Conservation in Mitochondrial DNA: Parallelized Estimation and Alignment Influence

Francisco Merino-Casallo, Jorge Álvarez-Jarreta and Elvira Mayordomo
Dept. de Informática e Ingeniería de Sistemas (DIIS) &
Instituto de Investigación en Ingeniería de Aragón (I3A),
Universidad de Zaragoza, María de Luna 1, 50018 Zaragoza, Spain
Email: {fmerino, jorgeal, elvira}@unizar.es

***Abstract*—The wide availability of sequenced biological data has challenged the conventional methods and tools used in molecular biology to compute the conservation index. The increasing size of input datasets is making the time cost of current conservation methods unaffordable.**

We propose a new software tool that combines several estimation methods applied to the conservation computation process with parallelization and divide-and-conquer techniques substantially improving its performance without affecting its accuracy.

We have also made an in-depth analysis of the impact of different methods and parameter selection on the alignment process applied prior to the conservation analysis of datasets. We have used sets of mitochondrial DNA sequences with different levels of heterogeneity and length, to provide a full case study. The software tool and the datasets are freely available at http://www.zaramit.org/conservation_index

***Keywords*—conservation index, parallelism, divide-and-conquer, alignment methods**

I. Introduction

Conservation analysis of biological sequences has led the scientific community to some meaningful advances on several fields of study. For decades researchers have been studying mitochondrial DNA (*mtDNA*). Human mtDNA (*hmtDNA*) was the first significant part of the human genome to be sequenced [1], and thereafter it has been a key element on a multitude of biological studies such as forensics [2], medical studies of mitochondrial diseases [3] and evolutionary studies of the human species [4]. mtDNA not only has a lower relative degradation than nuclear DNA (*nDNA*) [5], but also the mutation rate of the former exceeds approximately by a factor of 10 the one of nDNA [6].

In genetics, a mutation can be classified as nonneutral, which includes harmful and advantageous ones; or neutral based on its effect on fitness. According to the neutral model of molecular evolution, harmful mutations (also known as deleterious) are removed by negative selection while those classified as neutral are kept (advantageous mutations occur so rarely that they can be ignored). The positions with the highest functional importance in hemoglobin are those where heme is bounded and there is a remarkable conservation of the amino acids occupying these sites over millions of years of evolutionary history [7]. By contrast, other positions in the protein show a much higher rate of substitution. If the degree of functional constraint dictates how conserved a position is, then identifying conserved regions of a protein is tremendously useful [8].

Over the last decades, the scientific community has experienced an increasing rate of the biological data available, even more with the incorporation of the so-called *next generation sequencing methods* [9]. When tried to apply traditional methods and tools to these large datasets, overflow problems were uncovered, requiring new algorithmic techniques to handle this new scenario. Thus, many of those methods, like the ones used to calculate the conservation index, can take advantage of parallelization and divide-and-conquer techniques. Consequently, their performance will increase for small datasets and it will make them suitable to be applied to new large datasets.

The analysis of the conservation index of a set of sequences requires all of them to have the same length, i.e. to be aligned. This procedure is really common on evolutionary studies given the nature of the biological sequences: usually not every sequence of one set has the same length—sequences may suffer specific mutations called insertions or deletions, better known as *indels*, in some of their sites or sections (with higher impact in the noncoding DNA regions). The various heuristic methods one can apply to align the dataset may lead to different alignments from the same input, yielding an unknown effect on the calculation of the conservation of residues for each site.

In this paper we present a new software tool designed to calculate the conservation index of an input dataset of aligned sequences. The key values of our proposal are the combination of different methods into a single tool and the application of parallelization techniques to dramatically improve their performance. Furthermore, we have deeply studied the impact of the alignment in identifying the conservation of residues (nucleotides or amino acids). The alignment method selected along with its parameters can make the results highly variable [10]. Our results proving this statement are based on the analysis of different sets of aligned sequences—the differences between these sets are not only based on the type of residue, but also on the taxonomic classification of each of them, i.e. on the basis of shared characteristics.

II. Background

Conservation analysis is one of the most widely used methods for predicting functionally important residues in protein sequences [11]. In the last couple of decades there have been significant scientific advances based on the association of some nonneutral mutations to different types of cancer [12], [13] such as breast [14], gastric [15], lung [16], pancreatic [17] or prostate [18]. Nonneutral mutations have been also associated to other diseases such as Alzheimer's [19], diabetes [20], Parkinson's [21] and different cardiovascular ones [22], [23].

Even though a wide number of methods have been proposed during the last fifty years [24], there is no universally agreed upon technique [11]. The discrepancies among these methods can be clustered in different steps of the analysis. Usually, the analysis process is divided in two stages: estimation of frequencies of residues and calculation of the conservation score. For the first stage, some methods use a mathematical approach whilst others infer phylogenetic trees as the basis for this estimation process. For the latter, there are methods that apply entropy-based or variance-based approaches, and there are alternatives that use substitution matrices to perform the conservation calculus. Besides, there is another subset of methods that take into account the similarity between residues by grouping them into several classes prior to the estimation of the frequencies [8].

We could not find any paper that studied the need of parallel algorithms to analyze conservation. As aforementioned in the introduction, these tools were designed for a rather small set of biological sequences but, as the number of available sequences increases, their execution time is becoming unaffordable. Thus, it is required to implement parallelization and divide-and-conquer techniques to create a new set of conservation tools that do the same analysis, performing in really affordable execution times even for large input datasets.

As for conservation methods, there are several alignment methods as well. Their main goal is to arrange biological sequences so as to be able to compare them. The core of the alignment process is the score function, which differs for each method. Basically, this score function computes and minimizes the penalty value of different *gap* distributions at the input dataset. Moreover, this artificial element has associated a penalty factor when estimating the conservation score. Thus, the more accurate the alignment, the lesser the effect of the gap penalty factor will be in the conservation estimation. Besides, the performance of alignment methods is strongly affected not only by the size of the set of sequences, i.e. the number of sequences and their length, but also by the similarity between them. Furthermore, there is an obvious trade-off between accuracy and computing times: the more accurate, the higher the execution time of the alignment process.

