nature machine intelligence

Article

https://doi.org/10.1038/s42256-024-00872-0

DNA language model GROVER learns sequence context in the human genome

Received: 31 August 2023

Accepted: 26 June 2024

Published online: 23 July 2024

Check for updates

Melissa Sanabria [1], Jonas Hirsch[1], Pierre M. Joubert[1,2,3] & Anna R. Poetsch [1,4]

Deep-learning models that learn a sense of language on DNA have achieved a high level of performance on genome biological tasks. Genome sequences follow rules similar to natural language but are distinct in the absence of a concept of words. We established byte-pair encoding on the human genome and trained a foundation language model called GROVER (Genome Rules Obtained Via Extracted Representations) with the vocabulary selected via a custom task, next-*k*-mer prediction. The defined dictionary of tokens in the human genome carries best the information content for GROVER. Analysing learned representations, we observed that trained token embeddings primarily encode information related to frequency, sequence content and length. Some tokens are primarily localized in repeats, whereas the majority widely distribute over the genome. GROVER also learns context and lexical ambiguity. Average trained embeddings of genomic regions relate to functional genomics annotation and thus indicate learning of these structures purely from the contextual relationships of tokens. This highlights the extent of information content encoded by the sequence that can be grasped by GROVER. On fine-tuning tasks addressing genome biology with questions of genome element identification and protein–DNA binding, GROVER exceeds other models' performance. GROVER learns sequence context, a sense for structure and language rules. Extracting this knowledge can be used to compose a grammar book for the code of life.

The first draft of the human genome has been available for more than 20 years[1], and genomes of multiple species have become available since. We know the letters, but we still understand little about the 'genetic code'. DNA triplets in genes encode for amino acids[2] in 1–2% of the genome[1], but there are additional layers of 'code'. How those genes are regulated; how transcripts are structured, function and are kept stable; how, where and when the genome replicates; and how it is concurrently kept stable and functional are all encoded within the genome. Extracting these different layers of code comprehensively requires complex algorithms that only recently have become available in natural language processing. Large language models (LLMs), which are based on transformer architectures[3], are well suited for text data, with unprecedented performance and transparency. Pretrained LLMs like GPT-3 (ref. 4) and successors can also function as foundation models to be fine-tuned with classification, regression or generative tasks. These models have changed how we view language and can be very useful for a variety of purposes.

Although genomes are analogous to language in their structure that resembles grammar, syntax and semantics, they also differ. First, there is no clearly defined direction, unless viewed relative to biological processes like transcription or replication. Second, there is no natural definition of words. We know transcription factor binding motifs or the

[1]Biomedical Genomics, Biotechnology Center, Center for Molecular and Cellular Bioengineering, Technische Universität, Dresden, Germany. [2]Center for Advanced Systems Understanding (CASUS), Görlitz, Germany. [3]Helmholtz Zentrum Dresden-Rossendorf (HZDR), Dresden, Germany. [4]National Center for Tumor Diseases (NCT) partner site Dresden, German Cancer Research Center (DKFZ), Dresden, Germany. e-mail: arpoetsch@gmail.com

