Bioinformatics, 2017, 1–9
doi: 10.1093/bioinformatics/btx421
Advance Access Publication Date: 26 June 2017
Original Paper

OXFORD

Genome analysis

CoreTracker: accurate codon reassignment prediction, applied to mitochondrial genomes

Emmanuel Noutahi[1,†], Virginie Calderon[1,†], Mathieu Blanchette[2], Franz B. Lang[3] and Nadia El-Mabrouk[1,*]

[1]Département d'Informatique et de Recherche Opérationnelle (DIRO), Université de Montréal, Montréal, QC CP 6128, Canada, [2]School of Computer Science, McGill University, McConnell Engineering Bldg., Montréal, QC H3A 0E9, Canada and [3]Département de Biochimie, Centre Robert Cedergren, Université de Montréal, Montréal, QC CP 6128, Canada

*To whom correspondence should be addressed.
[†]The authors wish it to be known that, in their opinion, the first two authors should be regarded as Joint First Authors.
Associate Editor: John Hancock

Received on March 31, 2017; revised on June 7, 2017; editorial decision on June 22, 2017; accepted on June 23, 2017

Abstract

Motivation: Codon reassignments have been reported across all domains of life. With the increasing number of sequenced genomes, the development of systematic approaches for genetic code detection is essential for accurate downstream analyses. Three automated prediction tools exist so far: FACIL, GenDecoder and Bagheera; the last two respectively restricted to metazoan mitochondrial genomes and CUG reassignments in yeast nuclear genomes. These tools can only analyze a single genome at a time and are often not followed by a validation procedure, resulting in a high rate of false positives.

Results: We present CoreTracker, a new algorithm for the inference of sense-to-sense codon reassignments. CoreTracker identifies potential codon reassignments in a set of related genomes, then uses statistical evaluations and a random forest classifier to predict those that are the most likely to be correct. Predicted reassignments are then validated through a phylogeny-aware step that evaluates the impact of the new genetic code on the protein alignment. Handling simultaneously a set of genomes in a phylogenetic framework, allows tracing back the evolution of each reassignment, which provides information on its underlying mechanism. Applied to metazoan and yeast genomes, CoreTracker significantly outperforms existing methods on both precision and sensitivity.

Availability and implementation: CoreTracker is written in Python and available at https://github.com/UdeM-LBIT/CoreTracker.

Contact: mabrouk@iro.umontreal.ca

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

The genetic code sets the rules for translating the genetic information from coding sequences (genes or mRNAs) into proteins. It was initially deciphered in the 1960s (Nirenberg *et al.*, 1963; Söll *et al.*, 1965). Using a combination of *in vitro* and *in vivo* experiments conducted on *E. coli*, the complete set of 64 codons has been mapped to either amino acids or a translation termination signal. Long seen as universally conserved among all domains of life (Crick, 1968), the discovery that human and yeast mitochondria interpret the UGA stop codon as tryptophan (Barrell *et al.*, 1979; Fox, 1979) has challenged the hypothesis of a 'frozen' universal code, revealing its evolvability. Since then, many instances of other codon

reassignments have been reported across all three domains of life, in both organelle and nuclear genomes (Keeling, 2016; Kollmar and Mühlhausen, 2017; Knight *et al.*, 2001; Ling *et al.*, 2015; Santos *et al.*, 2004; Sengupta *et al.*, 2007; Watanabe and Yokobori, 2011).

The evolution of codon reassignment requires coordinated changes at the gene/RNA sequence level and the translation machinery in charge of recognizing and assigning a given codon to an amino acid. Two main, mutually non-exclusive mechanisms have been initially proposed for natural codon reassignment. According to the *codon capture mechanism*, the codon first disappears completely from the genome due to mutational pressure, followed by the complete disappearance of the corresponding tRNA or release factor (Osawa and Jukes, 1989; Osawa *et al.*, 1990). Then, re-use of this codon by a different amino acid (AA) would emerge, in conjunction with the appearance of a corresponding tRNA and/or tRNA synthetase, with different specificities. This mechanism is present mainly in small mitochondrial genomes encoding small gene sets, for which the disappearance of a codon (leading to an unassigned codon), is feasible. The second mechanism, called *ambiguous intermediate*, does not require the disappearance of the codon during reassignment (Schultz and Yarus, 1994). Instead, it is ambiguously decoded either by a single tRNA recognized by different aminoacyl tRNA synthetases, or by two different tRNAs (or alternatively, a tRNA and a release factor). In the case of the CTG reassignment to serine in some yeast nuclear genomes, it has been hypothesized that a selective advantage could have arisen from a decoding ambiguity, gradually allowing for the reassignment of the codon (Santos *et al.*, 1999). More recently, this hypothesis has been challenged by the discovery of CUG reassignment to alanine in *Pachysolen tannophilus* (Mühlhausen and Kollmar, 2014a; Mühlhausen *et al.*, 2016; Riley *et al.*, 2016), suggesting a different mechanism (*tRNA loss driven codon reassignment*) that could explain the polyphyly of the CUG codon usage in yeasts. Sengupta and Higgs have also proposed a classification through gain and loss scenarios (Sengupta and Higgs, 2005) which integrates the *codon capture* and *ambiguous intermediate* mechanisms, in addition to their *unassigned codon* and *compensatory change* scenarios.

Computational prediction of codon reassignments is straightforward when stop codons are involved, as proteins with either premature termination or multiple additional C-terminal domains will be predicted. Sense-to-sense codon identity changes are more difficult to infer, in distinction to ongoing mutations, and when an identity switch occurs among biochemically similar AAs. This motivates the development of appropriate bioinformatics methods. To our best knowledge, three tools based on comparative sequence analyses, have been developed to predict genetic codes. The GenDecoder web server (Abascal *et al.*, 2006a), exclusively designed for metazoan mitochondrial genomes, infers codon reassignments by comparing translations of the standard protein-coding genes of a genome of interest to a set of pre-aligned reference profiles including 54 metazoans. Each codon of the input genome is assigned to the AA to which it is most frequently aligned. The second tool, FACIL (Dutilh *et al.*, 2011), which is not specific to mitochondrial genomes, aligns the sequences of interest to PFAM protein domains, then uses Random Forests (RF) to infer the most probable AA—codon match. Finally, Bagheera (Mühlhausen and Kollmar, 2014b) is a web server for predicting CUG codons reassignment to serine in yeast nuclear genomes, based on the comparison of 38 cytoskeletal and motor proteins to a reference protein dataset. CUG reassignments are predicted by comparing CUG positions within the predicted genes to the reference dataset. The first two methods are restricted to predictions that apply to the codon capture mechanism, as they do not consider the possibility of ambiguous decoding of a codon to more than one AA (Li *et al.*, 2011; Swart *et al.*, 2016; Yadavalli and Ibba, 2013). Moreover, their predictions are not validated *a posteriori* by measuring the effect of predicted reassignments on the protein alignment quality. This lack of validation, plus a high sensitivity to substitutions between close AAs, usually leads to a high rate of false positives as we show in the result section. Bagheera, on the other hand, has a validation step based on $tRNA_{CAG}$ identity prediction by comparing its sequence to a set of reference tRNAs. The main drawback of this method however is its limited scope as it concerns exclusively CUG codon reassignments to serine in yeast nuclear genomes.

