria complete genome
110	2013339	0.649	984923562	Aerococcus urinaeequi strain CCUG28094
111	3472056	0.522	1011536582	Planococcus kocurii strain ATCC 43650
112	39842	0.460	1025806304	Lecanora strobilina mitochondrion
113	2292716	0.446	985585133	Rothia mucilaginosa DNA
114	41950	0.413	549717265	Xanthomonas fuscans subsp. fuscans str. 4834-R
115	4313264	0.390	973251686	Cedecea neteri strain ND02
116	5011796	0.352	1004716966	Obesumbacterium proteus strain DSM 2777
117	7835	0.333	9626692	Encephalomyocarditis virus
118	4574859	0.324	1060098565	Acidovorax sp. RAC01
119	3305371	0.291	1045836558	Planococcus donghaensis strain DSM 22276
120	59866	0.258	939482425	Stx1 converting phage DNA
121	3424893	0.238	1045837372	Planococcus halocryophilus strain DSM 24743
122	3782009	0.228	1045851098	Planococcus antarcticus DSM 14505
123	60238	0.204	116221992	Stx2-converting phage 86
124	38306	0.196	1043378898	Beauveria caledonica isolate fhr1 mitochondrion
125	62706	0.186	939482426	Stx2 converting phage II DNA
126	4605545	0.126	378764971	Pantoea ananatis LMG 5342 main chromosome complete genome
127	4674684	0.118	1059968446	Hydrogenophaga sp. RAC07
128	6038	0.107	998745912	Piscine myocarditis-like virus isolate Golden shiner/PMCLV/USA/MN/2014
129	4542863	0.092	1025881008	Hafnia alvei strain HUMV-5920
130	5188	0.017	108737103	Y73 sarcoma virus
131	4714359	0.011	982915660	Pseudomonas monteilii strain USDA-ARS-USMARC-56711
132	2802354	0.004	1011690330	Psychrobacter urativorans strain R10.10B

Table S4: Information of the genomes with similarity relatively to the Neanderthal (106-132).

Database intra-similarity

Figure S7 displays, in a compact map, the intra-similarity of the database genomes having similarity relatively to the Neanderthal. The GIs (genome identifiers) can be linked with the names from Tables S1, S2, S3 and S4. See **Supplementary Note 5** for more information about database intra-similarity.

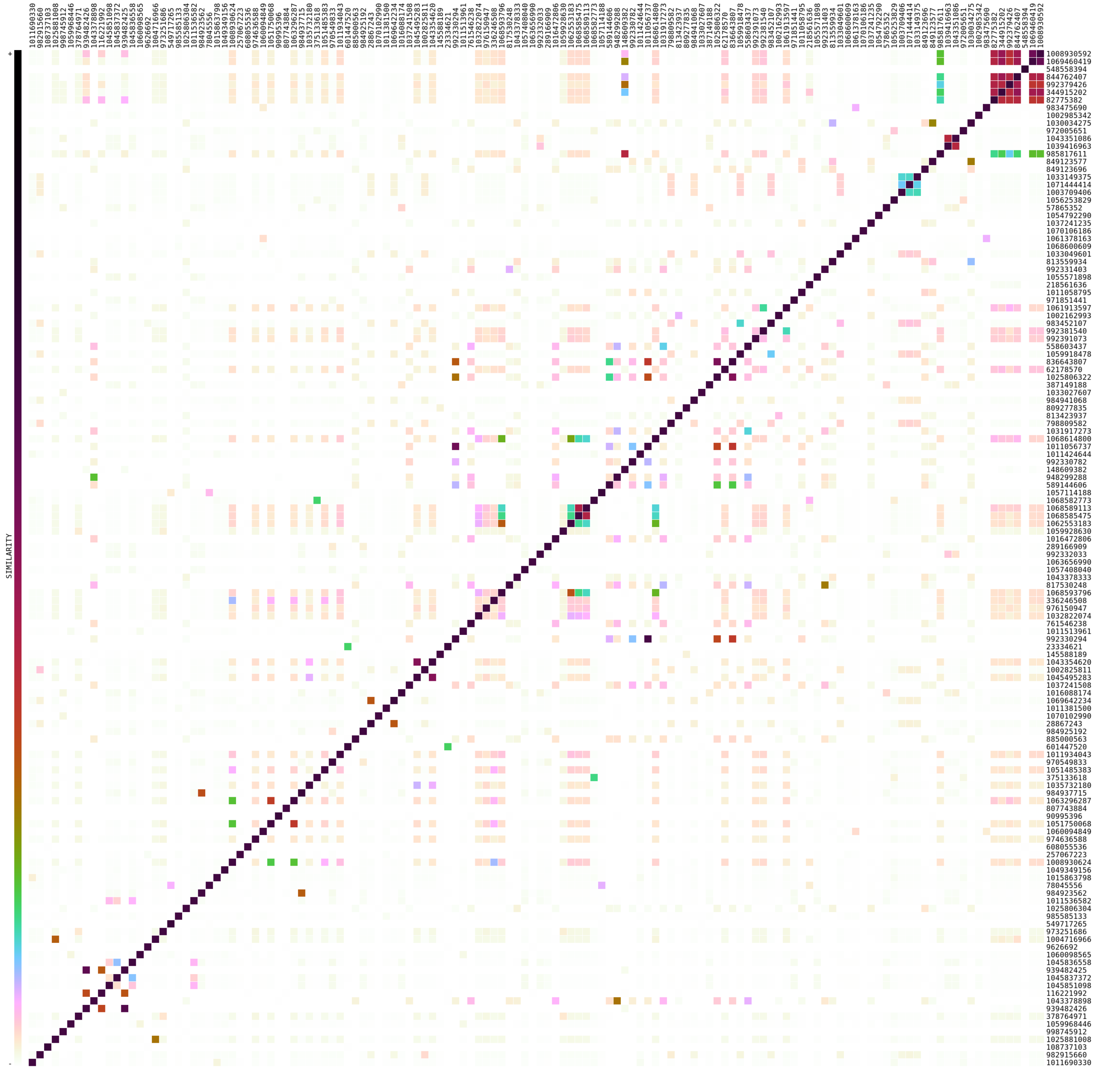

Figure S7: Relative similarity between the complete genomes that shared some degree of similarity with the Neanderthal raw data (ordered by similarity). Numbers stand for GIs. The map has been computed with GULL tool (`https://github.com/pratas/GULL`).

Note 8: Composition analysis of the Denisova

We have inferred the metagenomic sample composition of a Denisova whole genome [62]. We have downloaded the data from `http://cdna.eva.mpg.de/denisova/raw_reads/`, while the database has been built according to **Supplementary Note 2** (Creating the Database). For a script to run the full experiment, see: `https://raw.githubusercontent.com/pratas/falcon/master/ancient/runDenisova.sh`.

The Denisova top results, with the characteristics mentioned in parameters and computing environment of **Supplementary Note 2**, have been computed in 42 hours (desktop computer without SSD). The computation used more than 1 TB (approximately 925 GB) of reading data.

Figure S8 depicts the metagenomic composition result of the FASTQ reads from the Denisova in a compact map.

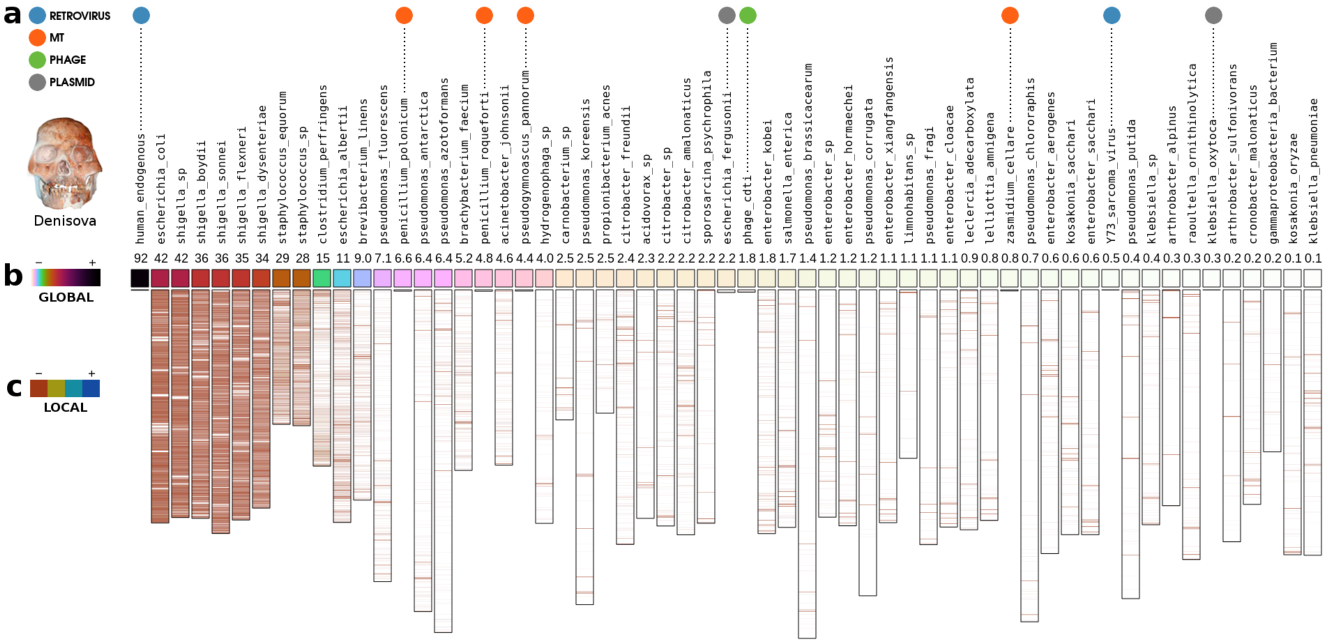

Figure S8: Metagenomic sample composition of a Denisova raw data infered by FALCON. (**a**) Classification of the complete genomes, where the absense of circle stands for bacterial genomes. (**b**) Relative similarity between a Denisova whole genome, using the FASTQ samples, and multiple NCBI databases, where relative similarities below 0.12 have been discarded from the figure. (**c**) Local similarity of filtered regions with relative similarity above 0.5.

NRS rankings

Tables S5 and S6 display the information of the genomes with similarity relatively to the Denisova. All tables are constituted by the ID, percentage (%) of similarity, size of the genome, GI (genome identifier) and name.

