coil: an R package for cytochrome C oxidase I (COI) DNA barcode data cleaning, translation, and error evaluation.

Cameron M. Nugent[1,2,*], Tyler A. Elliott[2], Sujeevan Ratnasingham[2], Sarah J. Adamowicz[1]

[1] Department of Integrative Biology, University of Guelph. Guelph, Ontario, Canada.

[2] Biodiversity Institute of Ontario, University of Guelph. Guelph, Ontario, Canada

[*]Corresponding author: nugentc@uoguelph.ca

Abstract

Biological conclusions based on DNA barcoding and metabarcoding analyses can be strongly influenced by the methods utilized for data generation and curation, leading to varying levels of success in the separation of biological variation from experimental error. The five-prime region of cytochrome c oxidase I (COI-5P) is the most common barcode gene with conserved structure and function that allows for biologically informed error identification. Here we present coil (https://CRAN.R-project.org/package=coil), an R package for the pre-processing and error assessment of COI-5P animal barcode and metabarcode sequence data. The package contains functions for placement of barcodes into a reading frame, accurate translation of sequences to amino acids, and highlighting insertion and deletion errors. The analysis of 10,000 barcode sequences of varying quality demonstrated how coil can place barcode sequences in reading frame and distinguish sequences containing indel errors from error free sequences with greater than 97.5% accuracy. The limitations of the coil's analysis pipeline were tested through the analysis of COI-5P sequences from the plant and fungal kingdoms as well as the analysis of potential contaminants: nuclear mitochondrial pseudogenes and *Wolbachia* COI-5P sequences.

Keywords: COI, DNA barcoding, error identification, translation

33 **Introduction**

34 DNA barcoding leverages sequence diversity within standardized gene regions for the
35 identification and classification of organisms (Hebert *et al.* 2003; Ratnasingham and Hebert 2007).
36 Answering questions about biodiversity through DNA barcode analyses depends on the comparison of
37 novel barcode sequences to reference libraries or the *de novo* comparison of sequences to one another
38 (Hebert *et al.* 2004; Ratnasingham and Hebert 2007; Hubert and Hanner 2015; Elbrecht *et al.* 2018).
39 Techniques such as DNA metabarcoding greatly expand the complexity of comparative analyses due to
40 the increased scale and associated challenges such as additional noise in datasets (Cristescu 2014).
41 Barcode and metabarcode output sequences can vary in terms of both length and accuracy due to a
42 mixture of true biological variation (Pentinsaari *et al.* 2016), the primers or sequencing platforms
43 utilized (Folmer *et al.* 1994; Hebert *et al.* 2018) and the data cleaning steps employed (Elbrecht *et al.*
44 2018). Defining the boundaries of the barcode region, trimming adjacent sequence information, and
45 aligning barcode information in a common context makes barcode sequences more directly
46 comparable, thereby allowing for more accurate analysis of the evolutionary history and relatedness of
47 organisms.

48 An approximately 657bp fragment of the 5' region of the cytochrome C oxidase protein (COI-
49 5P) is the main marker utilized in DNA barcoding of the animal kingdom (Hebert *et al.* 2003). COI-5P
50 is a component of the last enzyme in the electron transport chain, which is essential to metabolism
51 (Castresana *et al.* 1994; Pentinsaari *et al.* 2016). As a result, the COI-5P barcode region has particular
52 selective constraints, especially against changes to the corresponding amino acid sequence that modify
53 the protein structure and negatively affect metabolism (Tsukihara *et al.* 1995; Pentinsaari *et al.* 2016).
54 The conserved amino acid sequence corresponding to the COI-5P barcode region means that translation
55 of sequences can provide a powerful means of detecting insertion and deletion technical errors (indels)
56 that have resulted in reading frame shifts that alter the inferred amino acid profile.

57 The wealth of COI-5P sequence data available and the conserved profile of the sequence
58 provide the information necessary for pre-processing and evaluation of novel COI-5P barcode data
59 (Ratnasingham and Hebert 2007; Pentinsaari *et al.* 2016; Porter and Hajibabaei 2018). At present, the
60 comparison and evaluation of sequences in a common context can be accomplished through the *de*
61 *novo* alignment of novel sequences, but this approach grows in computational complexity with the
62 volume of data and the rate of errors in the data (Elbrecht *et al.* 2018). If systematic errors exist in
63 novel samples, for reasons such as the tendency of a given sequencing platform more often to insert or
64 delete bases in certain regions of a sequence (Hebert *et al.* 2018) or due to bias related to certain
65 sequence motifs (Schirmer *et al.* 2016), then *de novo* alignment of data may not identify systematic
66 errors and instead depict these as biological variants. Another method for processing and analyzing
67 barcode sequences is to upload them to a public repository such as BOLD (boldsystems.org)
68 (Ratnasingham & Hebert 2007). If the analyses being conducted are exploratory or still in the
69 development stage, then a researcher may not yet be prepared to upload the data to a public repository.
70 Additionally, if methods and associated results are being refined, the repeated upload of sequences to
71 BOLD for processing and error checking can be inefficient and best left as a final step in the workflow.
72 A locally available tool for the establishment of reading frame for COI-5P data could allow researchers
73 to compare barcode data from different samples efficiently in a common context. Additionally, placing
74 DNA sequences in the correct reading frame permits the translation of the nucleotide sequence to
75 amino acids, thereby providing a means of indel identification. Insertions and deletions that introduce
76 frameshifts can be easily detected through the resulting drastic changes to the amino acid sequence.
77 Here, we present a new R package for the pre-processing of COI-5P DNA barcode data: coil.
78 The package provides several functions for: (1) placement of sequences in a common reading frame,
79 (2) trimming of sequences to the COI-5P barcode region, (3) translation of nucleotides to amino acid
80 sequences, and (4) estimating the likelihood that a sequence contains an insertion or deletion error. The
81 tool is free and publically available through CRAN (https://CRAN.R-project.org/package=coil) and

82 GitHub (https://github.com/CNuge/coil). We demonstrate the effectiveness of coil by showing how it
83 can be used to align novel barcode sequences to the COI-5P profile and to identify sequences with
84 insertion or deletion errors, in most cases with greater than 97.5% accuracy.
85
86 **Implementation**
87 The coil package contains functions that allow new DNA barcode data to be placed into reading
88 frame, translated, and assessed for evidence of insertion or deletion errors (Fig. 1). The statistical
89 models the package uses to evaluate sequences are two profile hidden Markov models (PHMMs)
90 respectively trained on nucleotide and amino acid sequences of the COI-5P barcode region. Profile
91 hidden Markov models (PHMMs) are a specialized variant of a hidden Markov model well suited for
92 statistically representing biological sequences that share a common evolutionary origin (Durbin and
93 Eddy 1998; Eddy 1998; Wilkinson 2019). PHMMs capture information about the structure of
94 biological sequences, characterizing the probability of an observed character being present at the given
95 point in the sequence and the probability of transitioning from the observed character to the next
96 character in the sequence (Figs. 2, 3). The characters in the biological sequence (the nucleotide or
97 amino acid alphabets) constitute the observed characters (states), and these are related to a series of
98 unknown hidden states that further describe the characteristics of the sequence of characters (Figs. 2,
99 3). The PHMMs used within coil are meant to take a new DNA barcode sequence (the observed states)
100 and look for evidence of erroneously inserted or deleted base pairs in the sequence (the hidden states).
101 For coil, the observed character states are both the nucleotides and amino acids of the barcode sequence
102 and the hidden states are whether the given character: (1) matches the barcode profile, (2) is likely an
103 erroneous inserted character or (3) is evidence of a missing (deleted) character.
104 The coil analysis pipeline establishes the reading frame of sequences (Fig. 4) using the Viterbi
105 algorithm (Fig. 1) to compare the series of observed states (the DNA sequence) to the nucleotide
106 PHMM (Durbin and Eddy 1998; Wilkinson 2019). This algorithm determines the most likely sequence

107 of hidden states corresponding to the observed sequence by determining the optimal path through the
108 network of emission and transition probabilities, which are respectively defined as the probability of a
109 given base at a given position and the probability of a given base following the preceding base at the
110 given position. For framing of nucleotide data, the output is used to infer the presence of DNA not
111 matching the 657bp region defined by the boundaries of the model or missing information at the start
112 of the sequence (Figs. 1, 4). It is important to note that any data outside of the 657bp Folmer region
113 (upon which the model is trained) is removed and not utilized in subsequent translation or error
114 assessment; coil reports when information trimmed from the input sequence as part of this processing
115 step.

