Bioinformatics, 36, 2020, i353–i361
doi: 10.1093/bioinformatics/btaa447
ISMB 2020

EvoLSTM: context-dependent models of sequence evolution using a sequence-to-sequence LSTM

Dongjoon Lim and Mathieu Blanchette *

School of Computer Science, McGill University, Montreal, Quebec H3A 0G4, Canada

*To whom correspondence should be addressed.

Abstract

Motivation: Accurate probabilistic models of sequence evolution are essential for a wide variety of bioinformatics tasks, including sequence alignment and phylogenetic inference. The ability to realistically simulate sequence evolution is also at the core of many benchmarking strategies. Yet, mutational processes have complex context dependencies that remain poorly modeled and understood.

Results: We introduce EvoLSTM, a recurrent neural network-based evolution simulator that captures mutational context dependencies. EvoLSTM uses a sequence-to-sequence long short-term memory model trained to predict mutation probabilities at each position of a given sequence, taking into consideration the 14 flanking nucleotides. EvoLSTM can realistically simulate mammalian and plant DNA sequence evolution and reveals unexpectedly strong long-range context dependencies in mutation probabilities. EvoLSTM brings modern machine-learning approaches to bear on sequence evolution. It will serve as a useful tool to study and simulate complex mutational processes.

Availability and implementation: Code and dataset are available at https://github.com/DongjoonLim/EvoLSTM.

Contact: blanchem@cs.mcgill.ca

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

1 Introduction

Probabilistic models of sequence evolution are among the earliest areas of research in bioinformatics. These models aim to quantify the probability of specific types of mutations such as substitutions (Jukes *et al.*, 1969; Kimura, 1980) and small indels (Thorne *et al.*, 1991) in an evolving sequence. They provide the necessary probabilistic framework that allows formulating meaningful biological questions such as sequence alignment and phylogenetic inference. They also enable the realistic simulation of sequence evolution, and increasingly accurate tools have been introduced for this purpose over the years, including Rose (Stoye *et al.*, 1998), INDELible (Fletcher and Yang, 2009) and Evolver (Edgar *et al.*, 2019). By enabling the simulation of fake but evolutionarily related sequences, these approaches enable benchmarking and validating bioinformatics tools for problems such as multiple sequence alignment (Earl *et al.*, 2014; Edgar, 2004; Lassmann and Sonnhammer, 2005; Papadopoulos and Agarwala, 2007; Ramakrishnan *et al.*, 2018), ancestral genome reconstruction (Blanchette *et al.*, 2004b) and phylogenetic inference (Price *et al.*, 2010).

Mutation probabilities have long been known to be dependent on sequence context: the probability of a mutation happening at a certain site does not only depend on the type of mutation (e.g. transition versus transversion), but also on the nucleotides around it (Arenas, 2015). In some cases, this dependency is intrinsic to the mutational process itself. For example, perhaps the strongest type of context dependencies is the elevated C-to-T substitution rate in the context of a CpG dinucleotide, caused by the deamination of a methylated cytosine, transforming it to a thymine (Bird, 1980; Ehrlich and Wang, 1981). In other cases, it is the result of context-dependent DNA repair. Surrallés *et al.* (2002) found that the transcription-coupled repair machinery shows high localization on gene rich regions of the human genome, and Feng *et al.* (2002) found that transcription-coupled repair of cyclobutane pyrimidine dimers in Chinese hamster genome is context-dependent. Finally, it may be the result of context-dependent selective pressure. For example, coding sequence evolution is best captured by a 3-nucleotide codon model that accounts for the consequences of a nucleotide change on the amino acid encoded (Goldman and Yang, 1994). There is also a rich literature on co-evolution models in DNA (Makova and Hardison, 2015), RNA (Holmes, 2004) and proteins (Rodrigue *et al.*, 2005). Insertions and deletion rates also exhibit strong context dependencies, often linked to DNA polymerase slippage in locally repetitive sequences (Messer and Arndt, 2007).

Several types of models have been introduced to capture mutational context dependencies. Jensen and Pedersen (2000) proposed a Markov chain Monte Carlo method to calculate the substitution rate dependent on its two flanking bases inside a protein-coding DNA region. Later, a maximum likelihood analysis to infer nucleotide or dinucleotide mutation frequencies in non-coding regions of the genome was proposed (Arndt *et al.*, 2003). Siepel and Haussler (2003) introduced a context-dependent substitution model that has been developed to reflect context dependencies in both coding and non-coding regions of the genome, and a parameter-rich Bayesian network substitution model with parameters defined from flanking base contexts for genome-wide ancestral reconstruction were developed (Chachick and Tanay, 2012). However, these methods often limit the context effect to one flanking nucleotide/amino acid on each side due to the computational cost and sample size requirements growing exponentially with context size. The extent of

longer-range dependencies in mutation rates is thus poorly understood, although sparse Bayesian models (Ling *et al.*, 2019) have recently been introduced for that purpose and applied to viral evolution to determine the significance of certain configurations of nucleotide around mutation sites by inferring the mutation rate of 5-mers. Aggarwala and Voight (2016) proposed a statistical model that takes parameters from observed frequency of mutations within 5-mer or 7-mer, while Zhu *et al.* (2017) introduced a log-linear model for mutation frequency analysis that also considers up to 5-mers. It has also been found that variability of mutation in human genome between different populations of human genome are dependent on context beyond the immediate flanking bases (Aikens *et al.*, 2019).

To overcome the limitations of these probabilistic context-dependent substitution models, we propose an evolutionary model that harnesses recent developments in machine learning. In recent years, recurrent neural networks (RNNs) have enabled models to learn in the context of time series data much more efficiently than previously possible. In particular, the long short-term memory (LSTM) model architecture (Gers *et al.*, 2000) made it possible for the deep neural network in the area of natural language modeling to overcome the vanishing gradient problem of standard RNN (Sundermeyer *et al.*, 2012). More recently, RNN-encoder–decoder (Cho *et al.*, 2014) architectures, which separate the network into an RNN-encoder and RNN-decoder to better capture the context of the input sequence, were introduced, with applications to automated language translation. With a similar idea, sequence-to-sequence models (Sutskever *et al.*, 2014) have shown that using LSTM networks in the encoder–decoder architecture can further improve the capacity to capture the context of the input in neural machine translation.

In this paper, we introduce EvoLSTM, a sequence-to-sequence LSTM model of sequence evolution inspired by the aforementioned recent work in language modeling. We trained EvoLSTM from entire whole-genome primate alignments and show that it is able to capture context dependencies that are longer in range than what had been previous reported, for both substitutions and short indels. EvoLSTM's RNN evolutionary model paves the way for a variety of applications that rely on realistic modeling of sequence evolution.

2 Materials and methods

EvoLSTM is a machine-learning based probabilistic model of context-dependent sequence evolution. In its simplest form, it takes as input a sequence S of length K and outputs a randomly generated descendant sequence T that may either be identical to S or may differ from it through one or more substitutions or indels. In this section, we explain EvoLSTM's architecture, its training and evaluation.

2.1 Training data

EvoLSTM is trained from a set of pairs of aligned ancestral/descendant sequences. We used a 100-way alignment of whole vertebrate genomes (Blanchette *et al.*, 2004a; Miller *et al.*, 2007) and applied the Ancestors1.0 program (Blanchette *et al.*, 2004b; Diallo *et al.*, 2010) to infer maximum likelihood ancestral sequences genome-wide based on a simple context-independent substitution model. The approach has previously been shown to be highly accurate for most ancestors and, in particular, for primates (Blanchette *et al.*, 2004b). We then extracted induced pairwise alignments between the human genome and various primate ancestors: the old-world monkey ancestor (catarrhini; 0.03 expected substitution per site), and the simian ancestor (simiiformes; 0.05 exp. subst./site). Since our analyses did not reveal significant differences between using one or the other as training data, which proceeded to use the old-world monkey ancestor, which has the benefit of being the close enough to human to limit the risks of double mutations at the same site, while providing us with sufficiently many mutations to learn from.