ID	Size	% Sim	GI	Name
1	9472	92.597	548558394	Human endogenous retrovirus K113
2	4668621	42.184	1069460419	Escherichia coli strain 210221272
3	4558287	42.016	1008930592	Shigella sp. PAMC 28760
4	4574246	36.661	844762407	Shigella boydii strain ATCC 9210
5	4878853	36.531	992379426	Shigella sonnei strain FDAARGOS 90
6	4607196	35.654	344915202	Shigella flexneri 2a str. 301 chromosome
7	4369232	34.104	82775382	Shigella dysenteriae Sd197 chromosome
8	2693398	29.206	1043351086	Staphylococcus equorum strain KM1031
9	2721344	28.931	1039416963	Staphylococcus sp. AntiMn-1
10	3530409	15.052	1002985342	Clostridium perfringens strain JP838
11	4657167	11.586	985817611	Escherichia albertii DNA
12	4209935	8.981	1068600609	Brevibacterium linens strain SMQ-1335
13	5841722	7.058	1071444414	Pseudomonas fluorescens strain Pt14
14	28192	6.565	1030034275	Penicillium polonicum mitochondrion
15	6441449	6.433	1033149375	Pseudomonas antarctica strain PAMC 27494
16	6859618	6.370	1003709406	Pseudomonas azotoformans strain S4
17	3614992	5.174	257067223	Brachybacterium faecium DSM 4810 chromosome
18	29908	4.797	849123577	Penicillium roqueforti strain UASWS P1 mitochondrion
19	3509795	4.584	972005651	Acinetobacter johnsonii XBB1
20	26918	4.391	849123696	Pseudogymnoascus pannorum strain NN050741 mitochondrion
21	4674684	3.956	1059968446	Hydrogenophaga sp. RAC07
22	2605518	2.525	971851441	Carnobacterium sp. CP1
23	6301761	2.473	1033049601	Pseudomonas koreensis strain D26
24	2469997	2.461	983475690	Propionibacterium acnes strain PA 12 1 R1
25	5096586	2.424	1061913597	Citrobacter freundii strain SL151
26	4574859	2.319	1060098565	Acidovorax sp. RAC01
27	4727434	2.209	992381540	Citrobacter sp. FDAARGOS 156
28	4904610	2.184	992391073	Citrobacter amalonaticus strain FDAARGOS 122
29	4674191	2.175	1055571898	Sporosarcina psychrophila strain DSM 6497
30	55150	2.153	218561636	Escherichia fergusonii ATCC 35469 plasmid pEFER
31	47021	1.849	148609382	Phage cdtI DNA
32	4880257	1.794	1068614800	Enterobacter kobei strain DSM 13645
33	4755700	1.736	62178570	Salmonella enterica subsp. enterica serovar Choleraesuis str. SC-B67
34	6976764	1.394	983452107	Pseudomonas brassicacearum strain LBUM300
35	4551186	1.246	1062553183	Enterobacter sp. HK169
36	4724316	1.203	1068589113	Enterobacter hormaechei subsp. oharae strain DSM 16687
37	6124177	1.187	1059918478	Pseudomonas corrugata strain RM1-1-4
38	4661849	1.142	1068585475	Enterobacter xiangfangensis strain LMG27195
39	3372992	1.111	1011513961	Limnohabitans sp. 63ED37-2
40	5101809	1.080	984941068	Pseudomonas fragi strain P121

Table S5: Information of the genomes with similarity relatively to the Denisova (1-40).

ID	Size	% Sim	GI	Name
41	4748414	1.061	1068593796	Enterobacter cloacae complex 'Hoffmann cluster IV' strain DSM 16690
42	4803917	0.950	976150947	Leclercia adecarboxylata strain USDA-ARS-USMARC-60222
43	4616371	0.847	1032822074	Lelliottia amnigena strain ZB04
44	23743	0.800	1037241235	Zasmidium cellare mitochondrion
45	6646309	0.678	798809582	Pseudomonas chlororaphis strain PCL1606
46	5280349	0.637	336246508	Enterobacter aerogenes KCTC 2190 chromosome
47	4902106	0.635	1043354620	Kosakonia sacchari strain BO-1
48	4902027	0.600	1045495283	Enterobacter sacchari SP1
49	5188	0.525	108737103	Y73 sarcoma virus
50	6181873	0.443	1002825811	Pseudomonas putida KT2440 chromosome
51	4702949	0.358	1011934043	Klebsiella sp. G5
52	4322479	0.285	1011381500	Arthrobacter alpinus strain ERGS4:06
53	5398272	0.260	1051485383	Raoultella ornithinolytica strain A14
54	3897	0.257	1033027607	Klebsiella oxytoca plasmid pKO JKo3 4 DNA
55	5043757	0.240	971864227	Arthrobacter sulfonivorans strain Ar51
56	4294639	0.217	974636588	Cronobacter malonaticus LMG 23826
57	3243537	0.196	1057510000	Gammaproteobacteria bacterium TR3.2
58	5303342	0.141	1035732180	Kosakonia oryzae strain Ola 51
59	5312744	0.135	1063296287	Klebsiella pneumoniae strain UCLAOXA232KP Pt0
60	6570200	0.106	1043091695	Rhodococcus sp. 008
61	4916630	0.089	1004049116	Ramlibacter tataouinensis strain 5-10
62	6555569	0.082	1069642234	Pseudomonas syringae pv. actinidiae ICMP 18884
63	5521203	0.074	1051750068	Klebsiella variicola strain DSM 15968
64	6236862	0.045	972362081	Rhodococcus erythropolis R138
65	6397126	0.017	28867243	[Pseudomonas syringae] pv. tomato str. DC3000 chromosome
66	5314161	0.015	1008930624	Klebsiella quasipneumoniae strain ATCC 700603
67	2583617	0.014	992332033	Staphylococcus saprophyticus strain FDAARGOS 168

Table S6: Information of the genomes with similarity relatively to the Denisova (41-67).

Database intra-similarity

Figure S9 displays, in a compact map, the inter-similarity of the database genomes having top similarity relatively to the Denisova. The GIs (genome identifiers) can be linked with the names from Supplementary Tables S5 and S6. See **Supplementary Note 5** for more information about database intra-similarity.

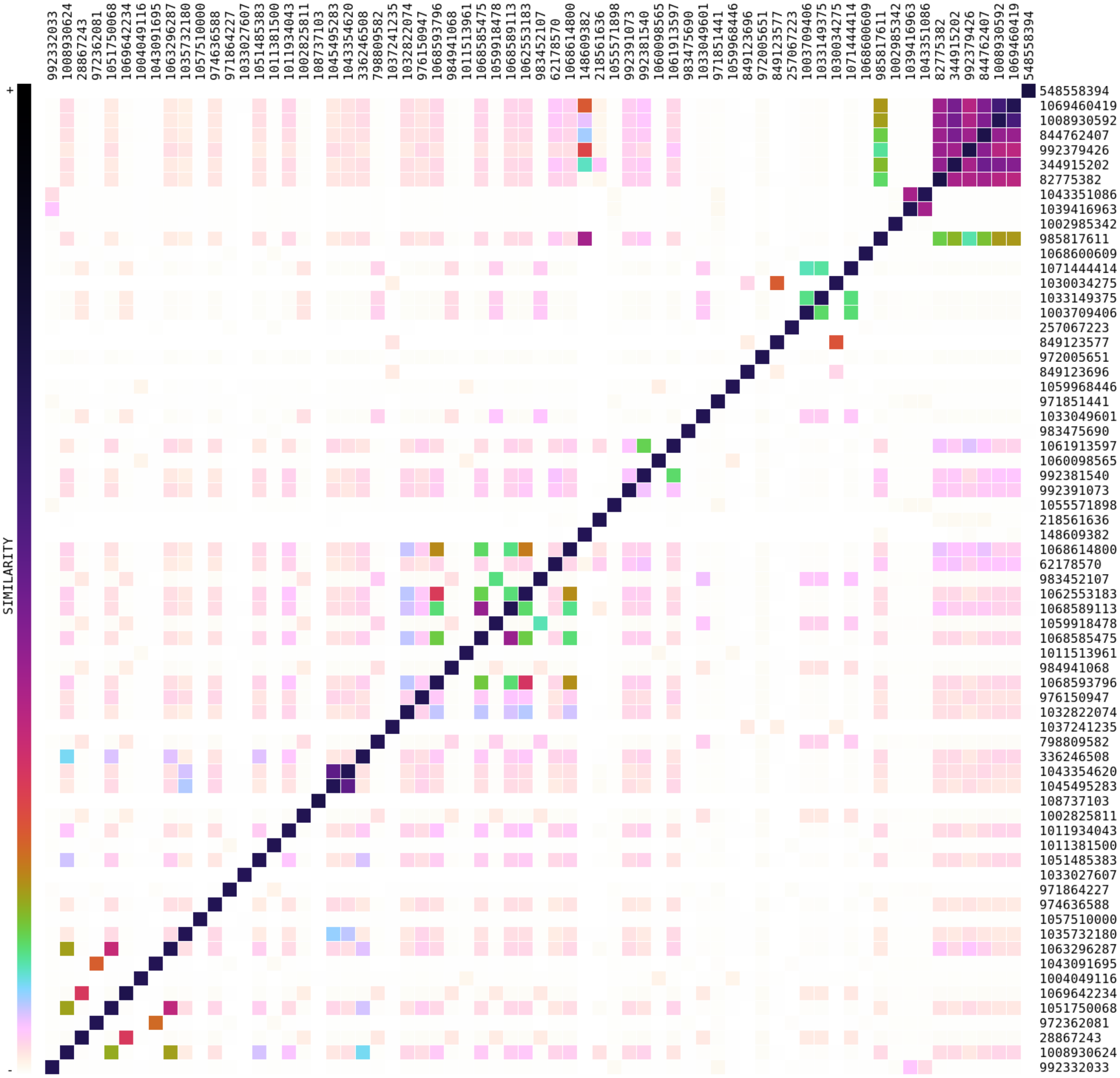

Figure S9: Relative similarity between the complete genomes that shared some degree of similarity with the Denisova raw data (ordered by similarity). Numbers stand for GIs. The map has been computed with GULL tool (`https://github.com/pratas/GULL`).

Note 9: Composition analysis of the Mammoth mitogenome

In this note, we report the inferred metagenomic sample composition of a Columbian mammoth (*Mammuthus columbi*) mitogenome [63], using FALCON. The mammoth inhabited North America from the northern United States to Costa Rica during the Pleistocene epoch. This genome, with approximately 11,000 years old, has been sequenced using NGS and analyzed in [64], yielding major progress in the phylogenetic positions of the species.

For the inference, we have used the process described in **Supplementary Note 2**. The mammoth top results, with the hardware characteristics mentioned in Parameters and computing environment, using the parameters "`-v -n 8 -t 1000 -F -Z -m 13:1:0:0/0 -m 20:100:1:5/10 -c 150`", has been computed using approximately 48 minutes.

The complete Columbian mammoth mitogenome [63], including all Illumina sequences in FASTQ format, has been downloaded from:

- `ftp://ftp.cbcb.umd.edu/pub/data/mammoth/c_2_sequence12.txt.gz`.

We have also downloaded the entire NCBI database for viruses, bacteria, archaea and fungi, resulting in four datasets with several GB. For each dataset, we have extracted only the sequences labeled as "complete genomes". The database was built according to **Supplementary Note 2** (Creating the Database).

The full experience (install, download, build, compute) can be replicated using the following script:

- `https://raw.githubusercontent.com/pratas/falcon/master/ancient/runMammoth.sh`.

Figure S10 depicts the metagenomic sample composition of the mammoth mitogenome, inferred by FALCON, and the corresponding relative similarity between the complete genomes that shared some degree of similarity with the mammoth (database intra-similarity).

