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] Centre for Biodiversity Genomics, 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 subunit I (COI-5P) is the most common barcode gene for animals, 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 common 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. Package limitations 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. Results demonstrated that coil is a strong technical error identification method but is not reliable for detecting all biological contaminants.

Keywords: COI, DNA barcoding, error identification, translation

34 **Introduction**

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

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

58 The wealth of COI-5P sequence data available and the conserved profile of the sequence
59 provide the information necessary for pre-processing and evaluation of novel COI-5P barcode data
60 (Ratnasingham and Hebert 2007; Stoeckle and Kerr 2012; Pentinsaari *et al.* 2016; Porter and
61 Hajibabaei 2018). At present, the comparison and evaluation of sequences in a common context can be
62 accomplished through the *de novo* alignment of novel sequences, but this approach grows in
63 computational complexity with the volume of data and the rate of errors in the data (Elbrecht *et al.*
64 2018). Software tools that allow for the local or web-based alignment of novel sequences to reference
65 datasets include MACSE (Multiple Alignment of Coding SEquences) (Ranwez *et al.* 2011) and
66 USEARCH (Edgar 2010). Analysis of novel sequences with these methods relies on the comparison of
67 inputs against large sets of reference sequences which a user must acquire, store, and curate. The
68 outputs must also be scrutinized in subsequent analysis so that the presence of errors can be inferred
69 from deviations in alignment. Additionally, if reference data sets are not sufficiently cleaned, then
70 sequences with errors may not be identified, and if references contain taxonomic bias then true
71 biological variants may be interpreted as errors. If systematic errors exist in novel samples, for reasons
72 such as the tendency of a given sequencing platform more often to insert or delete bases in certain
73 regions of a sequence (Hebert *et al.* 2018) or due to bias related to certain sequence motifs (Schirmer *et*
74 *al.* 2016), then *de novo* alignment of data may not identify systematic errors and instead depict these as
75 biological variants. Another method for processing and analyzing barcode sequences is to upload them
76 to a public repository such as BOLD (boldsystems.org), which uses proprietary error evaluation
77 software (Ratnasingham and Hebert 2007). If the analyses being conducted are exploratory or still in
78 the development stage, then a researcher may not yet be prepared to upload the data to a public
79 repository. Additionally, if methods and associated results are being refined, the repeated upload of
80 sequences to BOLD for processing and error checking can be inefficient and best left as a final step in
81 the workflow. A locally available tool for the establishment of reading frame for COI-5P data could
82 allow researchers to compare barcode data from different samples efficiently in a common context.

Additionally, placing DNA sequences in the correct reading frame permits the translation of the nucleotide sequence to amino acids, thereby providing a means of indel identification. Insertions and deletions that introduce frameshifts can be easily detected through the resulting drastic changes to the amino acid sequence.

A tool for pre-processing and technical error identification in DNA barcodes would be a useful addition to both DNA barcode and metabarcode workflows. In DNA barcoding, the barcode sequences generated for a set of specimens can be evaluated and data correction efforts quantitatively targeted. Barcodes displaying evidence of indel errors could be excluded, subjected to error correction, or recommended for resequencing. In DNA metabarcoding, the haplotypes contained within identified operational taxonomic units (OTUs) could be processed and tested for evidence of indels. Generation of consensus sequences for OTUs using only sequences without evidence of technical errors would improve quality. Additionally, if intraspecific diversity were being characterized through the examination of haplotypes within OTUs, low-abundance haplotypes would not have to be excluded solely based on their abundance (Elbrecht *et al.* 2018) and could be included in subsequent analyses with higher confidence if they displayed no evidence of technical errors.

Here, we present a new R package for the pre-processing of COI-5P DNA barcode data: coil. The package provides several functions for: (1) placement of sequences in a common reading frame, (2) trimming of sequences to the COI-5P barcode region, (3) translation of nucleotides to amino acid sequences, and (4) estimating the likelihood that a sequence contains an insertion or deletion error. The tool is free and publicly available through CRAN (https://CRAN.R-project.org/package=coil) and GitHub (https://github.com/CNuge/coil). We demonstrate the effectiveness of coil by showing how it can be used to align novel barcode sequences to the COI-5P profile and to identify sequences with insertion or deletion errors, in most cases with greater than 97.5% accuracy.

108 **Implementation**

109 The coil package contains functions that allow new DNA barcode data to be placed into reading
110 frame, translated, and assessed for evidence of insertion or deletion errors (Fig. 1). The statistical
111 models the package uses to evaluate sequences are two profile hidden Markov models (PHMMs)
112 respectively trained on nucleotide and amino acid sequences of the COI-5P barcode region. Profile
113 hidden Markov models (PHMMs) are a specialized variant of a hidden Markov model well suited for
114 statistically representing biological sequences that share a common evolutionary origin (Durbin and
115 Eddy 1998; Eddy 1998; Wilkinson 2019). A PHMM is a probabilistic representation (profile) of a
116 multiple sequence alignment. Training a PHMM captures information about the structure of a series of
117 related biological sequences, characterizing the probability of an observed character being present at
118 the given point in the sequence and the probability of transitioning from the observed character to the
119 next character in the sequence. Once trained, PHMMs are probabilistic representations of the aligned
120 training sequences that can be used to determine how closely new sequences match the profile. The
121 characters in the new biological sequence (the nucleotide or amino acid alphabets) constitute the
122 observed characters (states), and comparison to the PHMM relates the observed characters to the series
123 of unknown hidden states that describes how the observed sequence of characters most likely match to
124 the probabilistic profile (Figs. 2, 3). The PHMMs used within coil are meant to take a new DNA
125 barcode sequence (the observed states) and look for evidence of erroneously inserted or deleted base
126 pairs in the sequence (the hidden states). For coil, the observed character states are both the nucleotides
127 and amino acids of the barcode sequence and the hidden states are whether the given character: (1)
128 matches the barcode profile, (2) is likely an erroneous inserted character, or (3) is evidence of a missing
129 (deleted) character.