Each pairwise alignment was then processed as follows (Fig. 1). First, gaps present in both sequences were removed. Second, each portion of 1 or 2 nucleotides in the descendant sequence that is aligned to consecutive gaps in the ancestors (hence the results of a 1 or 2 bp insertion) were combined with the previous descendant nucleotide and represented as a single character from an extended alphabet of size 86 (4 normal nucleotides + 16 dinucleotides for 1 bp insertions + 64 trinucleotides for 2 bp insertions + gap + dummy character). This allows treating insertions of size 1 or 2 as a substitution, which means that we do not need to assume we know ahead of time the position and size of an insertion. Since our focus is on short indels, we did not consider regions with larger indels, due to the exponential blow up in extended alphabet size that would be required.

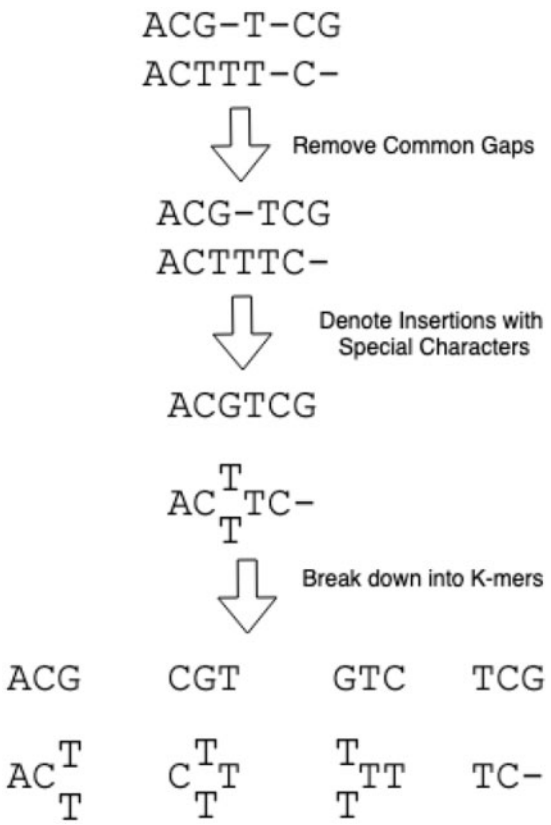

Fig. 1. Data preprocessing pipeline. An ancestor/descendant alignment is converted to a set of overlapping K-mer pairs from an extended alphabet of meta-nucleotides and gaps. Those K-mer pairs are used to train EvoLSTM

These modified ancestor-descendant alignments are sliced into K-mer pairs (for K ranging from 1 to 39). Those aligned K-mer pairs constitute the data from which EvoLSTM is trained. We selected the first 10 000 000 K-mer pairs from human chromosome 2 as the training set and the next 2 000 000 as validation set. Our test set consists of 149 860 432 K-mer pairs selected from all chromosomes except human chromosome 2, hence ensuring that the test set is entirely disjoint from the training and validation sets. This very large test set enables us to accurately estimate the accuracy of the trained models.

2.2 EvoLSTM's LSTM architecture

EvoLSTM is an RNN that takes as input a K-mer and outputs the probability of each of the nucleotides (and meta-nucleotides) in a hypothetical descendant sequence (Fig. 2). It is based on the LSTM architecture for RNNs, which was introduced to address the vanishing gradient problem of classical RNNs (Sundermeyer *et al.*, 2012). The LSTM cell corresponding to position t in the K-mer takes in an input value $x_{(t)}$ (the one-hot encoded version of meta-nucleotide at position t), as well as two types of recurrent states: the hidden state $h_{(t)}$ and the cell state $c_{(t)}$, which are vectors of predetermined dimensions. Following Gers *et al.* (2000), these states are combined with the input and are regulated by trainable input (W_{Inuc} and W_{Ih}),

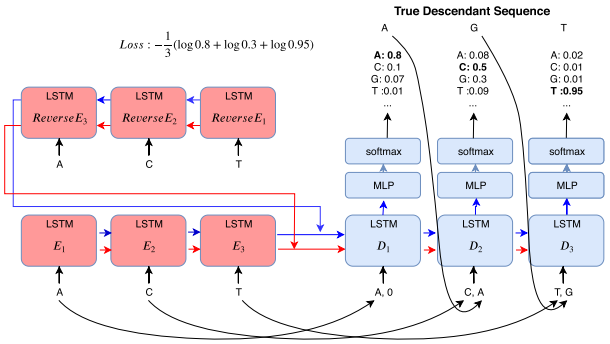

Fig. 2. EvoLSTM is a sequence-to-sequence bidirectional LSTM model made of an encoder (left portion, shown in red) and a decoder (right portion, shown in blue box). An ancestral sequences (here, ACT) are given as input to the encoder. Each cell recurrently receives information from the previously cell, in the form of a hidden state vector (blue arrows) and cell state vector (red arrows), and combines it with the one-hot encoded ancestral nucleotide to update those two vectors. The output of the encoder is the concatenation of the hidden and cell state vectors produced by the forward and reverse directions. Those vectors are an encoding of the sequence context, optimized so as to be maximally informative for the decoder. In the decoder, each cell receives as input (i) the one-hot encoded ancestral nucleotide and the previous descendant nucleotide generated by the model, as well as the state and cell vectors from the previous cell. It updates those two vectors and passes them to the next cell, but also feeds the hidden state vector to an MLP, which output a probability distribution over descendant characters (nucleotides or gap) at that position. A character is randomly drawn from that distribution, emitted and passed onto the next LSTM decoder cell

output (W_{Onuc} and W_{Oh}) and forget gates (W_{Fnuc} and W_{Fh}) weight vectors. Each LSTM cell remembers values of the hidden state and cell state over time intervals and the three gates modify the information received from the previous time step (Sundermeyer *et al.*, 2012).

Specifically, we have

$$f_{(t)} = \sigma(W_{Fnuc} \cdot x_{(t)} + W_{Fh} \cdot h_{(t-1)} + b_F)$$
$$i_{(t)} = \sigma(W_{Inuc} \cdot x_{(t)} + W_{Ih} \cdot h_{(t-1)} + b_I)$$
$$o_{(t)} = \sigma(W_{Onuc} \cdot x_{(t)} + W_{Oh} \cdot h_{(t-1)} + b_O).$$

Finally, the outputs of each gate are combined to update the hidden state and the cell state of the current cell:

$$c_{(t)} = f_{(t)} \odot c_{(t-1)} + i_{(t)} \odot \tanh(W_{Cnuc} \cdot x_{(t)} + W_{Ch} \cdot h_{(t-1)} + b_c)$$
$$h_{(t)} = o_{(t)} \odot \tanh(c_{(t)}),$$

where $\odot$ denotes the element-wise product.

2.3 Sequence-to-sequence model

Borrowing the idea from sequence-to-sequence learning (Sutskever *et al.*, 2014), EvoLSTM is composed of connected LSTM-based encoder and decoder networks (Fig. 2). The encoder is made of two LSTM looking at the same input K-mer but in opposite directions. Their last output hidden states h_K and cell states c_K are concatenated and passed to the decoder network, which uses it, along with the ancestral K-mer itself, to generate a descendant sequence. h_K and c_K capture the context information that will be used to inform the decoder.