https://doi.org/10.1038/s42256-024-00872-0

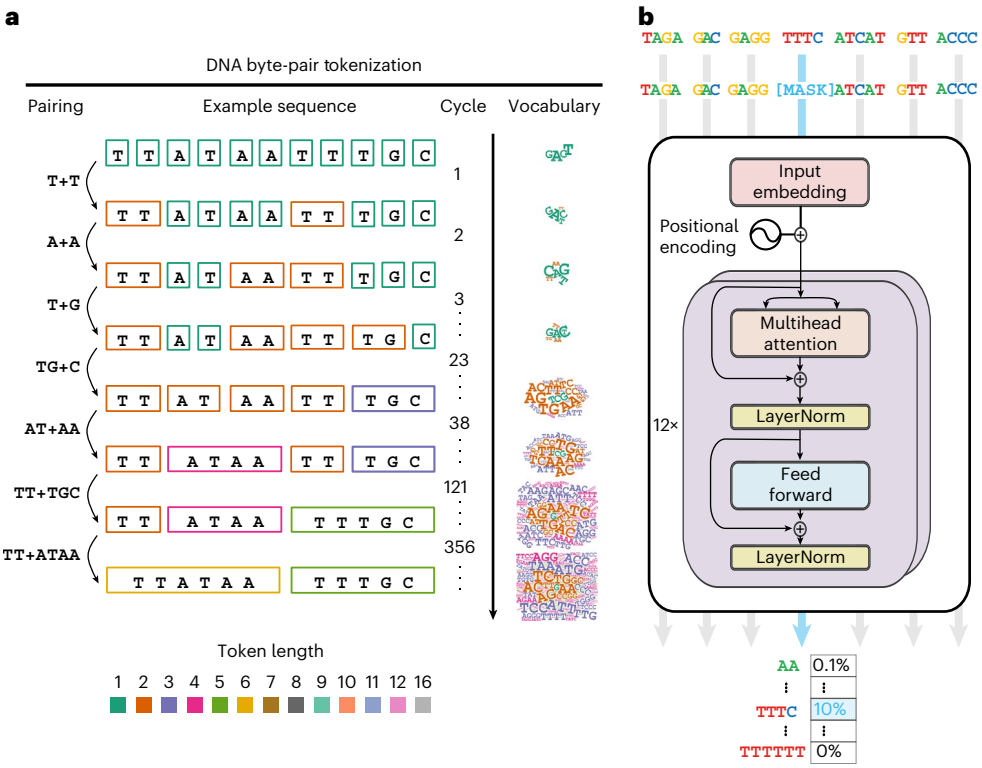


Fig. 1 | DNA BPE and model architecture. a, The principle of BPE highlighted on an example sequence with the tokenization steps relevant for this sequence. Resulting vocabularies are coloured by token length and depicted in a word cloud with relative weights of the words by their frequency. **b**, The model is a BERT architecture with 12 transformer blocks (in purple), which use multihead attention and a feed forward layer with normalisation steps (LayerNorm) in between. The model is embedding the tokens and is trained with cross-entropy loss to predict the masked token and updates the embedding while training. The output is probabilities of the masked token identity.

triplet code that encodes proteins. However, in the genome as a whole, there is no clear concept for words. To overcome those challenges for training transformer models on DNA, so-called DNA language models (DLM), there have been several approaches. Some are aimed at addressing specific tasks, such as the modelling of gene expression with Enformer[5], a model that combines convolutional layers with transformer blocks. Through the convolutional layer, no definition of words is necessary. Foundation models, however, are trained not directly on a specific genome biology task but rather are first pretrained on masked token prediction and subsequently fine-tuned. This strategy requires the definition of discrete tokens: that is, to build 'words' from DNA. Available models for the human genome include LOGO[6], DNABERT[7] and Nucleotide Transformer (NT)[8], which use a Bidirectional Encoder Representations from Transformers (BERT)[9] architecture and apply different strategies of generating the vocabulary. NT uses mainly 6-mers as its vocabulary. DNABERT uses *k*-mers of 3, 4, 5 and 6 nucleotides for four different models, of which the 6-mer model performs best[7]. The *k*-mers overlap, and the training is designed for the central nucleotide of a masked sequence not to overlap with any unmasked tokens. Consequently, the model largely learns the token sequence, rather than the larger context[10]. Semisupervised models include data beyond the genome sequence, such as GeneBERT[11]. HyenaDNA uses implicit convolutions in its architecture[12]. Taking genomes from multiple species increases the amount of training data, as for DNABERT-2 (ref. 13). Although these alternative strategies may improve performance for some tasks, the additional sources of information or different architectures make it harder to follow what the model is learning and to link the learned representations back to the relevant genome sequence. Therefore, we decided to train only with the human genome sequence, distributed into tokens. The ideal vocabulary for a DLM should have tokens with an appropriate length to capture the language structure of the genome. However, if length is chosen as a constant, the frequencies of the tokens become heterogeneous. 6-mers range from about 10^4 to 10^7 occurrences in the human genome (hg19). Such a frequency imbalance can inhibit model training through Rare Word Problems or lead to training on frequencies rather than genome language context. We therefore applied byte-pair encoding (BPE)[14] to the human genome to generate multiple frequency-balanced vocabularies and selected the vocabulary that carries the information content of the human genome in an optimal way. In combination with fine-tuning tasks and the inbuilt transparency of the model architecture, we can now start using the resulting foundation DLM, GROVER (Genome Rules Obtained Via Extracted Representations), to extract its learning and different layers of the genome's information content.

Results

Building a frequency-balanced vocabulary on the human genome

For a human DLM, grouping nucleotides into tokens with similar frequencies is a difficult task because of the heterogeneous sequence composition over the genome. A- and T-rich sequences are relatively frequent, whereas CG dinucleotides are depleted due to their susceptibility to mutation[15]. Therefore, tokens that contain rarer sequence content should be shorter and tokens with frequent sequence content should be longer. In the case of CG dinucleotides, this is of particular importance, given that through potential DNA methylation in the form of 5-methyl-cytosine[16], this dinucleotide fulfils a special biological role in gene regulation[17,18] and retrotransposon silencing[19]. Aiming for frequency balance, we employed BPE[14] (Fig. 1a). The algorithm prioritizes larger tokens of more frequent sequence content by sequentially combining the most frequent token pairs into new tokens. Starting with the four nucleotides A, C, G and T, in the first cycle of BPE, two Ts

are combined into a TT token, which adopts a new token identity. This pairing can in principle be continued for many cycles, continuously forming larger tokens. With the dictionary growing, new pairs become less frequent. We use these vocabularies to train a model with a BERT architecture (Fig. 1b) for masked token prediction with cross-entropy loss. This results in multiple models from 100–5,000 cycles of BPE from which the optimal model and therefore vocabulary is selected.