116 **Translation**

117 As a result of its ancient endosymbiotic origin, mitochondrial translation utilizes a different
118 genetic code from the nucleus, and this code additionally varies across taxa (Youle 2019). Coil is
119 therefore designed in a manner robust to the variability in animal mitochondrial genetic codes (NCBI
120 numeric identifiers: 2, 4, 5, 9, 13, 14, 21, 24, and 33)(Elzanowski and Ostell 2019). If the taxonomic
121 designation of a barcode sample is known and provided along with the input sequence, then coil calls
122 functions from the package SeqinR (https://CRAN.R-project.org/package=seqinr) to translate the
123 sequence using the appropriate genetic code (Osawa *et al.* 1992; Jukes and Opsawa 1993; Charif and
124 Lobry 2007). By default, a special censored translation option is utilized by coil in order to
125 accommodate instances when the taxonomy of a sample (and therefore the correct genetic code) is
126 unknown. Codons that vary in the amino acid they code for across known animal mitochondrial genetic
127 codes are translated to a placeholder character ('?') to indicate that the amino acid at this location in the
128 sequence cannot be stated with certainty (Fig. 5). This allows coil to assess the likelihood of the
129 sequence containing indel errors, without being biased by errors introduced due to the appropriation of
130 the wrong genetic code. As an example of why this is necessary, if a barcode sequence from a biting fly
131 (genetic code = 5) was amplified in the analysis of a mammal host specimen (proper genetic code = 2)

132 (Ratnasingham and Hebert 2007; Elzanowski and Ostell 2019) then the codon 'AGG' would be
133 interpreted as a stop codon, as opposed to the correct amino acid: serine. The presence of a stop codon
134 due to this translation error would make the resulting amino acid sequence highly improbable and lead
135 to it being flagged as likely erroneous, when it is in fact a true barcode sequence that a user may in
136 some instances wish to retain. To avoid this, censored translation outputs placeholder characters that do
137 not negatively impact the likelihood of the amino acid sequence because of mistranslated codons. The
138 censored translation outputs do however lead to a slight loss of power for subsequent error assessment
139 due to reduced sample size because placeholder characters (for 5/64 codons) are not evaluated in the
140 calculation of the likelihood score.

141 **Error assessment**

142 The inference of sequence errors is accomplished by comparison of a translated barcode
143 sequence against the amino acid PHMM (Fig. 6). The Viterbi algorithm (Fig. 1) is used to compare the
144 amino acid sequence to the PHMM and the likelihood of the optimal path that aligns the input sequence
145 to the PHMM is considered. The likelihood is expressed in logarithmic format; the higher a log
146 likelihood (all log likelihood values are negative) the more likely it is that the sequence is a match to
147 the profile. Conversely, the lower the log likelihood, the more likely the amino acid sequence is at
148 some point shifted out of reading frame due to an insertion or deletion error in the nucleotide sequence,
149 leading to a highly unlikely amino acid sequence. Low log likelihood values could result from
150 frameshifts due to technical errors (i.e. PCR amplification or DNA sequencing) or due to the detection
151 of non-barcode sequences, such as nuclear pseudogenes. Amino acid insertions or deletions (indel of a
152 full codon of nucleotides) do not result in frameshifts and lead to only minor perturbations in the log
153 likelihood score, the check of the amino sequence is therefore receptive of mutations of this kind, even
154 if they were unseen in the PHMM training data. The check for frameshifts via the PHMM comparison
155 is complemented by a simple query of the amino acid sequence for stop codons, which can provide
156 additional evidence of frameshift errors or identify instances where translation has been conducted with

157 the wrong genetic code. Since the enzyme that COI-5P codes for is highly conserved, there is an
158 extremely low likelihood of a biologically viable mutation that leads to early termination of the amino
159 acid sequence. The log likelihood of the sequence path and the query of the sequence for stop codons
160 can therefore be used in conjunction to assess sequence validity with high accuracy.
161

162 **Methods**

163 **BOLD data acquisition**

164 The Barcode of Life Data Systems (BOLD: boldsystems.org) public database was queried to obtain
165 a taxonomically representative sample of publically available COI-5P sequences. Unique COI-5P
166 haplotypes (Ratnasingham and Hebert 2013) were retained if the following criteria were met: (1) the
167 sequence was >600bp in length within the Folmer region (this was done to ensure sequences could be
168 aligned with high fidelity prior to model training) (Folmer *et al.* 1994), (2) the taxonomy of the
169 sequence was known to a genus level, (3) there were no missing bp ('N') in the Folmer region, (4) the
170 corresponding amino acid sequence did not contain stop codons, (5) the result of BOLD's internal
171 check for contaminants was negative. The query and subsequent filtering yielded approximately 1.3
172 million unique sequences (in BOLD's aligned format). To isolate the 658bp COI-5P region of the
173 sequence, data were initially analyzed in BOLD's aligned format. Sequences with information outside
174 of the COI-5P region were truncated and the placeholder dashes from BOLD's aligned format were
175 then removed from the sequences. The resulting unaligned sequence data for only the Folmer region
176 were then used in subsequent model training and assessment. This process was repeated for the amino
177 acid sequences, with only the 219 amino acids corresponding to the barcode region retained. To ensure
178 that the data used for model training and the data used for model testing and validation were
179 independent from one another, the barcode dataset was first split in a stratified fashion (on the
180 taxonomic level: family) into training (70%), test (15%), and validation (15%) sets for independent use
181 in model training, refinement, and final assessment. The taxonomically stratified split of the data was

done to ensure that the training dataset was not biased towards the largest taxonomic orders due to random chance and that subsequent models were trained on diverse data representative of the entire animal kingdom. It should be noted that some of the sequences in the BOLD dataset may be taxonomically misclassified, but that any errors of this kind would not have a significant impact on the PHMMs constructed. Training samples were used to train a single nucleotide and a single amino acid PHMM to represent the entire animal kingdom, so misclassified training data would only cause a slight change in the distribution of the taxonomic diversity of the test, validation, and training sets and not likely result in a significant impact on model performance.

Model training

A nucleotide and amino acid profile hidden Markov model (PHMMs) were respectively trained on COI-5P nucleotide and amino acid training sequences (Supplementary File 1). The models were not trained on the entire set of 936,971 barcodes in the training set; rather, in an attempt to prevent model overfit, different sized taxonomically representative subsamples of the full set of training data were used to train competing PHMMs. The full training set was subsampled in a stratified fashion (on the taxonomic level: family), and any families with fewer than 10 samples in the training set had all their representatives added to each subset to ensure that rare families were represented in model training. The two taxonomically representative training sets of ~1.2% (11,391 sequences) and ~3.0% (28,189) of the training data were produced and each was subjected to the following procedure. Training of the nucleotide and amino acid PHMMs was done using the R package aphid (Wilkinson 2019). Model training was conducted using aphid's derivePHMM function, utilizing the 'background' option for the pseudocounts parameter (see Durbin and Eddy 1998 – Chapter 5 for explanation of pseudocounts). Initially, models were trained using two 'maxsize' parameter pairs (a) a nucleotide PHMM maxsize of 657 and a corresponding amino acid maxsize of 219 and (b) a nucleotide PHMM maxsize of 673 and a corresponding amino acid maxsize of 224. The 'maxsize' parameter limits the upper bound of the number of modules in the PHMM; i.e. it provides the maximum length of the profile that represents the