III. METHODS

Conservation index (also known as conservation score) is considered to be a reliable metric for quantifying residue conservation. It is estimated on a column-by-column basis and helps scientists find out functionally important positions.

A valid conservation analysis for a set of biological sequences is made up of two main stages: estimation of frequencies and calculation of conservation score. The former estimates the number of times each residue appears in every position of each sequence. The latter returns a position-specific value, which represents how likely a mutation is expected to take place in that position. Finally, the representation of this analysis can be performed using different approaches.

We developed a new tool for carrying out conservation analyses that was implemented using Python (version 3.4). This decision was based on the widespread use of Python in bioinformatics as well as the Biopython project [25], which include a wide range of modules to manage biological information. Furthermore, due to Python's design philosophy, which emphasizes code readability, the scientific community could easily comprehend and extend the implemented algorithms.

Next, we provide a deeper description of the main aspects that affect the design and implementation of the proposed conservation analysis tool.

Data Structures

First, we studied the major data structure alternatives for the frequency storage during the conservation analysis. The methods we have chosen are almost constantly accessing the frequencies' data structure (at least once for every single residue and position of the input alignment). Hence, this decision had a key role on the final performance of our software tool. Two alternatives were considered: i) a dictionary of lists; and ii) a list of dictionaries.

TABLE I. EXECUTION TIME & MEMORY USAGE OF 10000 RUNS OF THE INITIALIZATION ALGORITHM USING DIFFERENT DATA STRUCTURES.

Data Structure	Size	Execution Time (s)	Memory Usage (MB)
Dict of Lists	10K items	$\overline{x}$ = 1.72 σ = 0.01	$\overline{x}$ = 8.48 σ = 0.18
	20K items	$\overline{x}$ = 3.67 σ = 0.01	$\overline{x}$ = 9.10 σ = 0.10
	30K items	$\overline{x}$ = 5.64 σ = 0.01	$\overline{x}$ = 9.71 σ = 0.15
	40K items	$\overline{x}$ = 7.59 σ = 0.01	$\overline{x}$ = 10.04 σ = 0.10
	50K items	$\overline{x}$ = 9.56 σ = 0.01	$\overline{x}$ = 10.71 σ = 0.14
	60K items	$\overline{x}$ = 11.52 σ = 0.01	$\overline{x}$ = 11.32 σ = 0.02
	70K items	$\overline{x}$ = 13.48 σ = 0.01	$\overline{x}$ = 11.82 σ = 0.10
	80K items	$\overline{x}$ = 15.45 σ = 0.01	$\overline{x}$ = 12.47 σ = 0.09
	90K items	$\overline{x}$ = 17.45 σ = 0.02	$\overline{x}$ = 12.61 σ = 0.00
List of Dicts	10K items	$\overline{x}$ = 47.86 σ = 0.09	$\overline{x}$ = 14.03 σ = 0.18
	20K items	$\overline{x}$ = 100.34 σ = 0.18	$\overline{x}$ = 20.43 σ = 0.17
	30K items	$\overline{x}$ = 140.06 σ = 0.12	$\overline{x}$ = 26.26 σ = 0.17
	40K items	$\overline{x}$ = 193.12 σ = 0.05	$\overline{x}$ = 32.38 σ = 0.17
	50K items	$\overline{x}$ = 247.23 σ = 0.12	$\overline{x}$ = 38.47 σ = 0.16
	60K items	$\overline{x}$ = 299.67 σ = 0.15	$\overline{x}$ = 44.66 σ = 0.16
	70K items	$\overline{x}$ = 354.21 σ = 0.73	$\overline{x}$ = 50.63 σ = 0.16
	80K items	$\overline{x}$ = 406.24 σ = 0.80	$\overline{x}$ = 56.82 σ = 0.14
	90K items	$\overline{x}$ = 419.79 σ = 0.41	$\overline{x}$ = 62.98 σ = 0.16

We tested both structures with a naive algorithm where each structure was created and initialized with zeros. We measured both execution time and memory usage for 10000 runs, and the outcome is shown in Table I. According to those results, using a dictionary of lists was the best option for both metrics.

Estimation of frequencies

The first step on the process to compute the conservation index is to estimate the residue frequencies. The methods chosen for this stage are totally compatible with the ones that will be used in the next one. Moreover, we selected two of the most consistent methods: weighted and unweighted. The former assigns a specific weight to each sequence based on its similarity with the rest of the aligned set, aiming to compensate for over-representation among multiple aligned sequences. The latter allocates the same weight to every sequence, considering each one equally significant.

Calculation of conservation score

Once all the frequencies are estimated, we can compute the conservation index. We selected two well-known and most used techniques: entropy-based and variance-based methods. There are several proposals for entropy-based methods [24], so we chose the one with better qualities in terms of simplicity of the computation process and accuracy of the results. On one hand, the entropy-based method returns a value that will reach its minimal value when all residues at a given site have equal frequencies. On the other hand, the variance-based method maximizes the returned value at the site occupied by an invariant residue whose overall frequency is minimal.

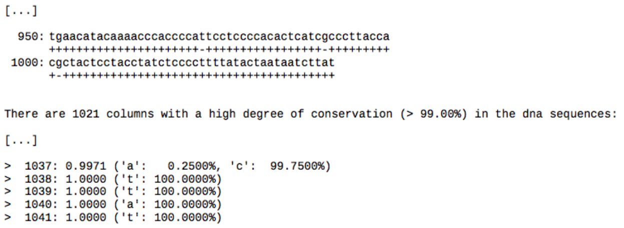
Parallelization

As we have claimed, one of the major benefits of our tool is the incorporation of parallelism, making it suitable even for large input alignments. The main idea underneath was to take advantage of the assumption of independence among sites of the input alignment, splitting the input into subsets of subsequences and generating one processing element per subset.