Selecting an optimal vocabulary with next-*k*-mer prediction

To select an optimal vocabulary and model, performance can in principle be assessed with two strategies, intrinsic or extrinsic validation: for example, on a specific genome biological task. To avoid biasing the model towards specific biology, we chose intrinsic validation. However, perplexity is dependent on dictionary size, so we applied next-*k*-mer prediction[10]. Predicting the next sequence token of a defined length requires a sense of sequence context. This allows relative comparisons without biological focus and is independent of tokenization strategy, vocabulary size and foundation model architecture. We used fine-tuning models for fixed-size *k*-mers with $k = [2,3,4,5,6]$ in combination with models trained on 100–5,000 cycles of BPE vocabulary (Fig. 2a). With minor differences, the different *k*-mer models perform most accurately within 400–800 cycles, so we picked cycle 600 for GROVER. To compare GROVER with models of fixed-size *k*-mer vocabularies, we established foundation models of non-overlapping 4-mers, 5-mers and 6-mers and performed the equivalent tasks. We also included the established models for the human genome NT[8] and HyenaDNA[12], as well as the multispecies model DNABERT-2 (ref. 13) (Fig. 2b and Supplementary Fig. 1a,c,e,g). They generally show inferior accuracy for next-token prediction, similar to the accuracy we have previously shown for DNABERT[10]. Relative differences become increasingly apparent when predicting larger tokens. GROVER achieves 2% accuracy predicting next-6-mers, whereas the next-best model, the multispecies model DNABERT-2, achieves 0.6% accuracy. Fixed-size *k*-mer BERT models, including NT, do not exceed 0.4% accuracy, independent of the *k*-mer sizes of the foundation model. Using term frequency-inverse document frequency (TF-IDF) models, we set a baseline of prediction dependencies on token frequencies for fine-tuning tasks. Applying TF-IDF to next-*k*-mer prediction with vocabulary on fixed-size *k*-mers and 600 cycles (BPE-600) (Fig. 2c and Supplementary Fig. 1b,d,f,h), we achieve increasing accuracy with longer token lengths, and BPE-600 shows intermediate accuracy. The best TF-IDF model for next-6-mers achieves 1.1% accuracy, outperforming all pretrained foundation models but GROVER. Sequence imbalances in the human genome are a dominant feature for some prediction tasks, and DLMs may focus on this feature rather than learning a sense of sequence context.

The GROVER foundation model training task of masked token prediction (Fig. 2d) achieves 21% accuracy. Allowing for the top 60 predicted tokens—that is, 10% of the dictionary—increases the accuracy to 75%. GROVER shows perplexity of 72 bits per token, which represents 12% of the dictionary size, whereas fixed-size *k*-mer models show perplexity of 65 (25%), 216 (21%) and 1472 (36%) for 4-mers, 5-mers and 6-mers, respectively (Fig. 2e). The selected BPE vocabulary is therefore outperforming the fixed-size *k*-mer models and the vocabulary is thus optimized to carry the information content of the genome with relevance for this type of model training. To interrogate the origin of such performance improvement, we next investigated vocabulary composition and learned representations.

The optimized vocabulary for DNA language training

BPE adds to the special tokens and the four nucleotides one token for each of the 600 cycles. All single A, C and T (not adjacent to N) are incorporated into larger tokens and thus removed from the dictionary, which consequently comprises 601 tokens. Fixed-size *k*-mer vocabularies show bimodal frequency distributions (Fig. 3a). The GROVER vocabulary (BPE-600) shows token frequencies that are mostly higher than 100,000, with a median of about 400,000. The majority of tokens are 4-mers with average token length 4.07 (Fig. 3b). Token length ranges from one 1-mer G to two 16-mers (A_{16}, T_{16}). Not all possible *k*-mer combinations are generated. CG dinucleotides, for example, have never formed, and this sequence is part of larger or several tokens. There is a heterogeneous representation of *k*-mers in the GROVER dictionary (Fig. 3c). Most token types are 5-mers ($n = 213$) and 6-mers ($n = 224$). Proportional representation of *k*-mers in the dictionary is therefore also heterogeneous (Fig. 3d). Of all 1-mers, only G is represented in the vocabulary, and 63% of the 2-mers are represented. Although 4-mers are the most frequent tokens in the vocabulary, only 32% of possible 4-mer sequences are represented. In total, the nucleotide representation within the dictionary reflects the nucleotide composition of the genome (Fig. 3e). There is, however, an imbalance with respect to the first nucleotide of the tokens in the dictionary, with 97% starting with A or T, where 60% would be expected. BPE initially prioritizes the more frequent As and Ts to generate new tokens and thus amplifies the nucleotide frequency imbalance of the genome in the representation of the tokens' first nucleotides.

Assessing performance metrics per token type, we discovered that 6-mers perform worse on average for area under the curve (AUC) and accuracy in comparison to both shorter and longer tokens (Fig. 3f,g). Importantly, every token has an AUC above 50% and thus contributes to predictions. Tokens with the most accurate predictions (>99%), are the 9-mer ATTACAGGC and 12-mer TGTAATCCCAGC. The least accurate predictions (<1%) are made for the 6-mer TTTAGG. This heterogeneity may be related either to heterogeneous sequence ambiguity of the tokens or differences within GROVER's learning. Therefore, we investigated what GROVER learns about the tokens themselves.