130 The coil analysis pipeline establishes the reading frame of sequences (Fig. 4) using the Viterbi
131 algorithm (Fig. 1) to compare the series of observed states (the DNA sequence) to the nucleotide
132 PHMM (Durbin and Eddy 1998; Wilkinson 2019). This algorithm determines the most likely sequence

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

142 **Translation**

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

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

167 **Error assessment**

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

183 the wrong genetic code. Since the enzyme that COI-5P codes for is highly conserved, there is an
184 extremely low likelihood of a biologically viable mutation that leads to early termination of the amino
185 acid sequence. The log likelihood of the sequence path and the query of the sequence for stop codons
186 can therefore be used in conjunction to assess sequence validity with high accuracy. This error
187 checking method has the additional advantage of not employing a computationally expensive step of
188 comparison of the input sequence against a large set of references (Edgar 2010; Ranwez *et al.* 2011)
189 and can be utilized without acquisition or user-curation of a large set of reference sequences.
190

191 **Methods**

192 **BOLD data acquisition**

193 The Barcode of Life Data Systems (BOLD: boldsystems.org) public database was queried to obtain
194 a representative sample of publicly available COI-5P sequences. Unique COI-5P haplotypes
195 (Ratnasingham and Hebert 2013) were retained if the following criteria were met: (1) the sequence
196 was >600bp in length within the Folmer region (this was done to ensure sequences could be aligned
197 with high fidelity prior to model training) (Folmer *et al.* 1994), (2) the taxonomy of the sequence was
198 known to a genus level, (3) there were no missing bp ('N') in the Folmer region, (4) the corresponding
199 amino acid sequence did not contain stop codons, (5) the result of BOLD's internal check for
200 contaminants was negative. The query and subsequent filtering yielded approximately 1.3 million
201 unique sequences (in BOLD's aligned format). To isolate the 657bp COI-5P region of the sequence,
202 data were initially analyzed in BOLD's aligned format. Sequences with information outside of the COI-
203 5P region were truncated and the placeholder dashes from BOLD's aligned format were then removed
204 from the sequences. The resulting unaligned sequence data for only the Folmer region were then used
205 in subsequent model training and assessment. PHMM training (described below) included *de novo*
206 multiple sequence alignment, so that the alignment of input sequences for PHMM generation was
207 repeatable (parameters for BOLD's alignment of sequences are not publicly available). This process

208 was repeated for the amino acid sequences, with only the 219 amino acids corresponding to the barcode
209 region retained. To ensure that the data used for model training and the data used for model testing and
210 validation were independent from one another, the barcode dataset was first split in a stratified fashion
211 (on the taxonomic level: family) into training (70%), test (15%), and validation (15%) sets for
212 independent use in model training, refinement, and final assessment. The taxonomically stratified split
213 of the data was done to ensure that the training dataset was not biased towards the largest taxonomic
214 orders due to random chance and that subsequent models were trained on diverse data representative of
215 currently-available DNA barcode data on BOLD for the entire animal kingdom. It should be noted that
216 some of the sequences in the BOLD dataset may be taxonomically misidentified, but that any errors of
217 this kind would not have a significant impact on the PHMMs constructed. Training samples were used
218 to train a single nucleotide and a single amino acid PHMM to represent the entire animal kingdom, so
219 misidentified training data would only cause a slight change in the distribution of the taxonomic
220 diversity of the test, validation, and training sets and not likely result in a significant impact on model
221 performance.

222 **Model training**

223 A nucleotide and amino acid profile hidden Markov model (PHMMs) were respectively trained on
224 COI-5P nucleotide and amino acid training sequences (Supplementary File 1). The models were not
225 trained on the entire set of 936,971 barcodes in the training set; rather, in an attempt to prevent model
226 over fit (and allow effective application on new animal taxa not yet represented on BOLD), different-
227 sized taxonomically representative subsamples of the full set of training data were used to train
228 competing PHMMs. The full training set was subsampled in a stratified fashion (on the taxonomic
229 level: family) so that different taxonomic groups were proportionally represented in the subsamples,
230 with the exception of families with fewer than 10 samples in the training set, which had all their
231 representatives added to each subset to ensure that these rare families were represented in model
232 training. The two taxonomically representative training sets of ~1.2% (11,391 sequences) and ~3.0%

233 (28,189) of the training data were produced, and each was subjected to the following procedure.
234 Training of the nucleotide and amino acid PHMMs was done using the R package aphid (Wilkinson
235 2019). Model training was conducted using aphid's derivePHMM function, which produces a *de novo*
236 multiple sequence alignment and generates an optimal PHMM using the Baum-Welch training
237 algorithm (Wilkinson 2019). Training with the derivePHMM function utilized default parameters,
238 except for the 'pseudocounts' and 'maxsize' parameters. The 'background' option was used for the
239 'pseudocounts' parameter (see Durbin and Eddy 1998 – Chapter 5 for explanation of pseudocounts).
240 Initially, models were trained using two 'maxsize' parameter pairs: (a) a nucleotide PHMM maxsize of
241 657 and a corresponding amino acid maxsize of 219 and (b) a nucleotide PHMM maxsize of 673 and a
242 corresponding amino acid maxsize of 224. The 'maxsize' parameter limits the upper bound of the
243 number of modules in the PHMM; i.e. it provides the maximum length of the profile that represents the
244 sequences. Alignment of test data to the models via the Viterbi algorithm revealed that when the larger
245 maxsize parameter was utilized, the PHMM expanded to the maximum size permitted. In subsequent
246 testing this led to variable framing of sequences (dependent on taxonomy) and made the establishment
247 of a common reading frame infeasible. The smaller maxsizes of 657/219 yielded a set of PHMMs that
248 could effectively establish the reading frame of taxonomically diverse sequences in the test data, and
249 this pair of PHMMs was therefore retained as the internal PHMMs for coil, against which new barcode
250 sequences are compared. Of the two training data sizes that were used, the PHMMs resulting from
251 training on the smaller training set (11,391 sequences) were found to perform marginally better on
252 initial tests, both in terms of the framing and error classification of sequences. Therefore, the nucleotide
253 and amino acid PHMMs trained on 11,391 training sequences were retained for internal use in the coil
254 package. The runtime (on a virtual machine hosted on Compute Canada cloud resources, with 32 CPU
255 cores (2.4GHz) and 120Gb of RAM) for training of the final nucleotide PHMM with 11,391 input
256 sequences was 2 hours and 22 minutes, and the time for training of the corresponding amino acid
257 model was 5 minutes.