207 sequences. Alignment of test data to the models via the Viterbi algorithm revealed that when the larger
208 maxsize parameter was utilized, the PHMM expanded to the maximum size permitted. In subsequent
209 testing this led to variable framing of sequences (dependent on taxonomy) and made the establishment
210 of a common reading frame infeasible. The smaller maxsizes of 657/219 yielded a set of PHMMs that
211 could effectively establish the reading frame of taxonomically diverse sequences in the test data and
212 this pair of PHMMs was therefore retained as the internal PHMMs for coil, against which new barcode
213 sequences are compared. Of the two training data sizes that were used, the PHMMs resulting from
214 training on the smaller training set (11,391 sequences) were found to perform marginally better on
215 initial tests, both in terms of the framing and error classification of sequences. Therefore, the nucleotide
216 and amino acid PHMMs trained on 11,391 training sequences were retained for internal use in the coil
217 package.
218
219 **Assessment of coil package performance**
220 Following successful model training and package construction, a random subsample of 10,000
221 barcode sequences from the previously withheld validation data was used to assess the performance of
222 coil, in terms of its ability to place sequences in the correct reading frame and determine if sequences
223 contain insertion or deletion errors. Additionally, we aimed to test coil's establishment of reading frame
224 and proper translation when excess sequence information in addition to the COI-5P region is present or
225 when barcode information is missing from the edges of sequences. The 10,000 samples were randomly
226 split and altered using a custom python script (Supplementary File 2) to construct the following 5
227 groups (2000 sequences each):
228 (a) No changes made.
229 (b) 1-100 random base pairs added to the front and/or back of the sequence, with equal addition
230 probabilities for each of the four nucleotides.
231 (c) 1-100 base pairs removed from the front and/or back of the sequence.

232 (d) Altered with a point mutation probability of 1% per bp (a 1% chance of a nucleotide being
233 changed to any other nucleotide) and a random indel probability of 1% per bp (equal
234 probability of either an insertion or deletion, in the case of insertions a random bp was
235 inserted).
236 (e) Sequences either shortened or elongated in the manner of groups (b) or (c), with errors then
237 introduced in the manner of group (d).
238 These validation data were then loaded into R and passed through the coil analysis pipeline
239 twice (Supplementary File 3). First, the translation step of the coil pipeline was run using the genetic
240 codes known to correspond to the sequences and second, the pipeline was run using the default
241 behaviour, assuming unknown genetic codes and utilizing the censored translation option.
242 The outputs of the pipeline were assessed to determine how well sequences were framed and
243 translated. The package's ability to frame sequence was evaluated by looking for stop codons in the
244 translation outputs of the 6000 validation sequences with no artificially introduced indel errors. The
245 presence of stop codons would indicate a sequence was not set in reading frame properly (as their
246 original amino acid sequences are known and do not contain stop codons), whereas the absence of stop
247 codons is strongly suggestive of correct translation. Additionally, the log likelihood values obtained
248 from comparison of all 10,000 sequences to the amino acid PHMM were assessed to determine the log
249 likelihood's effectiveness as an error classifier. Sequences with introduced indel errors and those
250 without errors were labelled in a binary fashion. The R package ROCit (CRAN.R-
251 project.org/package=ROCit) was used to determine the ability of the amino acid PHMM output log
252 likelihood values to discern sequences with errors from those without errors. Comparisons were made
253 between the following groups of validation data: full length sequence data ((a) vs (d)), variable length
254 sequence data ((b) and (c) vs (e)), and all sequence data ((a)(b)(c) vs. (d)(e)). To determine the effect of
255 sequence length on classification ability, the long sequences (b) were compared to the sequences from
256 the variable length error dataset (e) that were elongated, and the short sequences (c) were compared to

257 the sequences from the variable length error dataset (e) that were shortened. The area under the curve
258 (AUC – a metric of how well the log likelihood values separate the two categories) and the optimal F
259 score threshold (the log likelihood value that optimizes the trade-off between false positive and false
260 negative classifications) were determined for all comparisons.

261 **Processing non-animal barcode sequences**

262 In addition to identifying the presence of indels in animal COI-5P sequences (which would most
263 likely result from technical errors), we tested how coil performed on COI-5P sequences for which it
264 was not designed to process. A series of plant, fungi, and bacterial COI-5P sequences as well as animal
265 nuclear mitochondrial pseudogenes were analyzed with coil, to see if sequences from these categories
266 would be flagged as likely containing errors, or if they would be processed in the same manner as
267 animal COI-5P sequences. The first set of sequences investigated were COI-5P sequences from the
268 plant and fungi kingdoms. The NCBI nucleotide database (https://www.ncbi.nlm.nih.gov/nucleotide/)
269 was queried for 'cytochrome c oxidase 1' and the results were filtered to retain sequences from only
270 plants and fungi. Results were filtered by sequence length (minimum 550bp, maximum 2000bp) and
271 the sequence descriptions were all manually checked to ensure that all results were COI-5P sequences.
272 This resulted in a set of 570 plant and 174 fungal sequences (Supplementary File 4) that were then
273 analyzed using the coil analysis pipeline (coi5p_pipe function) with default parameters.

274 *Wolbachia* is a genus of bacterial endosymbionts (commonly found in arthropods) that can in
275 some instances confound attempts to isolate an animal specimen's barcode sequence (Smith *et al.*
276 2012). To see if coil would flag *Wolbachia* sequences as errors (either due to a low likelihood value or
277 the presence of a stop codon in the amino acid sequence), data associated with the accession numbers
278 of *Wolbachia* samples listed in Supplementary File S2 of Smith *et al.* (2012) were obtained from NCBI.
279 Data were filtered to omit sequences <550bp in length, yielding 345 *Wolbachia* sequences that were
280 analyzed via coil's 'coi5p_pipe' function (parameters: trans_table = 0 and indel_threshold = -440.24).

281 Nuclear mitochondrial pseudogenes (numts) are non-encoded sequences in the nucleus that are
282 the product of mitochondrial DNA fragments being incorporated into the nuclear genome (Bensasson *et*
283 *al.* 2001). Numts resulting from the COI-5P gene have been shown to interfere with accurate
284 assessment of the COI-5P sequences for a species complex of Australian stingless bees (*Tetragonula*
285 *carbonaria*, *Tetragonula davenporti*, *Tetragonula hockingsi*, and *Tetragonula mellipes*; hereafter the
286 'Carbonaria group'.) (Françoso *et al.* 2019). Sequence data for the 9 numts with reported accession
287 numbers in Figure 7 of Françoso *et al.* (2019) were obtained and analyzed with coil (using the
288 coi5p_pipe function and default parameters) to see if the package identified these sequences as likely
289 being erroneous.
290
291 **Results & Discussion**
292 **Framing and Translation**
293 The validation data outputs were assessed to determine how effective coil was at processing of
294 variable-length barcode data. The package's ability to set sequences in reading frame was evaluated by
295 looking for stop codons in the translation outputs of the 6000 validation sequences with no introduced
296 indel errors. If stop codons were present, the sequence would not have been set in reading frame
297 properly, whereas the absence of stop codons is strongly suggestive of a correct reading frame
298 establishment. Of the 6000 validation sequences without introduced errors, 5888 (98.13%) were
299 correctly framed. The sequences with incorrectly identified reading frames were not evenly distributed
300 across the three sequence length categories (short sequence, full-length sequences, and long
301 sequences). All barcode sequences in the full-length (a) category except for one were correctly placed
302 in reading frame, indicating that the nucleotide PHMM is able to recognize and frame taxonomically
303 variable barcode data and that it is not biased against certain taxonomic groups. The program appeared
304 to be effective at trimming artificially lengthened sequences (category b) and placing them in reading
305 frame, with only 38 framing errors in the 2000 long sequences (98.1% accuracy). The program was

306 slightly less effective at placing shortened sequences (category c) in reading frame, with 98 errors
307 observed (95.1% accuracy). Despite this imperfect performance, the program is effective at framing the
308 vast majority of sequences and can aid the user by successfully automating this task for most inputs
309 (Supplementary File 5). The summary data provided by the indel check step (stop codon presence and
310 error likelihood values) can aid users in targeting additional manual correction effort to the subset of
311 sequences coil is unable to place in reading frame.