Further, these methods are limited to the study of one genome at a time, completely ignoring its phylogenetic context (although Bagheera can perform an *a posteriori* phylogenetic grouping). In contrast, we argue that inferences based on the simultaneous study of multiple related genomes and their phylogenetic relatedness will lead to more sensitive predictions. Such an approach eliminates the dependency on an a priori reference set, thus allowing predictions in newly sequenced phylogenetic groups, and enabling the inclusion of non-standard proteins only shared by certain genomes. Furthermore, a phylogenetic framework can provide data for a better distinction between codon reassignments and ongoing mutations, and since codon reassignment is a progressive change, studying multiple genomes simultaneously will help identify footprints of ongoing reassignments. This innovative way of detecting codon reassignments can offer better insights toward the understanding of the underlying mechanisms of codon reassignments while systematically tracing back their evolutionary path.

Based on these ideas, we developed CoreTracker, a new algorithm exhaustively exploring sense-to-sense codon reassignments across any given group of genomes. It is the first automated approach for the prediction of codon reassignments that includes a phylogenetic framework and also extends to the context of ambiguous decoding of a codon to various amino acids.

Starting from a set of conserved positions in protein multiple alignments (derived by translating gene sequences with a given initial translation code) and a phylogenetic tree of the considered species, CoreTracker identifies candidate codons for reassignment, based on the recurring incidence of unexpected amino acids in conserved positions. Using a random forest classification approach, candidate codons are then evaluated according to a set of features related to various characteristics of the potential reassignment.

Although both use a random forest approach, CoreTracker and FACIL are significantly different. In contrast to FACIL, which is a complete parameter-free approach, CoreTracker has control over its level of precision and recall. In addition, CoreTracker integrates a correction using a similarity matrix accounting for frequent substitutions between close AAs, and a validation step, which evaluates the impact of a predicted reassignment on the alignment quality given a phylogenetic tree.

We applied CoreTracker to yeast and metazoan mitochondrial genomes and respectively predicted 54 and 85 codon reassignments. We were able to retrieve all known reassignment types and to extend them to newly analyzed genomes. On both datasets, CoreTracker achieved high precision and recall, outperforming FACIL and GenDecoder. We also compared CoreTracker to Bagheera and FACIL on a yeast nuclear dataset, on which CoreTracker also made accurate predictions for CUG codon reassignments and was even able to predict the CUG(Leu, Ala) in *Pachysolen tannophilus* missed by the other methods.

2 Materials and methods

2.1 Overview of CoreTracker

The algorithm steps are given below and illustrated on Figure 1 (see Supplementary Methods for more detailed information on each step). CoreTracker requires as input, groups of orthologous genes (nucleotide sequences) from the species of interest, and a corresponding phylogenetic tree. To identify orthologous positions, genes are first translated into proteins according to a preliminary, user-defined genetic code. They are then aligned and concatenated into a *raw alignment*, which is filtered by removing highly variable positions that may introduce noise. The obtained *filtered alignment* is used to identify potential codon reassignments. For each amino acid X_i an X_i-filtered alignment is obtained by keeping only columns in which X_i is the prevalent amino acid (Fig. 1A).

Denote by $C(X_i,\ X_j)$ the reassignment of a codon C from X_i to a different amino acid X_j . A candidate genome G for such reassignment is identified by comparing the *average* and *observed* conservation of X_j, respectively computed from the filtered and X_j-filtered alignments. An amino acid X_j that is less conserved than average in G might reflect a different genetic code. In our example (Fig. 1B), average (μ) and observed (μ_{Met}) conservation of methionine are significantly different in genomes $G2$ and $G3$. These genomes are candidates for the reassignment of isoleucine (which appear in the Met-filtered alignment) to methionine and are labeled as 'Met' on the tree (Fig. 1C). Fitch parsimony (Fitch, 1971) is then used on the species tree to trace back the history of each reassignment on the phylogeny. The analysis of the obtained history also helps recover the candidate genomes that would have been missed by steps A and B, due to an excess of filtering or a low amino acid usage.

On the other hand, an interesting reassignment from X_i to X_j in a genome G is selected according to codon usage. Codon usage is reported for all codons C encoding X_i in G (isoleucine in Fig. 1D) in two types of columns from the filtered alignment where X_i appears in G: I) columns prevalent in X_j (methionine in Fig. 1D), indicating a potential reassignment $C(X_i,\ X_j)$ in G, and II) columns prevalent in X_i (isoleucine in Fig. 1D). In a '*perfect*' dataset where C is fully reassigned from X_i to X_j, C would not be present anymore in X_i prevalent columns (type II columns). In practice, due to methodological issues (sequencing errors and alignment quality), ambiguous translation or mutations caused by true genetic divergence, the intersection between X_i and X_j prevalent columns is not expected to be