258

259 **Assessment of coil package performance**

260 Following successful model training and package construction, a random subsample of 10,000
261 barcode sequences from the previously withheld validation data was used to assess the performance of
262 coil, in terms of its ability to place sequences in the correct reading frame and determine if sequences
263 contain insertion or deletion errors. Additionally, we aimed to test coil's establishment of reading frame
264 and proper translation when excess sequence information in addition to the COI-5P region is present or
265 when barcode information is missing from the edges of sequences. The 10,000 samples were randomly
266 split and altered using a custom Python script (Supplementary File 2) to construct the following 5
267 groups (2000 sequences each):

268 (a) No changes made.
269 (b) 1-100 random base pairs added to the front and/or back of the sequence, with equal addition
270 probabilities for each of the four nucleotides.
271 (c) 1-100 base pairs removed from the front and/or back of the sequence.
272 (d) Altered with a random indel probability of 1% per bp (equal probability of either an
273 insertion or deletion, in the case of insertions a random bp was inserted). Point mutations
274 were also applied with a probability of 1% per bp (a 1% chance of a nucleotide being
275 changed to any other nucleotide) in order to produce more realistic error profiles and test
276 indel error detection in the presence of additional noise.
277 (e) Sequences either shortened or elongated in the manner of groups (b) or (c), with errors then
278 introduced in the manner of group (d).

279 These validation data were then loaded into R and passed through the coil analysis pipeline
280 twice (Supplementary File 3). First, the translation step of the coil pipeline was run using the genetic
281 codes known to correspond to the sequences, and second, the pipeline was run using the default
282 behaviour, assuming unknown genetic codes and utilizing the censored translation option.

283 The outputs of the pipeline were assessed to determine how well sequences were framed and
284 translated. The package's ability to frame sequences was evaluated by looking for stop codons in the
285 translation outputs of the 6000 validation sequences with no artificially introduced indel errors. The
286 presence of stop codons would indicate a sequence was not set in reading frame properly (as their
287 original amino acid sequences are known and do not contain stop codons), whereas the absence of stop
288 codons is strongly suggestive of correct translation. Additionally, the log likelihood values obtained
289 from comparison of all 10,000 sequences to the amino acid PHMM were assessed to determine the log
290 likelihood's effectiveness as an error classifier. Sequences with introduced indel errors and those
291 without errors were labelled in a binary fashion. The R package ROCit (CRAN.R-
292 project.org/package=ROCit) was used to determine the ability of the amino acid PHMM output log
293 likelihood values to discern sequences with errors from those without errors. Comparisons were made
294 between the following groups of validation data: full length sequence data ((a) vs (d)), variable length
295 sequence data ((b) and (c) vs (e)), and all sequence data ((a)(b)(c) vs. (d)(e)). To determine the effect of
296 sequence length on classification ability, the long sequences (b) were compared to the sequences from
297 the variable length error dataset (e) that were elongated, and the short sequences (c) were compared to
298 the sequences from the variable length error dataset (e) that were shortened. The area under the curve
299 (AUC – a metric of how well the log likelihood values separate the two categories) and the optimal F
300 score threshold (the log likelihood value that optimizes the trade-off between false positive and false
301 negative classifications) were determined for all comparisons.
302 To demonstrate how coil can be applied in metabarcoding analyses, the coil analysis pipeline
303 described above was rerun, and associated evaluation metrics were generated, for non-full-length
304 barcode sequences from a defined subsection for the COI-5P region. Subsets of the barcode sequences
305 in categories (a) and (c) were taken for an arbitrarily chosen 300bp window (base pairs 337-636 of the
306 full COI-5P region) to simulate non-full-length COI-5P sequences commonly generated in DNA
307 metabarcoding. Sequences from group (c) were only retained if they contained a documented indel

308 error within the defined 300bp region (1868 sequences). To determine coil's baseline classification
309 ability on these data, the coil analysis pipeline was run for the non-full-length sequences using the full-
310 length nucleotide and amino acid PHMMs. To test if the use of targeted PHMMs (corresponding to
311 only that specific 300bp section COI-5P region) improves coil's ability to frame sequences and identify
312 indels, the complete coil analysis pipeline was then rerun an additional time using a subsection of the
313 PHMMs representing only the given 300bp window to evaluate the non-full-length barcode sequences
314 (Supplementary File 4).

315 **Processing non-animal barcode sequences**

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

328 *Wolbachia* is a genus of bacterial endosymbionts (commonly found in arthropods) that can in
329 some instances confound attempts to isolate an animal specimen's barcode sequence (Smith *et al.*
330 2012). To see if coil would flag *Wolbachia* sequences as errors (either due to a low likelihood value or
331 the presence of a stop codon in the amino acid sequence), data associated with the accession numbers
332 of *Wolbachia* samples listed in Supplementary File S2 of Smith *et al.* (2012) were obtained from NCBI.

333 Data were filtered to omit sequences <550bp in length, yielding 345 *Wolbachia* sequences that were
334 analyzed via coil's 'coi5p_pipe' function (parameters: trans_table = 0 and indel_threshold = -440.24).
335 Nuclear mitochondrial pseudogenes (numts) are non-encoded sequences in the nucleus that are
336 the product of mitochondrial DNA fragments being incorporated into the nuclear genome (Bensasson *et*
337 *al.* 2001). Numts resulting from the COI-5P gene have been shown to interfere with accurate
338 assessment of the COI-5P sequences for a species complex of Australian stingless bees (*Tetragonula*
339 *carbonaria*, *T. davenporti*, *T. hockingsi*, and *T. mellipes*; hereafter the 'Carbonaria group') (Françoso *et*
340 *al.* 2019). Sequence data for the 9 numts with reported accession numbers in Figure 7 of Françoso *et al.*
341 (2019) were obtained and analyzed with coil (using the coi5p_pipe function and default parameters) to
342 see if the package identified these sequences as likely being erroneous.
343

344 **Results & Discussion**

345 **Framing and Translation**

346 The validation data outputs were assessed to determine how effective coil was at processing
347 variable-length barcode data. The package's ability to set sequences in reading frame was evaluated by
348 looking for stop codons in the translation outputs of the 6000 validation sequences with no introduced
349 indel errors. If stop codons were present, the sequence would not have been set in reading frame
350 properly, whereas the absence of stop codons is strongly suggestive of a correct reading frame
351 establishment. Of the 6000 validation sequences without introduced errors, 5888 (98.13%) were
352 correctly framed. The sequences with incorrectly identified reading frames were not evenly distributed
353 across the three sequence length categories (short sequence, full-length sequences, and long
354 sequences). All barcode sequences in the full-length (a) category except for one were correctly placed
355 in reading frame, indicating that the nucleotide PHMM is able to recognize and frame taxonomically
356 variable barcode data and that it is not biased against certain taxonomic groups. The program appeared
357 to be effective at trimming artificially lengthened sequences (category b) and placing them in reading