312 **Indel Error Identification**

313 Across the entire validation dataset, coil's indel_check function was able assess the amino acid
314 data and accurately discern sequences with insertion or deletion errors from correct barcode sequences
315 with greater than 97% accuracy. As with the framing of sequences, coil's effectiveness was influenced
316 by the length of the sequences (Table 1). When the full-length barcode sequences with and without
317 introduced errors were evaluated, they could be categorized with greater than 99.9% accuracy. For
318 sequences with excess DNA sequence added to the ends, the sequences with errors could be discerned
319 from the correct sequences with 98.0% accuracy. For the short sequences, performance was once again
320 the lowest, with 95.1% classification accuracy (Fig. 7). The similarity in these accuracy scores to the
321 percentage of sequences placed in reading frame correctly is not coincidental, but rather due to the fact
322 that the ability to correctly classify a sequence is highly dependent on the ability of coil to place it in
323 the common reading frame correctly.

324 Closer examination of the classification errors in the short sequence dataset (the category with
325 the highest error frequency) helped to characterize the limitations of coil's classification ability. The
326 majority of the classification errors (103/132 – 78%) were false negatives, i.e. sequences without
327 introduced errors that were flagged as likely to contain errors. The reason for this appears to be the
328 incorrect establishment of the reading frame and thus large-scale reading frame shifts, resulting in
329 incorrect amino acid sequences. The major limitation of coil therefore appears to be the inability to
330 always establish the reading frame properly, a problem that is magnified in short sequences where the

331 front of the sequence is truncated. The examination of the 29 falsely accepted sequences for the short
332 category revealed the three main patterns where coil failed to detect errors in a sequence: 1) When an
333 indel occurs late (>500 bp) in the sequence (observed for 7 sequences), 2) when three insertions or
334 three deletions are found in closely proximity (observed for 4 sequences), 3) when an insertion and
335 deletion occur in close proximity to one another (observed for 18 instances). All three of these patterns
336 do not lead to large-scale alterations to the amino acid sequence, and are therefore not easily detected
337 by examination of the amino acid sequence.

338 In all error classification tests, the accuracy (as shown by AUC) was highly similar when
339 translation was conducted with the known genetic codes and when censored translation was employed
340 (Table 1). This indicates that the censored translation method is effective at allowing error
341 identification to proceed with accuracy in instances where the taxonomic origin of a sequence is
342 unknown. It should be noted that the log likelihood values associated with sequences are overall
343 slightly lower when generated using censored translation, because lower numbers of amino acids
344 contribute to the likelihood value(5/64 codons are not translated – Fig. 5). For this reason, the threshold
345 used to identify sequences with indel errors should be adapted to the characteristics of the data being
346 analyzed. Table 1 provides users with a starting point for selecting a classification threshold, showing
347 the optimal log likelihood threshold (as determined through F-score maximization) for the different
348 comparisons of validation data. These values serve as statistically informed starting points from which
349 a user can deviate if they wish to attempt minimization of type 1 (lowering the logL threshold to
350 minimize rejection of correct sequences) or type 2 error (raising the logL threshold to minimize
351 acceptance of sequences with indel errors.

352

353 **Package limitations**

354 The results of processing non-animal COI-5P sequences with coil were variable; 167/174 (96%)
355 of fungi and 336/570 (58.8%) of plant sequences were flagged as likely erroneous based on the

356 resulting log likelihood scores and/or the presence of stop codons. The reason for coil labelling fungi
357 sequences as likely erroneous is the existence of introns in fungal COI-5P sequences (Avin *et al.* 2017).
358 Fungal sequences show drastic variability in length due to the presence of introns, compared to animal
359 sequences which do not contain introns and are relatively consistent in length (Avin *et al.* 2017). This
360 variation in the structure of fungal sequences is interpreted by coil's processing pipeline as evidence of
361 indel errors, resulting in the majority of true fungal sequences being flagged as erroneous. Plant COI-
362 5P sequences are less variable in structure than fungi, but more than half of the sequences tested
363 deviated from the animal COI-5P profile in a manner extreme enough for them being flagged as
364 erroneous. These results show that coil cannot be used to reliably pre-process and filter errors from
365 non-animal COI-5P sequences. The PHMMs are not trained to accommodate non-animal data and
366 processing sequences from the plant or fungal kingdoms results in a large number of false negatives. A
367 future extension of coil can be the addition of PHMMs trained on data from different kingdoms or
368 barcodes, which would allow coil's processing pipeline to be applied in error filtration of other protein
369 coding, intron-free barcodes.

370 Of the 345 *Wolbachia* COI-5P sequences analyzed, only five were flagged as likely to be
371 erroneous and only a single sequence contained stop codons. The five instances of *Wolbachia*
372 sequences being flagged as errors resulted from the incorrect framing and translation of sequences.
373 Since the *Wolbachia* sequences are real instances the COI-5P gene (albeit from bacteria as opposed to
374 animals) that do not possess any drastic changes to reading frame relative to animals, their nucleotide
375 and corresponding amino acid sequences are likely enough to be accepted as matches to the animal
376 COI-5P PHMMs. This demonstrates that coil cannot be effectively appropriated to separate animal
377 COI-5P sequences from the COI-5P sequences of endosymbiote contaminants.

378 All 9 of the numt sequences from the Carbonaria group analyzed with coil were flagged as
379 likely to contain an error. This was accomplished despite the fact that none of the translated sequences
380 contained a stop codon. Each pseudogene possessed an amino acid sequence that the program

381 characterized as improbable (all log likelihood values lower than -754.00). It should be noted that the
382 numts analyzed were shorter sequences (333bp each), which inherently possess lower log likelihood
383 values. However, animal COI-5P sequence fragments of this length tend to yield log likelihood values
384 of approximately -450.00, so the drastically lower log likelihood sequences are sufficient to distinguish
385 the pseudogene sequences from COI-5P sequences. The number of numts tested was low and limited to
386 sequences identified within a single species complex. Expansion of these tests to COI-5P-derived
387 numts from a more diverse set of taxa could reveal how well coil is able to characterize numts. At
388 present, it is hypothesized that the ability to discern pseudogenes from true barcode sequences will
389 depend on how diverged the structure and sequence of a pseudogene are, with frame-shift-causing
390 indels increasing the likelihood of detection by coil.

391 Taken together, these results show that coil Version 1.0 should only be used to process animal
392 COI-5P data and that it is less reliable at detecting biological contaminants than technical errors. The
393 genetic structure of biological contaminants such as *Wolbachia* COI-5P is not always dissimilar enough
394 from animal COI-5P sequences for the PHMMs to discern the two categories. Additionally, if an
395 animal specimen were contaminated with DNA not from another kingdom, but rather with DNA from
396 another animal then detection with coil in most instances would not be possible, as the models
397 employed by the package are trained on a taxonomically robust sample of animal sequences. The
398 results presented here regarding the identification of biological contaminants are conservative, as in all
399 tests we have used coil's default censored translation option, which assumes that taxonomic
400 information is not available. The ability to identify biological contaminants can be greatly increased if
401 the taxonomy (and therefore the genetic code) corresponding to a sample is known. Contaminants from
402 organisms that appropriate a different genetic code would have a higher likelihood of being identified
403 in this situation, as translation would be more likely to yield amino acid sequences with stop codons or
404 highly improbable amino acids. Overall the coil package should not be considered a panacea for all
405 DNA barcode data errors, but rather as a tool that can be incorporated into barcode analysis as a