The decoder consists of an LSTM similar to the encoder, coupled with a fully connected neural network. Each cell receives the cell and hidden states from the previous cell, except for the first cell, which obtains those from the encoder (h_K and c_K). Cell t from the decoder is used to predict a probability distribution over meta-nucleotides at position t of the descendant sequence. The hidden state is fed to a fully connected multi-layer perceptron (MLP) with 86 outputs, to which a softmax function is applied to obtain a normalized probability distribution. Also, in addition to receiving meta-nucleotide x_t as input, cell t (for $t > 1$) receives the observed (during training) or sampled (when using the network as a generative model) descendant nucleotide from position $t-1$, providing it with information about the evolutionary event that took place at the previous position. This is particularly critical to enable indels spanning more than one nucleotide.

2.4 Training EvoLSTM

We trained our model using the cross-entropy (CE) (negative log-likelihood) as the loss function to minimize. In short, we aim to minimize

$$-\sum_{(A,D)\in \text{Training set}} \log p(D|A),$$

where A is an input ancestral K-mer and D is its descendant K-mer. Trainable weights include the input, output, forget, cell state and hidden state weight matrices and bias terms for both the encoder and decoder, as well as the weights of the MLP in the decoder. Training is carried out using the Adam optimizer (Kingma and Ba, 2014) with a learning rate of 0.0001. We used batch learning with batch size 1024. To reduce overfitting, we used early stopping, ending training when the validation loss did not decrease for five consecutive epochs. To investigate the effect of context size, models were trained with six different values of K: 1, 5, 11, 15, 21 and 39.

Hyper-parameters of the model were set based on a compromise between training time and accuracy on the validation set. The size of the hidden and cell state vectors was set to 512. The MLP consists of 1 hidden layer of 86 neurons with ReLU (rectified linear activation unit) (Nair and Hinton, 2010) activation function and the output layer of 86 neurons.

2.5 Using EvoLSTM as an evolution simulator

To use a trained EvoLSTM model to simulate the evolution of a given ancestral K-mer sequence A, the model is used as described previously, with a few small modifications. At each position t, a meta-nucleotide is sampled from the distribution generated by the model at that position. It is that nucleotide (rather than the true descendant nucleotide, as was done during training phase) that is passed as input to the next cell. In contrast, in the context machine translation (Sutskever *et al.*, 2014), the goal is generally not to sample from a distribution over translations, but instead to identify the maximum-likelihood translation, which is achieved via a greedy or beam search algorithm (Neubig, 2017).

In order to use EvoLSTM to simulate the evolution of ancestral sequences longer than K, we proceed as follows. EvoLSTM breaks down the ancestral genome sequence of interest into K-mers similarly to the data preprocessing step described in Figure 1 but without overlap between each K-mer. Denote those ancestral K-mers as $S_a = [k_1, k_2 \ldots k_{n-1}, k_n]$. Each K-mer is given as input to EvoLSTM, in the order in which they appear in the sequence. To sample descendant K-mer $d_i (i > 1)$, EvoLSTM uses the cell and hidden state vectors obtained from passing K-mer k_i to the encoder network together with the last simulated nucleotide obtained from d_{i-1} to be passed down to the decoder network. Once all descendant K-mers are obtained, they are concatenated to yield the complete descendant sequence.

The simulation processed described until now only allows mimicking sequence evolution over a branch of the same length as that corresponding to the pair of ancestor/descendant genomes used for training. Let us call that branch length the *unit* branch length. To simulate the evolution of branches of non-unit length, we proceed as follows. Suppose the target branch length is λ units long. Decompose λ into its integer and factional portions: $\lambda = Int(\lambda) + Frac(\lambda)$. We apply EvoLSTM $Int(\lambda)$ times, each time using it to re-evolve the sequence produced from the previous iteration. The resulting K-mer (which would be the ancestral K-mer if $\lambda < 1$) is then fed one last time through EvoLSTM, but this time rejecting a proposed mutation with probability $Frac(\lambda)$. Supplementary Figure S1 shows that this iterative process captures well context dependencies of evolutionary events over longer branches.

2.6 Evaluation

2.6.1 Baseline approaches

Past studies on evolution models have mainly focused on immediate neighbor context-dependent substitutions and are thus not directly comparable to the EvoLSTM. Instead, we implemented two baseline models. Our *table-based* approach is perhaps the most natural context-dependent evolutionary model: It simply approximates and saves $\Pr[D|A]$ as $N(A,D)/N(A)$, where $N(A, D)$ and $N(A)$ are the observed frequencies of (D, A) and $(D, *)$ alignments in the training set. As K increases, $N(A)$ can become too small to provide a meaningful probability estimate. If $N(D,A) = 0$, we symmetrically trim the context sequences (by one nucleotide at each end), until we get $N(D,A) > 0$. This approach enables this algorithm to use large context sizes when sufficient data exist, but to returns to smaller context sizes when it does not.

We also implemented a second machine-learning approach using a standard bidirectional LSTM network (Fig. 3) coupled to a 1-hidden layer MLP (ReLU activation), with an output layer of 86 neurons. Unlike our EvoLSTM model, this model does not consider the prediction made in the previous time step and can make predictions based only on the bidirectional input context. The learning rate, hidden and cell state weight size, optimizer, initial learning rate, batch size and all other training details are identical to the sequence-to-sequence EvoLSTM.

2.7 Implementation

EvoLSTM was implemented using tf.keras (Chollet *et al.*, 2015) LSTM layers in Tensorflow 2.0 (Abadi *et al.*, 2015). Biopython (Cock *et al.*, 2009) was used for reading the MAF file and preprocessing genome sequences. All relevant code is available at https://github.com/DongjoonLim/EvoLSTM.

EvoLSTM is easy to train from whole-genome alignments and inferred ancestral sequences. It comes with code for interpretation and visualization of the models learned. It is also able to use a trained model to randomly evolve a given input sequence using substitutions and short indels. As such, it will easily integrate into more general genome evolution simulators such as Evolver (Edgar *et al.*, 2019).

3 Results

This section begins with the assessment of the accuracy of EvoLSTM and baseline models, followed by an empirical analysis of context-dependent mutation probabilities learned by EvoLSTM.

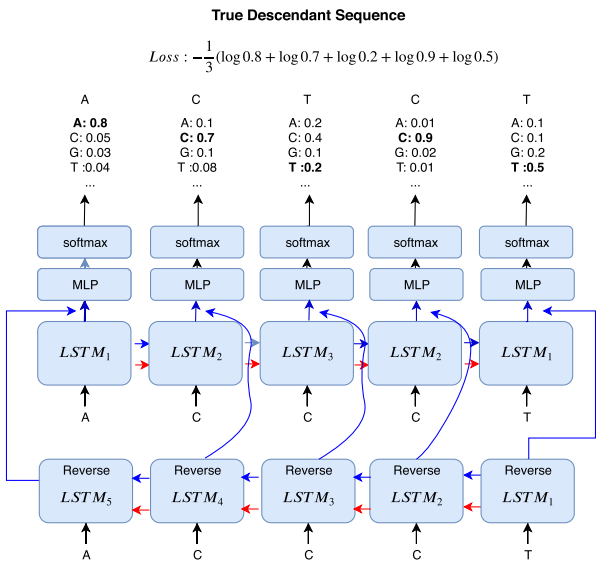

Fig. 3. Baseline bidirectional LSTM model structure. The two hidden states emitted from both directions are concatenated and passed down to an MLP

3.1 Model performance

We first evaluated the ability of different approaches to properly estimate mutation probability in a context-dependent manner (Fig. 4). The models investigated (see Section 2) included our biLSTM and EvoLSTM seq2seq models, as well as a simpler adaptive frequency-based approach, each with context sizes $K \in \{1, 5, 11, 15, 21, 39\}$. Each model was trained on 10 000 000 aligned K-mer pairs extracted from the whole-genome alignment of the computationally reconstructed old-world monkey genome to the human genome (from human chromosomes 2), and evaluated on a similar but much larger test set obtained from chromosomes 1 and 3–22, containing >149 million K-mer pairs. This atypical imbalance between the size of the training and test sets is intentional; having a very large test set allows us to accurately estimate context-dependent mutation probabilities using the simple frequency-based approach, to then be able to assess how the different approaches proposed can learn context dependencies from a relatively smaller training set.