358 frame, with only 38 framing errors in the 2000 long sequences (98.1% accuracy). The program was
359 slightly less effective at placing shortened sequences (category c) in reading frame, with 98 errors
360 observed (95.1% accuracy). Despite this imperfect performance, the program is effective at framing the
361 vast majority of sequences and can aid the user by successfully automating this task for most inputs
362 (Supplementary File 6). The summary data provided by the indel check step (stop codon presence and
363 error likelihood values) can aid users in targeting additional correction effort to the subset of sequences
364 coil is unable to place in reading frame.
365
366 **Indel Error Identification**
367 Across the entire validation dataset, coil's indel_check function was able assess the amino acid
368 data and accurately discern sequences with insertion or deletion errors from correct barcode sequences
369 with greater than 97% accuracy. As with the framing of sequences, coil's effectiveness was influenced
370 by the length of the sequences (Table 1). When the full-length barcode sequences with and without
371 introduced errors were evaluated, they could be categorized with greater than 99.9% accuracy. For
372 sequences with excess DNA sequence added to the ends, the sequences with errors could be discerned
373 from the correct sequences with 98.0% accuracy. For the short sequences, performance was once again
374 the lowest, with 95.1% classification accuracy (Fig. 7). The similarity in these accuracy scores to the
375 percentage of sequences placed in reading frame correctly is not coincidental, but rather due to the fact
376 that the ability to correctly classify a sequence is highly dependent on the ability of coil to place it in
377 the common reading frame correctly.
378 Closer examination of the classification errors in the short sequence dataset (the category with
379 the highest error frequency) helped to characterize the limitations of coil's classification ability. The
380 majority of the classification errors (103/132 – 78%) were false negatives, i.e. sequences without
381 introduced errors that were flagged as likely to contain errors. The reason for this appears to be the
382 incorrect establishment of the reading frame and thus large-scale reading frame shifts, resulting in

383 incorrect amino acid sequences. The major limitation of coil therefore appears to be the inability to
384 always establish the reading frame properly, a problem that is magnified in short sequences where the
385 front of the sequence is truncated. The examination of the 29 falsely accepted sequences for the short
386 category revealed the three main patterns where coil failed to detect errors in a sequence: 1) When an
387 indel occurs late (>500 bp) in the sequence (observed for 7 sequences), 2) when three insertions or
388 three deletions are found in closely proximity (observed for 4 sequences), and 3) when an insertion and
389 deletion occur in close proximity to one another (observed for 18 instances). All three of these patterns
390 do not lead to large-scale alterations to the amino acid sequence, and are therefore not easily detected
391 by examination of the amino acid sequence.

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

406 **Extension of coil to metabarcode data analysis**

407 The indel identification tests showed that coil's performance is suboptimal when the length of
408 the examined sequence is far shorter than the PHMM profile length. The proximal cause of this issue is
409 that similarity between probabilistic profiles for different parts of the barcode region can lead to
410 continuous series of base pairs from the sequence matching to an incorrect portion of the PHMM and
411 incorrect establishment of the reading frame. This is problematic in the analysis of metabarcode data,
412 where non-full-length (~200-400bp) sections of the COI-5P barcode are targeted for amplification and
413 sequencing by custom primers. The fact that these non-full-length barcodes are in most instances
414 derived from targeted subsections of the COI-5P region means that the framing limitation can be
415 overcome by comparing sequences to only a subsection of coil's complete PHMMs. Using a subsection
416 of the full PHMMs reduces or eliminates the discrepancy in size between the PHMM and the non-full-
417 length barcodes, drastically reducing the false positive indel signals caused by incorrect establishment
418 of the reading frame.

419 Using the subsection of the complete PHMMs (defined using coil's subsetPHMM function),
420 coil was able to separate non-full-length barcode sequences with indel errors from correct sequences
421 with greater than 99.4% accuracy (Fig 8.). This proved to be a drastic improvement over the use of the
422 full-length PHMMs in the analysis of the same 300bp sequence dataset (AUC = 0.77). Using the full-
423 length model, 513 sequences from group (a) were incorrectly flagged as containing stop codons,
424 indicating a high rate of incorrect reading frame establishment (25.6%). This high level of incorrect
425 framing is a result of the 300bp window starting late (base pair 337) in the COI-5P region, providing a
426 large set of leading profile positions to which incorrect matches could occur. Conversely, the false
427 positive rate using the subsection of the full PHMMs led to an extremely low false positive rate (0.1%).
428 This indicates that the coil analysis pipeline is appropriate for technical error identification in non-full-
429 length metabarcode outputs, if there is available information on the approximate subsection of the COI-
430 5P region from which the sequences are derived.

431 It should also be noted that amino acid PHMM log likelihood values reported for input
432 sequences are dependent on the size of the model to which they are compared. The optimal log
433 likelihood cut offs presented (in Table 1) are therefore not applicable when user-defined subsets of the
434 complete PHMMs are used. Supplementary File 4 has been included to aid in determining the optimal
435 log likelihood cut off for a user defined COI-5P subsection of interest.
436
437 **Package limitations**
438 The results of processing non-animal COI-5P sequences with coil were variable; 167/174 (96%)
439 of fungi and 336/570 (58.8%) of plant sequences were flagged as likely erroneous based on the
440 resulting log likelihood scores and/or the presence of stop codons. The reason for coil labelling fungi
441 sequences as likely erroneous is the existence of introns in fungal COI-5P sequences (Avin *et al.* 2017).
442 Fungal sequences show drastic variability in length due to the presence of introns, compared to animal
443 sequences which do not contain introns and are relatively consistent in length (Avin *et al.* 2017). This
444 variation in the structure of fungal sequences is interpreted by coil's processing pipeline as evidence of
445 indel errors, resulting in the majority of true fungal sequences being flagged as erroneous. Plant COI-
446 5P sequences are less variable in structure than fungi, but more than half of the sequences tested
447 deviated from the animal COI-5P profile in a manner extreme enough for them being flagged as
448 erroneous. These results show that coil cannot presently be used to pre-process and filter errors reliably
449 from non-animal COI-5P sequences. The PHMMs are not trained to accommodate non-animal data,
450 and processing sequences from the plant or fungal kingdoms results in a large number of false
451 negatives. A future extension of coil can be the addition of PHMMs trained on data from different
452 kingdoms or barcodes, which would allow coil's processing pipeline to be applied in error filtration of
453 other protein-coding, intron-free barcodes.
454 Of the 345 *Wolbachia* COI-5P sequences analyzed, only five were flagged as likely to be
455 erroneous, and only a single sequence contained stop codons. The five instances of *Wolbachia*

456 sequences being flagged as errors resulted from the incorrect framing and translation of sequences.
457 Since the *Wolbachia* sequences are real instances of the COI-5P gene (albeit from bacteria as opposed
458 to animals) that do not possess any drastic changes to reading frame relative to animals, their
459 nucleotide and corresponding amino acid sequences are likely enough to be accepted as matches to the
460 animal COI-5P PHMMs. This demonstrates that coil cannot be effectively appropriated to separate
461 animal COI-5P sequences from the COI-5P sequences of endosymbiote contaminants.