406 reliable pre-processing and data-cleaning step that can identify technical errors. Researchers using
407 barcode data should consider additional steps in their analyses, such as BLAST, which can help inform
408 the taxonomy of sequences and identify biological contaminants.
409
410 **Conclusion**
411 We have presented the R package coil and demonstrated how it can effectively set COI-5P
412 barcode data in reading frame, translate sequences in a taxonomically robust fashion and identify
413 sequences containing frame-shifting indel errors with greater than 97% accuracy. A vignette
414 demonstrating the software's functionality and suggested workflows is available within the package
415 (and included here as well – Supplementary File 6). The package can facilitate exploratory analysis of
416 barcode data and aid in data cleaning and error assessment. At present, the package's functionality is
417 limited to the main DNA barcode region for animals, COI-5P. In the future, we hope to extend the
418 package's functionality to additional DNA barcode regions, including alternative animal barcodes like
419 cytochrome b (Kochzius *et al.* 2010; Yacoub *et al.* 2015) and 16S rRNA (Vences *et al.* 2005), as well
420 barcodes employed in analysis of species from other kingdoms (Hollingsworth *et al.* 2009;
421 Hollingsworth *et al.* 2011). Adding this functionality would require the training of addition profile
422 hidden Markov models for these barcode regions and integration of these models into the coil package.
423 This is a feasible addition thanks to the large amount of additional barcode data available through
424 BOLD (Ratnasingham & Hebert 2007) and other protein-coding markers available through NCBI,
425 which can be queried and filtered using methods similar to those employed here for COI-5P. Another
426 extension of the functionality of the coil package is to move from error identification to error
427 correction. PHMMs are robust models of the conserved sequence of the COI-5P barcode region on both
428 a DNA and amino acid level. Accurately identifying where deviations from the expected sequence
429 occur and applying corrections to the sequences in an unbiased fashion would allow PHMMs to serve
430 as the basis of a DNA barcode data denoiser. Corrections based on PHMMs trained on the wealth of

431 barcode sequence information available could serve as a powerful complement, or alternative, to

432 current denoising methods which rely on read abundance and *de novo* clustering of sequences (Nearing

433 *et al.* 2018).

Acknowledgements

Funding for this research was provided by grants in Bioinformatics and Computational Biology from the Government of Canada through Genome Canada and Ontario Genomics and from the Ontario Research Fund. Funders played no role in software design or manuscript preparation. Thank you to Samantha E. Majoros for aiding in the initial testing of the coil package.

References

Avin, F.A., Subha B., Tan Y.S., Braukmann T.W., Vikineswary S., Hebert P.D. 2017. Escaping introns in COI through cDNA barcoding of mushrooms: *Pleurotus* as a test case. Ecology and Evolution. **7**(17):6972-80. doi: 10.1002/ece3.3049.

Bensasson D., Zhang D.X., Hartl D.L., Hewitt G.M. 2001. Mitochondrial pseudogenes: evolution's misplaced witnesses. 2001. Trends in Ecology & Evolution. **16**(6):314-21. doi: 10.1016/s0169-5347(01)02151-6.

Calderón-Sanou I., Münkemüller T., Boyer F., Zinger L., Thuiller W. 2019. From environmental DNA sequences to ecological conclusions: How strong is the influence of methodological choices?. Journal of Biogeography. doi: 10.1111/jbi.13681.

Castresana J., Lübben M., Saraste M., Higgins D.G. 1994. Evolution of cytochrome oxidase, an enzyme older than atmospheric oxygen. The EMBO journal. **13**(11):2516-25. doi: 10.1002/j.1460-2075.1994.tb06541.x

Charif D., Lobry J.R. 2007. SeqinR 1.0-2: a contributed package to the R project for statistical computing devoted to biological sequences retrieval and analysis. In Structural approaches to sequence evolution 2007 (pp. 207-232). Springer, Berlin, Heidelberg. Doi: 10.1007/978-3-540-35306-5_10

Durbin R., Eddy S.R., Krogh A., Mitchison G. 1998. Biological sequence analysis: probabilistic models of proteins and nucleic acids. Cambridge University Press, Cambridge.

Eddy S.R. 2009. A new generation of homology search tools based on probabilistic inference. In Genome Informatics Series. Vol. 23 pp. 205-211.

Elbrecht V., Vamos E.E., Steinke D., Leese F. 2018. Estimating intraspecific genetic diversity from community DNA metabarcoding data. PeerJ. **6**:e4644. doi: 10.7717/peerj.4644.

Elzanowski A., Ostell J. 2019. The Genetic Codes [online]. Available from https://www.ncbi.nlm.nih.gov/Taxonomy/Utils/wprintgc.cgi#SG5. [accessed Sept 9 2019].

Françoso E., Zuntini A.R., Ricardo P.C., Silva J.P., Brito R., Oldroyd B.P., Arias M.C. 2019. Conserved numts mask a highly divergent mitochondrial-COI gene in a species complex of Australian stingless bees *Tetragonula* (Hymenoptera: Apidae). Mitochondrial DNA Part A. **16**:1-2. doi: 10.1080/24701394.2019.1665036.

Folmer O., Black M., Hoeh W., Lutz R., Vrijenhoek R. 1994. DNA primers for amplification of mitochondrial cytochrome c oxidase subunit I from diverse metazoan invertebrates. Molecular Marine Biology and Biotechnology. **3**(5):294-9. PMID: 7881515.

Hebert P.D., Cywinska A., Ball S.L., deWaard J.R. 2003. Biological identifications through DNA barcodes. Proceedings of the Royal Society of London. Series B: Biological Sciences. **270**(1512):313-21. doi: 10.1098/rspb.2002.2218.

Hebert P.D., Penton E.H., Burns J.M., Janzen D.H., Hallwachs W. 2004. Ten species in one: DNA barcoding reveals cryptic species in the neotropical skipper butterfly *Astraptes fulgerator*. Proceedings of the National Academy of Sciences. **101**(41):14812-7. doi: 10.1073/pnas.0406166101.

Hebert P.D., Braukmann T.W., Prosser S.W., Ratnasingham S., deWaard J.R., Ivanova N.V., Janzen D.H., Hallwachs W., Naik S., Sones J.E., Zakharov E.V. 2018. A Sequel to Sanger: amplicon sequencing that scales. BMC Genomics. **19**(1):219. doi: 10.1186/s12864-018-4611-3.

Hollingsworth P.M., Forrest L.L., Spouge J.L., Hajibabaei M., Ratnasingham S., van der Bank M., Chase M.W., Cowan R.S., Erickson D.L., Fazekas A.J. 2009. A DNA barcode for land plants. Proceedings of the National Academy of Sciences. **106**(31):12794-7. doi: 10.1073/pnas.0905845106.

Hollingsworth P.M., Graham S.W., Little D.P. 2011. Choosing and using a plant DNA barcode. PloS One. **6**(5):e19254. doi: 10.1371/journal.pone.0019254.

Hubert N., Hanner R. 2015. DNA barcoding, species delineation and taxonomy: a historical perspective. DNA Barcodes. **3**(1):44-58. doi: 10.1515/dna-2015-0006.

Jukes T.H., Osawa S. 1993. Evolutionary changes in the genetic code. Comparative Biochemistry and Physiology. B, Comparative Biochemistry. **106**(3):489-94. doi: 10.1016/0305-0491(93)90122-l.

Kochzius M., Seidel C., Antoniou A., Botla S.K., Campo D., Cariani A., Vazquez E.G., Hauschild J., Hervet C., Hjörleifsdottir S., Hreggvidsson G. 2010. Identifying fishes through DNA barcodes and microarrays. PLoS One. **5**(9):e12620. doi: 10.1371/journal.pone.0012620

Nearing J.T., Douglas G.M., Comeau A.M., Langille M.G. 2018. Denoising the Denoisers: an independent evaluation of microbiome sequence error-correction approaches. PeerJ. **6**:e5364. doi: 10.7717/peerj.5364.

Osawa S., Jukes T.H., Watanabe K., Muto A. 1992. Recent evidence for evolution of the genetic code. Microbiology and Molecular Biology Reviews. **56**(1):229-64. PMCID: PMC372862.

Paradis E., Schliep K. 2018. ape 5.0: an environment for modern phylogenetics and evolutionary analyses in R. Bioinformatics. **35**(3):526-8. doi: 10.1093/bioinformatics/bty633.