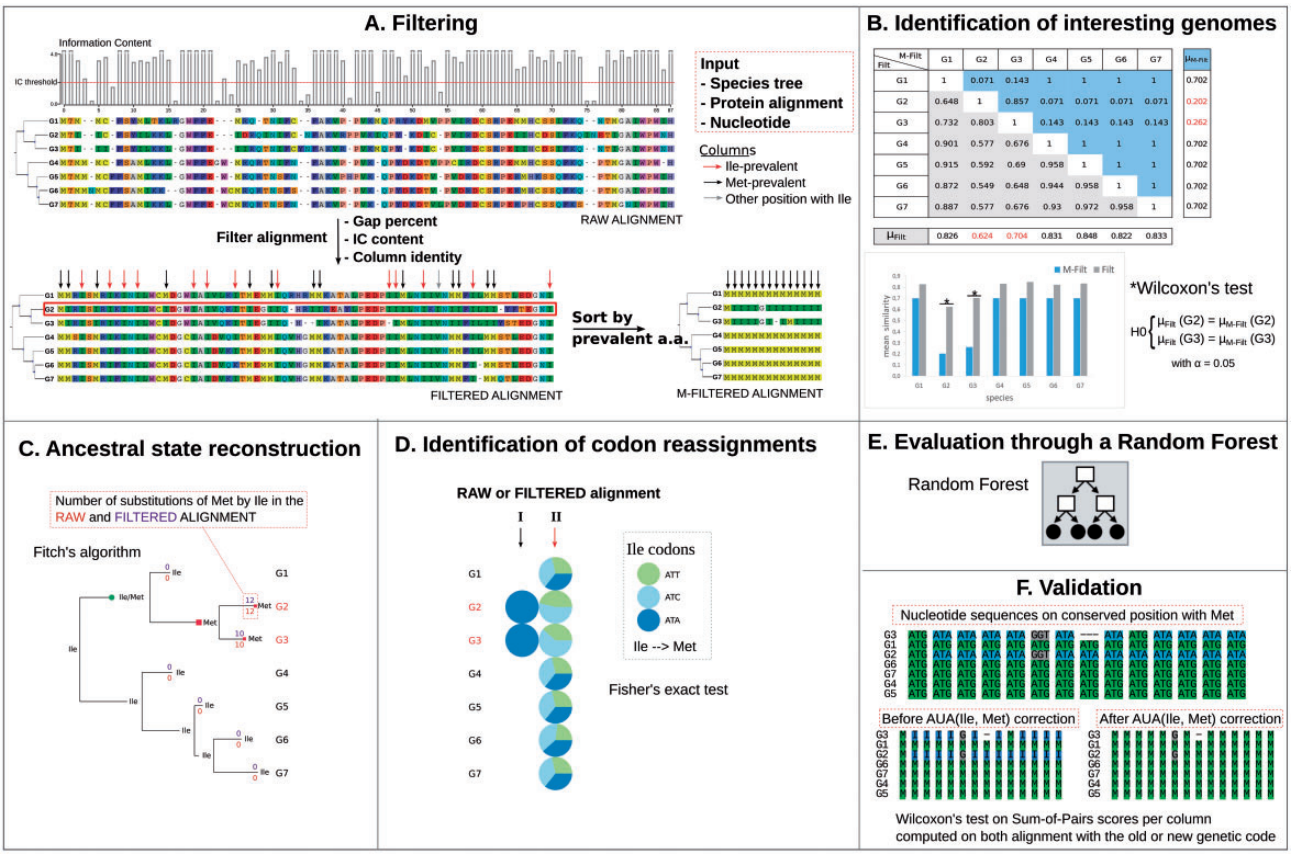

Fig. 1. Overview of CoreTracker algorithm, illustrated on a simulated example of the reassignment of AUA from isoleucine to methionine (**A**) the raw, filtered and Met-filtered protein alignments with the relationships between the input genomes. The histogram on top of the raw alignment represents information content at each position. Red, black and gray arrows on top of the filtered alignment's columns indicate respectively the Ile prevalent, Met prevalent and other columns containing Ile. The Met-filtered alignment on right is the concatenation of Met prevalent columns. (**B**) Similarity score matrix between all genome pairs, according to the filtered alignment (bottom gray part of the table), and Met-filtered alignment (top blue part), with mean values represented nearby. A Wilcoxon's signed-rank test is used to evaluate the difference between μ(Filt) and μ(Met-Filt), with stars indicating interesting genomes. (**C**) Leaves corresponding to genomes marked as interesting for a reassignment to Met (G2, G3) are labeled Met. The monophyletic group affected by such reassignment, identified by the red node, is inferred using Fitch algorithm. (**D**) Codon usage of Ile is reported in columns of the alignment (raw or filtered), prevalent in Met (I) or Ile (II). (**E**) To quantify the reliability of the predictions, a Random Forest approach is implemented using ten variables. (**F**) Sum-of-Pairs score for each column affected by the reassignment with the initial and new genetic code are computed then evaluated using Wilcoxon's test

empty. We use a generalized Fisher's exact test instead, to evaluate the difference in synonymous codon usage between the two columns. The *P*-values returned by this test are strong indicators for the identification of reassignments.

To quantify the reliability of each prediction, we implement a Random Forest (RF) approach (see Supplementary Methods). RF is a non-parametric classification algorithm (Breiman, 2001) which uses many classification trees in parallel. The features used here are described in Table 1. The RF was only trained on a set of 25 metazoan mitochondrial genomes (see Supplementary Table S1) extensively studied in the literature (Knight *et al.*, 2001; Sengupta *et al.*, 2007; Swire *et al.*, 2005), and for which we can assume that almost all reassignments have correctly identified. In order to determine the most relevant features, we measure their importance according to the Gini impurity index (Breiman *et al.*, 1984) (see Supplementary Fig. S1). From the ten selected features, the Fisher's exact test contributed the most to the predictive performance of the model. Surprisingly, the fraction of genes affected by the reassignment (Gene.fraction) and the distance to the closest reassigned node (Fitch) contributed the least.

Predictions made by the RF are then run through two validation steps that measure their impact on the proteome alignment (see Section 1.6 in Supplementary Information for more details). The first step validates the predictions per clade, while the second considers all genomes simultaneously. Both require a re-translation of gene sequences using the newly inferred genetic code. Since a reassignment shared by a whole clade is more likely to affect the same positions across all genomes in the clade, validating by clade ensures that randomly distributed sequence mutations are not inferred as codon reassignments. On the other hand, considering all genomes simultaneously reduces false predictions caused by clade-specific sequence mutations. For both validation steps, we expect an improvement of the similarity between sequences in the protein alignment for a genuine reassignment, of if the number of affected codons is significant (Fig. 1F). These validation tests are not relevant, however, for reassignments affecting too few positions in the genome.

2.2 Dataset of mitochondrial genomes

We annotated 104 genomes (see Supplementary Table S2) from a wild range of yeast mitochondrial genomes, using MFannot (http://megasun.bch.umontreal.ca/cgi-bin/mfannot/mfannotInterface.pl). The only sense-to-sense codon reassignments reported in the literature for these genomes involve codons CUN and AUA. The corresponding genetic code is number 3 in NCBI. As for the 40 metazoan mitochondrial genomes considered in our study (Supplementary Table S2), several genetic codes (2, 4, 5, 9 and 13 as referred in NCBI) are required for translation. As all these genetic codes are derived from the genetic code 4, we used it to translate the fourteen mtDNA-encoded protein (Cob, Cox1-2-3, Atp6-8-9 and Nad1-2-3-4-4L-5-6) from both the yeast and metazoan mitochondrial genomes and, when applicable, we correct frame-shifts.

Table 1. Variables used in the random forest for a reassignment $C(X_i, X_j)$ in G

Features	Description
Fitch*	Distance to the closest reassigned node
Freq.Rea*	Frequency of codon in X_j prevalent columns
Freq.Used*	Frequency of codon in X_i prevalent columns
Freq.Mixt	Frequency of codon in other columns containing X_i
Codon.usage*	Usage of the codon in genome G
Fisher.p-value*	*P*-value of the Fisher's test
Subst.count*	Number of (X_i, X_j) substitutions in the alignment
Codon.likelihood*	Telford score for the codon coding for X_j
Gene.fraction*	Fraction of genes containing the reassigned codon
G.length*	Length of the concatenated genomes
Codon.ID*	One-shot encoding representing the ID of the codon (1–64)
Suspected	Genome G is in the set of interesting genomes (0/1)

2.3 Phylogenies of metazoan and yeast species

The yeast phylogeny was constructed using thirteen standard mtDNA-encoded derived protein sequences (Cob, Cox1-2-3, Atp6-9 and Nad1-2-3-4-4L-5-6). The alignment was done in two steps. Sequences were first pre-aligned with Muscle (Edgar, 2004) and then refined with *hmmalign* from the HMMER package (Eddy, 2001). Alignments were then concatenated, resulting in a dataset that includes 104 species and has 5812 amino acid positions. The phylogenetic analysis was performed with PhyloBayes (Lartillot *et al.*, 2009), using the CAT/GTR model, six discrete categories, four independent chains, 6000 cycles and the –dc parameter to remove constant sites. The first 1000 cycles were discarded as burn-in.

The metazoan phylogeny is based on the phylogeny of Figure 4 from Sengupta *et al.* (2007). New genomes were added, while maintaining the relationships between groups (Adoutte *et al.*, 2000; Halanych, 2004).

2.4 Comparison of CoreTracker, FACIL and GenDecoder predictions on mitochondrial dataset

We compared CoreTracker, FACIL and GenDecoder predictions on the metazoan dataset. As GenDecoder is restricted to metazoan, only CoreTracker and FACIL were compared on the yeast dataset. We ran CoreTracker with default parameters and without the HMM alignment refinement step. FACIL was iteratively run on each genome in the dataset. A python script was written to query and retrieve the genetic code of each genome from the GenDecoder webserver. Since GenDecoder uses only sequence comparison to predict the genetic code, we used parameters slightly more stringent than the default (metazoan reference dataset, filtering out columns with more than 20% gap and keeping only 'highly conserved $S < 1.0$' sites according to the Shannon entropy) in order to increase precision. For all three programs, genetic code 4 was set as reference code. Non-determined predictions (marked by a '?' or '-' for GenDecoder or an 'X' for FACIL) were discarded. We kept GenDecoder's unreliable predictions (reported in lower cases) when comparing with CoreTracker and FACIL, as a sizeable proportion of its non-reassigned codons (305/2072) were reported as unreliable. This information on non-reassigned codons is hidden if we remove lower-case predictions, due to precision and recall being computed according to predicted codon reassignments only.