We assessed the ability of a model to accurately capture context-dependent mutation probabilities using the CE of the test data, which are equivalent to the negative log-likelihood of the data given the model. CE values that are reached using no context at all ($K = 1$) are much worse than those obtained with larger values of K, confirming that context dependencies are strong. The adaptive frequency table-based approach is limited in capturing long-range dependencies because of the exponential amount of data it would require in order to do so. In contrast, EvoLSTM (1 layer) is able to fully take advantage of large context sizes, reaching a minimal CE at $K = 15$. Larger values of K result in worse CE values, possibly because as the LSTM network becomes larger, gradients become unstable (Greff *et al.*, 2017). However, this does not mean that context sizes larger than $K = 15$ do not have an incidence on mutation probabilities. Overall, the difference in CE values obtained [0.18 for EvoLSTM ($K = 15$) versus 0.195 for table-based ($K = 21$)] is notable and reliably reproduced over different restarts. Note that the reason the performance of the adaptive table-based approach remains stable for large values of K is that this approach adaptively trims each K-mer until its count in the training set is sufficiently large to make accurate probability estimations; in practice, for $K \geq 15$, most K-mers get trimmed to 15-mers or shorter.

We also tested whether it may be beneficial to add a second LSTM layer to EvoLSTM, which is an approach that has been shown to be beneficial in language modeling (Sutskever *et al.*, 2014). However, this did not improve the performance, and was much longer to train. We also observe that the effect of capturing contexts by separating the encoder from the decoder in the sequence-to-sequence model used in EvoLSTM is important since the CE of EvoLSTM is much lower than that of the baseline

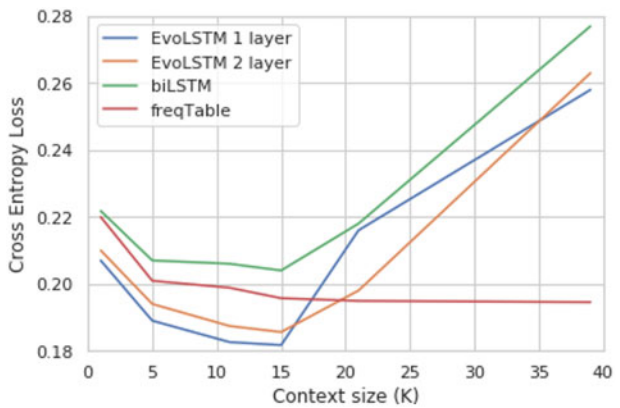

Fig. 4. CE loss (lower is better) of the test set (>149 million K-mers, corresponding to alignments of portions of the old-world monkeys' ancestral genome against the human genome), with different context sizes K. The models compared are two baseline models (adaptive frequency table and biLSTM) as well as two versions of EvoLSTM, with one or two biLSTM layers

biLSTM, across all context sizes. This suggests that relatively long-range context dependencies exist, and that EvoLSTM is able to capture many.

3.2 Flanking context strongly impacts mutation probability

We next aimed to characterize the impact of long-range context on mutation probability, and better understand the ability of EvoLSTM to take this context into consideration. We used a trained EvoLSTM ($K = 15$) to simulate the evolution of a set of 149 860 432 K-mers from an ancestral old-world monkey sequence. We then tabulated the frequency of simulated mutations in each of the 256 different contexts of the form wxNyz, i.e. including two flanking bases on each side of the mutating base, and contrasted those frequencies to those observed in actual alignments of the same regions.

Figure 5 shows the results for a subset of the 16 possible substitutions, 4 possible deletions and 4 possible insertions. See also Supplementary Tables S1–S4 for full results. Several observations emerge. First, EvoLSTM efficiently captures and simulates context dependencies of that size, with correlation coefficients ranging from 0.83 to 0.99 for substitutions. In particular, it clearly and accurately captures the well-known CpG to TpG mutation of methylated C's (Jabbari and Bernardi, 2004), or the equivalent CpG to CpA from the reverse strand, assigning significantly higher mutation probability (0.1–0.25) to these mutations when in the right dinucleotide context versus in other contexts (<0.05). Notably, even within the CpG dinucleotide context, the probability of C-to-T mutation is strongly dependent on the broader context. For example, CGCGz contexts are two times less mutagenic than AT-rich contexts. This may be explained by the presence of CpG islands in the genome, which is generally unmethylated and hence both CpG rich and substitution poor. Non-CpG contexts also exhibit strong context dependencies. For example, A-to-G transitions show a greater than 10-fold increase in the probability in the contexts of CAATw versus xAAAw.

Figure 6 illustrates this phenomenon in more details in the case of CpG dinucleotide (left) and nucleotide deletion (right). The massive difference in CpG versus non-CpG C-to-T substitution probabilities is only the beginning of a dive into further and further refinements of context dependencies. The ACG context is 70% more mutagenic than the TCG context. There is then a 35% difference in C-to-T mutagenicity between different xACGy contexts, and a 2-fold difference between high and low mutagenic xGACGGy contexts. Throughout, the correspondence between the predicted and observed mutation probabilities remains quite high, until the context size considered becomes less relevant to mutation probability (e.g. for xACCTTy). Deletions (right panel) exhibit these long-range dependencies even more strikingly, with a 20-fold difference in A-deletion probabilities between highly mutagenic AAAAAAA context and conservative context TCCAGCG.

Transversions generally show a slightly weaker context-dependency (5-fold variation between most and least mutagenic contexts). Because they are also rarer in general, prediction accuracy is slightly worse due to the relatively small number of training examples.

Figure 5 (bottom row) also shows the context-dependency of 1-nucleotide insertions and deletions. Those also display a very strong context-dependency. For example deletions of a T are roughly 8–12 times more likely in T-rich contexts than in GC-rich contexts such as zCTGz or zGTCz contexts. Insertions show a similar pattern of elevated probabilities for insertions of nucleotides in a context that resembles them, which is consistent with the well-known DNA polymerase slippage model (Messer and Arndt, 2007).

To study the impact of the broader context, we investigated how the probabilities of specific substitutions and indels vary as we look at increasingly large context sizes (Supplementary Fig. S7). Consider mutation $M \rightarrow N$, where M and N are nucleotides or gaps, which is taking place in the context $xMy \rightarrow xNy$, where x and y are context nucleotides sequences of length greater or equal to zero (hence context size $K = |x| + |y| + 1$). A meaningful assessment of the degree to which considering an extra context nucleotide at each end (i.e. $axMyb \rightarrow axNyb$) affects the predicted mutation probability is the log-odds ratio of the long context model to the short context model:

$$LOR(M \rightarrow N; axMyb; xMy) = \log(\Pr[axNyb|axMyb]/\Pr[xNy|xMy]).$$

If the mutation $M \rightarrow N$ does not depend on distant context nucleotides a and b, we get $LOR \approx 0$. However, if the presence of a and b significantly increases (resp. decreases) the odds of mutation, LOR takes on a positive (resp. negative) values.