Pentinsaari M., Salmela H., Mutanen M., Roslin T. 2016. Molecular evolution of a widely-adopted taxonomic marker (COI) across the animal tree of life. Scientific Reports. **13**;6:35275. doi: 10.1038/srep35275.

Porter T.M., Hajibabaei M. 2018. Over 2.5 million COI sequences in GenBank and growing. PLoS One. **13**(9):e0200177. doi: 10.1371/journal.pone.0200177.

Ratnasingham S., Hebert P.D. 2007. BOLD: The Barcode of Life Data System (http://www.barcodinglife.org). Molecular Ecology Notes. **7**(3):355-64. doi: 10.1111/j.1471-8286.2007.01678.

Schirmer M., D'Amore R., Ijaz U.Z., Hall N., Quince C. 2016. Illumina error profiles: resolving fine-scale variation in metagenomic sequencing data. BMC Bioinformatics. 17(1):125. doi: 10.1186/s12859-016-0976-y

Schoch C.L., Seifert K.A., Huhndorf S., Robert V., Spouge J.L., Levesque C.A., Chen W., Fungal Barcoding Consortium. 2012. Nuclear ribosomal internal transcribed spacer (ITS) region as a universal DNA barcode marker for Fungi. Proceedings of the National Academy of Sciences. **109**(16):6241-6. doi: 10.1073/pnas.1117018109.

543

544 Tsukihara T., Aoyama H., Yamashita E., Tomizaki T., Yamaguchi H., Shinzawa-Itoh K., Nakashima R.,
545 Yaono R., Yoshikawa S.. Structures of metal sites of oxidized bovine heart cytochrome c oxi-
546 dase at 2.8 A. Science. 1995 Aug 25;269(5227):1069-74. doi: 10.1126/science.7652554

547

548 Vences M., Thomas M., Van der Meijden A., Chiari Y., Vieites D.R.. 2005. Comparative performance
549 of the 16S rRNA gene in DNA barcoding of amphibians. Frontiers in Zoology. **2**(1):5. doi:
550 10.1186/1742-9994-2-5

551

552 Wilkinson S.P. 2019. aphid: an R package for analysis with profile hidden Markov models. Bioinfor-
553 matics. **35**(19):3829-30. doi: 10.1093/bioinformatics/btz159.

554
555 Yacoub H.A., Fathi M.M., Sadek M.A. 2015. Using cytochrome b gene of mtDNA as a DNA barcoding
556 marker in chicken strains. Mitochondrial DNA. **26**(2):217-23. doi:
557 10.3109/19401736.2013.825771

558

559 Youle R.J. 2019. Mitochondria—Striking a balance between host and endosymbiont. Science.
560 **16**;365(6454):eaaw9855. doi: 10.1126/science.aaw9855.

561

562

563 **Tables and Figures**

564 **Table 1.** Classification accuracy of coil's indel check function on test sequence groups of different
565 compositions. AUC – area under the curve for the ability of the amino acid PHMM log likelihood score
566 to discriminate sequences with insertion or deletion errors from sequences without errors. The optimal
567 LogL threshold for differentiating sequences with indels from those without indels is shown. The
568 optimal LogL cut off, as determined through F-score maximization, is shown.

Dataset	Translation	AUC	Optimal LogL Cut Off
Full length sequences	**Genetic code known**	**0.9989**	**-246.20**
	Censored	**0.9982**	**-358.88**
Variable length sequences	**Genetic code known**	**0.9662**	**-354.44**
	Censored	**0.9659**	**-457.39**
Short length sequences	**Genetic code known**	**0.9509**	**-354.44**
	Censored	**0.9504**	**-440.24**
Long length sequences	**Genetic code known**	**0.9802**	**-324.37**
	Censored	**0.9795**	**-412.88**
All data combined	**Genetic code known**	**0.9776**	**-308.05**
	Censored	**0.9773**	**-425.21**

569

coil workflow components

Load raw sequences (`coi5p` function)

- Takes a DNA sequence and builds a coi5p object

```
Standard sequence:  actttatatttctta.......................ttatatcaacattta
Extra sequence:     gcgcgcgacttctta.......................accaacatttacgcg
Missing sequence:   ttcttatttggagct.......................ttataccaacattta
Error sequence:     actttatagtatatt.......................ttatatcaacattta
```

Place in common reading frame (`frame` function)

- Trims excess base pairs from the sequence, adds dashes for sequences with missing base pairs at front

```
Standard sequence:  actttatatttctta.......................ttatatcaacattta
Extra sequence:     ttcttatttggagct.......................ttataccaacattta
Missing sequence:   ---------acttta.......................ttataccaacattta
Error sequence:     actttatagtatatt.......................ttatatcaacattta
```

Translate sequence to amino acids (`translate` function)

- Translate the framed sequence to amino acids, with known known genetic code or censored translation

```
Standard (know translation table =  5):  TLYFLFGMWSGMIGS.......................FDPAGGGDPILYQHL
Standard (unknown translation table):    TLYFLFG?WSG?IGS.......................FDPAGGGDPILYQHL
Extra sequence:                          TLYFLFGAWSG?VGT.......................FDPAGGGDPILYQHL
Missing sequence:                        ---FLFGAWSG?VGT.......................FDPAGGGDPILYQHL
Error sequence:                          TL*YIFGAFFWDD?N.......................FWSSW???PCFYINI
```

Evaluate amino acid sequence (`indel_check` function)

- Check for stop codons and check if the Markov model log likelihood is suggestive of an indel

```
Standard (known translation table): indel_likely: FALSE, stop_codons: FALSE
Standard (unknown translation table): indel_likely: FALSE, stop_codons: FALSE
Extra sequence: indel_likely: FALSE, stop_codons: FALSE
Missing sequence: indel_likely: FALSE, stop_codons: FALSE
Error sequence: indel_likely:  TRUE, stop_codons:  TRUE
```

Fig. 1. Depiction of the coil library's analysis pipeline. The tasks performed by the four main functions of the analysis pipeline on a variety of potential use cases are shown, and a brief explanation is given. For information on how to implement the analysis pipeline, please consult the package's vignette (Supplementary File 6).