Due to the lack of a gold standard dataset for codon reassignments (even the NCBI annotations cannot be trusted), an initial step was to build a composite reference standard for comparison. For this purpose, literature reviews on codon reassignments in yeast and metazoan, information on species phylogenetic positions (for genomes with no reported predictions but located in a clade affected by a particular codon reassignment) and predictions shared by the three methods were considered to establish a list of true positives consisting of 90 codon reassignments of seven types in metazoans and 72

of six types in yeasts (see Supplementary Table S3). Contradictories and ambiguous cases were discarded as well as expected reassignments in genomes avoiding the codon. Since some of the genomes in the metazoan dataset were used in the training set for CoreTracker, we also assessed performance when those genomes were excluded.

3 Results

We applied CoreTracker, using default parameters, to 40 metazoan (including the 25 used as the RF training set) and 104 yeast mitochondrial genomes (see Supplementary Table S2 for a list of the genomes used, and Table S4 for predicted reassignments and each feature value). CoreTracker was also applied to 23 nuclear yeast genomes. Results on the nuclear genomes are reported and discussed in section 2.1 of the Supplementary Material (see Supplementary Table S5 and Fig. S3).

3.1 Predicted codon reassignments in metazoan mitochondrial genomes

In metazoan mitochondrial genomes, CoreTracker predicted and validated 85 potential sense-to-sense codon reassignments of eight different types (see Fig. 2), among them the seven described in the literature: AUA(Ile, Met), AGG-AGA(Arg, Ser), AGG-AGA(Arg, Gly), AAA(Lys, Asn) and AGG(Arg, Lys). The new prediction is AUA(Ile, Leu), which was predicted in five genomes (*Euseius*, *Strigamia*, *Lumbricus*, *Epigonichthys* and *Ciona*).

We were able to extend the known codon reassignments to 13 new genomes. For example, the AUA(Ile, Met) reassignment has been reported in the literature for 14 genomes (see Fig. 4 of Sengupta *et al.*, 2007). Here, the prediction was extended to five more genomes located in monophyletic groups where the reassignment has already been detected.

Regarding AGR(Arg, Ser) and AGR(Arg, Gly) reassignments, most of them were correctly inferred and extended to the remaining genomes of the considered taxonomic group. Two additional genomes, *Aphrocallistes vastus* and *Metaseiulus occidentalis*, belonging to different monophyletic groups, were predicted to both reassign AGR codons to serine and glycine. CoreTracker also predicted AGA(Arg, Ser) in *Strigamia maritima*, while the synonymous arginine codon AGG was predicted to be reassigned to lysine. This pattern of reassignment has previously been observed in *Limulus polyphemus* and some other arthropods (Abascal, *et al.*, 2006b) highlighting the fact that synonymous codons can be independently reassigned to different amino acids.

As for AAA(Lys, Asn), the four previously known reassignments were retrieved and extended to eight of the nine analyzed *Platyhelminthes* genomes. This reassignment was not predicted in *Echinococcus equinus* since its AAA codon usage is very low, and the codon is absent in asparagine predominant columns (column I). In this genome and also in *Haplorchis taichui*, AAA(Lys, Ser) was predicted but not validated (not shown) because too few codons were involved. A closer look at the alignment showed that in the positions affected by AAA(Lys, Ser), asparagine codons were found in other closely related *Platyhelminthes* genomes. This further supports AAA(Lys, Asn) reassignment in *Platyhelminthes*, under the hypothesis of an ancestral substitution from serine to asparagine in some positions.

The new identified reassignment type AUA(Ile, Leu), although supported by the validation steps, remains questionable for various reasons. Aside from the fact that the two amino acids are highly similar and frequently substituted each other, the reassignment

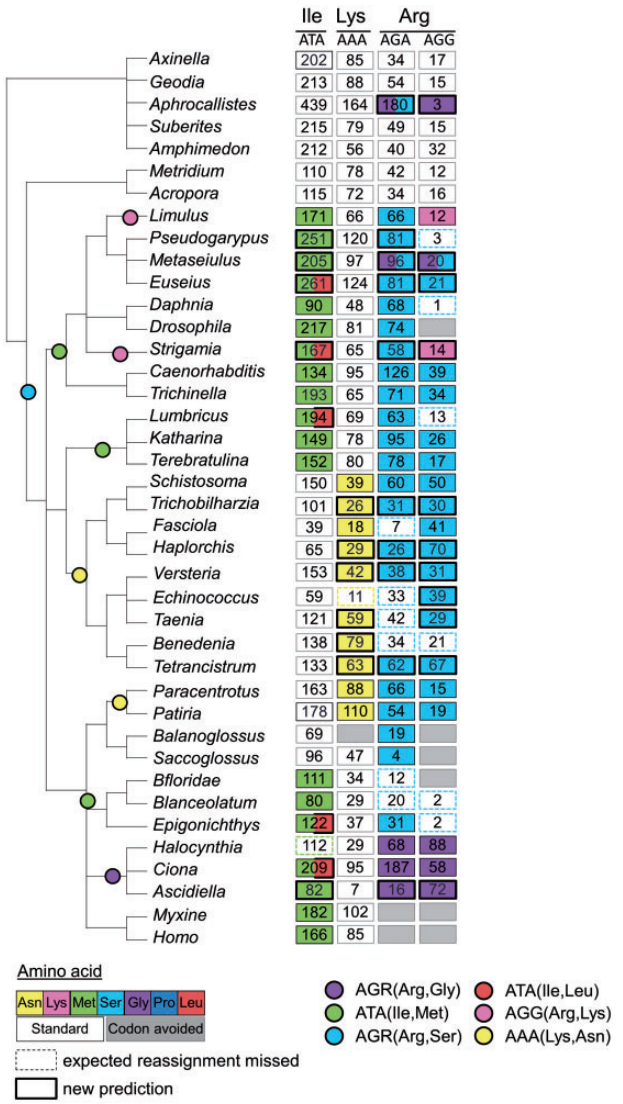

Fig. 2. Known and inferred reassignments in metazoan mitochondrial genomes. The species tree on the left is based on the literature (Sengupta et al., 2007). Numbers in rectangle indicate, for each genome, the reassigned codon (columns) count in the 13 standard mitochondrial genes

concerns five genomes that are spread apart in the phylogeny and already predicted to have reassigned AUA to methionine. Furthermore, according to the validation steps, AUA(Ile, Met) seems to improve the alignment more than AUA(Ile, Leu) (*P*-value of 1.54e-10 versus 4.06e-04). A closer look at positions affected by AUA(Ile, Leu) in the alignment shows that leucine's CUA and UUA codons are used in related genomes, suggesting sequence substitutions at the nucleotide level rather than codon reassignment. In some of these positions, we also found valine and methionine in a few genes (Cox1, Cob and Nad3). These mutations mostly occur in transmembrane domains where substitutions between hydrophobic residues is tolerated (Liu *et al.*, 2002; McClellan and McCracken, 2001), further supporting nucleotide substitutions over reassignment. However, a possible decoding of AUA codons as leucine cannot be ruled out completely without biochemical evidence.