GROVER learns token characteristics and some annotations

What transformer models learn can be extracted by analysing token embeddings. Updated during the training process, they reflect how the trained model sees the tokens. We compared the average trained GROVER embeddings with static Word2Vec (W2V) embeddings[20] for each token type. W2V was built to reflect average word associations in a multidimensional space and reflects average word associations within the vocabulary. To assess how much context learning can be inferred, we extract the maximum explainable variance (MEV)[21], which takes from a principal component analysis (PCA) the variance explained from the first principal component (PC1) (Fig. 4a). Although W2V embedding shows with 15% a rather high MEV value for an LLM, GROVER lies with 3.5% in a range typical for LLMs[21]. The first two PCs of W2V shows that there are no clear token clusters forming (Fig. 4b). Due to the high variance explained on PC1, its ranks were used to colour-code tokens as a visual measure of token similarity, which correlates with GC content (Spearman's $R = 0.93$) (Fig. 4c). Some outlier tokens are also coloured blue, despite not being of high GC content. These are predominantly tokens that reside in Alu sequences and are thus surrounded by high GC content. A nonlinear dimensionality reduction of W2V embedding with Uniform Manifold Approximation and Projections (UMAP) largely reflects the colour coding and also does not reveal major token clusters (Fig. 4d). PCA and UMAP on the trained embedding of GROVER also does not reveal major token clusters. This indicates that GROVER does not simply learn to spell out tokens[10], but the trained embeddings reflect learned content beyond pure token identity. Dimensionality reduction reflects the token colours derived from the W2V embedding (Fig. 4e,f) on the second dimension. The first 20 PCs correlate with some quantifiable characteristics (Fig. 4g,h). W2V learns GC content on PC1 (Spearman's $R = 0.93$) and AG content on PC2 (Spearman's $R = 0.75$). AG content is also a reflection of strand specificity relative to replication, transcription and repeats. Although the PCs explain less variance for GROVER, associations with token characteristics and annotation are more pronounced. Because most genome functional annotations correlate with GC content, this was corrected for. PC1 strongly correlates