When K is relatively large ($K > 5$), verifying EvoLSTM's predictions about the frequencies of mutations in specific contexts become difficult, because our test data does not have sufficiently many examples of each mutation/context pairs for accurate estimation. Here, we take advantage of the fact that most mutational processes are agnostic of strandedness, which should result in context-dependencies to be invariant to reverse complementation:

$$LOR(M \rightarrow N; axMyb; xMy) \approx LOR(M' \rightarrow N'; b'y'Mx'a'; y'M'x'),$$

where prime indicates reverse complement. This provides us with a way to verify the internal consistency of the model, without the need for a test set. This serves as a proxy for evaluating the tool's accuracy, because high correlations would be unlikely to arise by chance.

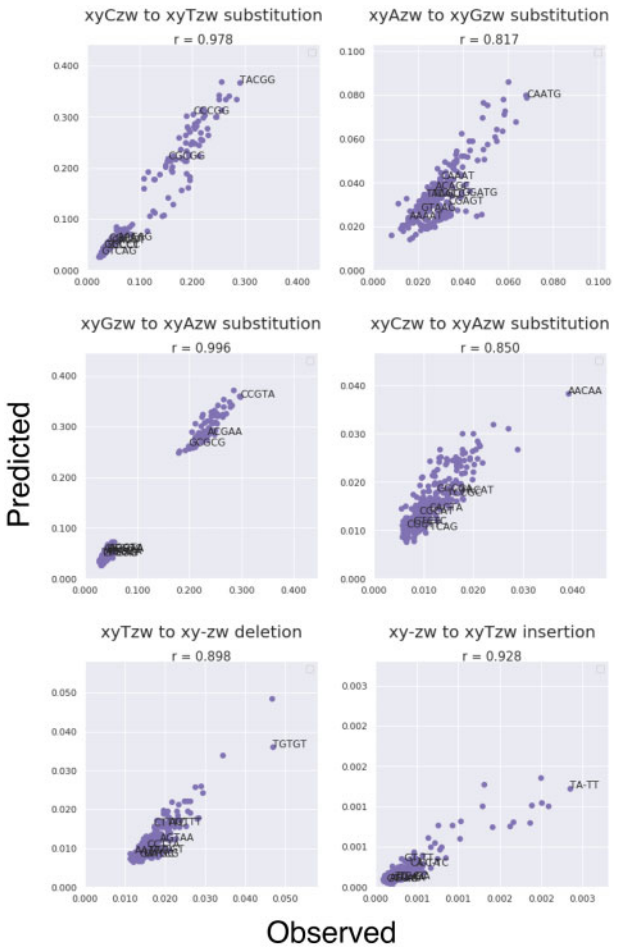

Fig. 5. Comparison of the observed and EvoLSTM-predicted mutation probabilities for context size $K = 5$. Each graph shows the results for a given mutation occurring at the center of the K-mer. Each plot has 256 points, corresponding to the 256 possible contexts for that mutation. Some of the contexts are labeled. The Pearson correlation coefficient r is given for each case

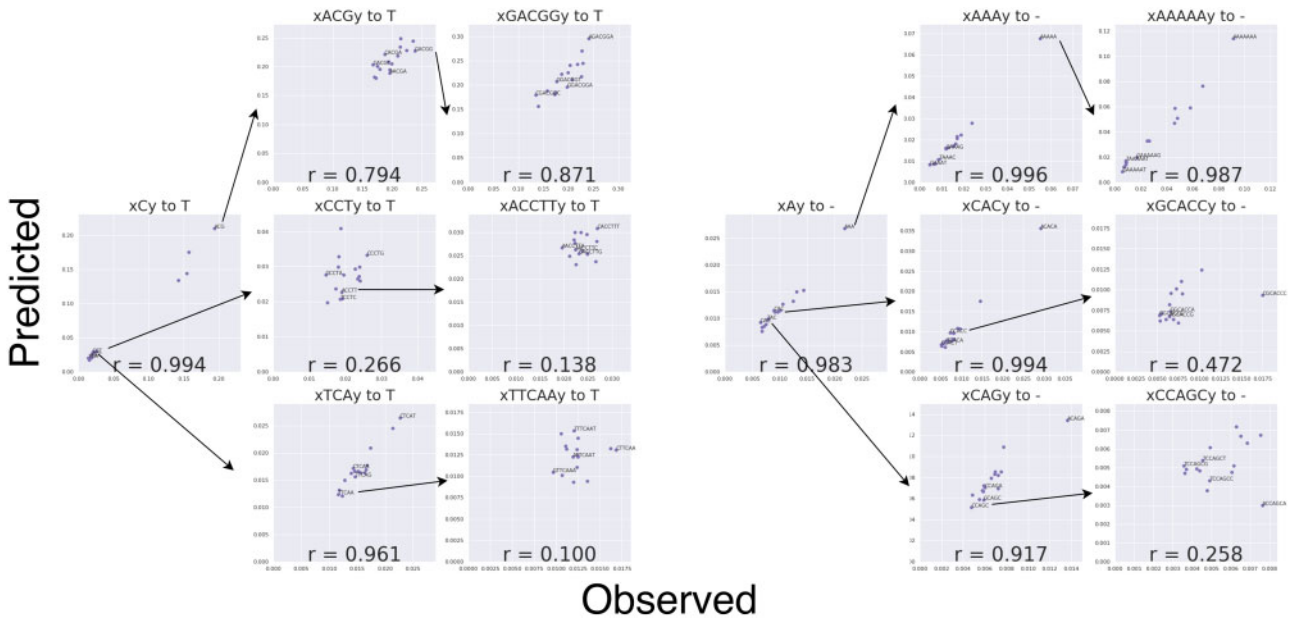

Fig. 6. EvoLSTM-predicted mutation probability and observed mutation probability in increasingly large contexts. Each dot represents one of the 16 possible context extensions. Out of those flanking base configurations, the top row shows the effect of adding the most mutagenic flanking bases, the second row shows the effect of adding the median mutagenic flanking bases and the bottom row shows the effect of the least mutagenic flanking bases

Figure 7 shows how LOR values for mutation/context pairs relate to the LOR values for their reverse complement. For $K \leq 5$, LOR values are highly consistent between a mutation/context pair and its reverse complement. For example, $LOR(T \rightarrow A; TTTAA; TTA) = 0.55$, suggesting that a T-to-A substitution is approximately $10^{0.55} = 3.5$ times more likely in the context TTTAA than in the shorter context TTA. The reverse-complement mutation/context pair displays a similarly strong bias, with $LOR(A \rightarrow T; TTAAA; TAA) = 0.44$. With context size $K = 7$, the correspondence between reverse complements becomes less strong, suggesting that EvoLSTM's estimations may be less accurate. Nonetheless, several results are striking and reproducible across reverse complements. Surprisingly, a C-to-A substitution in the context CCCCCCC is 3 times more likely than in the shorter CCCCC context. A-to-T substitutions are 5 times less likely in the context of ATTAAAG than in the context of TTAAA. These results show that long-range context dependencies are strong for certain mutations, and that EvoLSTM is able to capture many of them, although the task becomes increasingly difficult as K increases. Similar results are observed for deletions, but long-range context-dependencies for insertions appear to be less reliably captured, probably because of their rarity in our training set.