Regarding its implementation, a deep analysis of the parallel algorithms and their execution was performed. One of the fundamental elements to take into account was Python's Global Interpreter Lock (*GIL*), which showed a limitation in threads performance, especially in CPU-intensive algorithms. This limitation was due to the fact that GIL restricts to one the number of threads that can be running in the interpreter at once. Thus, the only situation in which it is suitable to simulate a parallel execution with threads is when the algorithm has a lot of blocking operations such as I/O or network communication. However, our method doesnt belong to this sort of problems. The alternative was to use Python's multiprocessing module, which creates independent processes instead of a multithreading scheme. It suited perfectly our needs and it was the tool we used to implement the parallelization of our application.

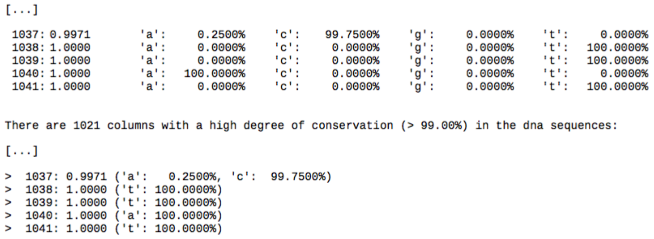
Report generation

The last stage of our application is the generation of a report containing the most valuable information about the conservation analysis. There are several parameters the user can select in order to get the report that better fits its needs: a threshold to highlight only those sites that cross it, and the detail level of the report. Currently the application offers two different report levels: basic and detailed. The first one includes the consensus sequence of the input alignment and the list of sites that cross the chosen threshold with their conservation distribution. The second one includes the same information of the basic report, extending the list of sites to all the sites of the input alignment. An example overview of a basic report is shown in Figure 1. On the other hand, Figure 2 show an example overview of a detailed report for the same case.



```
[...]

  950: tgaacatacaaaacccaccccattcctccccacactcatcgcccttacca
       ++++++++++++++++++++++-+++++++++++++++++-+++++++++
 1000: cgctactcctacctatctccccttttatactaataatcttat
       +-++++++++++++++++++++++++++++++++++++++++


There are 1021 columns with a high degree of conservation (> 99.00%) in the dna sequences:

[...]

>  1037: 0.9971 ('a':   0.2500%, 'c':  99.7500%)
>  1038: 1.0000 ('t': 100.0000%)
>  1039: 1.0000 ('t': 100.0000%)
>  1040: 1.0000 ('a': 100.0000%)
>  1041: 1.0000 ('t': 100.0000%)
```

Fig. 1. Example overview of a basic report.



```
[...]

 1037: 0.9971       'a':     0.2500%    'c':    99.7500%    'g':     0.0000%    't':     0.0000%
 1038: 1.0000       'a':     0.0000%    'c':     0.0000%    'g':     0.0000%    't':   100.0000%
 1039: 1.0000       'a':     0.0000%    'c':     0.0000%    'g':     0.0000%    't':   100.0000%
 1040: 1.0000       'a':   100.0000%    'c':     0.0000%    'g':     0.0000%    't':     0.0000%
 1041: 1.0000       'a':     0.0000%    'c':     0.0000%    'g':     0.0000%    't':   100.0000%

There are 1021 columns with a high degree of conservation (> 99.00%) in the dna sequences:

[...]

>  1037: 0.9971 ('a':   0.2500%, 'c':  99.7500%)
>  1038: 1.0000 ('t': 100.0000%)
>  1039: 1.0000 ('t': 100.0000%)
>  1040: 1.0000 ('a': 100.0000%)
>  1041: 1.0000 ('t': 100.0000%)
```

Fig. 2. Example overview of a detailed report.

IV. Results and Discussion

This section has a twofold division. First, we show all the information, tests performed and results of the execution time speedup obtained with our new tool. Secondly, we provide all the details of the studies made to test the influence of the alignment process on the conservation analysis.

A. Performance evaluation: Sequential vs Parallel approach

Sequences datasets: The performance tests were carried out using 9 subsets extracted from a set of all the complete hmtDNA sequences stored at GenBank [26], a comprehensive and well-known database that contains publicly available biological sequences. Before the extraction, the whole dataset was aligned using MAFFT [27] in its *auto* configuration (–auto). Furthermore, these subsets contained not only different number of sequences but also different lengths (number of residues per sequence). All the sequences and fragments were selected randomly with Python's random module.

Experimental setup: We developed two versions of the same algorithm for the conservation analysis. The first version of this algorithm was based on a sequential approach whereas the second one did so on a parallel approach and thus, used every available CPU. We tested both versions for all the possible combinations of methods available, both at the frequency estimation and the conservation score computation stages. We also run the tests several times to add statistical significance to our results.

The purpose of this experiment was to measure the improvement in time cost of the parallel version versus the sequential one. Hence, we wanted to estimate an upper boundary of the theoretical speedup in order to assess how close were we to the best solution possible. Given the applied divide-and-conquer technique to achieve the best parallelism implementation possible, the ideal speedup is equal to the number of cores or processors the CPU has. It is important to emphasize that this upper boundary is unreachable because we were only parallelizing the estimation of frequencies and the conservation score computation. Thus, there were still fragments of the algorithm that were sequentially executed.

All the tests were executed on an Intel(R) Core(TM) i5-4440 CPU @ 3.10GHz with 16G (2x8) DIMM DDR3 1600 MHz machine running Debian 8.1, Python 3.4 and BioPython 1.65.

Results evaluation: The performance tests carried out proved that the execution time of both approaches was more sensitive to a ten-fold increment of the sequences length (number of residues per sequence) than by the same increment in the number of sequences (See Table II and Table III). Yet, regarding to the memory requirements, this increment was almost constant regardless of the number of sequences (See Table IV and Table V).