https://doi.org/10.1038/s42256-024-00872-0

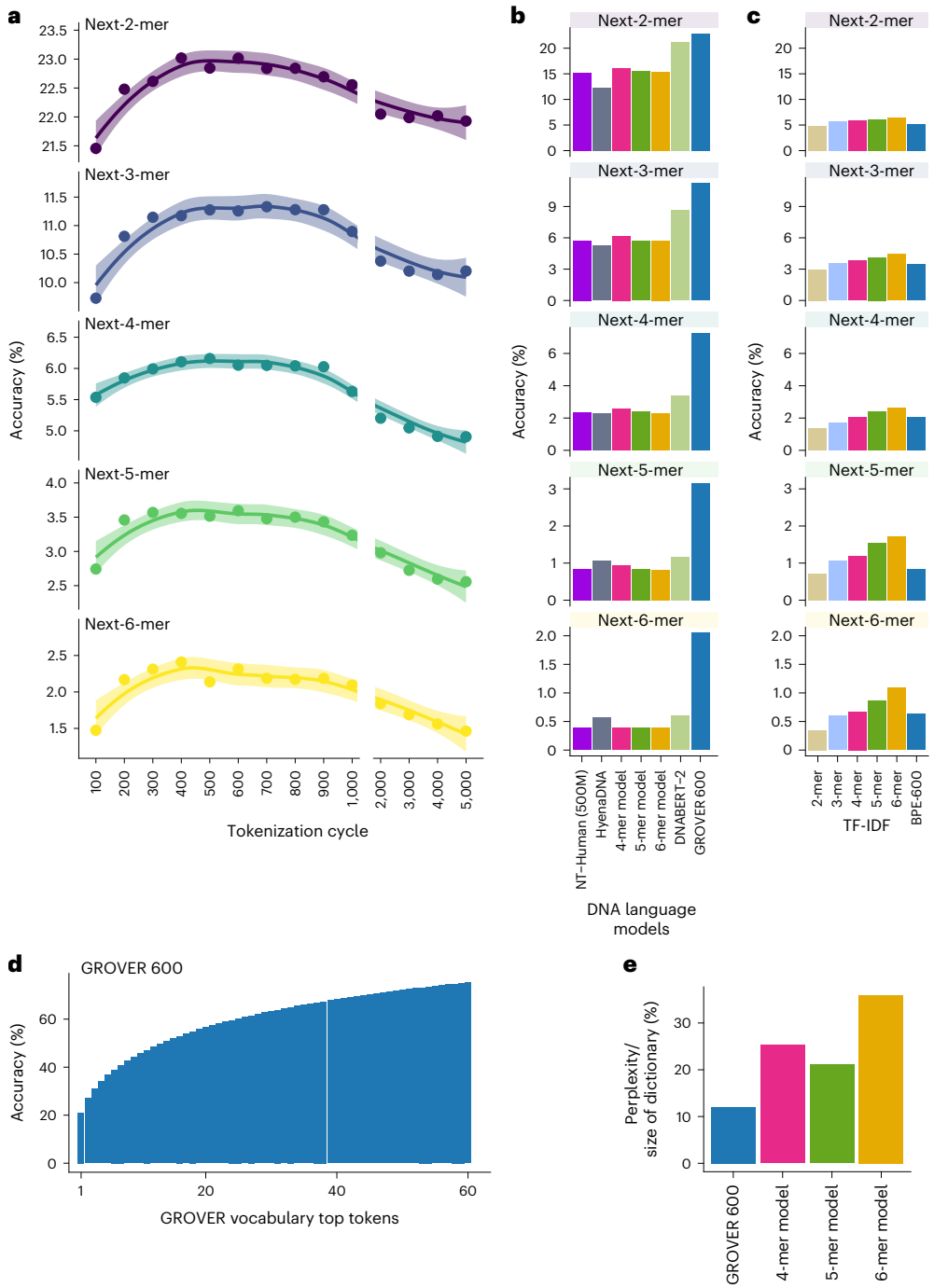


Fig. 2 | Performance based selection of the vocabulary identifies 600 cycles of BPE as optimal. a, Selection of the optimal vocabulary through accuracy of next-token prediction as a fine-tuning task for the foundation models using prediction of two- to six-nucleotide-long next-*k*-mers as readout. Depicted with solid points is accuracy with a solid line depicting a loess fit and the 95% confidence interval as shading. The interruption of the line illustrates the change of scale on the *x* axis. **b**, Performance comparison using accuracy of next-*k*-mer prediction as a fine-tuning task. Compared are GROVER with 600 cycles of BPE (BPE-600) with models based on *k*-mer tokenization, with lengths of four, five and six nucleotides, the human model of NT with 500 million parameters (NT-Human), HyenaDNA and DNABERT-2. **c**, Comparison of accuracy to TF-IDF models, which use two- to six-nucleotide-long *k*-mers and the GROVER vocabulary (BPE-600). These models take only token frequencies into account, which are used to train a random forest model. They are not learning context between tokens. **d**, Performance assessment of GROVER with 600 cycles of BPE using accuracy for the masked token being predicted as the top 1 token, up to top 60: that is, the top 10%. **e**, Performance assessment of GROVER with 600 cycles of BPE using perplexity divided by the total number of words in the dictionary. Comparison with models based on *k*-mer-tokenization, with lengths of four, five and six nucleotides.

with token frequency (Spearman's $R = 0.88$) (Fig. 4i), which played no visible role for W2V. BPE was used to reduce frequency imbalance in the vocabulary, but PC1 shows that remaining imbalances of token frequency is still a major feature learned by GROVER. PC2 correlates with GC content (Spearman's $R = -0.96$) (Fig. 4j), similar to W2V's PC1. PC3 correlates with AG content (Spearman's $R = 0.94$) (Fig. 4k), which probably reflects the learning of DNA strand information. PC4 and PC6 mildly correlate with token length (Spearman's $R = 0.39$ and Spearman's