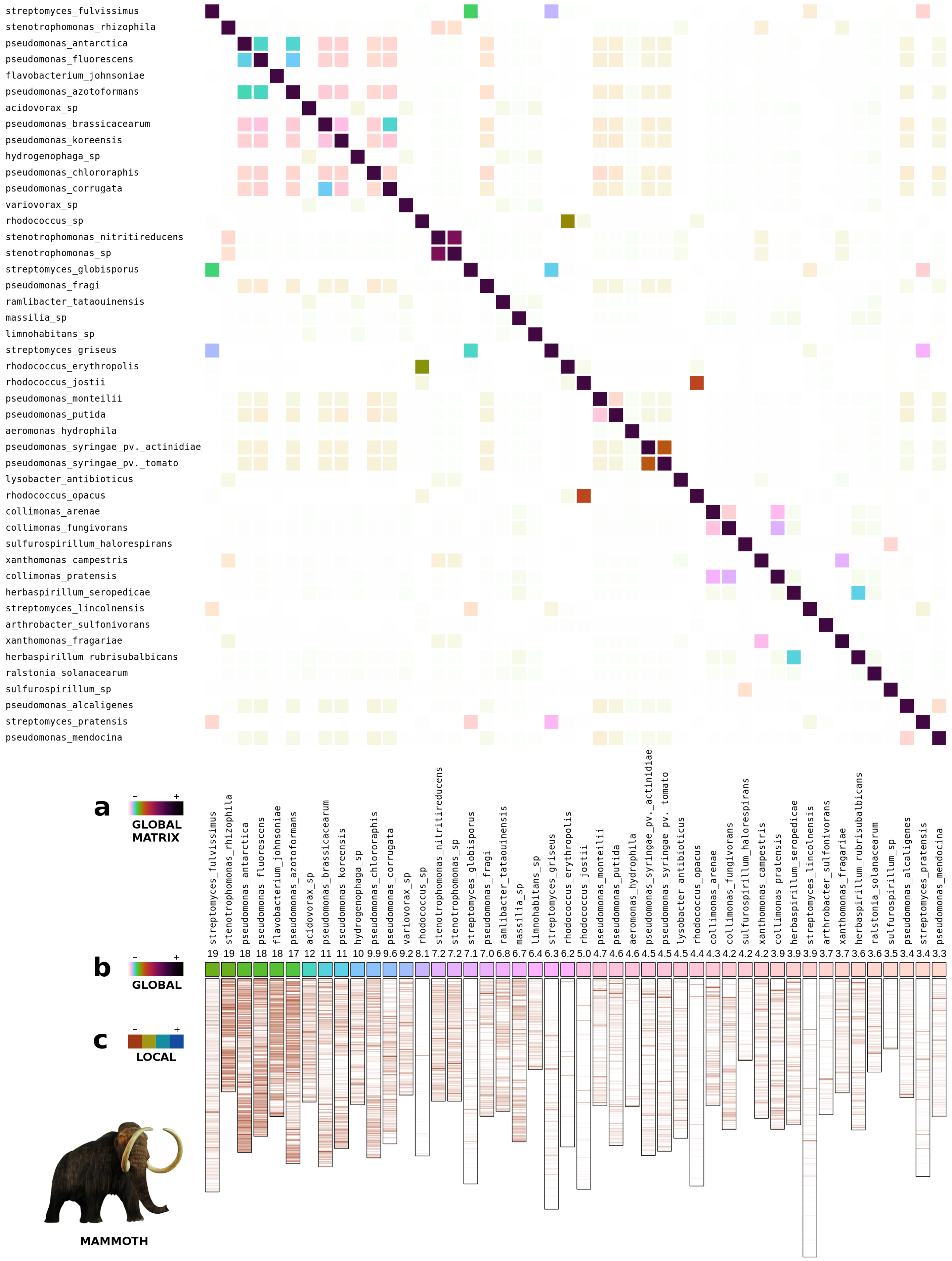

Figure S10: Metagenomic sample composition of mammoth mitogenome (All Illumina sequences in FASTQ format) inferred by FALCON. (a) Relative similarity between the complete genomes that shared some degree of similarity with the mammoth. (b) Relative similarity between mammoth whole genome and multiple NCBI databases. (c) Local similarity of filtered regions with relative similarity above 0.5. The Figure can be replicated using the script: `https://raw.githubusercontent.com/pratas/falcon/master/ancient/runMammoth.sh`.

Note 10: Local relative similarity

As an extension to **Supplementary Note 1**, the local relative similarity, $\mathrm{LRS}(x_i \| y)$, defines the relative similarity, $\mathrm{RS}(x\|y)$, according to

$$\mathrm{RS}(x\|y) = \sum_{i=1}^{|x|} \mathrm{LRS}(x_i\|y) = \sum_{i=1}^{|x|} \log_2 |\Theta| - C(x_i\|y). \quad (17)$$

Therefore, if we plot each value of the local relative similarity, $\mathrm{LRS}(x_i\|y)$, for each base we have a description of how similar it seems relatively to y. Moreover, we are able to segment and classify regions [24].

Figure S11 shows the process of creating the local relative similarity maps. These maps identify regions in sequence x that share similarity with regions of y below a certain threshold T. Then, the filtered regions are compressed (reference-free) in order to evaluate their self-similarity. This enables to destinguish repetitive from complex regions.

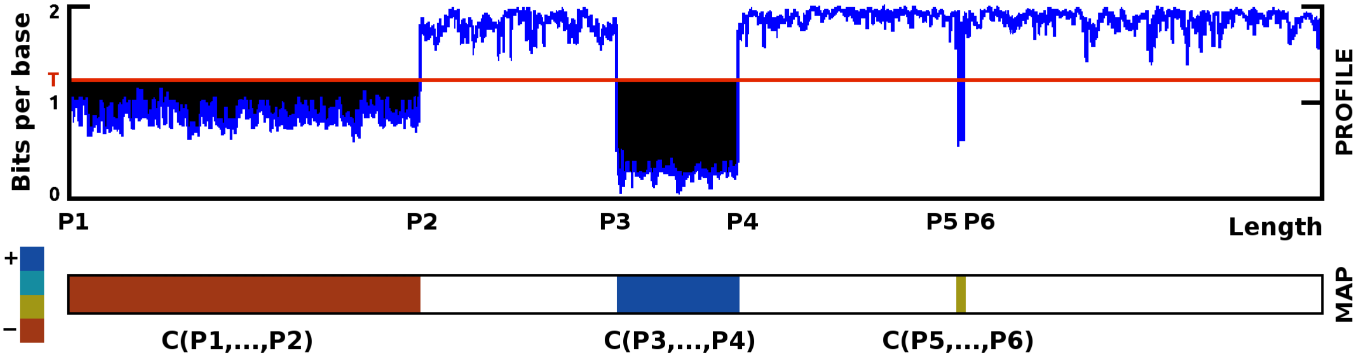

Figure S11: From local complexity profiles to local relative similarity maps. The complexity profile is segmented using a threshold T. The segmented regions are individually compressed (without using prior information) and classifyed according to their local self-similarity [65]. The colors of the regions in the maps are according to the color-scale at bottom left. As an example, $C(P1, ..., P2)$ stands for the (reference-free) compression of the region starting in position $P1$ to $P2$.

In this note we provide some examples of mapped local similarity, namely using *Penicillium polonicum* and *Human adenovirus*, and its correlation with the respective NCBI annotated data.

Penicillium polonicum

In this example we mapped the regions of the *Penicillium polonicum* mitochondrial genome sequence having local similarity relatively to the Neanderthal (N) and Denisova (D). Figure S12 provides a map with those regions and its correlation with the NCBI annotated data.

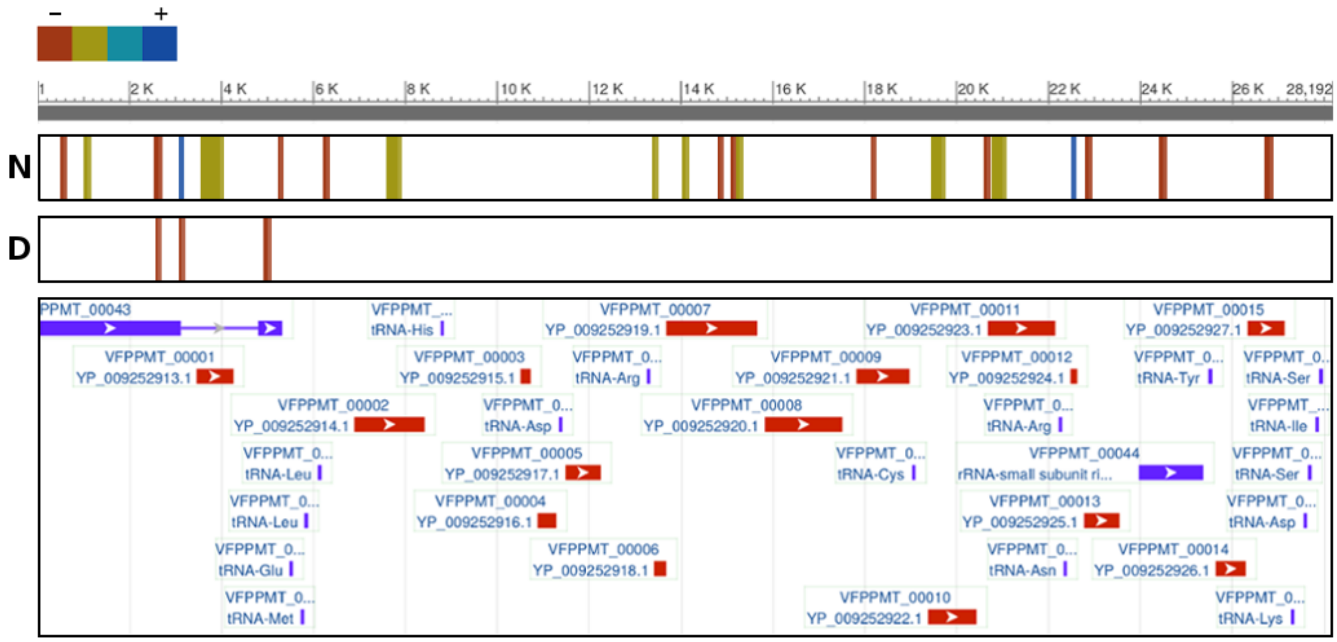

Figure S12: Local relative similarity of the *Penicillium polonicum* mitochondrial genome compared to a Neanderthal (N) and a Denisova (D) samples. The local similarities have been segmented above 0.5 and evaluated, for each region, given its self-similarity (top left legend expresses the intervals self-similarity, where '-' stands for less). Bottom map depicts the NCBI map aligned with the above maps.

Human adenovirus

In this example, we used FALCON with more complex models, namely adding an extra model of depth 13 and increasing the deepest model cache-hash to "`c=250`". Although it needs more computational resources it also provides more inferred genomes. One of them, relatively to the Denisova, is the *Human adenovirus* C, having relative similarity around 7%. The *Human adenovirus* C is a dsDNA virus that that is usually associated with respiratory tract infections in humans.

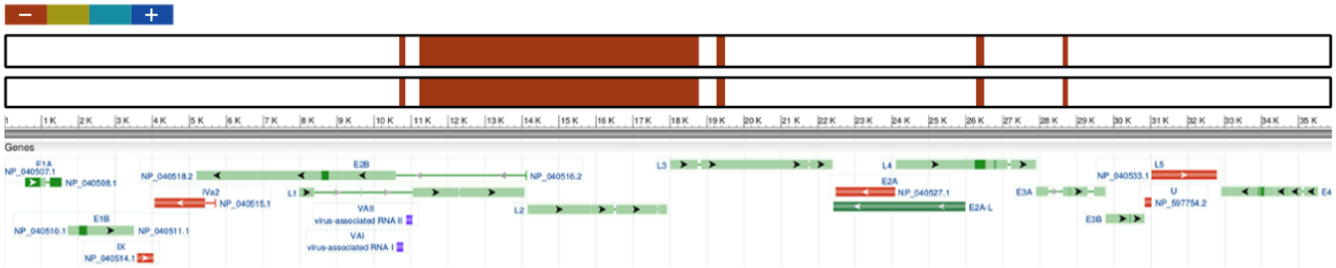

Figure S13: Local relative similarity of *Human adenovirus* C (upper map) and *Human adenovirus* 2 (lower map) viral sequences relatively to a Denisova genome. Colors indicate the interval sef-similarity of each regions after being filtered and segmented with a threshold of 1.5. Blue color (+) indicates self-similarity greather than 1.95, while marine blue, green and red (-), respectively 1.6, 1.25, 0. The bottom map depicts the aligned visual information of gene content of the *Human adenovirus* C provided by the respective NCBI query.