TABLE II. EXECUTION TIME (S) OF THE SEQUENTIAL VERSION OF THE ALGORITHM USING 1 CORE.

	entropy		variance	
Dataset	unweight.	weight.	unweight.	weight.
100s_100n	$\overline{x}$ = 0.12 σ = 0.00	$\overline{x}$ = 0.13 σ = 0.00	$\overline{x}$ = 0.12 σ = 0.00	$\overline{x}$ = 0.13 σ = 0.01
100s_1000n	$\overline{x}$ = 0.18 σ = 0.03	$\overline{x}$ = 0.28 σ = 0.04	$\overline{x}$ = 0.20 σ = 0.00	$\overline{x}$ = 0.29 σ = 0.06
100s_10000n	$\overline{x}$ = 0.82 σ = 0.32	$\overline{x}$ = 1.80 σ = 0.38	$\overline{x}$ = 0.99 σ = 0.02	$\overline{x}$ = 1.92 σ = 0.56
1000s_100n	$\overline{x}$ = 0.19 σ = 0.04	$\overline{x}$ = 0.30 σ = 0.04	$\overline{x}$ = 0.20 σ = 0.01	$\overline{x}$ = 0.31 σ = 0.06
1000s_1000n	$\overline{x}$ = 0.73 σ = 0.32	$\overline{x}$ = 1.72 σ = 0.39	$\overline{x}$ = 0.91 σ = 0.03	$\overline{x}$ = 1.87 σ = 0.57
1000s_10000n	$\overline{x}$ = 6.22 σ = 3.19	$\overline{x}$ = 16.04 σ = 3.87	$\overline{x}$ = 8.08 σ = 0.22	$\overline{x}$ = 17.61 σ = 5.67
10000s_100n	$\overline{x}$ = 0.86 σ = 0.37	$\overline{x}$ = 1.93 σ = 0.42	$\overline{x}$ = 1.01 σ = 0.02	$\overline{x}$ = 2.05 σ = 0.62
10000s_1000n	$\overline{x}$ = 6.20 σ = 3.23	$\overline{x}$ = 16.10 σ = 3.87	$\overline{x}$ = 8.02 σ = 0.17	$\overline{x}$ = 17.61 σ = 5.71
10000s_10000n	$\overline{x}$ = 60.69 σ = 31.86	$\overline{x}$ = 158.84 σ = 38.37	$\overline{x}$ = 79.58 σ = 2.59	$\overline{x}$ = 175.27 σ = 56.88

These performance tests also showed, as expected, that if the dataset to analyze was rather small (100 sequences and up to 1000 residues or 1000 sequences and up to 100 residues), using the parallel version instead of the sequential one resulted in an actual degradation of performance (See Table II and Table III). This is due to the additional infrastructure required to manage the parallelism. As the number of sequences and/or the length of each one of them increased, which is to be expected in the foreseeable future, the performance improvement achieved with the parallel version of the algorithm increased. On one hand, the increase in the number of sequences is already a reality[26]. On the other hand, as the computational power keeps growing over the years, it does not seem a reckless idea to think that these algorithms could be used to analyze DNA sequences instead of mitochondrial ones—the human genome has an approximate length of 3.2 billion base pairs but researchers usually work with fragments of around 150,000 residues, which is a ten-fold increment.

As shown in Table VI the experimental speedup is really low for small datasets. In this case, the estimation of both frequencies and conservation represents a small percentage of the total execution time. In contrast, as both the number of sequences and its length grow, the algorithm spends more time with these two tasks, that is, they represent a higher percentage of the total execution time. As a result, the experimental speedup approximates the theoretical one.

B. Alignment influence on conservation analysis

Sequence datasets: In order to analyze the influence of the alignment process on the conservation analysis we used several sets of complete mtDNA sequences downloaded from GenBank. We used three source sets of 400 mtDNA sequences: the first one formed only by hmtDNA sequences, the second one including primate sequences (*pmtDNA*) but not hmtDNA, and the last one composed by mammal sequences (*mmtDNA*) but neither pmtDNA nor hmtDNA. The statistics of the three sets are shown in Table VII.

Furthermore and in order to perform an in-depth analysis, we extracted the ND2 gene and its translation to protein sequence from all the datasets, taking advantage of the biological information that GenBank provides with each sequence. Then we generated six new datasets, three formed by all the ND2 gene fragments of each dataset, and three with all the protein sequences. The statistics of this datasets are displayed in Table VIII and Table IX.

Experimental setup: For this experiment, the alignment tools were not selected based on a thorough analysis of

TABLE III. EXECUTION TIME (S) OF THE PARALLEL VERSION OF THE ALGORITHM USING 4 CORES.

	entropy		variance	
Dataset	unweight.	weight.	unweight.	weight.
100s_100n	$\overline{x}$ = 0.33 σ = 0.04	$\overline{x}$ = 0.43 σ = 0.04	$\overline{x}$ = 0.33 σ = 0.01	$\overline{x}$ = 0.43 σ = 0.06
100s_1000n	$\overline{x}$ = 0.34 σ = 0.05	$\overline{x}$ = 0.45 σ = 0.04	$\overline{x}$ = 0.33 σ = 0.01	$\overline{x}$ = 0.44 σ = 0.0
100s_10000n	$\overline{x}$ = 0.53 σ = 0.08	$\overline{x}$ = 0.83 σ = 0.11	$\overline{x}$ = 0.62 σ = 0.01	$\overline{x}$ = 0.89 σ = 0.17
1000s_100n	$\overline{x}$ = 0.33 σ = 0.05	$\overline{x}$ = 0.44 σ = 0.04	$\overline{x}$ = 0.33 σ = 0.00	$\overline{x}$ = 0.44 σ = 0.06
1000s_1000n	$\overline{x}$ = 0.45 σ = 0.09	$\overline{x}$ = 0.75 σ = 0.12	$\overline{x}$ = 0.54 σ = 0.01	$\overline{x}$ = 0.85 σ = 0.18
1000s_10000n	$\overline{x}$ = 2.21 σ = 0.95	$\overline{x}$ = 5.00 σ = 1.10	$\overline{x}$ = 2.63 σ = 0.06	$\overline{x}$ = 5.32 σ = 1.59
10000s_100n	$\overline{x}$ = 0.58 σ = 0.17	$\overline{x}$ = 1.09 σ = 0.22	$\overline{x}$ = 0.66 σ = 0.01	$\overline{x}$ = 1.19 σ = 0.30
10000s_1000n	$\overline{x}$ = 2.17 σ = 0.99	$\overline{x}$ = 5.05 σ = 1.12	$\overline{x}$ = 2.59 σ = 0.09	$\overline{x}$ = 5.39 σ = 1.65
10000s_10000n	$\overline{x}$ = 18.48 σ = 9.19	$\overline{x}$ = 45.95 σ = 10.72	$\overline{x}$ = 23.15 σ = 0.63	$\overline{x}$ = 49.79 σ = 15.82