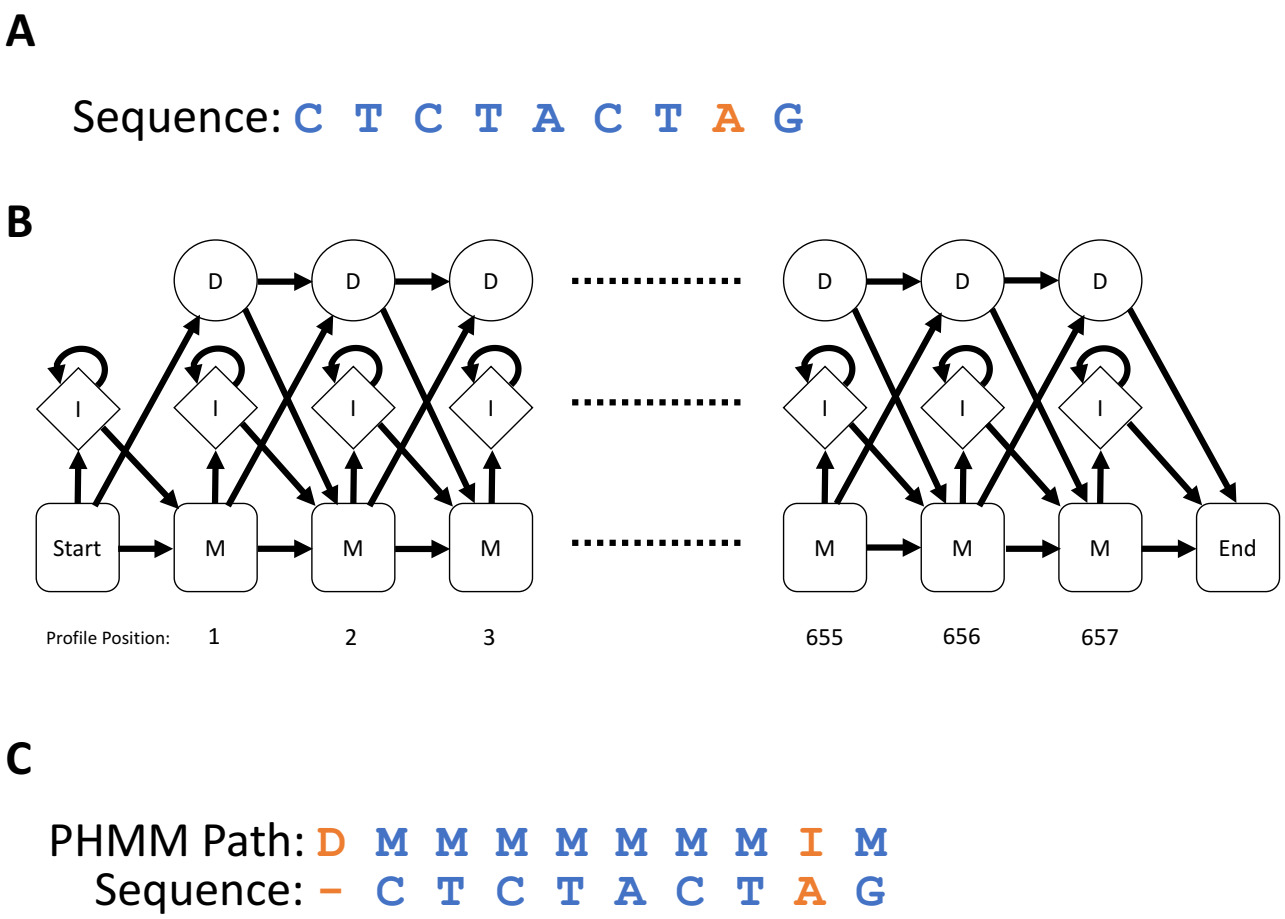

Fig. 2. Description of how a profile hidden Markov model (PHMM) determines the hidden state of nucleotides in a biological sequence. (A) An truncated example of a DNA sequence (the nucleotides are the observed states) to be evaluated using the PHMM. An erroneous nucleotide (indicated in orange) has been inserted into the sequence; this fact would be unknown prior to evaluation. (B) A diagram of the potential paths of hidden states through the PHMM. The possible hidden states are: M (indicates a nucleotide is a match to the sequence profile), D (indicates the nucleotide for the given position in the profile has been deleted) and I (indicates a nucleotide has been inserted into the sequence at that position). The arrows indicate the possible transitions from one hidden state to another and the numbers below the diagram indicate the positions in the profile. (C) The Viterbi algorithm is used to compare the observed states (nucleotide sequence) to the PHMM to determine the most probable set of

589 hidden states corresponding to the sequence. When a DNA sequence is evaluated against the PHMM
590 using the Viterbi algorithm, the position-specific emission and transition probabilities (essentially the
591 probability of a given nucleotide being found at that point in the sequence and the probability of
592 transitioning from the current nucleotide to the next observed nucleotide, see Fig. 3) are used to assign
593 hidden states to the nucleotides in the sequence. In this example, a nucleotide was missing at the start
594 of the sequence, so the hidden state for position 1 in the profile is delete ('D'). The orange (inserted)
595 nucleotide has been identified as an insertion between positions 8 and 9 of the profile and therefore this
596 nucleotide is assigned the hidden state of insert ('I'), while all other nucleotides are found to match
597 positions in the profile and have been assigned hidden states of match ('M').
598
599

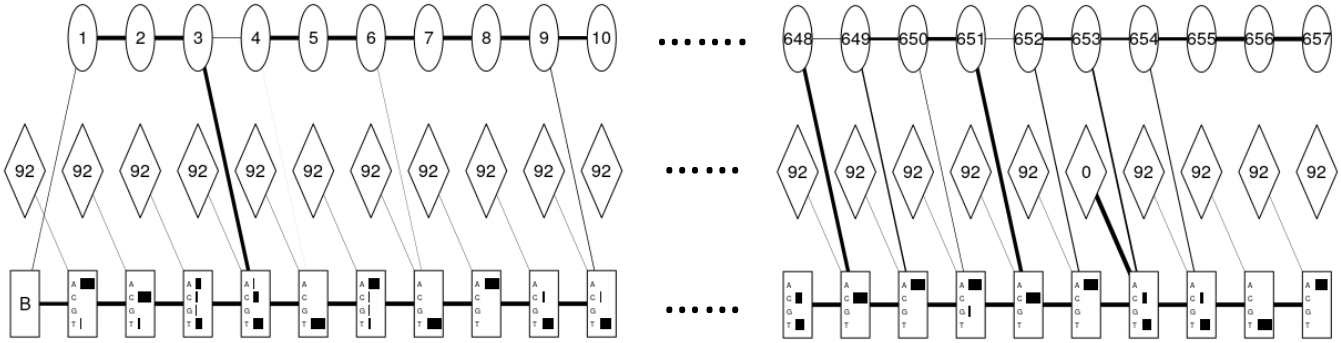

Fig. 3. Diagram of the emission and transition probabilities for the first and last 10 positions in the nucleotide PHMM used in the coil package. The circles represent the hidden delete states and contain the profile positions, the diamonds indicate the hidden insert states and the numbers within the diamonds indicate the probability of insert-insert transitions. The rectangles indicate the match states and the horizontal bars within represent the emission probabilities for the possible nucleotides at the given profile position. The weight of the lines between shapes represent the transition probabilities.