Notice that few known AGR(Arg, Ser) and one AUA(Ile, Met) reassignments were missed. As shown in Figure 2, these missed predictions coincide with low codon usage in corresponding genomes.

It is possible that an excess of filtering plus a very low usage of these codons in conserved positions conceal the reassignment signal.

3.2 Predicted codon reassignments in yeast mitochondrial genomes

In yeast mitochondrial genomes, CoreTracker predicted and validated 54 codons reassignments of seven types (see Fig. 3). These types include known CUA-CUU-CUG(Leu, Thr), AUA(Ile, Met) and CUA-CUU(Leu, Ala) reassignments and a new AUA (Ile, Leu) reassignment.

Among the 54 inferred reassignments, ambiguity only lies for the translation of the codon AUA to either methionine or leucine in the two *Ashbya gossypii* species. As in the metazoan dataset, AUA(Ile, Leu) is most likely a false positive.

In yeast mitochondrial genomes, CUN codon reassignments from leucine to threonine were previously detected in *Saccharomycetaceae* mitochondrial genomes (Osawa *et al.*, 1990; Su *et al.*, 2011). CoreTracker predicted this reassignment for 23 of the 104 analyzed genomes. The predicted least common ancestor of all genomes affected by this reassignment is shown on the phylogeny (Fig. 3). The corresponding monophyletic group involves 26 genomes. This subgroup also contains two genomes, *Kluyveromyces lactis* and *Kluyveromyces marxianus*, where CUN(Leu, Thr) was not predicted, due to their complete lack of CUN codons in the 13 mitochondrial genes we considered.

CUU and CUA codons have been reported to be reassigned to alanine in *Ashbya gossypii ATCC10895* (Ling *et al.*, 2014). Our dataset includes two additional *Ashbya* genomes: *Ashbya gossypii FDAG1* and *Ashbya aceri*. Both CUU and CUA codons are predicted to be reassigned to alanine with a high probability (0.99), in all *Ashbya* species. Since CUC and CUG codons are avoided in most yeast mitochondrial genomes, they were often not found after filtering the alignment and were therefore rarely predicted to be reassigned. In the literature, the AUA codon was reported reassigned from isoleucine to methionine in *Saccharomyces* and *A. gossypii* mitochondrial genomes (Miranda *et al.*, 2006). Our predictions are in agreement with these results. According to the ancestral state reconstruction, such a reassignment appears to have occurred twice during the evolution of two *Saccharomycetaceae* monophyletic lineages (indicated by a green circle on Fig. 3). However, the low codon usage in *Saccharomycetaceae* genomes outside *Saccharomyces* and *Ashbya* clades (for example *Lachancea*), suggest that a single event may have affected the whole group leading eventually to AUA reassignment in certain subgroups.

Finally, as in metazoans, some expected reassignments are missed by CoreTracker due to a low codon usage in conserved coding regions. In particular, the avoidance of CUC, CUG and even CUU codons in most *Saccharomycetaceae* make the prediction and validation of their reassignments difficult.

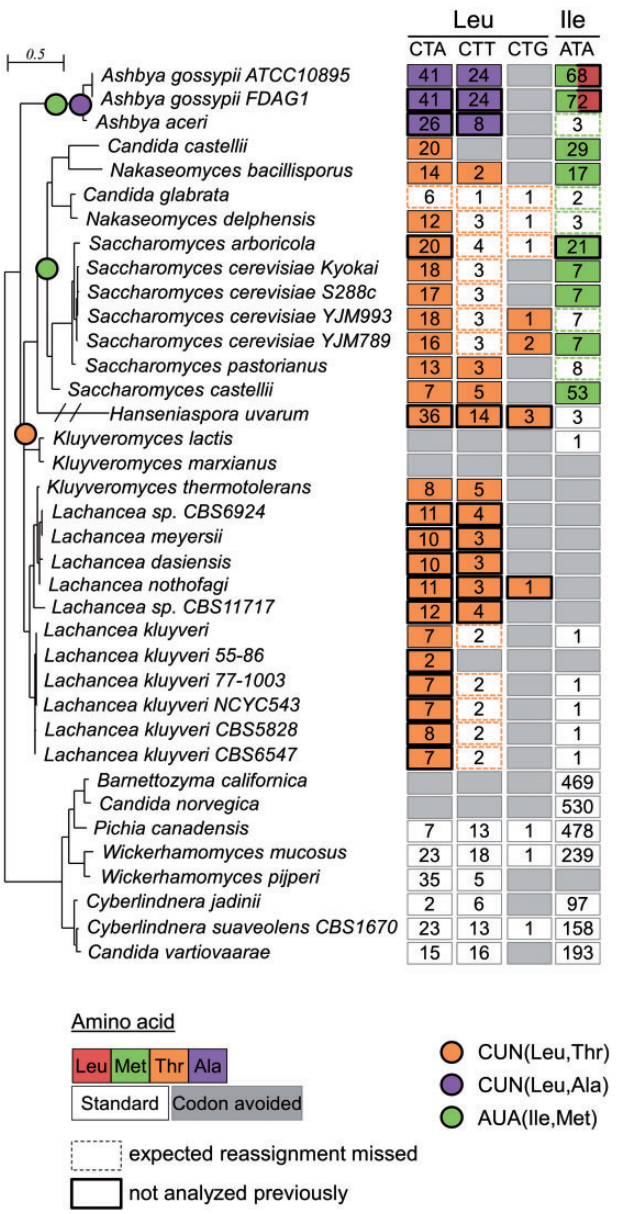

Fig. 3. Known and inferred reassignments in yeast mitochondrial genomes. The species tree on the left was computed using the 13 standard mitochondrial proteins (See Methods) and has been pruned to keep only genomes relevant to the discussion section. Numbers in rectangle indicate for each reassigned codon (columns) in each genome, the codon usage in the 13 standard mitochondrial genes. Empty grey rectangles indicate that the codon is completely avoided in the corresponding genome

3.3 Efficiency of CoreTracker compared with FACIL and GenDecoder

We compared CoreTracker to GenDecoder and FACIL on metazoan and yeast mitochondrial datasets (Table 2). The latter algorithm is more comparable to CoreTracker, as it is not restricted to any particular phylum or type of genome. The three methods were compared on a manually curated composite reference dataset (see Section 2.4 in Methods), in terms of precision, recall and F-score (harmonic mean of precision and recall). This dataset contains 90 codon reassignments of seven types in metazoans and 72 of six types in yeasts (see Supplementary Table S3).