3.3 Context-dependencies in other mammals and in plants

To demonstrate the applicability of EvoLSTM outside of primates, we used it to learn mutation context-dependencies in other species. First, we trained a model on bats sequences (training set: 10 million K-mers; test set: 158 million K-mers), using the branch from the most recent common ancestor David's Myotis bat (*Myotis davidii*) and Microbat (*Myotis lucifugus*) to descendant Microbat. Supplementary Figure S9 shows that EvoLSTM is able to capture the same type of dependencies as in primates, although the correlations observed are weaker. We attribute those differences to the fact that the number of mutated sites available for training is substantially lower in bats, despite the branch lengths being similar to those used primates. This is likely an artifact of the way the multiple genome alignment used for ancestral genome reconstruction was built, using human as a reference, which results in highly diverged bat sequences to sometimes be missing from the alignment. Nevertheless, the predicted context dependencies learned in primates and bats are quite similar (Supplementary Fig. S3).

We then repeated the analysis on plants (*Brassicaceae*), using a whole-genome alignment produced by Haudry *et al.* (2013). Here, we used as ancestral sequence the most recent ancestor of *Arabidopsis thaliana* and *Arabidopsis lyrata*, reconstructed using *Capsella rubella* as an outgroup, and studied its evolution toward the *A.lyrata* genome. We excluded coding regions, resulting in 10 million examples being used for training, but only 6 million for testing. Again, EvoLSTM is able to detect strong context dependencies, especially for insertions and deletions. The correlation coefficients of predicted and observed (in the test set) mutation frequencies are lower than in human, which we attribute in part to the fact that the test set used to estimate observed mutation frequencies is >20 times smaller than in mammals. Notably, we observe an absence of CpG to TpG elevated substitution rate, due to the fact that DNA methylation is rare in plants, outside of transposable elements (Zhang *et al.*, 2018).

3.4 Running time

EvoLSTM was trained on an Intel(R) Xeon(R) Silver 4210 CPU @ 2.20 GHz CPU and NVIDIA GeForce RTX 2080Ti GPU. Training on a set of 10 000 000 15-mer examples took on average 1374 s per epoch with the batch size of 1024 and the hidden state size of 512. The training was stopped after 132 epochs.

4 Discussion and conclusion

Context-dependent mutation rates have been known and documented for a long time, starting with the elevated CpG to TpG substitution rate (Bird, 1980; Ehrlich and Wang, 1981), and more recently in many other cases (Arndt and Hwa, 2005; Averof *et al.*, 2000; Messer and Arndt, 2007; Morton, 2003; Siepel and Haussler, 2003). While several computational models have been proposed to characterize these dependencies [e.g. hidden Markov models (Siepel and Haussler, 2003) or Bayesian networks (Chachick and Tanay, 2012; Cohn *et al.*, 2010)], those are generally unable to capture complex long-range dependencies. The EvoLSTM model introduced here builds on prior work in deep learning and natural language processing to learn and reveal such dependencies.

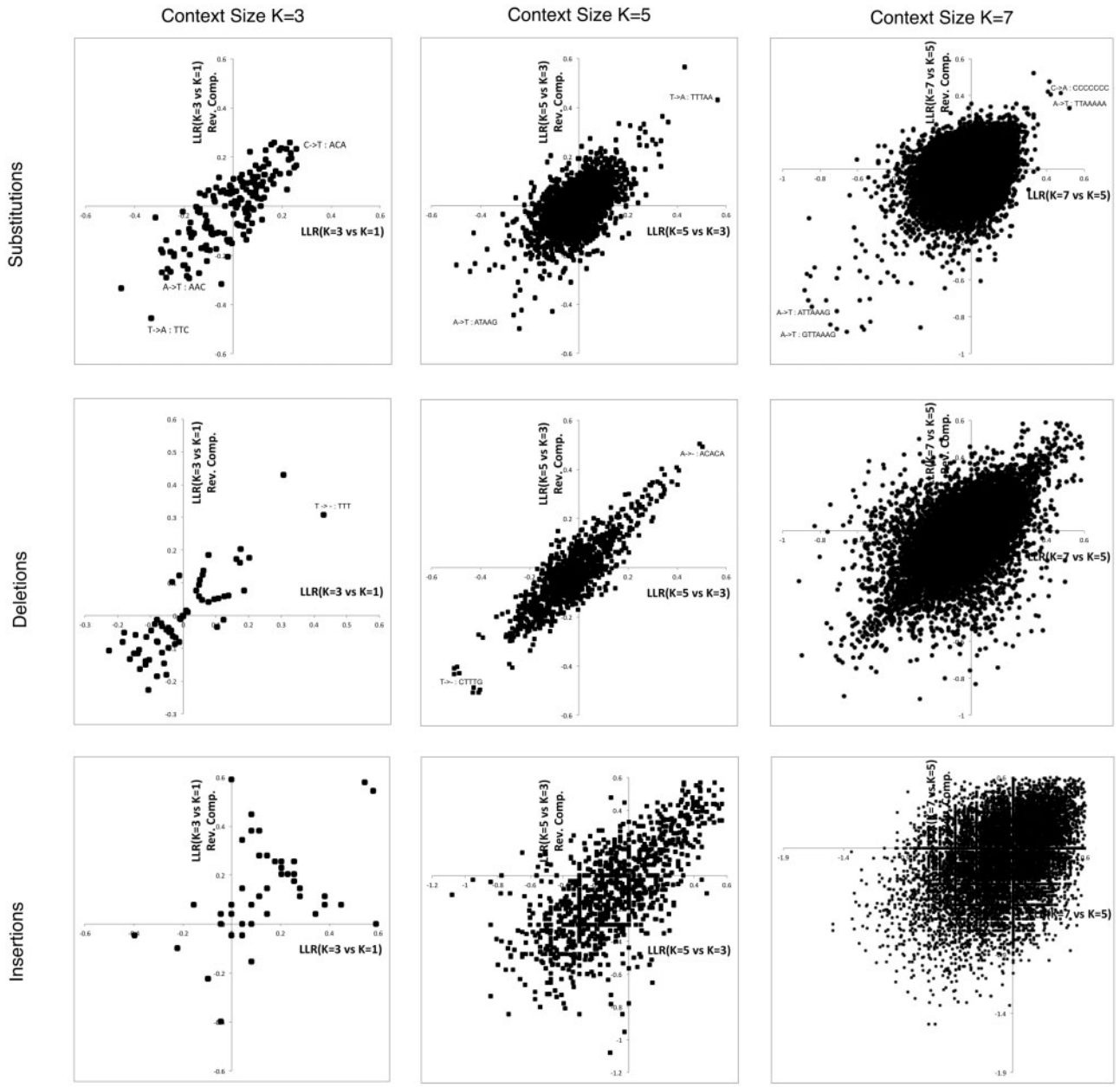

Fig. 7. Increasing context size allows capturing strong dependencies not captured at lower context sizes. For each type of mutations (substitutions, deletions, insertions) and each context size $K = 3, 5, 7$, we show the log-odds ratio of a mutation/context pair in a model with context K versus one with context size $K-2$. Each dot corresponds to a mutation/context pair, with x-coordinate being the log-odds ratio of the mutation in those two context sizes (K versus $K-2$) and the y-axis being the same measure, but for the reverse complement of that mutation/context pair. Since most context-dependencies would be expected to be strand-independent, one would expect a strong correlation between those two values