TABLE IV. MEMORY USAGE (MB) OF THE SEQUENTIAL VERSION OF THE ALGORITHM USING 1 CORE.

	entropy		variance	
Dataset	unweight.	weight.	unweight.	weight.
100s_100n	$\overline{x}$ = 18.86 σ = 0.06	$\overline{x}$ = 18.88 σ = 0.07	$\overline{x}$ = 18.85 σ = 0.06	$\overline{x}$ = 18.88 σ = 0.02
100s_1000n	$\overline{x}$ = 19.58 σ = 0.14	$\overline{x}$ = 19.60 σ = 0.07	$\overline{x}$ = 19.29 σ = 0.06	$\overline{x}$ = 19.34 σ = 0.17
100s_10000n	$\overline{x}$ = 27.67 σ = 0.23	$\overline{x}$ = 27.80 σ = 0.80	$\overline{x}$ = 26.93 σ = 0.07	$\overline{x}$ = 26.05 σ = 0.81
1000s_100n	$\overline{x}$ = 18.86 σ = 0.07	$\overline{x}$ = 18.88 σ = 0.06	$\overline{x}$ = 18.86 σ = 0.06	$\overline{x}$ = 18.89 σ = 0.02
1000s_1000n	$\overline{x}$ = 19.59 σ = 0.14	$\overline{x}$ = 19.59 σ = 0.13	$\overline{x}$ = 19.30 σ = 0.06	$\overline{x}$ = 19.57 σ = 0.15
1000s_10000n	$\overline{x}$ = 27.99 σ = 0.23	$\overline{x}$ = 27.86 σ = 0.95	$\overline{x}$ = 27.12 σ = 0.06	$\overline{x}$ = 26.51 σ = 0.70
10000s_100n	$\overline{x}$ = 18.87 σ = 0.10	$\overline{x}$ = 18.98 σ = 0.07	$\overline{x}$ = 18.86 σ = 0.07	$\overline{x}$ = 18.95 σ = 0.06
10000s_1000n	$\overline{x}$ = 19.59 σ = 0.23	$\overline{x}$ = 19.82 σ = 0.23	$\overline{x}$ = 19.29 σ = 0.07	$\overline{x}$ = 19.82 σ = 0.26
10000s_10000n	$\overline{x}$ = 28.43 σ = 0.27	$\overline{x}$ = 28.33 σ = 1.15	$\overline{x}$ = 27.64 σ = 1.12	$\overline{x}$ = 27.62 σ = 0.44

TABLE V. MEMORY USAGE (MB) OF THE PARALLEL VERSION OF THE ALGORITHM USING 4 CORES.

	entropy		variance	
Dataset	unweight.	weight.	unweight.	weight.
100s_100n	$\overline{x}$ = 19.91 σ = 0.06	$\overline{x}$ = 19.93 σ = 0.06	$\overline{x}$ = 19.90 σ = 0.05	$\overline{x}$ = 19.93 σ = 0.02
100s_1000n	$\overline{x}$ = 20.78 σ = 0.09	$\overline{x}$ = 20.80 σ = 0.06	$\overline{x}$ = 20.62 σ = 0.06	$\overline{x}$ = 20.64 σ = 0.10
100s_10000n	$\overline{x}$ = 29.58 σ = 0.14	$\overline{x}$ = 29.62 σ = 0.86	$\overline{x}$ = 28.97 σ = 0.10	$\overline{x}$ = 27.92 σ = 0.80
1000s_100n	$\overline{x}$ = 19.90 σ = 0.07	$\overline{x}$ = 19.99 σ = 0.08	$\overline{x}$ = 19.89 σ = 0.05	$\overline{x}$ = 19.98 σ = 0.06
1000s_1000n	$\overline{x}$ = 20.78 σ = 0.10	$\overline{x}$ = 20.82 σ = 0.06	$\overline{x}$ = 20.63 σ = 0.05	$\overline{x}$ = 20.67 σ = 0.10
1000s_10000n	$\overline{x}$ = 29.65 σ = 0.12	$\overline{x}$ = 29.73 σ = 0.93	$\overline{x}$ = 29.06 σ = 0.10	$\overline{x}$ = 28.31 σ = 0.66
10000s_100n	$\overline{x}$ = 19.90 σ = 0.63	$\overline{x}$ = 21.44 σ = 0.63	$\overline{x}$ = 19.90 σ = 0.08	$\overline{x}$ = 21.44 σ = 0.89
10000s_1000n	$\overline{x}$ = 20.80 σ = 0.35	$\overline{x}$ = 21.44 σ = 0.34	$\overline{x}$ = 20.61 σ = 0.09	$\overline{x}$ = 21.43 σ = 0.43
10000s_10000n	$\overline{x}$ = 30.18 σ = 0.30	$\overline{x}$ = 30.10 σ = 1.12	$\overline{x}$ = 29.31 σ = 1.16	$\overline{x}$ = 29.33 σ = 0.48

the state of the art on the corresponding field. Instead, we chose them based on their extended usage by the scientific community and their applicability on medium-sized datasets. Hence, we used MAFFT [27], Clustal Omega [28] and Muscle [29]. We extended MAFFT with three really common and extended different parameter settings: MAFFT –auto, MAFFT –linsi and MAFFT –parttree. The first one is the default option when running MAFFT; the second performs an accuracy-oriented alignment; and the third one realizes a performance-oriented alignment.