462 All 9 of the numt sequences from the Carbonaria group analyzed with coil were flagged as
463 likely to contain an error. This was accomplished despite the fact that none of the translated sequences
464 contained a stop codon. Each pseudogene possessed an amino acid sequence that the program
465 characterized as improbable (all log likelihood values lower than -754.00). It should be noted that the
466 numts analyzed were shorter sequences (333bp each), which inherently possess lower log likelihood
467 values. However, animal COI-5P sequence fragments of this length tend to yield log likelihood values
468 of approximately -450.00, so the drastically lower log likelihood sequences are sufficient to distinguish
469 the pseudogene sequences from COI-5P sequences. The number of numts tested was low and limited to
470 sequences identified within a single species complex. Expansion of these tests to COI-5P-derived
471 numts from a more diverse set of taxa could reveal how well coil is able to characterize numts. At
472 present, it is hypothesized that the ability to discern pseudogenes from true barcode sequences will
473 depend on how diverged the structure and sequence of a pseudogene are, with frame-shift-causing
474 indels increasing the likelihood of detection by coil.

475 Taken together, these results show that coil Version 1.0 should only be used to process animal
476 COI-5P data and that it is less reliable at detecting biological contaminants than technical errors. The
477 genetic structure of biological contaminants such as *Wolbachia* COI-5P is not always dissimilar enough
478 from animal COI-5P sequences for the PHMMs to discern the two categories. Additionally, if an
479 animal specimen were contaminated with DNA not from another kingdom, but rather with DNA from
480 another animal, then detection with coil in most instances would not be possible, as the models

481 employed by the package are trained on a taxonomically robust sample of animal sequences. The
482 results presented here regarding the identification of biological contaminants are conservative, as in all
483 tests we have used coil's default censored translation option, which assumes that taxonomic
484 information is not available. The ability to identify biological contaminants can be greatly increased if
485 the taxonomy (and therefore the genetic code) corresponding to a sample is known. Contaminants from
486 organisms that appropriate a different genetic code would have a higher likelihood of being identified
487 in this situation, as translation would be more likely to yield amino acid sequences with stop codons or
488 highly improbable amino acids. Overall the coil package should not be considered a panacea for all
489 DNA barcode data errors, but rather as a tool that can be incorporated into barcode analysis as a
490 reliable pre-processing and data-cleaning step that can identify technical errors. Researchers using
491 barcode data should consider additional steps in their analyses, such as BLAST, which can help inform
492 the taxonomy of sequences and identify biological contaminants.
493
494 **Conclusion**
495 We have presented the R package coil and demonstrated how it can effectively set COI-5P
496 barcode data in reading frame, translate sequences in a taxonomically robust fashion, and identify
497 sequences containing frame-shifting indel errors with greater than 97% accuracy. A vignette
498 demonstrating the software's functionality and suggested workflows is available within the package
499 (and included here as well – Supplementary File 7). The package can facilitate exploratory analysis of
500 barcode data and aid in data cleaning and error assessment. At present, the package's functionality is
501 limited to the main DNA barcode region for animals, COI-5P. In the future, we hope to extend the
502 package's functionality to additional DNA barcode regions, including alternative animal barcodes like
503 cytochrome b (Kochzius *et al.* 2010; Yacoub *et al.* 2015), as well barcodes employed in analysis of
504 species from other kingdoms (Hollingsworth *et al.* 2009; Hollingsworth *et al.* 2011). Adding this
505 functionality would require the training of addition profile hidden Markov models for these barcode

506 regions and integration of these models into the coil package. This is a feasible addition thanks to the
507 large amount of additional barcode data available through BOLD (Ratnasingham & Hebert 2007) and
508 other protein-coding markers available through NCBI, which can be queried and filtered using methods
509 similar to those employed here for COI-5P. Another extension of the functionality of the coil package
510 is to move from error identification to error correction. PHMMs are robust models of the conserved
511 sequence of the COI-5P barcode region on both a DNA and amino acid level. Accurately identifying
512 where deviations from the expected sequence occur and applying corrections to the sequences in an
513 unbiased fashion would allow PHMMs to serve as the basis of a DNA barcode data denoiser.
514 Corrections based on PHMMs trained on the wealth of barcode sequence information available could
515 serve as a powerful complement, or alternative, to current denoising methods which rely on read
516 abundance and *de novo* clustering of sequences (Nearing *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. This research was enabled in part by resources provided by Compute Canada (www.computecanada.ca). 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.N. 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. 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. **47**(1):193-206. 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

Cristescu M.E. 2014. From barcoding single individuals to metabarcoding biological communities: towards an integrative approach to the study of global biodiversity. Trends in Ecology & Evolution. **29**(10):566-71. doi: 10.1016/j.tree.2014.08.001

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.

Edgar R.C. 2010. Search and clustering orders of magnitude faster than BLAST. Bioinformatics. 26(19):2460-1. doi: 10.1093/bioinformatics/btq461.

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.

575 Hebert P.D.N., Cywinska A., Ball S.L., deWaard J.R. 2003. Biological identifications through DNA
576 barcodes. Proceedings of the Royal Society of London. Series B: Biological Sciences.
577 **270**(1512):313-21. doi: 10.1098/rspb.2002.2218.
578
579 Hebert P.D.N., Penton E.H., Burns J.M., Janzen D.H., Hallwachs W. 2004. Ten species in one: DNA
580 barcoding reveals cryptic species in the neotropical skipper butterfly *Astraptes fulgerator*. Pro-
581 ceedings of the National Academy of Sciences. **101**(41):14812-7. doi:
582 10.1073/pnas.0406166101.
583
584 Hebert P.D.N., Braukmann T.W., Prosser S.W., Ratnasingham S., deWaard J.R., Ivanova N.V., Janzen
585 D.H., Hallwachs W., Naik S., Sones J.E., Zakharov E.V. 2018. A Sequel to Sanger: amplicon
586 sequencing that scales. BMC Genomics. **19**(1):219. doi: 10.1186/s12864-018-4611-3.
587
588 Hollingsworth P.M., Forrest L.L., Spouge J.L., Hajibabaei M., Ratnasingham S., van der Bank M.,
589 Chase M.W., Cowan R.S., Erickson D.L., Fazekas A.J. 2009. A DNA barcode for land plants.
590 Proceedings of the National Academy of Sciences. **106**(31):12794-7. doi:
591 10.1073/pnas.0905845106.
592
593 Hollingsworth P.M., Graham S.W., Little D.P. 2011. Choosing and using a plant DNA barcode. PloS
594 One. **6**(5):e19254. doi: 10.1371/journal.pone.0019254.
595
596 Hubert N., Hanner R. 2015. DNA barcoding, species delineation and taxonomy: a historical perspec-
597 tive. DNA Barcodes. **3**(1):44-58. doi: 10.1515/dna-2015-0006.
598