https://doi.org/10.1038/s42256-024-00872-0

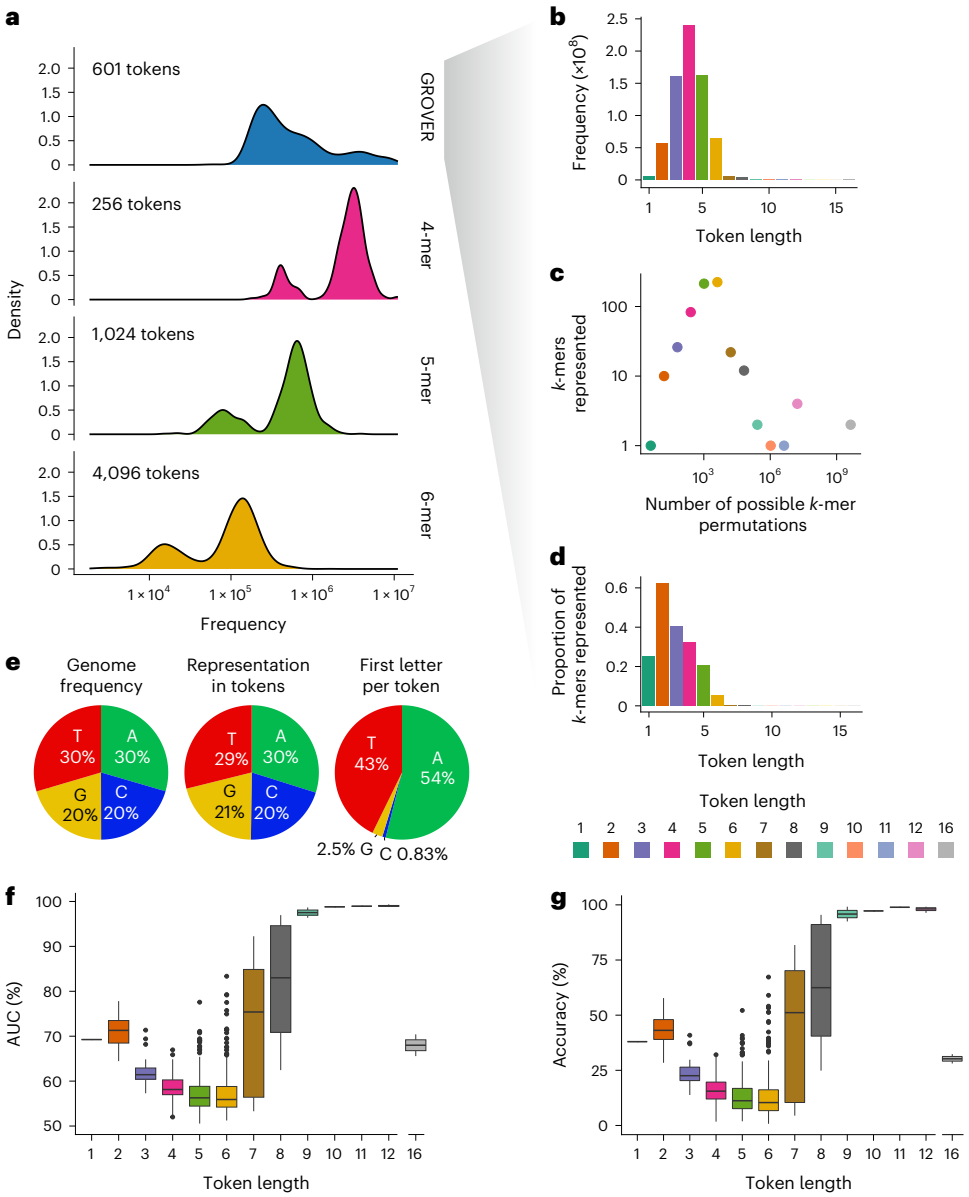


Fig. 3 | The frequency-balanced GROVER vocabulary shows differential learning performance by token length. a, Token frequency in the genome for GROVER and *k*-mer-tokenization. **b**, Composition of the GROVER vocabulary, differentiating token length versus frequency in the genome. **c**, Composition of the GROVER dictionary, differentiating the frequency of possible *k*-mer permutations and how many *k*-mers are represented in the GROVER dictionary. **d**, Composition of the GROVER dictionary, differentiating the proportion of *k*-mers represented in the dictionary, dependent by length. **e**, Representation of the four nucleotides in the genome and in the GROVER dictionary. **f**, Prediction performance as AUC by token length. $n_{Total} = 601$; subdifferentiation by *k*-mer length is according to *k*-mer representation in **c**. Box plots depict the median with the quartiles and as whiskers, 1.5 times the interquartile range of the of the upper quartile/lower quartile. **g**, Prediction performance as accuracy by token length. $n_{Total} = 601$; subdifferentiation by *k*-mer length is according to *k*-mer representation in **c**. Box plots depict the median with the quartiles and as whiskers, 1.5 times the interquartile range of the of the upper quartile/lower quartile.

$R = 0.43$, respectively) (Fig. 4l,n). Token length is also related to token frequencies, which may result in these mild correlations. Finally, PC5 correlates with AC content (Spearman's $R = -0.81$) (Fig. 4m), which complements GC and AG content. Lower PCs show additional mild associations with certain repeat classes, gene transcription and replication timing (Fig. 4h). Although GROVER learns with an unsupervised strategy, it clearly learns to separate token characteristics. Hierarchical clustering on the Euclidean distance of the average trained token embeddings leads to similar results (Supplementary Fig. 2). The most prominent cluster is composed on high-frequency 4-mers and 5-mers. GC content, AG content and token length contribute to the token distances. There are tokens that almost exclusively localize to repeat elements. Although there is also clustering visible for tokens with variable contribution to chromatin colours, beyond repeats there is no clear assignment of tokens to specific genome functional elements. Taken together, GROVER learns from token identity, but learning of functional genomics features needs to rely on a larger sequence context.

GROVER learns context

To assess how much GROVER sees context for individual token types, we use self-similarity: that is, cosine similarity between trained embeddings across a token type's different contexts (Fig. 5a). The more contextualized the representations are, the lower self-similarity we expect. Using hierarchical clustering on the Euclidean distance over

self-similarities for each transformer layer, clustering resembles the analogous results of clustering based on average trained embeddings. Tokens that localize largely to repeats show the highest self-similarity almost throughout the transformer layers. They are also distinct in their good per-token performance metrics and long token lengths. Otherwise, high self-similarity in layer 12 also highlights a special token group, short tokens with high frequency, which also show good performance metrics. For the other tokens, it can be concluded that dependent on their cluster, there is low self-similarity in different layers, which indicates contextualized learning. To investigate whether there is sufficient context to reflect particular genome biology, we used trained embeddings for the classify token (CLS) summarising the trained embeddings per window of 510 tokens (average 2.1 kb) over the genome to interrogate which regions GROVER identifies as similar or dissimilar (Fig. 5b–d). UMAP shows a spread of sequence context with some clustering. Annotation of repeats over the windows (Fig. 5b) indeed shows that long interspersed nuclear elements (LINEs) form distinct clusters dependent on their orientation relative to the tokenization direction. Similar patterns are formed by short interspersed nuclear elements (SINEs), of which Alu elements are the most prominent representative. Long terminal repeats (LTRs), satellite repeats, as well as low-complexity and simple repeats cover distinct territories. Annotation of chromatin colours (Fig. 5c) also shows distinct locations for most chromatin features. Within these features there has not been a strong enrichment of particular token sequences (Fig. 5a), which suggests that sequence context has been learned. Differentiating some selected elements for their predominant direction relative to tokenization as well as replication timing reveals that GROVER not only recognizes the elements in question by giving them distinct territories in the trained embedding space but also separates them by their directionality (Fig. 5d). Representation of directionality becomes even more pronounced for repeat elements with little fragmentation (LINE1 6 kb ± 300 bp, LTR > 1 kb, Alus 300 bp ± 30 bp). Correlation between direction of replication and direction of genes becomes visible as well as the anticorrelation of both with direction of LINE elements, which preferentially localize on the strand antisense to replication[22]. GROVER also assigns distinct territories to replication timing without explicitly being given information beyond DNA sequence. Therefore, GROVER can learn biological information and epigenetic phenomena directly from sequence.