Once we aligned all the aforementioned datasets with the different tools and parameters chosen, we generated their corresponding conservation analysis for all the different methods available. We also included different thresholds in order to provide a complete study of almost any possible scenario. The following evaluation has been made crossing all the results of the same unaligned dataset individually.

TABLE VI. EXPERIMENTAL SPEEDUP.

	entropy		variance	
Dataset	unweight.	weight.	unweight.	weight.
100s_100n	0.32	0.27	0.32	0.27
100s_1000n	0.48	0.56	0.52	0.59
100s_10000n	1.52	2.05	1.90	2.22
1000s_100n	0.51	0.60	0.54	0.61
1000s_1000n	1.47	2.03	1.83	2.19
1000s_10000n	2.98	3.40	3.30	3.52
10000s_100n	1.60	1.71	1.52	1.79
10000s_1000n	2.97	3.39	3.21	3.47
10000s_10000n	3.49	3.69	3.69	3.75

TABLE VII. STATISTICS FROM THE THREE SOURCE SETS OF COMPLETE MTDNA SEQUENCES DOWNLOADED.

Seqs.	Num. seqs.	Length mean (bp)	Length StDev (bp)
hmtDNA	400	16568.99	±2.58
pmtDNA	400	16687.24	±280.09
mmtDNA	400	16492.28	±340.54

Results evaluation: One of the first things to notice was that, as expected, when aligning sequences from a wide range of species like the ones in the primate or the mammal set, the accuracy of the alignment decreased. This was clearly noticeable looking at the original length of the complete mtDNA sequences in Table VII and then comparing them with the alignments' length in Table X; as the sequences to be aligned were less similar between each other (they belonged to organisms of different species) the number of gaps inserted increased. Regarding the nucleotides sequences of the ND2 gene, these differences were not that obvious (See Table VIII and Table XI).

This is why we performed an in-depth analysis, to take a closer look into those sets of sequences in order to find more subtle differences, and we did find them. The gaps were inserted at different positions regarding the alignment tool. Therefore, if we are interested in accuracy, we should take this into account and choose MAFFT –linsi or Clustal Omega as our alignment tool. However, if we are more concern about execution time, we should clearly choose MAFFT –auto or MAFFT –parttree, which had a 44-fold or greater improvement in execution time compared to MAFFT –linsi. These conservation analyses not only let us find out there were important differences between the alignments regarding the positions of gaps but they also showed us there were some positions where, regarding the alignment tool chosen, conservation varied. We have included some of these cases in Table XII.

One of the differences we found was that regarding the alignment tool used, some gaps were inserted in different positions. An example of this behavior was the case of pmtDNA sequences using unweighted frequencies and entropy-based conservation with a 0.99 threshold: Clustal Omega had inserted gaps in positions 264 and 265 whereas MAFFT (MAFFT –auto, MAFFT –linsi and MAFFT –parttree) had inserted gaps in positions 252 to 254. Another subtle difference was that there were positions where the conservation was really

TABLE VIII. Statistics from the ND2 gene fragments of the three source sets of mtDNA sequences downloaded.

Seqs.	Num. seqs.	Length mean (bp)	Length StDev (bp)
hmtDNA	400	1042.00	±0.00
pmtDNA	400	1042.28	±1.64
mmtDNA	400	1042.42	±2.29

TABLE IX. Statistics from the ND2 protein sequences downloaded.

Seqs.	Num. seqs.	Length mean (bp)	Length StDev (bp)
hmtDNA	400	347.00	±0.00
pmtDNA	400	346.90	±0.44
mmtDNA	400	346.95	±0.69

similar but using some alignment tools, the score was greater than the given threshold, whilst with others, the score was slightly lower than this threshold. This behavior was found, for example, while studying the results for the mmtDNA set using unweighted frequencies and entropy-based conservation with a 0.99 threshold; with MAFFT –linsi and Clustal Omega position 7 had a conservation score of 0.9895 and thus, it were not included in the appropriate report. Finally, there was another important difference: we found out that there were some sites where the conservation score had really different values. For example, in the case of mmtDNA sequences using weighted frequencies and entropy-based conservation with a 0.99 threshold, position 320 had a conservation index of 1.00 using Muscle, 0.9847 using Clustal Omega and 0.8180 using MAFFT. This last type of difference proved that the alignment tool has a really important influence when analyzing conservation of a set of biological sequences and thus, the decision of which one to use is more complex than it may seem at first glance.

V. Conclusion

We have presented a new software application to estimate the conservation index of an alignment of biological sequences. It combines some of the best-published techniques to perform a conservation analysis with the generation of readable and useful reports. We have included parallelization and divide-and-conquer techniques in the implementation process in order to improve the performance of those techniques without affecting the final accuracy. Moreover, our application is capable of handling large sequence datasets in a feasible execution time. This new software application can be executed in any OS that supports Python 3.4 and BioPython 1.65.

Besides, we have performed an in-depth study about the impact of the alignment process on the conservation analysis. We have used several sets of mtDNA sequences with different evolution distances, proving the correlation between the differences among alignment applications and parameter selection, and their conservation score.

For future improvements we aim to include new methods in all the stages involved in the conservation analysis, specifically those related with phylogeny inference processes, and more types of reports to make our tool more suitable for not considered scenarios.

TABLE X. Statistics from the three source sets of complete mtDNA sequences aligned using different tools.