The ability to efficiently learn and model context-dependent mutation rates is of high importance for several tasks. First, substitution models are at the core of sequence alignment tasks. Most commonly used alignment algorithms use context-independent models [e.g. Needleman-Wunsch algorithm (Needleman and Wunsch, 1970), Blast (Altschul *et al.*, 1990) and their variations (Schwartz *et al.*, 2000; Smith and Waterman, 1981)] or consider limited context for codon-based alignment (Ranwez *et al.*, 2011). This is partly due to the algorithmic challenge linked to computing maximum likelihood alignments under context-dependent substitution or indel models (Hickey and Blanchette, 2011). Yet many scoring-scheme agnostic pairwise and multiple alignment heuristics have been proposed recently, e.g. using reinforcement learning (Jafari *et al.*, 2019; Mircea *et al.*, 2018; Ramakrishnan *et al.*, 2018). This paves the way for the possible adoption of complex mutation models such as EvoLSTM. Using accurate substitution and indel models is particularly important when aligning highly diverged sequences, where using the right model enables both more accurate alignment computation and higher remote homology detection, both of which are of high importance for whole-genome alignment (Blanchette *et al.*, 2004a) and ancient transposable element detection (RepeatMasker; Smit *et al.* http://www.repeatmasker.org). Accurate modeling of context dependencies is also important to obtain improved sequence evolution simulators [such as Evolver (Edgar *et al.*, 2019)], which are instrumental in benchmarking a variety of bioinformatics tools such as whole-genome aligners (Earl *et al.*, 2014). These models are also relevant to phylogenetic inference, where the choice of mutation models has been shown to have a high impact on the accuracy of the trees inferred, especially for highly divergent species (Delsuc *et al.*, 2005). Finally, a more detailed study of the models learned by

EvoLSTM is likely to reveal valuable information about mutagenesis and DNA repair. In particular, different types of cancer have been shown to be associated with different mutational signatures (Helleday *et al.*, 2014); a detailed analysis of EvoLSTM models trained on such cancer mutations may help reveal the mechanisms at play.

Many potentially fruitful directions may be explored to improve EvoLSTM's accuracy and scalability. Attention mechanisms have shown promising results in neural machine translation (Bahdanau *et al.*, 2014) for capturing larger sentence contexts and could be beneficial in our context. Other directions may include using transformers (Vaswani *et al.*, 2017), which have recently revolutionized the field of natural language processing, as well as word embedding (Mikolov *et al.*, 2013). Code optimization should also enable EvoLSTM to be trained on larger datasets; memory requirements currently limit us to using at most 10 million training examples.

Several new biological applications would also be of interest. First, one may consider training an ensemble of models, to capture different types of genomic contexts (methylated versus non-methylated, or transcribed versus non-transcribed, protein-coding versus non-coding regions, etc.), which are believed to have different mutational signatures either due to different mutational or DNA repair processes, or to natural selection. Extending EvoLSTM to other types of mutational events such as transposable element insertions [which have been shown to be highly context-dependent (Beggs *et al.*, 2000; Wall *et al.*, 1999)] and tandem or segmental duplication would also be worthwhile.

Applying EvoLSTM to the study of the mutational processes at play outside of primates would also be valuable. One challenge in that direction is data availability. To train EvoLSTM, one needs the possibility of accurately reconstructing an ancestral sequence, which is only feasible if at least two relatively closely related species and a close outgroup are available. These genomes need to be sufficiently closely related that they can be accurately aligned to each other, but diverged enough that the number of mutational events available for training is sufficient. As such, it works best for large genomes with a densely populated phylogenetic tree.

In conclusion, machine-learning advances have only begun to impact evolutionary biology and genomics, but this represents an application area of potentially high impact, due to the complexity of the mechanisms at play and a large amount of genomic data available to train sophisticated models. EvoLSTM represents the first step in that direction, enabling a detailed study of context-dependent mutational mechanisms and their integration in sequence evolution simulations, with applications in genomics, evolution, phylogenetics and potentially human health.

Acknowledgement

We thank Sean McRae, Zichao Yan and Elliot Layne for useful suggestions on the manuscript.

Funding

This work was funded in part by a Genome Canada Large-Scale Applied Research Project (LSARP) grant and by a Discovery grant from the National Science and Engineering Research Council of Canada. We thank Calcul Quebec and Compute Canada for computing resources.

Conflict of Interest: none declared.

References

Abadi,M. *et al.* (2015) TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. http://tensorflow.org (21 January 2020, date last accessed).

Aggarwala,V. and Voight,B.F. (2016) An expanded sequence context model broadly explains variability in polymorphism levels across the human genome. *Nat. Genet.*, **48**, 349–355.

Aikens,R.C. *et al.* (2019) Signals of variation in human mutation rate at multiple levels of sequence context. *Mol. Biol. Evol.*, **36**, 955–965.

Altschul,S.F. *et al.* (1990) Basic local alignment search tool. *J. Mol. Biol.*, **215**, 403–410.

Arenas,M. (2015) Trends in substitution models of molecular evolution. *Front. Genet.*, **6**, 319.

Arndt,P.F. and Hwa,T. (2005) Identification and measurement of neighbor-dependent nucleotide substitution processes. *Bioinformatics*, **21**, 2322–2328.

Arndt,P.F. *et al.* (2003) DNA sequence evolution with neighbor-dependent mutation. *J. Comput. Biol.*, **10**, 313–322.

Averof,M. *et al.* (2000) Evidence for a high frequency of simultaneous double-nucleotide substitutions. *Science*, **287**, 1283–1286.

Bahdanau,D. *et al.* (2014) Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv*:1409.0473

Beggs,M.L. *et al.* (2000) Mapping of IS6110 insertion sites in two epidemic strains of *Mycobacterium tuberculosis*. *J. Clin. Microbiol.*, **38**, 2923–2928.

Bird,A.P. (1980) DNA methylation and the frequency of CpG in animal DNA. *Nucleic Acids Res.*, **8**, 1499–1504.

Blanchette,M. *et al.* (2004a) Aligning multiple genomic sequences with the threaded blockset aligner. *Genome Res.*, **14**, 708–715.

Blanchette,M. *et al.* (2004b) Reconstructing large regions of an ancestral mammalian genome in silico. *Genome Res.*, **14**, 2412–2423.

Chachick,R. and Tanay,A. (2012) Inferring divergence of context-dependent substitution rates in drosophila genomes with applications to comparative genomics. *Mol. Biol. Evol.*, **29**, 1769–1780.

Cho,K. *et al.* (2014) Learning phrase representations using RNN encoder-decoder for statistical machine translation. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724–1734. Association for Computational Linguistics, Doha, Qatar.

Chollet,F. *et al.* (2015) Keras. https://github.com/fchollet/keras (21 January 2020, date last accessed).

Cock,P.J.A. *et al.* (2009) Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics*, **25**, 1422–1423.

Cohn,I. *et al.* (2010) Mean field variational approximation for continuous-time Bayesian networks. *J. Mach. Learn. Res.*, **11**, 2745–2783.

Delsuc,F. *et al.* (2005) Phylogenomics and the reconstruction of the tree of life. *Nat. Rev. Genet.*, **6**, 361–375.

Diallo,A.B. *et al.* (2010) Ancestors 1.0: a web server for ancestral sequence reconstruction. *Bioinformatics*, **26**, 130–131.

Earl,D. *et al.* (2014) Alignathon: a competitive assessment of whole-genome alignment methods. *Genome Res.*, **24**, 2077–2089.

Edgar,R.C. (2004) MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Res.*, **32**, 1792–1797.