Specifically, in Figure S13 we have mapped the local similarity of the *Human adenovirus* C relatively to the Denisova and we have found, besides several regions of similarity distributed along the viral sequence, strong similarity concentrated in three genes of the adenovirus, namely L1 (2652993), L2 (265996) and the beginning of L3 (2652998).

The Human Adenovirus C cause roughly 5% of symptomatic upper respiratory tract and 15% of lower respiratory tract infections in children younger than 5 years [66]. Moreover, species C viruses display prolonged fecal excretion for months, and even years, after the virus is no longer detected in nasopharyngeal washings [67].

Therefore, the identification of a virus similar to the Human Adenovirus C might indicate that the Denisova individual was a young child. However, since the virus may persist for some years, the Denisova may had a few more years. This genetic insights are according to the Denisova bone analysis. Specifically, the bone growth plates and size led to a prediction of an age between 6 and 13.5 years [68].

Note 11: Ancient pathogens authentication

The first aDNA sequences started to appear in the mid-1990s, mainly using mitochondrial DNA (mtDNA) from a Neanderthal bone [69]. After several years of advances in sequencing technology and in anti-contamination protocols, the first full high coverage ancient genomes of a Neanderthal [60] and a Denisova [62] were made available. Other ancient genomes have also been sequenced, such as Tyrolean's iceman [70], Egyptian mummies [71], Columbian mammoth [63], among others. Moreover, new projects addressing this matter are currently on the way and many more are expected in a near future [72].

DNA extracted from ancient samples may contain information beyond genomic data from the individual [73]. In fact, it has also been shown to harbor preserved signatures of bacteria or archaea, possibly hosted before the ancient individual's death [74, 75, 76, 40].

After the identification of the metagenomic composition by FALCON, there is the need to classify each genome regarding age, for proper authentication of the ancient ones. Most of the genomes come from present-day's contamination, while just a few come from ancient-days. Next, we describe a procedure to authenticate ancient DNA.

Authentication procedure

Ancient genomes reveal several specific characteristics, namely DNA pattern damages [77, 78]. These patterns are: short sequence length, an excess of cytosine to thymine (C-to-T) misincorporations at 5' ends of sequences, complementary guanine to adenine (G-to-A) misincorporations at 3'-termini and an excess of purines at the genomic coordinate located just before the sequencing starts, indicative of post-mortem depurination, followed by strand fragmentation [77]. Therefore, when the sequences from each genome reveal some of these characteristics, the genome is authenticated as ancient [79].

For this purpose, we need to identify which genomes have patterns of damage. We follow the pipeline provided in Figure S14, based on the steps of inference, aligning and classification of pattern damage.

For the measurement and classification of pattern damage there are already available tools, namely mapDamage [79, 80] and PMDTools [81]. However, they rely on prior alignments.

The aligning of the filtered reads against the reference micro genome may be performed using BOWTIE [53] or BWA [44]. However, for aligning efficiently, the parameters should be properly set to ancient data [41], namely turning off the seeding option, allowing more substitutions and gaps, and setting relaxed thresholds (BWA example: "`-I 0 -O 2 -N 0.02 -L 1024 -E 7`").

Assume that, with FALCON, we have found an *Escherichia coli* (EC) with high similarity relatively

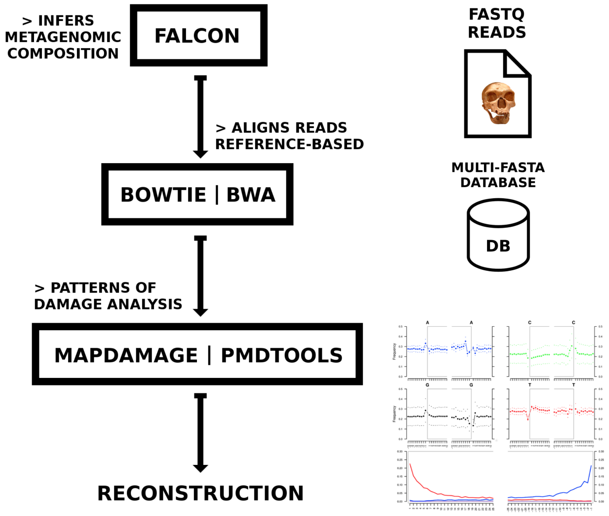
> INFERS METAGENOMIC COMPOSITION
FALCON
FASTQ READS
> ALIGNS READS REFERENCE-BASED
MULTI-FASTA DATABASE
BOWTIE | BWA
DB
> PATTERNS OF DAMAGE ANALYSIS
MAPDAMAGE | PMDTOOLS
RECONSTRUCTION

Figure S14: A pipeline for ancient pathogen authentication and reconstruction. The program BOWTIE and BWA enables to map the reads with similarity relatively to a refrence, while the MAPDAMAGE and PMDTOOLS enables to measure the pattern damage of the mapped reads.

to the ancient reads (AR). For aligning the reads with BWA, we use:

```
bwa index EC.fa
bwa mem -t 8 -I 0 -O 2 -N 0.02 -L 1024 -E 7 EC.fa AR.fq > F-PA-OUT.sam
samtools view -Shb F-EC-OUT.sam > F-EC-OUT.bam
samtools view -b -F4 F-EC-OUT.bam > FIL-EC.bam
```

After, the aligned reads are analysed according to the reference micro genome for finding pattern damages using, for example, MAPDAMAGE [79] or PMDTOOLS [81]. Only the ones having characteristics of pattern damage are authenticated as ancient, and further reconstructed [82]. In the case of using MAPDAMAGE [79], after aligning the reads, we run:

```
mapDamage -i FIL-EC.bam -r EC.fa
```

A few new microbial genomes have already been reconstructed, such as an archaeal genome of the *Methanobrevibacter oralis Neanderthalensis* [40]. The reconstruction of the ancient microbial genomes is also a very complex computational challenge. For this purpose, some alignment-based methods have been proposed [83, 84].

Ancient Pseudomonas authentication

From the composition analysis of the Mammoth mitogenome (Figure S10), we used the first two *Pseudomonas* to classify them regarding age using MAPDAMAGE. The two *Pseudomonas* with the highest NRS are *P. antarctica* PAMC 27494 [85] and *P. fluorescens* F113 [86] having genomes with about 6.5 MB of size. The *P. antarctica* has been collected in Antarctica: Barton Peninsula, King George Island, in fresh water (2014.01.19). Its GC content is about 59.8%. The Pseudomonas fluorescens F113 has been isolated from the sugar-beet rhizosphere in Ireland [87]. Its GC content is about 60.8%.

The commands used in the authentication of the *P. antarctica*, were

```
bwa index PA.fa
bwa mem -t 8 -I 0 -O 2 -N 0.02 -L 1024 -E 7 PA.fa c_2_sequence12.txt > F-PA-OUT.sam
samtools view -Shb F-PA-OUT.sam > F-PA-OUT.bam
samtools view -b -F4 F-PA-OUT.bam > FIL-PA.bam
mapDamage -i FIL-PA.bam -r PA.fa;
```

while for *P. fluorescens*, were

```
bwa index PF.fa
bwa mem -t 8 -I 0 -O 2 -N 0.02 -L 1024 -E 7 PF.fa c_2_sequence12.txt > F-PF-OUT.sam
samtools view -Shb F-PF-OUT.sam > F-PF-OUT.bam
samtools view -b -F4 F-PF-OUT.bam > FIL-PF.bam
mapDamage -i FIL-PF.bam -r PF.fa;
```

The output of MAPDAMAGE for *P. antarctica* is depicted is in Figure S15, while for *P. fluorescens* in Figure S16. As it can be seen, both patterns of damage are very similar, mostly because the *P. fluorescens* and *P. antarctica* share 40.4314 % of similarity, according to the NCBI. In both maps, the frequency of the substitutions (C-to-T and A-to-G) increase in the tips of the reads, showing moderate levels of pattern damage.

There are two reasons for a moderate frequency of damage. The first one is given by the high presence of GC content, while the second by the lack of similarity between the reference sequence with the ancient *Pseudomonas*. From Figure S10-b, we are able to see that, although the ancient reads match the references in almost 20 %, it has a distribution without being concentrated in a single or in a small number of regions (Figure S10-c). This means that the alignment methods are not so efficient as FALCON while mapping similarity, and, therefore, some of the reads might not have been mapped efficiently, partially influencing the authentication.

Dismissal of correlation between mtDNA and bacterial DNA

The works of Lynn Margulis in the 1960s, attempted to explain the origins of eukaryotic cell organelles such as mitochondria and chloroplasts. Other authors supported the theory, mostly describing the evolution of the mitochondria [88, 89].

In order to dismiss correlation between the mtDNA of the Mammoth and bacterial DNA of the *P. antarctica*, we compress them relatively to each other. Moreover, we added the mtDNA of the African savanna elephant (as a control). Any relative similarity between the Mammoth mtDNA and the bacterial DNA may show a false positive ancient authentication. To assess this possibility, we have manually downloaded the assembled version of the Mammoth mtDNA from `https://www.ncbi.nlm.nih.gov/nuccore/333236151?report=fasta`. Then, we have renamed the file to Mammoth.fa.

For estimation of the relative similarity, we used the state-of-the-art DNA compressor (GeCo [25]). To install GeCo, we ran the following:

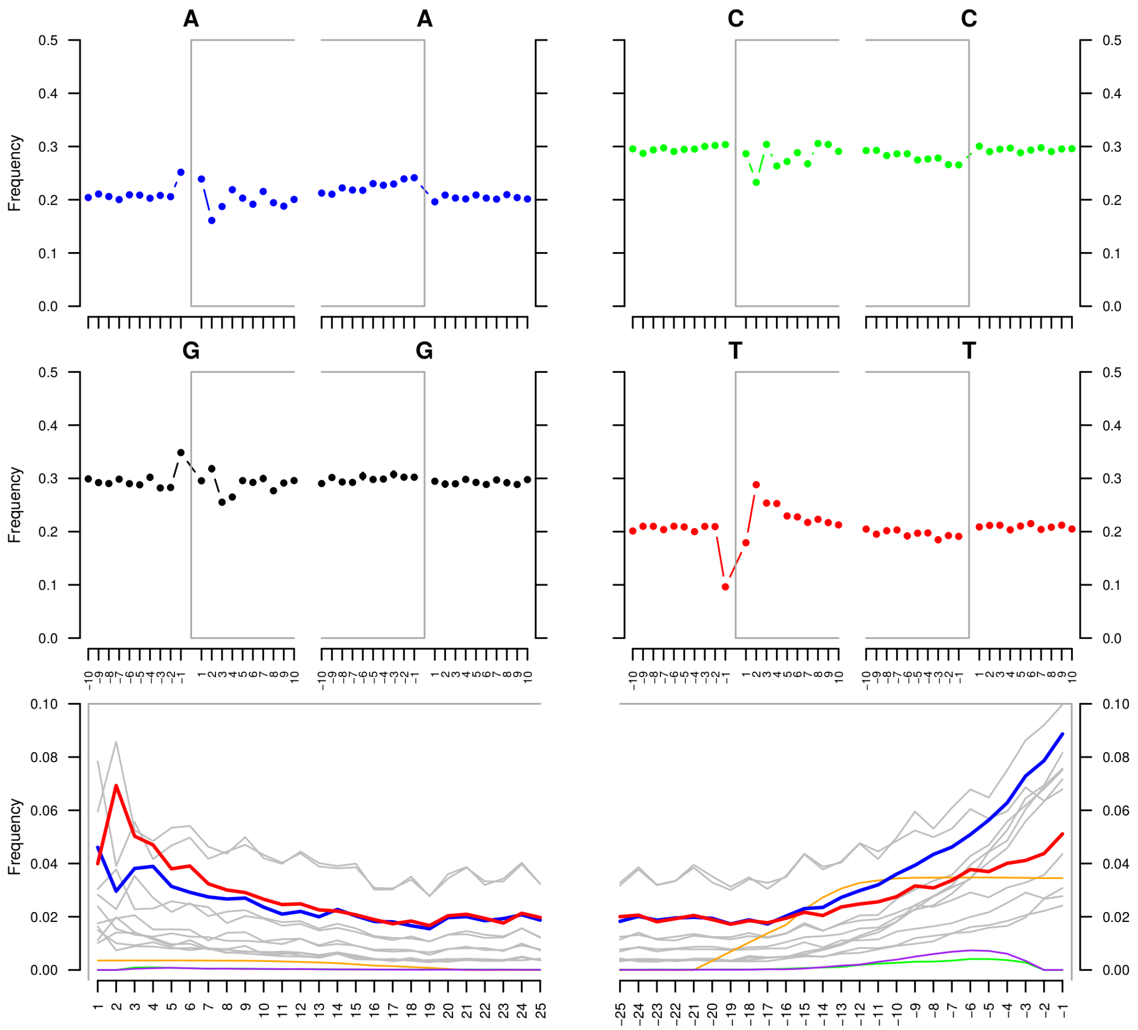

Figure S15: Patterns of damage in the reads aligned with the *P. antarctica* reference genome. The tool used was MAPDAMAGE v2.

```
git clone https://github.com/pratas/geco.git
cd geco/src/
cmake .
make
```

Then, we ran the following

```
# Download
wget ftp://ftp.ncbi.nlm.nih.gov/genomes/refseq/bacteria/Pseudomonas_antarctica/all_assembly_vers
ions/GCF_001647715.1_ASM164771v1/GCF_001647715.1_ASM164771v1_genomic.fna.gz
# Uncompress and Parsing
gunzip GCF_001647715.1_ASM164771v1_genomic.fna.gz
mv GCF_001647715.1_ASM164771v1_genomic.fna PA.fa
cat PA.fa | grep -v ">" | tr -d -c "ACGT" > PA.dna
cat Elephant.fa | grep -v ">" | tr -d -c "ACGT" > Elephant.dna
cat Mammoth.fa | grep -v ">" | tr -d -c "ACGT" > Mammoth.dna
# Run compression
./GeCo -rm 6:10:0:0/0 -rm 11:20:1:0/0 -rm 14:500:1:3/100 -r Elephant.dna PA.dna
./GeCo -rm 6:10:0:0/0 -rm 11:20:1:0/0 -rm 14:500:1:3/100 -r PA.dna Elephant.dna
```

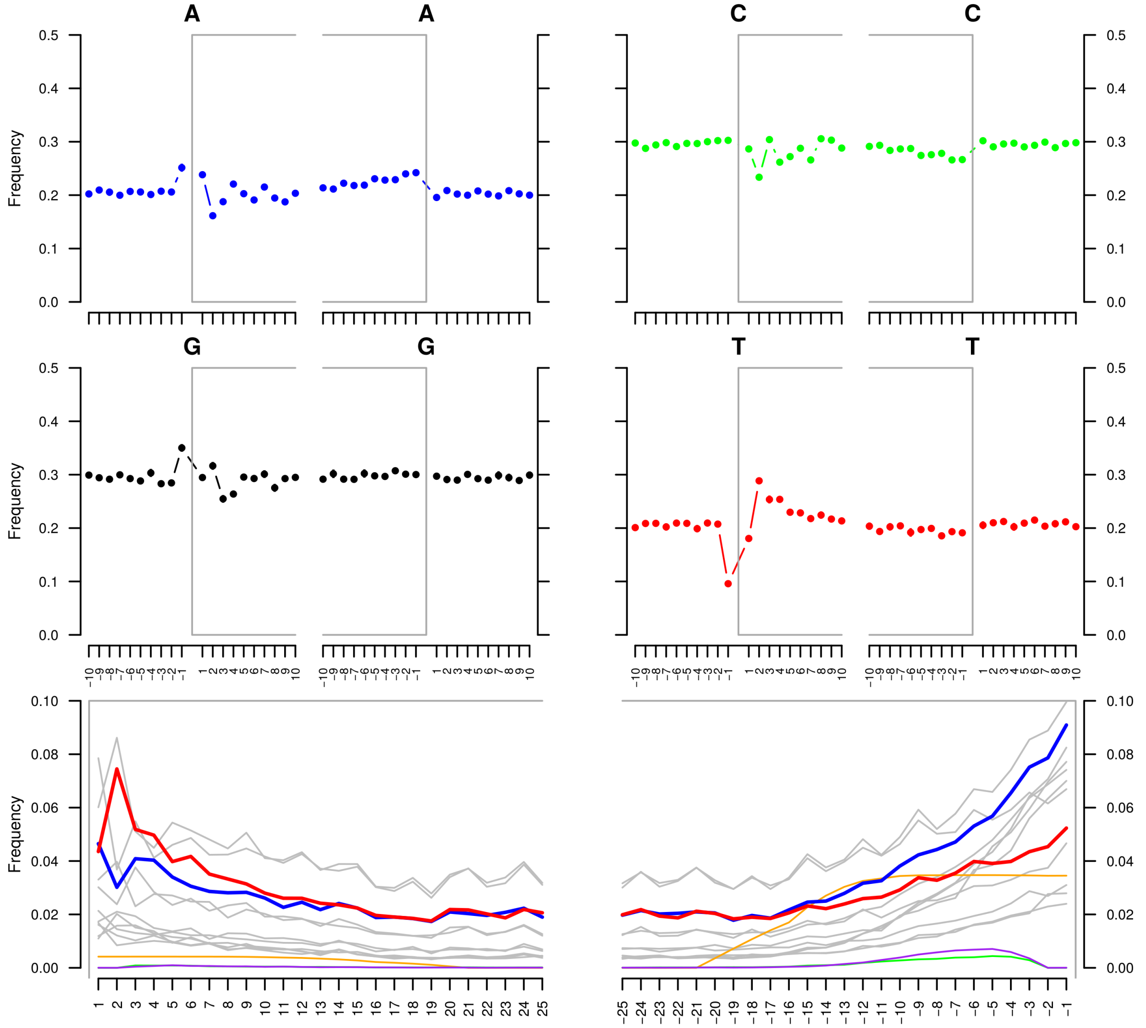

Figure S16: Patterns of damage in the reads aligned with the *P. fluorescens* reference genome. The tool used was MAPDAMAGE v2.

```
./GeCo -rm 6:10:0:0/0 -rm 11:20:1:0/0 -rm 14:500:1:3/100 -r Mammoth.dna PA.dna
./GeCo -rm 6:10:0:0/0 -rm 11:20:1:0/0 -rm 14:500:1:3/100 -r PA.dna Mammoth.dna
```

that resulted in:

```
Reference: Elephant.dna, Target: PA.dna
Total bytes: 1619467 (1.5 MB), 2.011 bpb, Normalized Dissimilarity Rate: 1.00565
#
Reference: PA.dna, Target: Elephant.dna
Total bytes: 4423 (4.3 KB), 2.098 bpb, Normalized Dissimilarity Rate: 1.04897
#
Reference: Mammoth.dna, Target: PA.dna
Total bytes: 1654921 (1.6 MB), 2.004 bpb, Normalized Dissimilarity Rate: 1.00191
#
Reference: PA.dna, Target: Mammoth.dna
Total bytes: 4295 (4.2 KB), 2.089 bpb, Normalized Dissimilarity Rate: 1.04444
```

As it can be seen, the relative similarity of the *P. antarctica* reference genome relatively to the Mammoth or Elephant, and vice-versa, is null according to the compressor GeCo (Normalized Dissimilarity Rate

>= 1 means maximum statistical dissimilarity). Therefore, this supports the possibility of a true positive ancient authentication, given the absense of correlation between the mtDNA and the bacterial DNA.

Note 12: Software availability and characteristics

All source codes for FALCON are available at:

- `https://github.com/pratas/falcon`;
- `https://pratas.github.io/falcon` (secondary link);

To install the complete package, run:

```
git clone https://github.com/pratas/falcon.git
cd falcon/src/
cmake .
make
```

FALCON includes the following programs:

1. **FALCON** - infers metagenomic composition;
2. **FALCON-FILTER** - filters local relative similarity of inferred sequences;
3. **FALCON-EYE** - displays the output from FALCON and FALCON-FILTER in a compact map;

For a common pipeline using these programs see **Supplementary Note 2**.

FALCON program

The FALCON program enables to identify and quantify the relative similarity between (usually extant) reference sequences and the FASTQ reads from ancient genomes. For secure encryption of the FASTA and FASTQ files, use CRYFA [90]. In the following subsections, we explain the input and output parameters.

Input parameters

The FALCON program needs two files for the computation. The FILE1, with the reads provided from the NGS platform, and the FILE2, with the multi-FASTA cointaining the sequences of the genomes and respective headers. These parameters are mandatory arguments. The rest of the parameters are not mandatory.

```
Usage: FALCON [OPTION]... [FILE1] [FILE2]
A compression-based method to infer metagenomic sample composition.

Non-mandatory arguments:

  -h                         give this help,
  -F                         force mode (overwrites top file),
  -V                         display version number,
  -v                         verbose mode (more information),
  -Z                         database local similarity,
  -s                         show compression levels,
  -l <level>                 compression level [1;44],
  -p <sample>                subsampling (default: 1),
  -t <top>                   top of similarity (default: 20),
  -n <nThreads>              number of threads (default: 2),
  -x <FILE>                  similarity top filename,
  -y <FILE>                  local similarities filename,

Mandatory arguments:

  [FILE1]                    metagenomic filename (FASTA or FASTQ),
  [FILE2]                    database filename (FASTA or Multi-FASTA).
```

There are two hidden parameters. One for setting the cache size:

```
  -c <cache>               maximum collisions for hash cache. Memory
                           values are higly dependent of the parameter
                           specification.
```

This is a parameter that affects the precision and computational resources needed for the computation. Lower values use less memory. For a complete description see [25, 65]. The previous parameter is only needed when the context of the Markov model is higher than 13. For setting, specifically, the models through the command line, use the following explanation (more information can be found at [25, 65]):

```
  -m <c>:<d>:<i>:<m/e>   context (Markov) model (ex:-m 13:100:0:0/0),
  -m <c>:<d>:<i>:<m/e>   context (Markov) model (ex:-m 18:1000:1:1/500),
  ...
                         templates use <c> for context-order size, <d> for
                         alpha (1/<d>), <i> (0 or 1) to set the usage of
                         inverted repeats (1 to use) and <m> to the
                         maximum allowed mutation on the context without
                         being discarded (usefull in deep contexts), under
                         the estimator <e>.
```