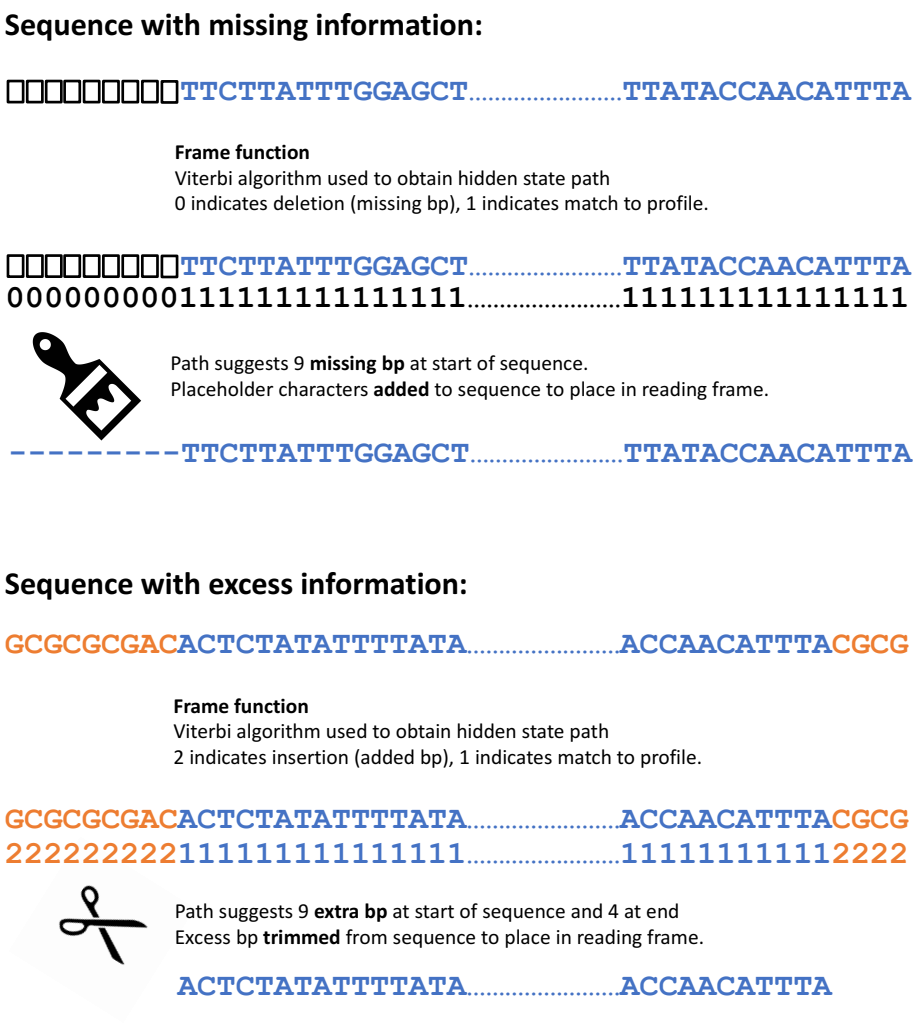

Fig. 4. Diagram showing how coil establishes the reading frame of nucleotide sequences through assessment of the hidden state path produced through the Viterbi algorithm's comparison of input sequence to the nucleotide PHMM.

```
Amino Acid =   FFLLSSSSYY?*CCWWLLLLPPPPHHQQRRRRII?MTTTTNN?KSS??VVVVAAAADDEEGGGG

   Base 1  =   TTTTTTTTTTTTTTTTCCCCCCCCCCCCCCCCAAAAAAAAAAAAAAAAGGGGGGGGGGGGGGGG
   Base 2  =   TTTTCCCCAAAAGGGGTTTTCCCCAAAAGGGGTTTTCCCCAAAAGGGGTTTTCCCCAAAAGGGG
   Base 3  =   TCAGTCAGTCAGTCAGTCAGTCAGTCAGTCAGTCAGTCAGTCAGTCAGTCAGTCAGTCAGTCAG
```

Fig. 5. Censored genetic code utilized by coil when the genetic code corresponding to an input sequence is not known. The '?' placeholder characters correspond to the ambiguous codons which code for different amino acids in different animal mitochondrial genetic codes. These amino acids are therefore censored to avoid translation errors.

Sequence without errors:

TLYFLFGMWSGMIGS.....FDPAGGGDPILYQHL

Viterbi algorithm used to obtain hidden state path
0 indicates deletion, 1 indicates match to profile, 2 indicates insertion.

111111111111111.....111111111111111

Path suggests all match states, suggestive of a barcode sequence
without errors. The log likelihood of the sequence is therefore high.

Log likelihood: **-237.68** , stop codons: **False** ✓

Additionally, the amino acid sequence does not contain stop codons,
which is further suggestive of a correct amino acid sequence.

Sequence with errors:

TLYIW*GAFFWDDGS.....FWSSWGGGPCFYINI

Viterbi algorithm used to obtain hidden state path
0 indicates deletion, 1 indicates match to profile, 2 indicates insertion.

1112221122222211.....222221112122222

Path is rife with insert states, suggests a phase shift has occurred
in the nucleotide sequence, moving the sequence out of correct
reading frame. The log likelihood of the sequence is therefore low.

Log likelihood: **-1138.69**, stop codons: **True** X

Additionally, the amino acid sequence contains stop codons,
which is further suggestive of a shift out of correct reading frame.

Fig. 6. Diagram showing how coil's indel_check function utilizes the hidden state path output by the Vitertbi algorithm to determine the likelihood that a sequence contains an insertion or deletion error that has led to a shift in reading frame.

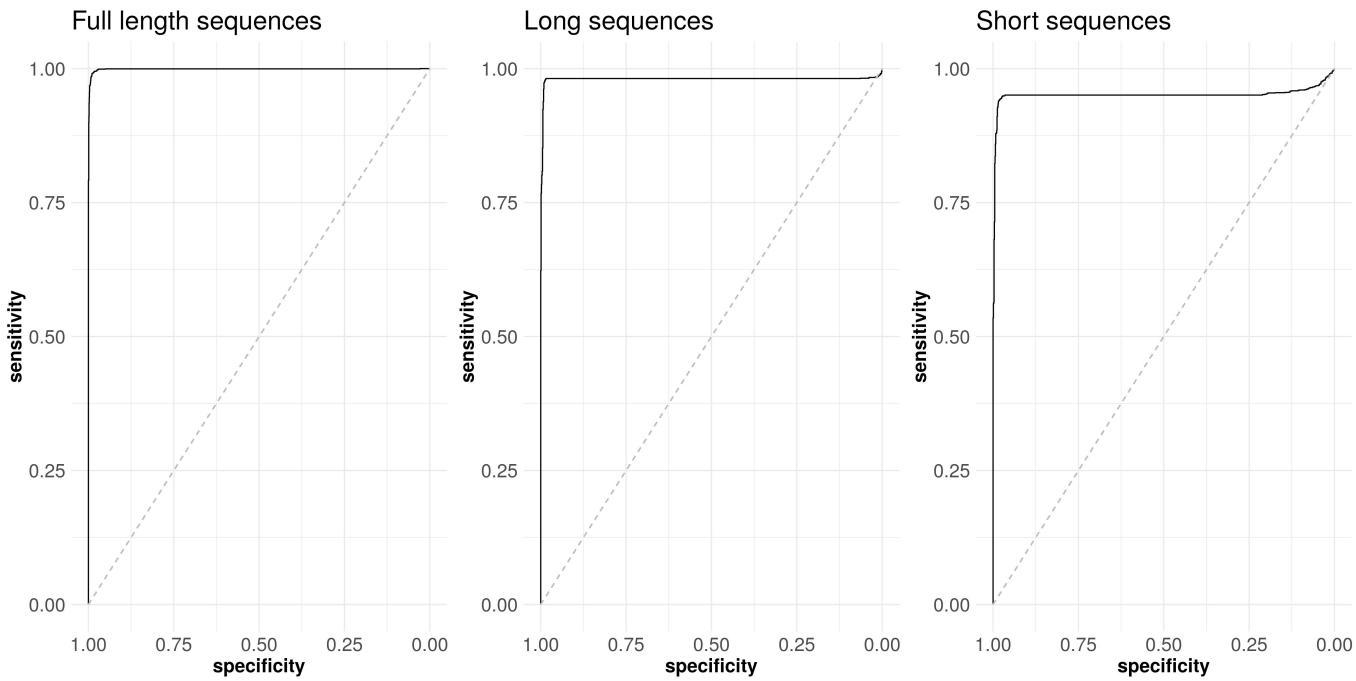

Fig. 7. ROC (receiver operating characteristic) curves showing the ability of coil to correctly classify sequences as correct or containing insertion or deletion errors. Indel classification ability for three types of sequences (all translated with known genetic codes) are displayed: full-length barcode sequences (barcode sequences of unaltered lengths), long sequences (1-100bp added to both ends of the sequences) and short sequences (1-100bp removed from each end of the sequence). Black lines indicate the ability of the classifier to separate true sequences from those with indels. A perfect classifier would have an area under the curve (AUC) of 1 and intersect the top left corner of the graph (See table 1 for corresponding AUC values). The dotted grey lines represent classification through chance (AUC = 0.5).

636 **List of Supplementary Files**

637 **Supplementary File 1 –** 'S1-train_bold_coi_public_data_subsample.zip' A .tsv file with the
638 representative subsample of the BOLD public database used to train the nucleotide and amino acid
639 PHMMs.

640

641 **Supplementary File 2** – 'S2-create_assessment_data.py' Custom python script used in the creation of
642 the variable quality dataset for testing coil.

643

644 **Supplementary File 3** – 'S3-coil-utilization.r' R script showing how the coil pipeline was applied to
645 the validation data

646

647 **Supplementary File 4 –** 'S4-non_animal_tests_output.tsv' Output from the analysis of publically
648 available *Wolbachia,* plant and fungi COI-5P barcode sequences and nuclear mitochondrial
649 pseudogenes (numts) using the coil analysis pipeline.

650

651 **Supplementary File 5** – 'S5-validation_data' Folder containing the validation data inputs and outputs
652 produced for the assessment of coil effectiveness.

653

654 **Supplementary File 6** – 'S6-coil-vignette.pdf' A pdf demonstrating the functionality of the coil
655 package. This is the static version of the package vignette, which is available through R. The original R
656 markdown file is available for download from: https://github.com/CNuge/coil/tree/master/vignettes

657