Edgar,R.C. *et al.* (2019) Evolver. http://www.drive5.com/evolver (21 January 2020, date last accessed).

Ehrlich,M. and Wang,R. (1981) 5-methylcytosine in eukaryotic DNA. *Science*, **212**, 1350–1357.

Feng,Z. *et al.* (2002) Transcription-coupled DNA repair is genomic context--dependent. *J. Biol. Chem.*, **277**, 12777–12783.

Fletcher,W. and Yang,Z. (2009) INDELible: a flexible simulator of biological sequence evolution. *Mol. Biol. Evol.*, **26**, 1879–1888.

Gers,F.A. *et al.* (2000) Learning to forget: continual prediction with LSTM. Continual prediction with lstm. *Neural computation*, **12**, 2451–2471.

Goldman,N. and Yang,Z. (1994) A codon-based model of nucleotide substitution for protein-coding DNA sequences. *Mol. Biol. Evol.*, **11**, 725–736.

Greff,K. *et al.* (2017) LSTM: a search space odyssey. *IEEE Trans. Neural Networks Learn. Syst.*, **28**, 2222–2232.

Haudry,A. *et al.* (2013) An atlas of over 90,000 conserved noncoding sequences provides insight into crucifer regulatory regions. *Nat. Genet.*, **45**, 891–898.

Helleday,T. *et al.* (2014) Mechanisms underlying mutational signatures in human cancers. *Nat. Rev. Genet.*, **15**, 585–598.

Hickey,G. and Blanchette,M. (2011) A probabilistic model for sequence alignment with context-sensitive indels. *J. Comput. Biol.*,18 **11**, 1449–1464.

Holmes,I. (2004) A probabilistic model for the evolution of RNA structure. *BMC Bioinform.*, **5**, 166.

Jabbari,K. and Bernardi,G. (2004) Cytosine methylation and cpg, tpg (cpa) and tpa frequencies. *Gene*, **333**, 143–149.

Jafari,R. *et al.* (2019) Using deep reinforcement learning approach for solving the multiple sequence alignment problem. *SN Appl. Sci.*, **1**, 592.

Jensen,J.L. and Pedersen,A.-M.K. (2000) Probabilistic models of DNA sequence evolution with context dependent rates of substitution. *Adv. Appl. Prob*., **32**, 499–517.

Jukes,T.H. *et al*. (1969) Evolution of protein molecules. *Mammalian Protein Metab*., **3**, 132.

Kimura,M. (1980) A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences. *J. Mol. Evol*., **16**, 111–120.

Kingma,D.P. and Ba,J. (2014) Adam: a method for stochastic optimization. In: *3rd International Conference on Learning Representations, ICLR 2015*, San Diego, CA, USA, May 7-9, 2015, ConferenceTrack Proceedings.

Lassmann,T. and Sonnhammer,E.L. (2005) Kalign—an accurate and fast multiple sequence alignment algorithm. *BMC Bioinform*., **6**, 298.

Ling,G. *et al*. (2019) A Bayesian framework for inferring the influence of sequence context on point mutations. *Mol. Biol. Evol*., **37**, 893–903.

Makova,K.D. and Hardison,R.C. (2015) The effects of chromatin organization on variation in mutation rates in the genome. *Nat. Rev. Genet*., **16**, 213–223.

Messer,P.W. and Arndt,P.F. (2007) The majority of recent short DNA insertions in the human genome are tandem duplications. *Mol. Biol. Evol*., **24**, 1190–1197.

Mikolov,T. *et al*. (2013) Efficient estimation of word representations in vector space. *arXiv preprint arXiv*:1301.3781

Miller,W. *et al*. (2007) 28-way vertebrate alignment and conservation track in the UCSC genome browser. *Genome Res*., **17**, 1797–1808.

Mircea,I.-G. *et al*. (2018) A reinforcement learning based approach to multiple sequence alignment. In: *Proceedings of the 7th International Workshop Soft Computing Applications (SOFA2016),* vol. 2, pp. 54–70. Cham, Switzerland, Springer.

Morton,B.R. (2003) The role of context-dependent mutations in generating compositional and codon usage bias in grass chloroplast DNA. *J. Mol. Evol*., **56**, 616–629.

Nair,V. and Hinton,G.E. (2010) Rectified linear units improve restricted Boltzmann machines. In: *Proceedings of the 27th International Conference on Machine Learning*, pp. 807–814. Omnipress, Madison, WI, USA.

Needleman,S.B. and Wunsch,C.D. (1970) A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol*., **48**, 443–453.

Neubig,G. (2017) Neural machine translation and sequence-to-sequence models: a tutorial. *arXiv preprint arXiv:1703.01619*

Papadopoulos,J.S. and Agarwala,R. (2007) Cobalt: constraint-based alignment tool for multiple protein sequences. *Bioinformatics*, **23**, 1073–1079.

Price,M.N. *et al*. (2010) Fasttree 2—approximately maximum-likelihood trees for large alignments. *PLoS One*, **5**, e9490.

Ramakrishnan,R.K. *et al*. (2018) Rlalign: a reinforcement learning approach for multiple sequence alignment. In: *2018 IEEE 18th International Conference on Bioinformatics and Bioengineering (BIBE)*, pp. 61–66. IEEE Computer Society, Conference Publishing Services, Los Alamitos, CA, USA.

Ranwez,V. *et al*. (2011) MACSE: multiple alignment of coding sequences accounting for frameshifts and stop codons. *PLoS One*, **6**, e22594.

Rodrigue,N. *et al*. (2005) Site interdependence attributed to tertiary structure in amino acid sequence evolution. *Gene*, **347**, 207–217.

Schwartz,S. *et al*. (2000) Pipmaker—a web server for aligning two genomic DNA sequences. *Genome Res*., **10**, 577–586.

Siepel,A. and Haussler,D. (2003) Phylogenetic estimation of context-dependent substitution rates by maximum likelihood. *Mol. Biol. Evol*., **21**, 468–488.

Smith,T.F. and Waterman,M.S. (1981) Identification of common molecular subsequences. *J. Mol. Biol*., **147**, 195–197.

Stoye,J. *et al*. (1998) Rose: generating sequence families. *Bioinformatics*, **14**, 157–163.

Sundermeyer,M. *et al*. (2012) LSTM neural networks for language modeling. In: *13th Annual Conf of the International Speech Communication Association*, pp. 194-197. Portland, OR, USA. ISCA

Surrallés,J. *et al*. (2002) Clusters of transcription-coupled repair in the human genome. *Proc. Natl. Acad. Sci. USA*, **99**, 10571–10574.

Sutskever,I. *et al*. (2014) Sequence to sequence learning with neural networks. In: *Proceedings of the 27th International Conference on Neural Information Processing Systems* - vol. 2, NIPS'14, page 3104–3112, Cambridge, MA, USA. MIT Press.

Thorne,J.L. *et al*. (1991) An evolutionary model for maximum likelihood alignment of DNA sequences. *J. Mol. Evol*., **33**, 114–124.

Vaswani,A. *et al*. (2017) Attention is all you need. In: *Advances in Neural Information Processing Systems*, pp. 5998–6008.

Wall,S. *et al*. (1999) Context-sensitive transposition of IS6110 in mycobacteria. *Microbiology*, **145**, 3169–3176.

Zhang,H. *et al*. (2018) Dynamics and function of DNA methylation in plants. *Nat. Rev. Mol. Cell Biol*., **19**, 489–506.

Zhu,Y. *et al*. (2017) Statistical methods for identifying sequence motifs affecting point mutations. *Genetics*, **205**, 843–856.