599 Jukes T.H., Osawa S. 1993. Evolutionary changes in the genetic code. Comparative Biochemistry and
600 Physiology. B, Comparative Biochemistry. **106**(3):489-94. doi: 10.1016/0305-0491(93)90122-l.
601
602 Kochzius M., Seidel C., Antoniou A., Botla S.K., Campo D., Cariani A., Vazquez E.G., Hauschild J.,
603 Hervet C., Hjörleifsdottir S., Hreggvidsson G. 2010. Identifying fishes through DNA barcodes
604 and microarrays. PLoS One. **5**(9):e12620. doi: 10.1371/journal.pone.0012620
605
606 Nearing J.T., Douglas G.M., Comeau A.M., Langille M.G. 2018. Denoising the Denoisers: an inde-
607 pendent evaluation of microbiome sequence error-correction approaches. PeerJ. **6**:e5364. doi:
608 10.7717/peerj.5364.
609
610 Osawa S., Jukes T.H., Watanabe K., Muto A. 1992. Recent evidence for evolution of the genetic code.
611 Microbiology and Molecular Biology Reviews. **56**(1):229-64. PMCID: PMC372862.
612
613 Paradis E., Schliep K. 2018. ape 5.0: an environment for modern phylogenetics and evolutionary anal-
614 yses in R. Bioinformatics. **35**(3):526-8. doi: 10.1093/bioinformatics/bty633.
615
616 Pentinsaari M., Salmela H., Mutanen M., Roslin T. 2016. Molecular evolution of a widely-adopted tax-
617 onomic marker (COI) across the animal tree of life. Scientific Reports. **13**;6:35275. doi:
618 10.1038/srep35275.
619
620 Porter T.M., Hajibabaei M. 2018. Over 2.5 million COI sequences in GenBank and growing. PLoS
621 One. **13**(9):e0200177. doi: 10.1371/journal.pone.0200177.
622

Ranwez V., Harispe S., Delsuc F., Douzery E.J. 2011. MACSE: Multiple Alignment of Coding SEquences accounting for frameshifts and stop codons. PLoS One. 6(9):e22594. doi: 10.1371/journal.pone.0022594.

Ratnasingham S., Hebert P.D.N. 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.

Stoeckle M.Y, Kerr K.C. 2012. Frequency matrix approach demonstrates high sequence quality in avian BARCODEs and highlights cryptic pseudogenes. PLoS One. **7**(8). doi: 10.1371/journal.pone.0043992

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

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

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

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

657 **Tables and Figures**

658 **Table 1.** Classification accuracy of coil's indel check function on test sequence groups of different
659 compositions. AUC – area under the curve for the ability of the amino acid PHMM log likelihood score
660 to discriminate sequences with insertion or deletion errors from sequences without errors. The optimal
661 LogL threshold for differentiating sequences with indels from those without indels is shown. The
662 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**

663

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 7).

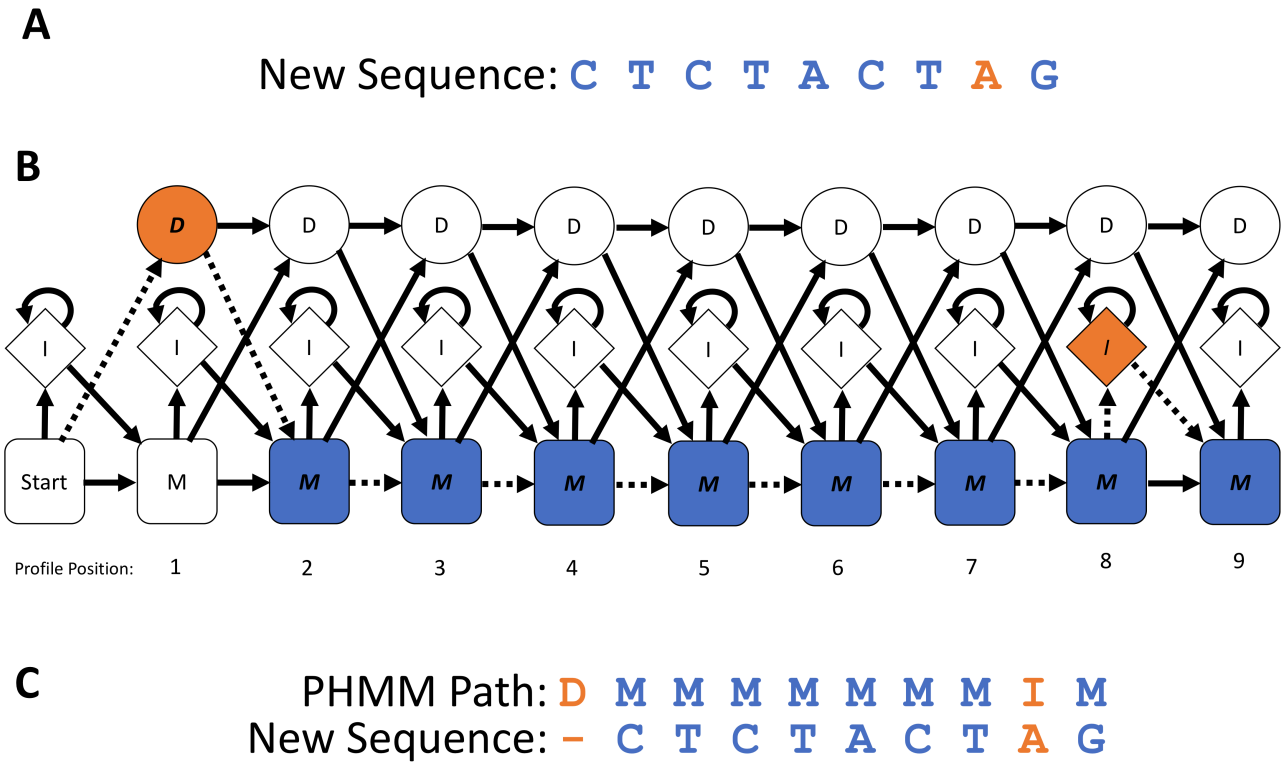

Fig. 2. Description of how a trained profile hidden Markov model (PHMM) determines the hidden state of nucleotides for a new biological sequence. The input (A) is a truncated example of a DNA sequence (the nucleotides are the observed states) to be evaluated using the PHMM. An erroneous nucleotide (indicated 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 trained PHMM (only first 9 of 657 profile positions shown). 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 COI-5P profile. The Viterbi algorithm is used to compare the observed states (nucleotide sequence) to the PHMM to determine the most probable set of hidden states corresponding to the sequence. When a DNA sequence is evaluated against the PHMM using the Viterbi algorithm, the position-specific emission and transition probabilities (essentially the probability