On the metazoan mitochondrial dataset, CoreTracker achieved the highest precision (89.4%) and F1-score (86.9%), while FACIL, the closest in terms of precision has a F1-score of only 69.9%. This high precision was also achieved when genomes from CoreTracker's training set were removed. Only GenDecoder was slightly more sensitive than CoreTracker (87.8% versus 84.4%), but it had a lower precision. By removing GenDecoder's unreliable predictions (reported in lower-cases), it's precision jumps from 54.9% to 89.3%, at a cost of a decrease in sensitivity (74.4%), making it the second best tool according to the F1-Score (81.2%) (see Section 2.4 for a discussion on keeping or removing GenDecoder's lower cases). It is noteworthy that CoreTracker can achieve better recall by decreasing

Table 2. CoreTracker, FACIL and GenDecoder are compared in terms of precision (P), recall (R) and F1-score (F1) on metazoan and yeast mitochondrial genomes dataset

	CoreTracker						FACIL						GenDecoder					
	TP	FP	FN	P	R	F1	TP	FP	FN	P	R	F1	TP	FP	FN	P	R	F1
Metazoan (all)	76	9	14	0.894	0.844	0.869	64	29	26	0.688	0.711	0.699	79	65	11	0.549	0.878	0.675
Metazoan(new genomes)	31	6	10	0.838	0.756	0.795	30	13	11	0.698	0.732	0.714	33	40	8	0.452	0.805	0.579
Yeast	52	2	20	0.963	0.722	0.825	34	99	38	0.256	0.472	0.332	-	–	–	–	–	–

CoreTracker performs better than the other methods. For direct comparison, the numbers of true positives (TP), false positives (FP) and false negatives (FN) are also provided.

the stringency of the alignment filtering parameters. As for the yeast mitochondrial dataset, CoreTracker significantly outperformed FACIL by displaying precision as high as 96.3% and a recall of 72.2%, compared to FACIL with a precision of only 25.6% and a recall of 47.2%. A closer look at the two programs' predictions on the yeast dataset showed that they both have difficulties predicting reassignments involving codons with extremely low usage, such as CUU(Leu, Thr) and CUG(Leu, Thr) in *Saccharomycetaceae*, which explains the low recall. However, CoreTracker was able to detect the CUG(Leu, Thr) reassignments with only one codon count in *L. nothofagi* that FACIL missed (see Discussion). FACIL also failed to predict most AUA(Ile, Met) reassignments, whereas CoreTracker was able to predict them in 9 genomes.

In addition, the alternative comparison of Supplementary Figure S2, show that GenDecoder and FACIL predicted several unlikely codon reassignments types. In particular, GenDecoder made 48 unheard-of codon reassignment prediction types, not shared with any of the two other methods, and never reported in the literature. Although the three methods have difficulty to distinguish codon reassignments from substitutions between amino acids with similar properties due to neutral evolution, this effect is more noticeable in GenDecoder's and FACIL's prediction.

4 Discussion

CoreTracker is the first automated tool developed for codon reassignment prediction that uses a phylogenetic context. It is a generic method that allows exploring a large number of genomes from any taxonomic group, with distinct reassignment scenarios. Application to mitochondrial and nuclear genomes highlights its ability to efficiently predict known sense-to-sense reassignments.

4.1 Reassignments in metazoan and yeast mitochondrial genomes

In metazoan genomes, CoreTracker correctly predicted the five previously known and well characterized sense-to-sense reassignment types, and was able to extend them to newly analyzed genomes. Here, we discuss the unexpected AGR(Arg, Ser) and AGR(Arg, Gly) simultaneous predictions in the hexactinellid sponge *Aphrocallistes vastus* and in the arachnid *Metaseiulus occidentalis*. It has been previously suggested (Sengupta *et al.*, 2007) that AGR(Arg, Gly) is caused by the gain of a new $tRNA^{Gly}_{UCU}$ after the reassignment of AGR(Arg, Ser). This tRNA then outcompetes $tRNA^{Ser}_{GCU}$ to decode AGR codons, leading eventually to AGR(Arg, Gly). However, it is also known that in several invertebrate groups, the G base of $tRNA^{Ser}_{GCU}$ is mutated into a U, resulting in $tRNA^{Ser}_{UCU}$ which is also able to decode AGR codons, sometimes with better affinity (Abascal, *et al.*, 2006b).

Previous studies have revealed the presence of $tRNA^{Ser}_{UCU}$ in the mitochondrial genome of *Aphrocallistes* (Haen *et al.*, 2007; Rosengarten *et al.*, 2008). This tRNA has also been previously predicted in *Metaseiulus occidentalis* (Jeyaprakash and Hoy, 2007). The two genomes also have only one $tRNA^{Arg}_{UCG}$ and a single $tRNA^{Gly}_{UCC}$, supporting further the decoding of AGR codons as serine by $tRNA^{Ser}_{UCU}$. Considering these results and the fact that only a single point mutation from G to A is required to transform glycine's GGR codons into AGR codons (which can occur due to sequencing errors or sequence mutation), it can be argued that the AGR(Arg, Gly) prediction in *A.vastus* and *M.occidentalis* are most likely false positives. However, in *A. vastus*, according to CoreTracker's output, while AGR codons are completely avoided in arginine-conserved positions, AGA was found in 14 conserved serine positions compared to 16 conserved glycine positions and AGG in three glycine conserved positions. In most of those glycine positions (particularly abundant in Cox1 and Cob gene), GGN codons were found in other sponges, while the three tunicates known for AGR(Arg, Gly) used AGR codons. Moreover, AGG(Arg, Gly) was also predicted in *A. vastus* and *M. occidentalis* by GenDecoder. Further investigation through tRNA phylogenetic analysis and enzymatic study, which is beyond the scope of this paper, is thus needed.

In yeast mitochondrial genomes, CoreTracker confirmed and extended known CUA-CUU(Leu, Ala) in the *Ashbya* mitochondrial genome and CUN(Leu, Thr) in *Saccharomycetaceae* monophyletic group except *K.lactis* and *K.marxianus* in which CUN codons are absent. CUN reassignments have been extensively studied, and it has been shown that reassigned CUN codons are decoded by a $tRNA_{UAG}$ that emerges from the duplication of $tRNA^{His}_{GUG}$ then further diverges into two distinct identities to decode CUN codons as either threonine or alanine. It is believed that this reassignment is preceded by the disappearance of CUN codons, as illustrated on Figure 3 by their absence in *Kluyveromyces* and their low usage in other *Saccharomycetaceae* genomes. Although CoreTracker was able to predict most CUU and CUA reassignments, reassignments involving CUG and CUC were harder to predict, due to their extremely low usage. This low usage was expected however, since yeast mitochondrial genomes are AT-rich genomes with strong mutation pressure toward A and U. In *S. cerevisiae YJM993* and *L. nothofagi* where CUG codons appear only once, CoreTracker predicted CUG (Leu, Thr) with strong support. This reassignment was also predicted by FACIL in *S. cerevisiae YJM993* but missed in *L. nothofagi*. From CoreTracker's output, it can be observed that the leucine codon usage difference in leucine and threonine conserved columns is extremely high (*P*-value of 3.41e-10 in *S. cerevisiae* and 1.57e-07 in *L. nothofagi*). In fact UUA was almost the only codon used in leucine conserved positions, while CUA and the only CUG were found in highly conserved threonine positions. As synonymous codon usage is one of the most important feature of

the RF model, this weight a lot. Furthermore, in both genomes, the CUG codon appeared in an extremely conserved threonine positions with only a few *S. cerevisiae* using CUA while all the remaining genomes used threonine's ACA or ACU codons. Therefore, the clade validation test for CUG was successful. The second validation test was also successful since it simultaneously consider all synonymous codons predicted reassigned to the same amino acid, thereby CUG was validated alongside the more frequent CUA and CUU codons. If this second validation test were to be performed separately for each codon, there would not be enough positions for the improvement in the alignment quality to be significant (*P*-value of 2.57e-01) for CUG (Leu, Thr), and the reassignment would have failed the validation test. Note that CoreTracker has a parameter to set the minimum occurrence of a codon required before prediction, which default value is one.