Seqs.	Alignment tool	Execution Time (s)	Length (bp)
hmtDNA	MAFFT –auto	27.38	17226
	MAFFT –linsi	48912.23	17215
	MAFFT –parttree	4390.33	17231
	Clustal Omega	26133.06	17206
	Muscle	2480.37	17225
pmtDNA	MAFFT –auto	111.49	23345
	MAFFT –linsi	59695.85	23021
	MAFFT –parttree	4027.77	23780
	Clustal Omega	32445.27	23451
	Muscle	2132.29	22957
mmtDNA	MAFFT –auto	149.46	21884
	MAFFT –linsi	57716.7	21398
	MAFFT –parttree	2677.79	22105
	Clustal Omega	28439.41	21978
	Muscle	2079.46	21504

TABLE XI. Statistics from the ND2 gene of the three source sets of sequences aligned using different tools.

Seqs.	Alignment tool	Execution Time (s)	Length (bp)
hmtDNA	MAFFT –auto	6.86	1042
	MAFFT –linsi	192.11	1042
	MAFFT –parttree	3.11	1042
	Clustal Omega	146.18	1042
	Muscle	29.35	1042
pmtDNA	MAFFT –auto	4.88	1050
	MAFFT –linsi	215.52	1050
	MAFFT –parttree	7.43	1050
	Clustal Omega	160.64	1050
	Muscle	54.71	1054
mmtDNA	MAFFT –auto	6.56	1077
	MAFFT –linsi	220.08	1077
	MAFFT –parttree	3.64	1077
	Clustal Omega	155.08	1077
	Muscle	55.84	1106

Acknowledgment

Funding: The Spanish Ministry of Science and Innovation (Project TIN2011-27479-C04-01) and the Spanish Ministry of Education which provided J. Álvarez-Jarreta with a grant [FPU AP2010-1058] to carry out this investigation.

References

[1] S. Anderson, A. T. Bankier, B. G. Barrell, M. H. L. de Bruijn, A. R. Coulson, J. Drouin, I. C. Eperon, D. P. Nierlich, B. A. Roe, F. Sanger, P. H. Schreier, A. J. H. Smith, R. Staden, and I. G. Young, "Sequence and organization of the human mitochondrial genome," *Nature*, vol. 290, no. 5806, pp. 457–465, Apr. 1981.

[2] A. Carracedo, W. Br, P. Lincoln, W. Mayr, N. Morling, B. Olaisen, P. Schneider, B. Budowle, B. Brinkmann, P. Gill, M. Holland, G. Tully, and M. Wilson, "{DNA} commission of the international society for forensic genetics: guidelines for mitochondrial {DNA} typing," *Forensic Science International*, vol. 110, no. 2, pp. 79 – 85, 2000. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0379073800001614

[3] D. C. Wallace, "Diseases of the mitochondrial DNA," *Annual Review of Biochemistry*, vol. 61, no. 1, pp. 1175–1212, 1992.

[4] M. Ingman, H. Kaessmann, S. Paabo, and U. Gyllensten, "Mitochondrial genome variation and the origin of modern humans," *Nature*, vol. 408, no. 6813, pp. 708–713, Dec. 2000.

TABLE XII. NUMBER OF POSITIONS WITH A HIGHER CONSERVATION THAN THE CHOSEN THRESHOLD.

		entropy		variance	
		unweight.	weight.	unweight.	weight.
Seqs.	Alignment tool	0.99		0.50	
pmtDNA	MAFFT –auto	284	271	238	0
	MAFFT –linsi	284	271	238	0
	MAFFT –parttree	284	271	238	0
	Clustal Omega	283	270	238	0
	Muscle	287	274	242	0
mmtDNA	MAFFT –auto	374	282	436	0
	MAFFT –linsi	373	287	436	0
	MAFFT –parttree	375	283	437	0
	Clustal Omega	372	282	437	0
	Muscle	283	292	471	0

[5] D. R. Foran, "Relative degradation of nuclear and mitochondrial DNA: An experimental approach*," *Journal of Forensic Sciences*, vol. 51, no. 4, pp. 766–770, 2006.

[6] W. M. Brown, M. George, and A. C. Wilson, "Rapid evolution of animal mitochondrial dna." *Proceedings of the National Academy of Sciences of the United States of America*, vol. 76, no. 4, pp. 1967–1971, Apr. 1979. [Online]. Available: http://www.pnas.org/content/76/4/1967.abstract

[7] R. D. M. Page and E. C. Holmes, *Molecular evolution: a phylogenetic approach*, 2nd ed. Blackwell Publishing Ltd., 1998.

[8] W. S. J. Valdar, "Scoring residue conservation," *Proteins: Structure, Function, and Bioinformatics*, vol. 48, no. 2, pp. 227–241, 2002.

[9] A. Grada and K. Weinbrecht, "Next-generation sequencing: Methodology and application," *Journal of Investigative Dermatology*, vol. 133, no. 11, 2013.

[10] S.-W. Zhang, Y.-L. Zhang, Q. Pan, Y.-M. Cheng, and K.-C. Chou, "Estimating residue evolutionary conservation by introducing von neumann entropy and a novel gap-treating approach," *Amino Acids*, vol. 35, no. 2, pp. 495–501, 2008.

[11] J. A. Capra and M. Singh, "Predicting functionally important residues from sequence conservation," *Bioinformatics*, vol. 23, no. 15, pp. 1875–1882, 2007. [Online]. Available: http://bioinformatics.oxfordjournals.org/content/23/15/1875.abstract

[12] J. Carew and P. Huang, "Mitochondrial defects in cancer," *Molecular Cancer*, vol. 1, no. 1, p. 9, 2002. [Online]. Available: http://www.molecular-cancer.com/content/1/1/9

[13] M. Brandon, P. Baldi, and D. C. Wallace, "Mitochondrial mutations in cancer," *Oncogene*, vol. 25, no. 34, pp. 4647–4662, 2006.