GROVER can be fine-tuned for tasks on genome biology

To show the suitability of GROVER for genome biology questions, we selected three representative fine-tuning tasks. Prom300 was adapted with some minor modifications from ref. 7 (Fig. 6a). In short, promoters are selected for sequences around transcription start sites (TSS) −250 bp/+50 bp and classified as actual promoters versus promoters with shuffled tokens. Shuffling was implemented rather than taking non-promoter sequences or nucleotide exchanges to give token identity and frequency less relevance for the task. Shuffling relative to BPE-600, the task performs with a Matthew's correlation coefficient (MCC) of 99.6% as compared to 79% for the 4-mer model, the model with second-best performance (Fig. 6b). MCC is comparable to other performance metrics and results relative to shuffling with fixed-*k*-mer tokens (Supplementary Fig. 3). Interestingly, for this task, token frequency is still very informative, with an MCC of 67% for the TF-IDF 5-mer model (Fig. 6c). A more challenging task is PromScan (Fig. 6d), where 1 kb windows are selected from 10 kb regions around the TSS and classified for overlap with the TSS. Due to unbalanced classes, this task is more challenging, yet GROVER recognizes the TSS windows with an MCC of 63% as compared to 52% for NT, the second-best-performing human model (Fig. 6e) and 39% for TF-IDF 3-mers, the best-performing context-free model (Fig. 6f). Other performance metrics show similar results (Supplementary Fig. 4). Finally, we developed a task of predicting protein–DNA binding focusing on the CCCTC-binding factor (CTCF) (Fig. 6g). The task is to recognize which sites that contain a CTCF binding motif are indeed bound by the protein according to ChIP–seq data from HepG2 cells obtained from ENCODE[23]. Although there are ~85,000 binding motifs in the human genome, only ~32,000 are actually bound by CTCF. Beyond the motif, sequence context with particular physicochemical properties of the DNA, as well as binding of other proteins, determines whether a protein indeed binds its assigned motif. GROVER achieves for this task an MCC of 60% as compared to 59% of the multispecies model DNABERT-2, the next-best-performing model (Fig. 6h). For this task, a TF-IDF 4-mer model reaches an MCC of 26% (Fig. 6i). The performance is comparable when using other performance metrics (Supplementary Fig. 5).

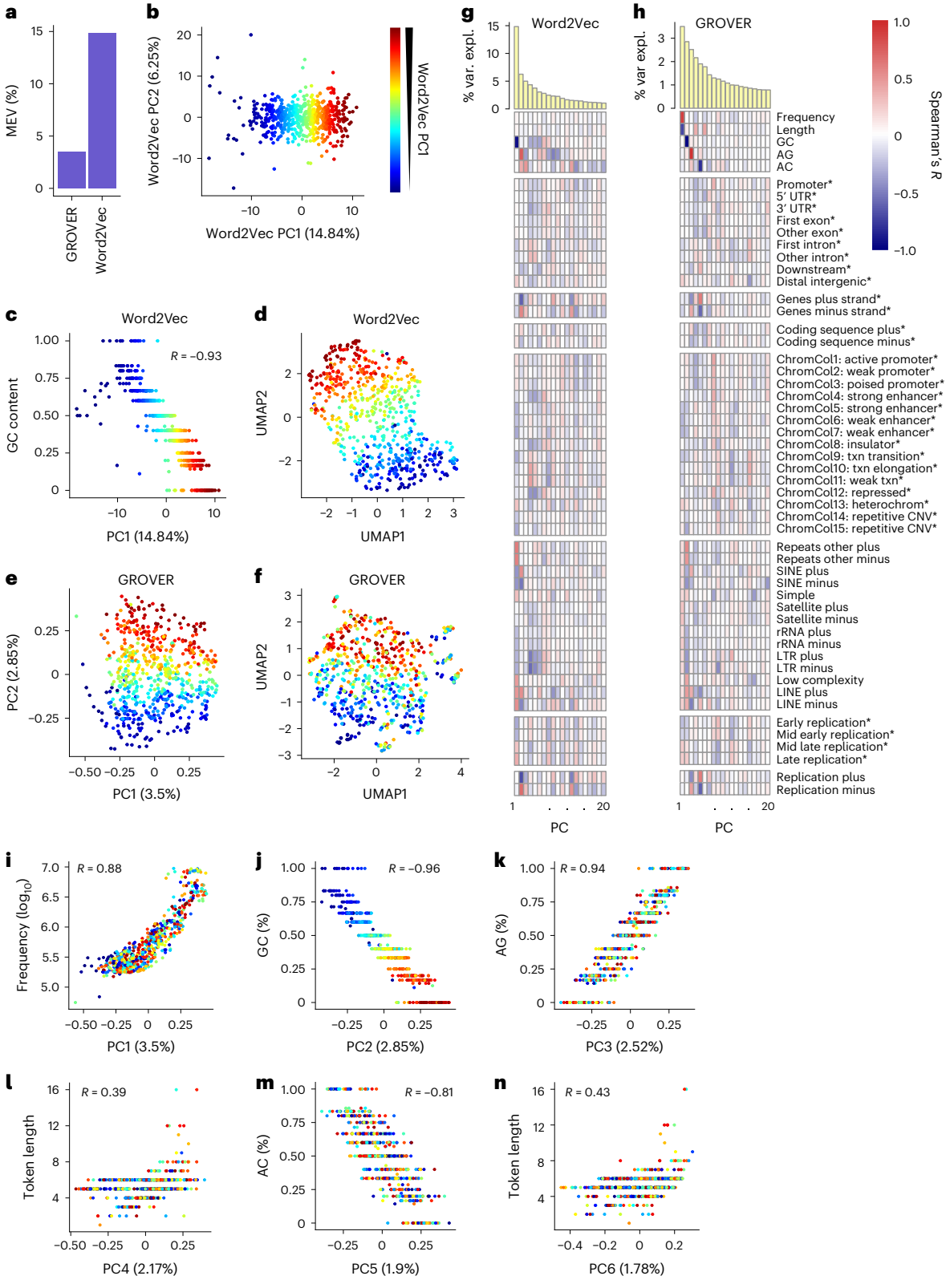
To compare with other published benchmarking tasks, we derived the scores from refs. 8,13, who defined the benchmarking tasks 'NT tasks' and 'Genome Understanding Evaluation (GUE) tasks', respectively (Fig. 6j and Supplementary Fig. 6). Comparing MCCs between human models, multispecies models and GROVER shows that the performance is comparable for most tasks. The two NT-enhancer prediction tasks represent the hardest tasks with an MCC for GROVER of 58% and 46%, which outperforms all other DLMs. However, comparison to the TF-IDF models shows that this is only marginally improved over the BPE-600 TF-IDF model (55%) and even inferior for the second enhancer prediction task (57%). The NT-promoter models all perform similarly both for the human and multispecies models. However, this task can already be explained through token frequency with an MCC up to 87%. Splice sites are the main task where DLMs strongly outperform TF-IDF models and among the human models, GROVER is slightly outperformed by NT and DNABERT with an MCC of 94%, 96% and 96%, respectively, for all splice sites. DNABERT is a model that mainly learns token identity due to the design of overlapping tokens[10]. Therefore, this task is likely dependent on short sequence motifs rather than larger sequence context. Similar