Although the models might be set directly from the command line, we have already set 44 levels with multiple combinations of models and memory caches. These levels are the following:

```
Level  1: -m 12:20:0:0/0
Level  2: -m 12:20:1:0/0
Level  3: -m 13:50:0:0/0
Level  4: -m 13:50:1:0/0
Level  5: -m 14:100:0:0/0
Level  6: -m 14:100:1:0/0
Level  7: -m 14:100:1:1/20
Level  8: -m 18:200:0:1/50 -c 10 -g 0.9
Level  9: -m 18:200:1:1/50 -c 10 -g 0.9
Level 10: -m 18:200:1:1/50 -c 20 -g 0.9
Level 11: -m 18:200:1:1/50 -c 30 -g 0.9
Level 12: -m 18:200:1:1/50 -c 40 -g 0.9
Level 13: -m 20:500:1:1/100 -c 10 -g 0.9
Level 14: -m 20:500:1:1/100 -c 20 -g 0.9
Level 15: -m 20:500:1:1/100 -c 30 -g 0.9
Level 16: -m 20:500:1:1/100 -c 40 -g 0.9
Level 17: -m 20:500:1:1/100 -c 50 -g 0.9
Level 18: -m 20:500:1:2/50  -c 50 -g 0.9
Level 19: -m 20:500:1:2/50  -m 13:100:1:0/0 -c 30 -g 0.9
Level 20: -m 20:500:1:1/100 -m 13:100:1:0/0 -c 35 -g 0.9
Level 21: -m 20:500:1:1/100 -m 13:100:1:0/0 -c 40 -g 0.9
Level 22: -m 20:500:1:1/100 -m 13:100:1:0/0 -c 45 -g 0.9
Level 23: -m 20:500:1:1/100 -m 14:100:1:0/0 -c 20 -g 0.9
Level 24: -m 20:500:1:1/100 -m 14:100:1:0/0 -c 25 -g 0.9
Level 25: -m 20:500:1:1/100 -m 14:100:1:0/0 -c 30 -g 0.9
Level 26: -m 20:500:1:1/100 -m 14:100:1:0/0 -c 35 -g 0.9
Level 27: -m 20:500:1:1/100 -m 14:100:1:0/0 -c 40 -g 0.9
Level 28: -m 20:500:1:1/100 -m 14:100:1:0/0 -c 45 -g 0.9
Level 29: -m 20:500:1:1/100 -m 14:100:1:0/0 -c 50 -g 0.9
Level 30: -m 20:500:1:1/100 -m 14:100:1:0/0 -c 60 -g 0.9
Level 31: -m 20:500:1:1/100 -m 14:100:1:0/0 -c 70 -g 0.9
Level 32: -m 20:500:1:1/100 -m 14:100:1:0/0 -m 12:1:0:0/0 -m 8:1:0:0/0 -c 30
Level 33: -m 20:500:1:1/100 -m 14:100:1:0/0 -m 12:1:0:0/0 -m 8:1:0:0/0 -c 40
Level 34: -m 20:500:1:1/100 -m 14:100:1:0/0 -m 12:1:0:0/0 -m 8:1:0:0/0 -c 50
Level 35: -m 20:500:1:1/100 -m 14:100:1:0/0 -m 12:1:0:0/0 -m 8:1:0:0/0 -m 4:1:0:0/0 -c 50
Level 36: -m 20:500:1:3/20 -m 14:100:1:0/0 -m 12:1:0:0/0 -c 50 -g 0.95
Level 37: -m 20:500:1:3/20 -m 14:100:1:0/0 -c 50 -g 0.95
Level 38: -m 20:500:1:3/20 -m 14:100:1:0/0 -c 70 -g 0.95
Level 39: -m 20:500:1:3/20 -c 50 -g 0.95
Level 40: -m 14:50:1:1/10
Level 41: -m 8:1:0:0/0 -m 14:50:1:0/0
Level 42: -m 4:1:0:0/0 -m 14:50:1:0/0
Level 43: -m 6:1:0:0/0 -m 14:50:1:0/0
Level 44: -m 11:1:0:0/0 -m 14:50:1:0/0
```

Output

The output of the FALCON program are two files. One file contains the top-n values with the highest similarity relatively to the reads. The following output shows an example of a top-5:

```
1   9472    92.597  548558394   Human endogenous retrovirus K113
2   4668621 42.184  1069460419  Escherichia coli strain 210221272
3   4558287 42.016  1008930592  Shigella sp. PAMC 28760
4   4574246 36.661  844762407   Shigella boydii strain ATCC 9210
5   4878853 36.531  992379426   Shigella sonnei strain FDAARGOS 90
```

These column values stand for the ranking of higher similarity, the sequence size, the percentage of similarity, the global identifyer (GI) and the name of the FASTA read (genome name), respectively.

The other file contains the relative local similarities (in a compressed format) and it is only available when FALCON runs with the "-Z" flag. This file is the input of the FALCON-FILTER program that we will describe later. This file is packed in a compact format in order to not increase the storage, substantially.

FALCON-FILTER program

The FALCON-FILTER program enables to identify and quantify where the relative similarity between (extant) reference genomes and the FASTQ reads from ancient genomes, below a certain threshold, occurs. In the following subsections, we explain the input and output paramters.

Input parameters

The FALCON-FILTER program needs only one file for the computation. The file is provided from the output of the FALCON program (when the flag "-Z" is set on the FALCON program). The non-mandatory arguments are mostly to parameterize filters and the threshold options.

```
Usage: FALCON-FILTER [OPTION]... [FILE]
Filter and segment FALCON output.

Non-mandatory arguments:

  -h                         give this help,
  -F                         force mode (overwrites top file),
  -V                         display version number,
  -v                         verbose mode (more information),
  -s  <size>                 filter window size,
  -w  <type>                 filter window type,
  -x  <sampling>             filter window sampling,
  -sl <lower>              similarity lower bound,
  -su <upper>              similarity upper bound,
  -dl <lower>              size lower bound,
  -du <upper>              size upper bound,
  -t  <threshold>          threshold [0;2.0],
  -o  <FILE>               output filename,

Mandatory arguments:

  [FILE]                   profile filename (from FALCON).
```

Output

The output of the file contains the coordinates of the relative similar regions with the respective self-similarity classification.

FALCON-EYE program

The FALCON-EYE program enables to vizualise both results from the computation of FALCON and FALCON-FILTER. It creates a vectorial image (SVG) with the respective information. In the following subsections, we explain the input and output parameters.

Input parameters

The mandatory input of the FALCON-EYE program is provided by the output of the FALCON-FILTER program. The non-mandatory parameters are mainly related with the shape, colors and disposal of the image output.

```
Usage: FALCON-EYE [OPTION]... [FILE]
Visualize FALCON-FILTER output.

Non-mandatory arguments:

  -h                       give this help,
  -F                       force mode (overwrites top file),
  -V                       display version number,
  -v                       verbose mode (more information),
  -w  <width>              square width (for each value),
  -s  <ispace>             square inter-space (between each value),
  -i  <indexs>             color index start,
  -r  <indexr>             color index rotations,
  -u  <hue>                color hue,
  -sl <lower>              similarity lower bound,
  -su <upper>              similarity upper bound,
  -dl <lower>              size lower bound,
  -du <upper>              size upper bound,
  -bg                      show only the best of group,
  -g  <color>              color gamma,
  -e  <size>               enlarge painted regions,
  -ss                        do NOT show global scale,
  -sn                        do NOT show names,
  -o  <FILE>                 output image filename,

Mandatory arguments:

  [FILE]                     profile filename (from FALCON-FILTER).
```

Output

The output is an image with a summary of all the computations from the programs FALCON, FALCON-FILTER and FALCON-EYE. According to Figure S17, the image is mainly divided in two regions A and B. The region A stands for the Normalized Relative Similarity (NRS) of the reference genomes relatively to the ancient reads, while the B stands for the local relative similarity (**Supplementary Note 10**) classification of regions with high Normalized Relative Similarity below a certain threshold provided from A.

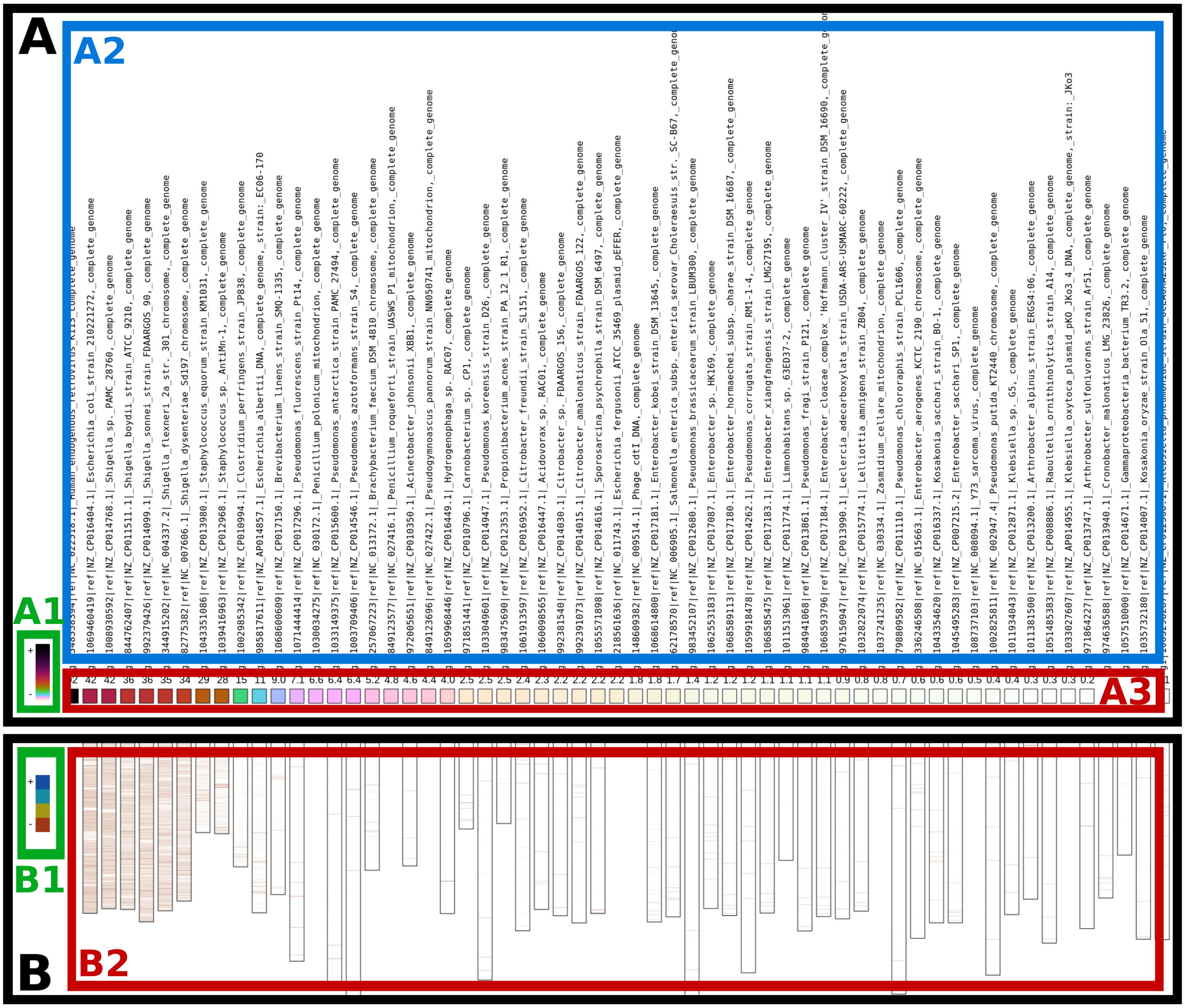

Figure S17: FALCON-EYE image output. **A)** Normalized Relative Similarity (NRS) of the reference genomes relatively to the ancient reads. A1) NRS scale, where "-" and "+" stand for lower and higer NRS values, respectively. A2) names (and GIs) of each reference genome (provided in the FASTA file). A3) NRS values and corresponding colors according to the scale. **B)** local relative similarity (**Supplementary Note 10**) classification of regions with high Normalized Relative Similarity provided from A). B1) local similarity scale where "-" stands for high complex regions, while "+" for very repetitive regions. B2) local similarity classification along the sequence ordered from the top to the bottom.