[14] D.-J. Tan, R.-K. Bai, and L.-J. C. Wong, "Comprehensive scanning of somatic mitochondrial DNA mutations in breast cancer," *Cancer Research*, vol. 62, no. 4, pp. 972–976, 2002. [Online]. Available: http://cancerres.aacrjournals.org/content/62/4/972.abstract

[15] L. J. Burgart, J. Zheng, Q. Shu, J. G. Strickler, and D. Shibata, "Somatic mitochondrial mutation in gastric cancer." *The American Journal of Pathology*, vol. 147, no. 4, pp. 1105–1111, Oct. 1995. [Online]. Available: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1871018/

[16] W. Matsuyama, M. Nakagawa, J. Wakimoto, Y. Hirotsu, M. Kawabata, and M. Osame, "Mitochondrial DNA mutation correlates with stage progression and prognosis in non-small cell lung cancer," *Human Mutation*, vol. 21, no. 4, pp. 441–443, 2003.

[17] J. B. Jones, J. J. Song, P. M. Hempen, G. Parmigiani, R. H. Hruban, and S. E. Kern, "Detection of mitochondrial DNA mutations in pancreatic cancer offers a "mass"-ive advantage over detection of nuclear DNA mutations," *Cancer Research*, vol. 61, no. 4, pp. 1299–1304, 2001. [Online]. Available: http://cancerres.aacrjournals.org/content/61/4/1299.abstract

[18] J. A. Petros, A. K. Baumann, E. Ruiz-Pesini, M. B. Amin, C. Q. Sun, J. Hall, S. Lim, M. M. Issa, W. D. Flanders, S. H. Hosseini, F. F. Marshall, and D. C. Wallace, "mtdna mutations increase tumorigenicity in prostate cancer," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 102, no. 3, pp. 719–724, 2005. [Online]. Available: http://www.pnas.org/content/102/3/719.abstract

[19] K. Hirai, G. Aliev, A. Nunomura, H. Fujioka, R. L. Russell, C. S. Atwood, A. B. Johnson, Y. Kress, H. V. Vinters, M. Tabaton, S. Shimohama, A. D. Cash, S. L. Siedlak, P. L. R. Harris, P. K. Jones, R. B. Petersen, G. Perry, and M. A. Smith, "Mitochondrial abnormalities in alzheimer's disease," *The Journal of Neuroscience*, vol. 21, no. 9, pp. 3017–3023, 2001. [Online]. Available: http://www.jneurosci.org/content/21/9/3017.abstract

[20] T. Kadowaki, H. Kadowaki, Y. Mori, K. Tobe, R. Sakuta, Y. Suzuki, Y. Tanabe, H. Sakura, T. Awata, Y.-i. Goto *et al.*, "A subtype of diabetes mellitus associated with a mutation of mitochondrial DNA," *New England Journal of Medicine*, vol. 330, no. 14, pp. 962–968, 1994.

[21] A. Bender, K. J. Krishnan, C. M. Morris, G. A. Taylor, A. K. Reeve, R. H. Perry, E. Jaros, J. S. Hersheson, J. Betts, T. Klopstock, R. W. Taylor, and D. M. Turnbull, "High levels of mitochondrial DNA deletions in substantia nigra neurons in aging and parkinson disease," *Nature Genetics*, vol. 38, no. 5, pp. 515–517, May 2006.

[22] M. Corral-Debrinski, J. M. Shoffner, M. T. Lott, and D. C. Wallace, "Association of mitochondrial {DNA} damage with aging and coronary atherosclerotic heart disease," *Mutation Research/DNAging*, vol. 275, no. 36, pp. 169–180, 1992. [Online]. Available: http://www.sciencedirect.com/science/article/pii/092187349290021G

[23] J. Wang, H. Wilhelmsson, C. Graff, H. Li, A. Oldfors, P. Rustin, J. C. Bruning, C. R. Kahn, D. A. Clayton, G. S. Barsh, P. Thoren, and N.-G. Larsson, "Dilated cardiomyopathy and atrioventricular conduction blocks induced by heart-specific inactivation of mitochondrial DNA gene expression," *Nature Genetics*, vol. 21, no. 1, pp. 133–137, Jan. 1999.

[24] F. Johansson and H. Toh, "A comparative study of conservation and variation scores," *BMC Bioinformatics*, vol. 11, no. 1, p. 388, 2010. [Online]. Available: http://www.biomedcentral.com/1471-2105/11/388

[25] P. J. A. Cock, T. Antao, J. T. Chang, B. A. Chapman, C. J. Cox, A. Dalke, I. Friedberg, T. Hamelryck, F. Kauff, B. Wilczynski, and M. J. L. de Hoon, "Biopython: freely available python tools for computational molecular biology and bioinformatics," *Bioinformatics*, vol. 25, no. 11, pp. 1422–1423, 2009. [Online]. Available: http://bioinformatics.oxfordjournals.org/content/25/11/1422.abstract

[26] D. A. Benson, K. Clark, I. Karsch-Mizrachi, D. J. Lipman, J. Ostell, and E. W. Sayers, "Genbank," *Nucleic Acids Research*, vol. 42, no. D1, pp. D32–D37, 2014. [Online]. Available: http://nar.oxfordjournals.org/content/42/D1/D32.abstract

[27] K. Katoh, K. Misawa, K.-i. Kuma, and T. Miyata, "Mafft: a novel method for rapid multiple sequence alignment based on fast Fourier transform," *Nucleic Acids Research*, vol. 30, no. 14, pp. 3059–3066, 2002. [Online]. Available: http://nar.oxfordjournals.org/content/30/14/3059.abstract

[28] F. Sievers, A. Wilm, D. Dineen, T. J. Gibson, K. Karplus, W. Li, R. Lopez, H. McWilliam, M. Remmert, J. Söding, J. D. Thompson, and D. G. Higgins, "Fast, scalable generation of high-quality protein multiple sequence alignments using clustal omega," *Molecular Systems Biology*, vol. 7, no. 1, 2011.

[29] R. C. Edgar, "Muscle: multiple sequence alignment with high accuracy and high throughput," *Nucleic Acids Research*, vol. 32, no. 5, pp. 1792–1797, 2004. [Online]. Available: http://nar.oxfordjournals.org/content/32/5/1792.abstract