686 of a given nucleotide being found at that point in the sequence and the probability of transitioning from
687 the current nucleotide to the next observed nucleotide, see Fig. 3) are used to assign hidden states to the
688 nucleotides in the sequence. The most probable hidden state path aligning the new sequence to the
689 displayed section of the PHMM is indicated by the series of coloured hidden state cells (with italicized
690 letters) and the dotted arrows indicating the transitions between states. (C) In this example, a nucleotide
691 was missing at the start of the sequence, so the hidden state for position 1 in the profile is delete ('D').
692 The orange (inserted) nucleotide has been identified as an insertion between positions 8 and 9 of the
693 profile, and therefore this nucleotide is assigned the hidden state of insert ('I'), while all other
694 nucleotides are found to match positions in the profile and have been assigned hidden states of match
695 ('M').
696

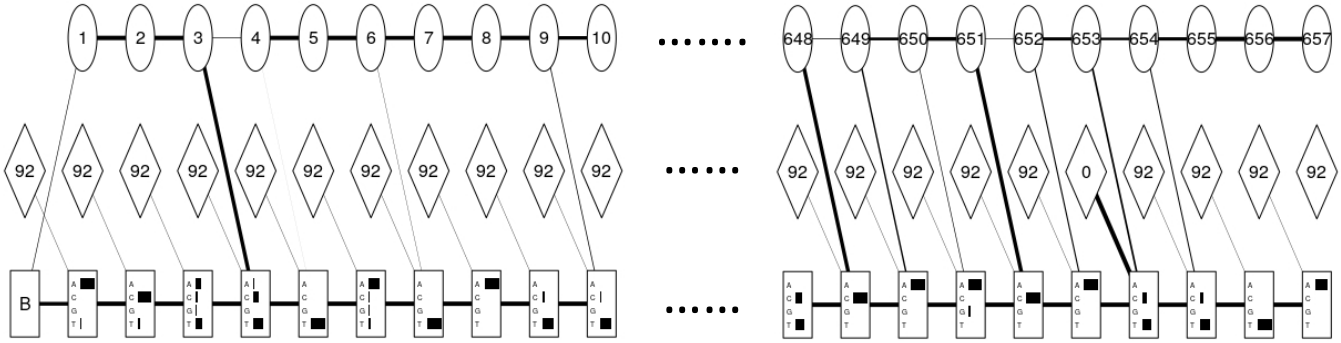

Fig. 3. Diagram of the emission and transition probabilities for the first and last 10 positions in the nucleotide PHMM used in the coil package. The circles represent the hidden delete states and contain the profile positions, the diamonds indicate the hidden insert states, and the numbers within the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

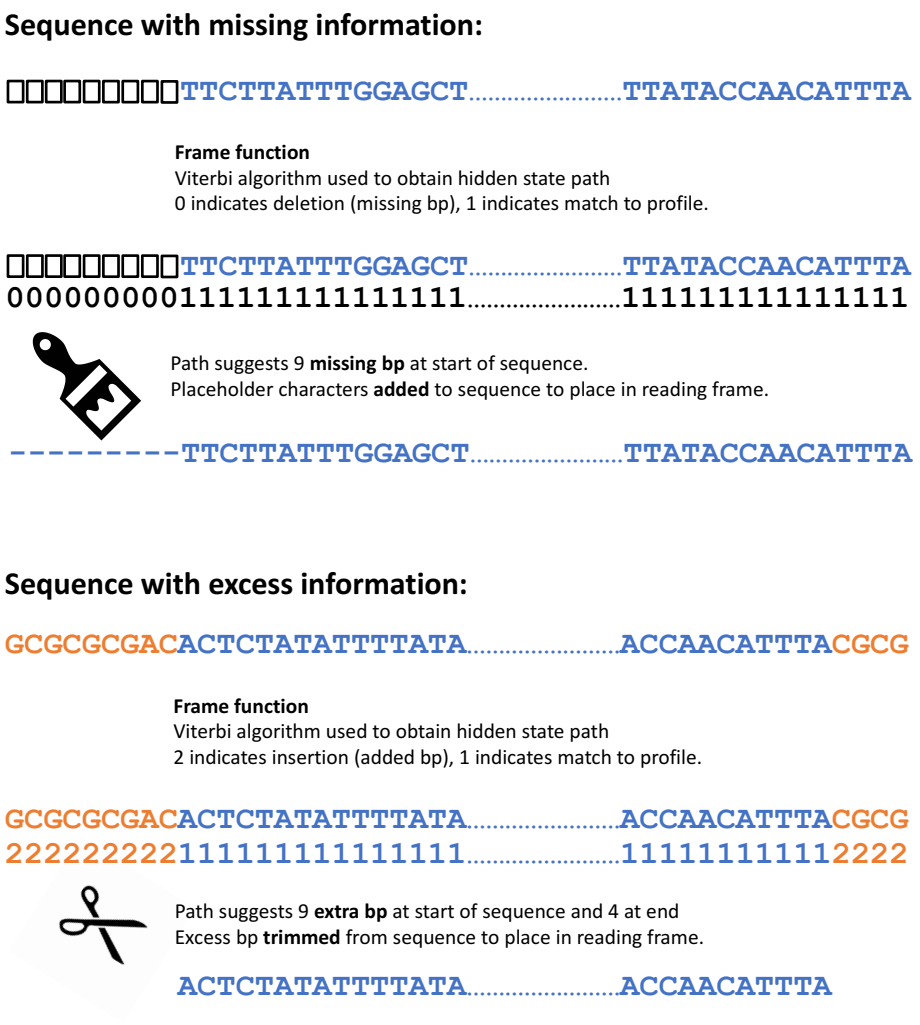

Fig. 4. Diagram showing how coil establishes the reading frame of nucleotide sequences through assessment of the hidden state path produced through the Viterbi algorithm's comparison of input sequence to the nucleotide PHMM.