As for AUA (Ile, Met), it was inferred in both *Ashbya* and *Saccharomyces*. This reassignment was previously reported to be linked to the loss of the $tRNA^{Ile}_{CAU}$ followed by a gain of function in the $tRNA^{Met}_{CAU}$ which is then able to decode both AUG and AUA codons (Sengupta *et al.*, 2007). Such hypothesis requires the avoidance of AUA codons at one point of the reassignment process. As shown on Figure 3, AUA codons are effectively either completely absent or avoided in other *Saccharomycetaceae*, suggesting that AUA(Ile, Met) in *Ashbya* and *Saccharomyces* is initiated by the loss of the gene encoding $tRNA^{Ile}_{CAU}$ in an ancestral *Saccharomycetaceae* genome.

4.2 Limitations, flexibility and possible extensions

Although CoreTracker was able to detect some reassignments of weakly used codons and reassignments occurring in single genomes (such as AGG(Arg, Lys) in *Strigamia* on Fig. 2), codon usage remains a limitation of the method. In fact, in the validation step, measuring the impact of the new genetic code on the proteome is informative only if the new genetic code affects enough positions to significantly alter the alignment quality. To overcome this limitation, prior knowledge on functional domains may be used to attribute different weights to reassigned positions according to their location in the gene. In our study, we checked the location of the predicted reassignments according to PFAM domains. However, as almost all predicted reassignments were in such domains, this step did not offer any significant filtering advantage. Alternatively, using more proteins when available, as it is the case for nuclear genomes, can help reduce the effect of codon usage limitation. However, the trade-off will be the increase in running time needed to analyze this larger dataset.

As input protein sequences are obtained from the translation of annotated genes, CoreTracker solely predicts sense-to-sense reassignments. Although this can be seen as a limitation, reassignments involving a stop codon are easily predicted by most existing annotation tools. In fact, missing C-terminal domains or proteins with abnormally long or short length, compared to others in the same family, are strong indicator of stop-to-sense and sense-to-stop codon reassignments. By default, CoreTracker also removes from the input dataset, genes with frame-shifts. Since this might not be the desired action, we provide tools to help identify and correct frame-shifts.

In order to measure the performance of CoreTracker, we evaluated its precision and sensitivity compared to GenDecoder and FACIL on mitochondrial genomes. In contrast with CoreTracker, neither GenDecoder, nor FACIL use a phylogenetic framework. Instead, each genome is analyzed independently. Applying the three algorithms on metazoan and yeast mitochondrial genomes, CoreTracker performed better than both GenDecoder and FACIL. Indeed, although GenDecoder displayed a slightly higher sensitivity than CoreTracker, both GenDecoder and FACIL demonstrated lower precision. As GenDecoder and FACIL are based on a pre-established reference datasets that do not necessarily allow an appropriate taxon sampling, both algorithms are highly sensitive to amino acid substitutions, which explains in large part their lack of precision. CoreTracker addresses this issue by offering a wide range of control over its precision, and built-in steps that ensure accuracy. Aside from the possibility to perform appropriate taxa sampling, facilitated by the provision of dataset merging tools, the added validation steps help filtering out unreliable predictions. For instance, on the metazoan dataset, 97 predictions were originally made but only 85 were validated, increasing precision by 11.06%. As CoreTracker is user oriented, it outputs several additional information (both visual and statistical) on each prediction, and even on some non-predicted reassignments in order to help users decide if a predicted (or non-predicted) reassignment should be further inquired into.

CoreTracker's efficiency on nuclear genomes was also assessed by comparing its predictions on a yeast nuclear dataset to Bagheera's and FACIL's. On this dataset, CoreTracker accurately predicted CUG codon reassignments, missing only CUG (Leu, Ser) in *L.elongisporus* (see Supplementary Fig. S3). More importantly, it was the only method able to detect the CUG(Leu, Ala) in *Pachysolen tannophilus*. This application to nuclear genomes demonstrates its consistent high accuracy and its wide range of applicability.

It is worth noticing that CoreTracker's predictions are strongly dependent upon the sequence alignment. However, as shown on Supplementary Figure S4, the method is robust even towards unrealistic high rate of errors in the alignment. Nevertheless, we provide a default alignment pipeline (see methods section) that use HMMs to refine alignments. Although CoreTracker does not require the full resolution of the input phylogenetic tree (as shown by the metazoan phylogenetic tree used) predictions depend on the considered phylogenetic tree. In particular, a reassignment affecting a whole clade will only be predicted if a genome outside this clade and missing the reassignment is included in the analysis.

One useful information that was not included in CoreTracker is the analysis of tRNAs, given that codon reassignments are often linked to changes in tRNAs. Analyzing tRNA is particularly useful when a set of reference tRNAs that have changed identity is available. In this case, it will be possible to predict reassignments by comparing the predicted tRNA of a query genome to the reference tRNA set, as done by Bagheera. This approach is suitable when predictions are restricted to a specific codon reassignment but less applicable when all potential codon reassignments are being considered as it is the case for CoreTracker. In this latter scenario, analysis of the full tRNA repertoire is necessary. This requires both a complete characterization of the tRNA identity determinants, and a reliable tRNA phylogenetic tree. Unfortunately, these information are often not available (Giegé and Frugier, 2003; Rogers and Griffiths-Jones, 2014) or difficult to reliably obtain without human intervention. Furthermore, tRNAs are not the only components involved in codon reassignments, since mutations in aminoacyl tRNA synthetases can also potentially lead to codon reassignments. Therefore, despite being linked, there is no necessary a one-to-one correspondence between tRNA evolution and codon reassignments. Although tRNA analysis could not be automatically included in CoreTracker, it still represents an important validation test that should be done when possible.

Future extensions of CoreTracker will involve accounting for branch length variation in the ancestral reconstruction step and the random forest as well. Other changes at the mRNA level that could be confounded with codon reassignments, such as RNA editing, will also be taken into account, in order to make CoreTracker a truly universal tool for codon reassignment predictions. Possible ways to achieve distinction between RNA editing, codon reassignment and artifact due to amino acid substitution, include consideration of relative codon usage between genomes, codon substitution models and more importantly tRNA phylogenetic analysis.

Acknowledgements

We thank G. Thauvette for helpful discussions. We are grateful to J. Nosek for providing his comprehensive collection of yeast mitochondrial genomes and to three anonymous reviewers for their comments and suggestions that greatly improved the article.

Funding

This work was supported by "Fonds de recherche du Québec - Nature et technologies" (FRQNT).

Conflict of Interest: none declared.

References

Abascal,F. *et al*. (2006a) GenDecoder: genetic code prediction for metazoan mitochondria. *Nucleic Acids Res*., **34**, W389–W393.

Abascal,F. *et al*. (2006b) Parallel evolution of the genetic code in arthropod mitochondrial genomes. *PLoS Biol*., **4**, e127.

Adoutte,A. *et al*. (2000) The new animal phylogeny: reliability and implications. *Proc. Natl. Acad. Sci. USA*, **97**, 4453–4456.

Barrell,B. G., Bankier,A. T. & Drouin J. (1979) A different genetic code in human mitochondria. *Nature*., **282**, 189–194.