Bibliography

[1] R. J. Solomonoff, “A formal theory of inductive inference. Part I,” *Information and Control*, vol. 7, no. 1, pp. 1–22, Mar. 1964.

[2] ——, “A formal theory of inductive inference. Part II,” *Information and Control*, vol. 7, no. 2, pp. 224–254, Jun. 1964.

[3] G. J. Chaitin, “On the length of programs for computing finite binary sequences,” *Journal of the ACM*, vol. 13, pp. 547–569, 1966.

[4] C. E. Shannon, “A mathematical theory of communication,” *Bell System Technical Journal*, vol. 27, pp. 379–423,623–656, 1948.

[5] A. N. Kolmogorov, “Three approaches to the quantitative definition of information,” *Problems of Information Transmission*, vol. 1, no. 1, pp. 1–7, 1965.

[6] M. Hutter, “Algorithmic information theory: a brief non-technical guide to the field,” *Scholarpedia*, p. 9620, Mar. 2007.

[7] A. Turing, “On computable numbers, with an application to the Entscheidungsproblem,” *Proceedings of the London Mathematical Society*, vol. 42, no. 2, pp. 230–265, 1936.

[8] M. Li and P. Vitányi, *An introduction to Kolmogorov complexity and its applications*, 3rd ed. Springer, 2008.

[9] C. H. Bennett, P. Gács, M. L. P. M. B. Vitányi, and W. H. Zurek, “Information distance,” *IEEE Trans. on Information Theory*, vol. 44, no. 4, pp. 1407–1423, Jul. 1998.

[10] M. Li, X. Chen, X. Li, B. Ma, and P. M. B. Vitányi, “The similarity metric,” *IEEE Trans. on Information Theory*, vol. 50, no. 12, pp. 3250–3264, Dec. 2004.

[11] R. Cilibrasi *et al.*, “Statistical inference through data compression,” Ph.D. dissertation, Institute for Logic, Language and Computation, Universiteit van Amsterdam, 2007.

[12] P. Gács, “Quantum algorithmic entropy,” *Journal of Physics A: Math. Gen.*, vol. 34, no. 2001, pp. 6859–6880, 2001.

[13] F. Soler-Toscano, H. Zenil, J.-P. Delahaye, and N. Gauvrit, “Calculating Kolmogorov complexity from the output frequency distributions of small Turing machines,” *PLoS ONE*, vol. 9, no. 5, p. e96223, 2014.

[14] H. Zenil, F. Soler-Toscano, N. Kiani, S. Hernández-Orozco, and A. Rueda-Toicen, “A decomposition method for global evaluation of Shannon entropy and local estimations of algorithmic complexity,” arXiv preprint arXiv:1609.00110, 2016.

[15] J. Ziv and N. Merhav, “A measure of relative entropy between individual sequences with application to universal classification,” *IEEE Trans. on Information Theory*, vol. 39, no. 4, pp. 1270–1279, Jul. 1993.

[16] D. Benedetto, E. Caglioti, and V. Loreto, "Language trees and zipping," *Physical Review Letters*, vol. 88, no. 4, pp. 048 702–1–048 702–4, Jan. 2002.

[17] D. Cerra and M. Datcu, "Algorithmic relative complexity," *Entropy*, vol. 13, pp. 902–914, 2011.

[18] D. P. Coutinho and M. Figueiredo, "Text classification using compression-based dissimilarity measures," *Int. Journal of Pattern Recognition and Artificial Intelligence*, vol. 29, no. 5, 2015.

[19] A. J. Pinho, D. Pratas, and P. J. S. G. Ferreira, "Authorship attribution using relative compression," in *Proc. of the Data Compression Conf., DCC-2016*, Snowbird, Utah, Mar. 2016.

[20] A. Rybalov, "On the strongly generic undecidability of the halting problem," *Theoretical Computer Science*, vol. 377, no. 1, pp. 268–270, 2007.

[21] D. Coutinho and M. Figueiredo, "An information theoretic approach to text sentiment analysis," in *ICPRAM*, 2013, pp. 577–580.

[22] G. A. Fink, *Markov models for pattern recognition: from theory to applications.* Springer Science & Business Media, 2014.

[23] S. Brás and A. J. Pinho, "ECG biometric identification: A compression based approach," in *Engineering in Medicine and Biology Society (EMBC).* IEEE, 2015, pp. 5838–5841.

[24] D. Pratas, R. M. Silva, A. J. Pinho, and P. J. S. G. Ferreira, "An alignment-free method to find and visualise rearrangements between pairs of DNA sequences," *Scientific Reports*, vol. 5, p. 10203, May 2015.

[25] D. Pratas, A. J. Pinho, and P. J. S. G. Ferreira, "Efficient compression of genomic sequences," in *Proc. of the Data Compression Conf., DCC-2016*, Snowbird, Utah, Mar. 2016, pp. 231–240.

[26] K. Sayood, *Introduction to data compression*, 3rd ed. Morgan Kaufmann, 2006.

[27] A. J. Pinho, D. Pratas, and P. J. S. G. Ferreira, "Bacteria DNA sequence compression using a mixture of finite-context models," in *Proc. of the IEEE Workshop on Statistical Signal Processing*, Nice, France, Jun. 2011.

[28] D. Pratas and A. J. Pinho, "Exploring deep Markov models in genomic data compression using sequence pre-analysis," in *Proc. of the 22th European Signal Processing Conf., EUSIPCO-2014*, Lisbon, Portugal, Sep. 2014, pp. 2395–2399.

[29] W. Zhao, J. Wang, and H. Lu, "Combining forecasts of electricity consumption in China with time-varying weights updated by a high-order Markov chain model," *Omega*, vol. 45, pp. 80–91, 2014.

[30] J. Kwak, C.-H. Lee *et al.*, "A high-order Markov-chain-based scheduling algorithm for low delay in CSMA networks," *IEEE/ACM Transactions on Networking*, vol. 24, no. 4, pp. 2278–2290, 2016.

[31] M. Kárnỳ, "Recursive estimation of high-order Markov chains: approximation by finite mixtures," *Information Sciences*, vol. 326, pp. 188–201, 2016.

[32] D. Pratas, M. Hosseini, and A. J. Pinho, "Substitutional tolerant markov models for relative compression of DNA sequences," in *11th Int. Conf. on Practical Applications of Computational Biology & Bioinformatics.* Springer, 2017, pp. 265–272.

[33] T. C. Bell, J. G. Cleary, and I. H. Witten, *Text compression.* Prentice Hall, 1990.

[34] D. Pratas and A. J. Pinho, "Compressing the human genome using exclusively Markov models," in *Advances in Intelligent and Soft Computing, Proc. of the 5th Int. Conf. on Practical Applications of Computational Biology & Bioinformatics, PACBB 2011*, vol. 93, Apr. 2011, pp. 213–220.

[35] A. J. Pinho, P. J. S. G. Ferreira, A. J. R. Neves, and C. A. C. Bastos, "On the representability of complete genomes by multiple competing finite-context (Markov) models," *PLoS ONE*, vol. 6, no. 6, p. e21588, 2011.

[36] C. E. Shannon, "Channels with side information at the transmitter," *IBM Journal of Research and Development*, vol. 2, no. 4, pp. 289–293, 1958.

[37] R. Ounit, S. Wanamaker, T. J. Close, and S. Lonardi, "Clark: fast and accurate classification of metagenomic and genomic sequences using discriminative k-mers," *BMC genomics*, vol. 16, no. 1, p. 1, 2015.

[38] I. Borozan and V. Ferretti, "Cssscl: a python package that uses combined sequence similarity scores for accurate taxonomic classification of long and short sequence reads," *Bioinformatics*, p. btv587, 2015.

[39] J. G. Caporaso, J. Kuczynski, J. Stombaugh, K. Bittinger, F. D. Bushman, E. K. Costello, N. Fierer, A. G. Pena, J. K. Goodrich, J. I. Gordon *et al.*, "QIIME allows analysis of high-throughput community sequencing data," *Nature methods*, vol. 7, no. 5, pp. 335–336, 2010.

[40] L. S. Weyrich, S. Duchene, J. Soubrier, L. Arriola, B. Llamas, J. Breen, A. G. Morris, K. W. Alt, D. Caramelli, V. Dresely *et al.*, "Neanderthal behaviour, diet, and disease inferred from ancient DNA in dental calculus," *Nature*, 2017.

[41] M. Schubert, A. Ginolhac, S. Lindgreen, J. F. Thompson, K. A. Al-Rasheid, E. Willerslev, A. Krogh, and L. Orlando, "Improving ancient DNA read mapping against modern reference genomes," *BMC genomics*, vol. 13, no. 1, p. 178, 2012.

[42] M. Schubert, L. Ermini, C. Der Sarkissian, H. Jónsson, A. Ginolhac, R. Schaefer, M. D. Martin, R. Fernández, M. Kircher, M. McCue *et al.*, "Characterization of ancient and modern genomes by SNP detection and phylogenomic and metagenomic analysis using PALEOMIX," *Nature protocols*, vol. 9, no. 5, p. 1056, 2014.

[43] A. Herbig, F. Maixner, K. I. Bos, A. Zink, J. Krause, and D. H. Huson, "MALT: Fast alignment and analysis of metagenomic DNA sequence data applied to the Tyrolean Iceman," *bioRxiv preprint*, 2017.

[44] H. Li and R. Durbin, "Fast and accurate short read alignment with Burrows–Wheeler transform," *Bioinformatics*, vol. 25, no. 14, pp. 1754–1760, 2009.

[45] D. H. Huson, A. F. Auch, J. Qi, and S. C. Schuster, "MEGAN analysis of metagenomic data," *Genome research*, vol. 17, no. 3, pp. 377–386, 2007.

[46] G. Renaud, U. Stenzel, and J. Kelso, "leeHom: adaptor trimming and merging for Illumina sequencing reads," *Nucleic acids research*, p. gku699, 2014.

[47] M. L. Metzker, "Sequencing technologies–the next generation," *Nature reviews. Genetics*, vol. 11, no. 1, p. 31, 2010.

[48] M. Kircher, "Analysis of high-throughput ancient DNA sequencing data," in *Ancient DNA*. Springer, 2012, pp. 197–228.

[49] A. M. Bolger, M. Lohse, and B. Usadel, "Trimmomatic: a flexible trimmer for illumina sequence data," *Bioinformatics*, vol. 30, no. 15, pp. 2114–2120, 2014.

[50] D. Pratas, A. J. Pinho, and J. M. O. S. Rodrigues, "XS: a FASTQ read simulator," *BMC Research Notes*, vol. 7, no. 1, p. 40, 2014.