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

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

Fig. 5. The censored genetic code utilized by coil to translate codons of nucleotides to amino acids when the genetic code corresponding to an input sequence is not known. For each column, the three bottom rows represent the nucleotide composition of the codon (base pair 1, 2, and 3 of the codon read from top to bottom). The amino acid characters in the top row represents the result of translating the codon listed below. The '?' placeholder characters in the Amino Acid row 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, while translation of all other codons is conducted normally.

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.

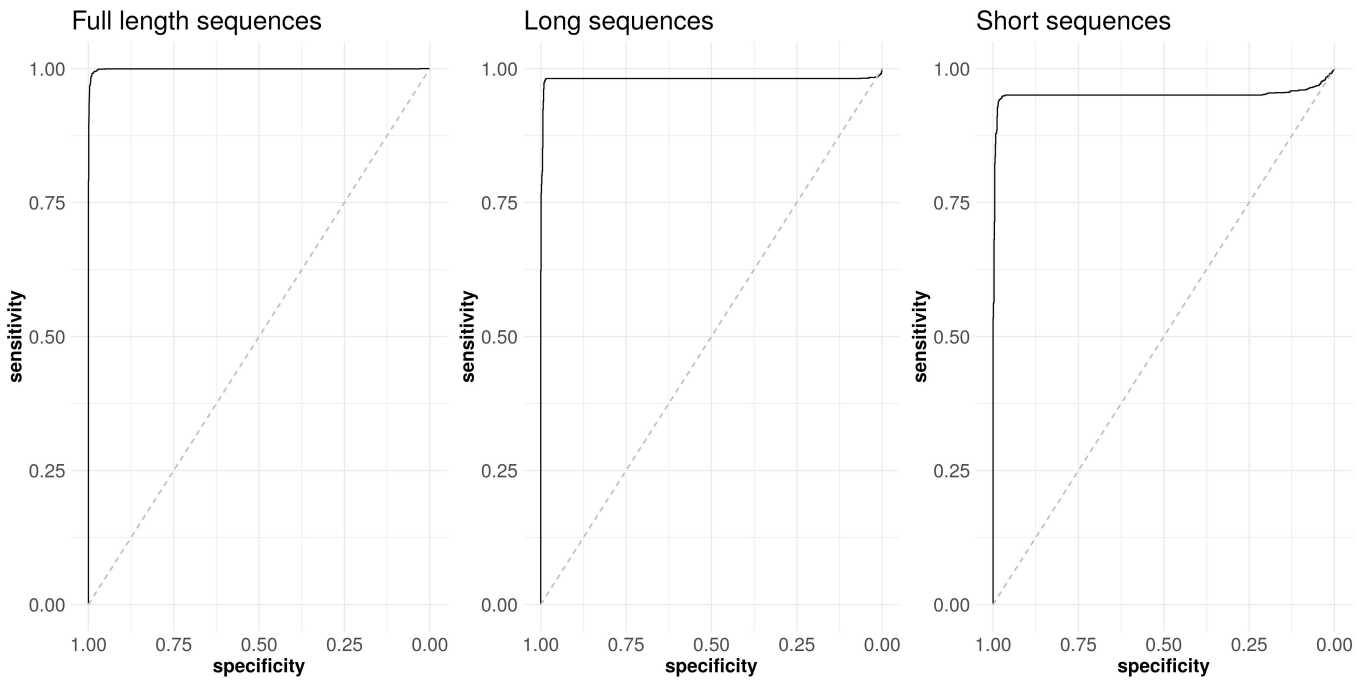

Fig. 7. ROC (receiver operating characteristic) curves showing the ability of coil to classify correctly sequences as being correct or containing insertion or deletion errors. Indel classification ability for three types of sequences (all translated with known genetic codes) are displayed: full-length 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 indel errors. 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).

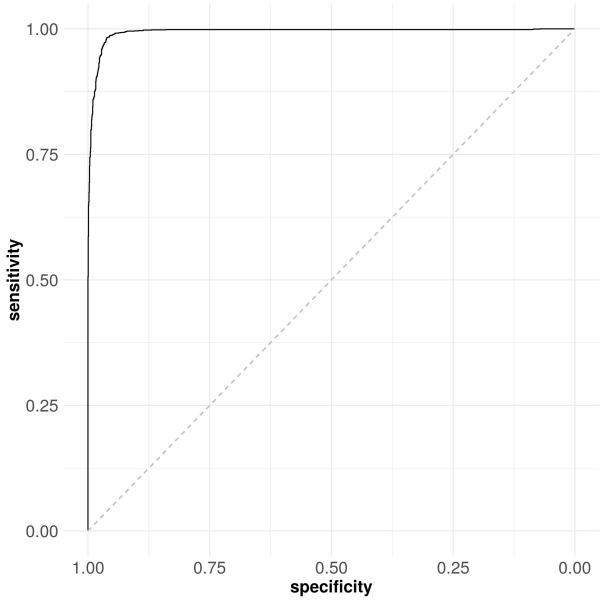

Fig 8. ROC (receiver operating characteristic) curve showing coil's ability to classify correctly non-full-length barcode sequences (300bp) from a defined sub-region of the COI-5P barcode, using a subsection of the nucleotide and amino acid PHMMs corresponding to the same sub-region. The ROC values shown correspond to analysis using censored translation, which assumes unknown genetic code (AUC = 0.994). When the analysis was repeated using known genetic codes, the AUC value improved slightly (AUC = 0.996).

List of Supplementary Files

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

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

Supplementary File 3 – 'S3-coil-utilization.r' R script for application of the coil pipeline in the analysis of the validation data.

Supplementary File 4 – 'S4-constrained_eval.r' R script that contains the code for analyzing short 300bp barcode sequences with subsets of the complete PHMMs.

Supplementary File 5 – 'S5-non_animal_tests_output.tsv' Output from the analysis of publicly available *Wolbachia,* plant and fungi COI-5P barcode sequences and nuclear mitochondrial pseudogenes (numts) using the coil analysis pipeline.

Supplementary File 6 – 'S6-validation_data' Folder containing the validation data inputs and outputs produced for the assessment of coil effectiveness.

Supplementary File 7 – 'S7-coil-vignette.pdf' A pdf demonstrating the functionality of the coil package. This is the static version of the package vignette, which is available through R. The original R markdown file is available for download from: https://github.com/CNuge/coil/tree/master/vignettes