Breiman,L. (1984) *Classification and Regression Trees*. Wadsworth International Group, Belmont, CA.

Breiman,L. (2001) Random forests. *Mach. Learn*., **45**, 5–32.

Crick,F.H. (1968) The origin of the genetic code. *J. Mol. Biol*., **38**, 367–379.

Dutilh,B.E. *et al*. (2011) FACIL: fast and accurate genetic code inference and logo. *Bioinformatics*, **27**, 1929–1933.

Eddy,S.R. (2001) HMMER: Profile hidden Markov models for biological sequence analysis. Washington University School of Medicine, St Louis, MO (http://hmmer.wustl.edu/).

Edgar,R.C. (2004) MUSCLE: a multiple sequence alignment method with reduced time and space complexity. *BMC Bioinformatics*, **5**, 113.

Fitch,W.M. (1971) Toward defining the course of evolution: minimum change for a specific tree topology. *Syst. Biol*., **20**, 406–416.

Fox,T.D. (1979) Five TGA 'stop' codons occur within the translated sequence of the yeast mitochondrial gene for cytochrome c oxidase subunit II. *Proc. Natl. Acad. Sci. USA*, **76**, 6534–6538.

Giegé,R. and Frugier,M. (2003) Transfer RNA structure and identity. In Lapointe,J. and Brakier-Gingras,L. (eds), *Translation Mechanisms*. Landes Biosciences, Georgetown, TX, pp. 1–24.

Haen,K.M. *et al*. (2007) Glass sponges and bilaterian animals share derived mitochondrial genomic features: a common ancestry or parallel evolution?. *Mol. Biol. Evol*., **24**, 1518–1527.

Halanych,K.M. (2004) The new view of animal phylogeny. *Annu. Rev. Ecol. Evol. Syst*., **35**, 229–256.

Jeyaprakash,A. and Hoy,M.A. (2007) The mitochondrial genome of the predatory mite *Metaseiulus occidentalis* (Arthropoda: Chelicerata: Acari: Phytoseiidae) is unexpectedly large and contains several novel features. *Gene*, **391**, 264–274.

Keeling,P.J. (2016) Genomics: evolution of the genetic code. *Curr. Biol*., **26**, R851–R853.

Knight,R.D. *et al*. (2001) Rewiring the keyboard: evolvability of the genetic code. *Nat. Rev. Genet*., **2**, 49–58.

Kollmar,M. and Mühlhausen,S. (2017) Nuclear codon reassignments in the genomics era and mechanisms behind their evolution. *BioEssays*, **39**, 1600221–n/a.

Lartillot,N. *et al*. (2009) PhyloBayes 3: a Bayesian software package for phylogenetic reconstruction and molecular dating. *Bioinformatics*, **25**, 2286–2288.

Li,L. *et al*. (2011) Naturally occurring aminoacyl-tRNA synthetases editing-domain mutations that cause mistranslation in Mycoplasma parasites. *Proc. Natl. Acad. Sci. USA*, **108**, 9378–9383.

Ling,J. *et al*. (2015) Genetic code flexibility in microorganisms: novel mechanisms and impact on physiology. *Nat. Rev. Microbiol*., **13**, 707–721.

Ling,J. *et al*. (2014) Natural reassignment of CUU and CUA sense codons to alanine in Ashbya mitochondria. *Nucleic Acids Res*., **42**, 499–508.

Liu,Y. *et al*. (2002) Genomic analysis of membrane protein families: abundance and conserved motifs. *Genome Biol*., **3**, research0054.

McClellan,D.A. and McCracken,K.G. (2001) Estimating the influence of selection on the variable amino acid sites of the cytochrome b protein functional domains. *Mol. Biol. Evol*., **18**, 917–925.

Miranda,I. *et al*. (2006) Evolution of the genetic code in yeasts. *Yeast*, **23**, 203–213.

Mühlhausen,S. *et al*. (2016) A novel nuclear genetic code alteration in yeasts and the evolution of codon reassignment in eukaryotes. *Genome Res*., **26**, 945–955.

Mühlhausen,S. and Kollmar,M. (2014a) Molecular phylogeny of sequenced saccharomycetes reveals polyphyly of the alternative yeast codon usage. *Genome Biol. Evol*., **6**, 3222–3237.

Mühlhausen,S. and Kollmar,M. (2014b) Predicting the fungal CUG codon translation with Bagheera. *BMC Genomics*, **15**, 411.

Nirenberg,M.W. *et al*. (1963) On the coding of genetic information. *Cold Spring Harb. Symp. Quant. Biol*.. **28**, 549–557.

Osawa,S. *et al*. (1990) Evolution of the mitochondrial genetic code III. Reassignment of CUN codons from leucine to threonine during evolution of yeast mitochondria. *J. Mol. Evol*., **30**, 322–328.

Osawa,S. and Jukes,T.H. (1989) Codon reassignment (codon capture) in evolution. *J. Mol. Evol*., **28**, 271–278.

Riley,R. *et al*. (2016) Comparative genomics of biotechnologically important yeasts. *Proc. Natl. Acad. Sci. USA*, **113**, 9882–9887.

Rogers,H.H. and Griffiths-Jones,S. (2014) tRNA anticodon shifts in eukaryotic genomes. *RNA*, **20**, 269–281.

Rosengarten,R.D. *et al*. (2008) The mitochondrial genome of the hexactinellid sponge Aphrocallistes vastus: evidence for programmed translational frameshifting. *BMC Genomics*, **9**, 33.

Santos,M.A. *et al*. (2004) Driving change: the evolution of alternative genetic codes. *TRENDS Genet*., **20**, 95–102.

Santos,M.A. *et al*. (1999) Selective advantages created by codon ambiguity allowed for the evolution of an alternative genetic code in Candida spp. *Mol. Microbiol*., **31**, 937–947.

Schultz,D.W. and Yarus,M. (1994) Transfer RNA mutation and the malleability of the genetic code. *J. Mol. Biol*., **235**, 1377–1380.

Sengupta,S. *et al*. (2007) The mechanisms of codon reassignments in mitochondrial genetic codes. *J. Mol. Evol*., **64**, 662–688.

Sengupta,S. and Higgs,P.G. (2005) A unified model of codon reassignment in alternative genetic codes. *Genetics*, **170**, 831–840.

Söll,D. *et al*. (1965) Studies on polynucleotides, XLIX. Stimulation of the binding of aminoacyl-sRNA's to ribosomes by ribotrinucleotides and a survey of codon assignments for 20 amino acids. *Proc. Natl. Acad. Sci. USA*, **54**, 1378–1385.

Su,D. *et al*. (2011) An unusual tRNAThr derived from tRNAHis reassigns in yeast mitochondria the CUN codons to threonine. *Nucleic Acids Res*., gkr073.

Swart,E.C. *et al*. (2016) Genetic codes with no dedicated stop codon: context-dependent translation termination. *Cell*, **166**, 691–702.

Swire,J. *et al*. (2005) Mitochondrial genetic codes evolve to match amino acid requirements of proteins. *J. Mol. Evol*., **60**, 128–139.

Watanabe,K. and Yokobori,S. (2011) tRNA modification and genetic code variations in animal mitochondria. *J. Nucleic Acids*, **2011**, 623095.

Yadavalli,S.S. and Ibba,M. (2013) Selection of tRNA charging quality control mechanisms that increase mistranslation of the genetic code. *Nucleic Acids Res*., **41**, 1104–1112.