[51] M. Escalona, S. Rocha, and D. Posada, "A comparison of tools for the simulation of genomic next-generation sequencing data," *Nature Reviews Genetics*, vol. 17, no. 8, pp. 459–469, 2016.

[52] A. J. Pinho, D. Pratas, and S. P. Garcia, “GReEn: a tool for efficient compression of genome resequencing data,” *Nucleic Acids Research*, vol. 40, no. 4, p. e27, 2012.

[53] B. Langmead, C. Trapnell, M. Pop, and S. L. Salzberg, “Ultrafast and memory-efficient alignment of short DNA sequences to the human genome,” *Genome biology*, vol. 10, no. 3, p. R25, 2009.

[54] M. McClelland, L. Florea, K. Sanderson, S. W. Clifton, J. Parkhill, C. Churcher, G. Dougan, R. K. Wilson, and W. Miller, “Comparison of the escherichia coli k-12 genome with sampled genomes of a klebsiella pneumoniae and three salmonella enterica serovars, typhimurium, typhi and paratyphi,” *Nucleic acids research*, vol. 28, no. 24, pp. 4974–4986, 2000.

[55] J. F. Hughes and J. M. Coffin, “Evidence for genomic rearrangements mediated by human endogenous retroviruses during primate evolution,” *Nature genetics*, vol. 29, no. 4, pp. 487–489, 2001.

[56] E. Morandi, R. E. Tarlinton, and B. Gran, “Multiple sclerosis between genetics and infections: human endogenous retroviruses in monocytes and macrophages,” *Frontiers in immunology*, vol. 6, 2015.

[57] I. M. Campbell, T. Gambin, P. Dittwald, C. R. Beck, A. Shuvarikov, P. Hixson, A. Patel, A. Gambin, C. A. Shaw, J. A. Rosenfeld *et al.*, “Human endogenous retroviral elements promote genome instability via non-allelic homologous recombination,” *BMC biology*, vol. 12, no. 1, p. 74, 2014.

[58] G. Magiorkinis, D. Blanco-Melo, and R. Belshaw, “The decline of human endogenous retroviruses: extinction and survival,” *Retrovirology*, vol. 12, no. 1, p. 8, 2015.

[59] M. Peeters, M. D’Arc, and E. Delaporte, “The origin and diversity of human retroviruses,” *AIDS reviews*, vol. 16, no. 1, p. 23, 2014.

[60] K. Prüfer, F. Racimo, N. Patterson, F. Jay, S. Sankararaman, S. Sawyer, A. Heinze, G. Renaud, P. H. Sudmant, C. De Filippo *et al.*, “The complete genome sequence of a Neanderthal from the Altai Mountains,” *Nature*, vol. 505, no. 7481, pp. 43–49, 2014.

[61] 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, no. 16, p. 2078, 2009.

[62] M. Meyer, M. Kircher, M.-T. Gansauge, H. Li, F. Racimo, S. Mallick, J. G. Schraiber, F. Jay, K. Prüfer, C. De Filippo *et al.*, “A high-coverage genome sequence from an archaic Denisovan individual,” *Science*, vol. 338, no. 6104, pp. 222–226, 2012.

[63] J. Enk, A. Devault, R. Debruyne, C. E. King, T. Treangen, D. O’Rourke, S. L. Salzberg, D. Fisher, R. MacPhee, and H. Poinar, “Complete Columbian mammoth mitogenome suggests interbreeding with woolly mammoths,” *Genome biology*, vol. 12, no. 5, p. 1, 2011.

[64] J. L. Paijmans, M. T. P. Gilbert, and M. Hofreiter, “Mitogenomic analyses from ancient DNA,” *Molecular phylogenetics and evolution*, vol. 69, no. 2, pp. 404–416, 2013.

[65] D. Pratas, “Compression and analysis of genomic data,” Ph.D. dissertation, University of Aveiro, 2016.

[66] M. K. Scott, C. Chommanard, X. Lu, D. Appelgate, L. Grenz, E. Schneider, S. I. Gerber, D. D. Erdman, and A. Thomas, “Human adenovirus associated with severe respiratory infection, Oregon, USA, 2013–2014,” *Emerging infectious diseases*, vol. 22, no. 6, p. 1044, 2016.

[67] C. Garnett, D. Erdman, W. Xu, and L. R. Gooding, “Prevalence and quantitation of species C adenovirus DNA in human mucosal lymphocytes,” *Journal of virology*, vol. 76, no. 21, pp. 10 608–10 616, 2002.

[68] D. Reich, R. E. Green, M. Kircher, J. Krause, N. Patterson, E. Y. Durand, B. Viola, A. W. Briggs, U. Stenzel, P. L. Johnson *et al.*, "Genetic history of an archaic hominin group from Denisova Cave in Siberia," *Nature*, vol. 468, no. 7327, pp. 1053–1060, 2010.

[69] M. Krings, A. Stone, R. W. Schmitz, H. Krainitzki, M. Stoneking, and S. Pääbo, "Neandertal DNA sequences and the origin of modern humans," *Cell*, vol. 90, no. 1, pp. 19–30, 1997.

[70] F. Maixner, B. Krause-Kyora, D. Turaev, A. Herbig, M. R. Hoopmann, J. L. Hallows, U. Kusebauch, E. E. Vigl, P. Malfertheiner, F. Megraud *et al.*, "The 5300-year-old Helicobacter pylori genome of the Iceman," *Science*, vol. 351, no. 6269, pp. 162–165, 2016.

[71] R. Khairat, M. Ball, C.-C. H. Chang, R. Bianucci, A. G. Nerlich, M. Trautmann, S. Ismail, G. M. Shanab, A. M. Karim, Y. Z. Gad *et al.*, "First insights into the metagenome of Egyptian mummies using next-generation sequencing," *Journal of applied genetics*, vol. 54, no. 3, pp. 309–325, 2013.

[72] J. Daura, M. Sanz, J. L. Arsuaga, D. L. Hoffmann, R. M. Quam, M. C. Ortega, E. Santos, S. Gómez, A. Rubio, L. Villaescusa *et al.*, "New Middle Pleistocene hominin cranium from Gruta da Aroeira (Portugal)," *Proceedings of the National Academy of Sciences*, vol. 114, no. 13, pp. 3397–3402, 2017.

[73] F. M. Key, C. Posth, J. Krause, A. Herbig, and K. I. Bos, "Mining Metagenomic Data Sets for Ancient DNA: Recommended Protocols for Authentication," *Trends in Genetics*, vol. 33, no. 8, pp. 508–520, 2017.

[74] A. Keller, A. Graefen, M. Ball, M. Matzas, V. Boisguerin, F. Maixner, P. Leidinger, C. Backes, R. Khairat, M. Forster *et al.*, "New insights into the Tyrolean Iceman's origin and phenotype as inferred by whole-genome sequencing," *Nature Communications*, vol. 3, p. 698, 2012.

[75] K. I. Bos, K. M. Harkins, A. Herbig, M. Coscolla, N. Weber, I. Comas, S. A. Forrest, J. M. Bryant, S. R. Harris, V. J. Schuenemann *et al.*, "Pre-Columbian mycobacterial genomes reveal seals as a source of New World human tuberculosis," *Nature*, vol. 514, no. 7523, pp. 494–497, 2014.

[76] C. Warinner, J. F. M. Rodrigues, R. Vyas, C. Trachsel, N. Shved, J. Grossmann, A. Radini, Y. Hancock, R. Y. Tito, S. Fiddyment *et al.*, "Pathogens and host immunity in the ancient human oral cavity," *Nature genetics*, vol. 46, no. 4, pp. 336–344, 2014.

[77] A. W. Briggs, U. Stenzel, P. L. Johnson, R. E. Green, J. Kelso, K. Prüfer, M. Meyer, J. Krause, M. T. Ronan, M. Lachmann *et al.*, "Patterns of damage in genomic DNA sequences from a Neandertal," *Proceedings of the National Academy of Sciences*, vol. 104, no. 37, pp. 14 616–14 621, 2007.

[78] J. Dabney, M. Meyer, and S. Pääbo, "Ancient DNA damage," *Cold Spring Harbor Perspectives in Biology*, vol. 5, no. 7, p. a012567, 2013.

[79] A. Ginolhac, M. Rasmussen, M. T. P. Gilbert, E. Willerslev, and L. Orlando, "mapDamage: testing for damage patterns in ancient DNA sequences," *Bioinformatics*, vol. 27, no. 15, pp. 2153–2155, 2011.

[80] H. Jónsson, A. Ginolhac, M. Schubert, P. L. Johnson, and L. Orlando, "mapDamage2.0: fast approximate Bayesian estimates of ancient DNA damage parameters," *Bioinformatics*, p. btt193, 2013.

[81] P. Skoglund, B. H. Northoff, M. V. Shunkov, A. P. Derevianko, S. Pääbo, J. Krause, and M. Jakobsson, "Separating endogenous ancient DNA from modern day contamination in a Siberian Neandertal," *Proceedings of the National Academy of Sciences*, vol. 111, no. 6, pp. 2229–2234, 2014.

[82] L. Orlando, M. T. P. Gilbert, and E. Willerslev, "Reconstructing ancient genomes and epigenomes," *Nature Reviews. Genetics*, vol. 16, no. 7, p. 395, 2015.

[83] A. Peltzer, G. Jäger, A. Herbig, A. Seitz, C. Kniep, J. Krause, and K. Nieselt, “EAGER: efficient ancient genome reconstruction,” *Genome biology*, vol. 17, no. 1, p. 60, 2016.

[84] B. R. Jones, A. Rajaraman, E. Tannier, and C. Chauve, “ANGES: reconstructing ANcestral GEnomeS maps,” *Bioinformatics*, vol. 28, no. 18, pp. 2388–2390, 2012.

[85] J. Lee, Y.-J. Cho, J. Y. Yang, Y.-J. Jung, S. G. Hong, and O.-S. Kim, “Complete genome sequence of Pseudomonas antarctica PAMC 27494, a bacteriocin-producing psychrophile isolated from Antarctica,” *Journal of biotechnology*, vol. 259, pp. 15–18, 2017.

[86] M. Redondo-Nieto, M. Barret, J. P. Morrisey, K. Germaine, F. Martínez-Granero, E. Barahona, A. Navazo, M. Sánchez-Contreras, J. A. Moynihan, S. R. Giddens *et al.*, “Genome sequence of the biocontrol strain Pseudomonas fluorescens F113,” *Journal of bacteriology*, vol. 194, no. 5, pp. 1273–1274, 2012.

[87] P. Shanahan, D. J. O’Sullivan, P. Simpson, J. D. Glennon, and F. O’Gara, “Isolation of 2, 4-diacetylphloroglucinol from a fluorescent pseudomonad and investigation of physiological parameters influencing its production,” *Applied and Environmental Microbiology*, vol. 58, no. 1, pp. 353–358, 1992.

[88] M. W. Gray, G. Burger, and B. F. Lang, “Mitochondrial evolution,” *Science*, vol. 283, no. 5407, pp. 1476–1481, 1999.

[89] J. M. Archibald, “Endosymbiosis and eukaryotic cell evolution,” *Current Biology*, vol. 25, no. 19, pp. R911–R921, 2015.

[90] D. Pratas, M. Hosseini, and A. J. Pinho, “Cryfa: A Tool to Compact and Encrypt FASTA Files,” in *11th Int. Conf. on Practical Applications of Computational Biology & Bioinformatics*. Springer, 2017, pp. 305–312.