Fig. 4 | Average GROVER token embedding shows learning of genome information content. a, MEV derived from PC1 of the GROVER embedding averaged for each token, compared to W2V static embedding, derived from the same vocabulary. **b**, PCA of the GROVER vocabulary-derived W2V embedding for the first two PCs, with their explained variance. Colour represents the rank within W2V PC1. **c**, Correlation of W2V PC1 with token GC content. Colour represents the rank within W2V PC1. *R* = Spearman's *R* correlation coefficient. **d**, UMAP of the W2V embedding of the GROVER vocabulary. Colour represents the rank within W2V PC1. **e**, PCA of the GROVER vocabulary embedding, averaged per token, for the first two PCs, with their explained variance. Colour represents the rank within W2V PC1. **f**, UMAP of the GROVER vocabulary embedding, averaged per token. Colour represents the rank within W2V PC1. **g**,**h**, Correlation of vocabulary characteristics and genome biology annotation of the vocabulary with the GROVER vocabulary-derived W2V embedding (**g**) and with the GROVER token averaged embedding (**h**). Depicted is variance explained throughout the first 20 PCs of a PCA, along with the Spearman correlation with token characteristics and percentage of tokens of a specific token sequence that belong to genome annotation categories. Gene element annotations with gene promoters (transcriptional start site ±1 kb), 5′ and 3′ untranslated regions, exons, introns, gene downstream regions (10 kb) and distal intergenic regions, as well as gene strand, coding sequence strand, chromatin colours, replication timing and replication strand, were corrected for GC content (as marked with an asterisk) using linear regression. Repeat annotations are obtained from RepeatMasker: SINE, short interspersed nuclear elements; rRNA, ribosomal RNA; LTR, long terminal repeats; LINE, long interspersed nuclear elements. **i**–**n**, Correlation of the PCs from a PCA of the GROVER token averaged embedding with features that were identified to explain much of the variance explained in the PC: that is, PC1 and token frequency (**i**), PC2 and GC content (**j**), PC3 and AG content (**k**), PC4/PC6 and token length (**l**,**n**) and PC5 and AC content (**m**). ChromCol, chromatin colour; CNV, copy number variation; txn, transcribed; UTR, untranslated region; var. expl., variance explained.

https://doi.org/10.1038/s42256-024-00872-0

to the NT tasks, the GUE tasks (Supplementary Fig. 6) also show largely marginal performance gains over TF-IDF models, with the exception of tasks on splice sites. GROVER and DNABERT-2 do particularly well on the transcription factor binding tasks (MCC_{tf4} = 75% and 77%, respectively), especially when considering that the BPE-600 TF-IDF model explains the task less (MCC_{tf4} = 59%) than the 6-mer model (MCC_{tf4} = 72%), which would be the corresponding model to NT (MCC_{tf4} = 61%). However, the generally high-performance metrics for the TF-IDF models raise the question of how much of the performance can be explained by learning token frequencies versus relevant sequence context. Still, GROVER generally shows similar performance to the other models that had already been assessed for the GUE and NT tasks.

In conclusion, GROVER not only achieves good performance with tasks that are agnostic to genome biology questions but also achieves good performance with tasks that directly address genome function, particularly those that have been designed with a special focus on



https://doi.org/10.1038/s42256-024-00872-0

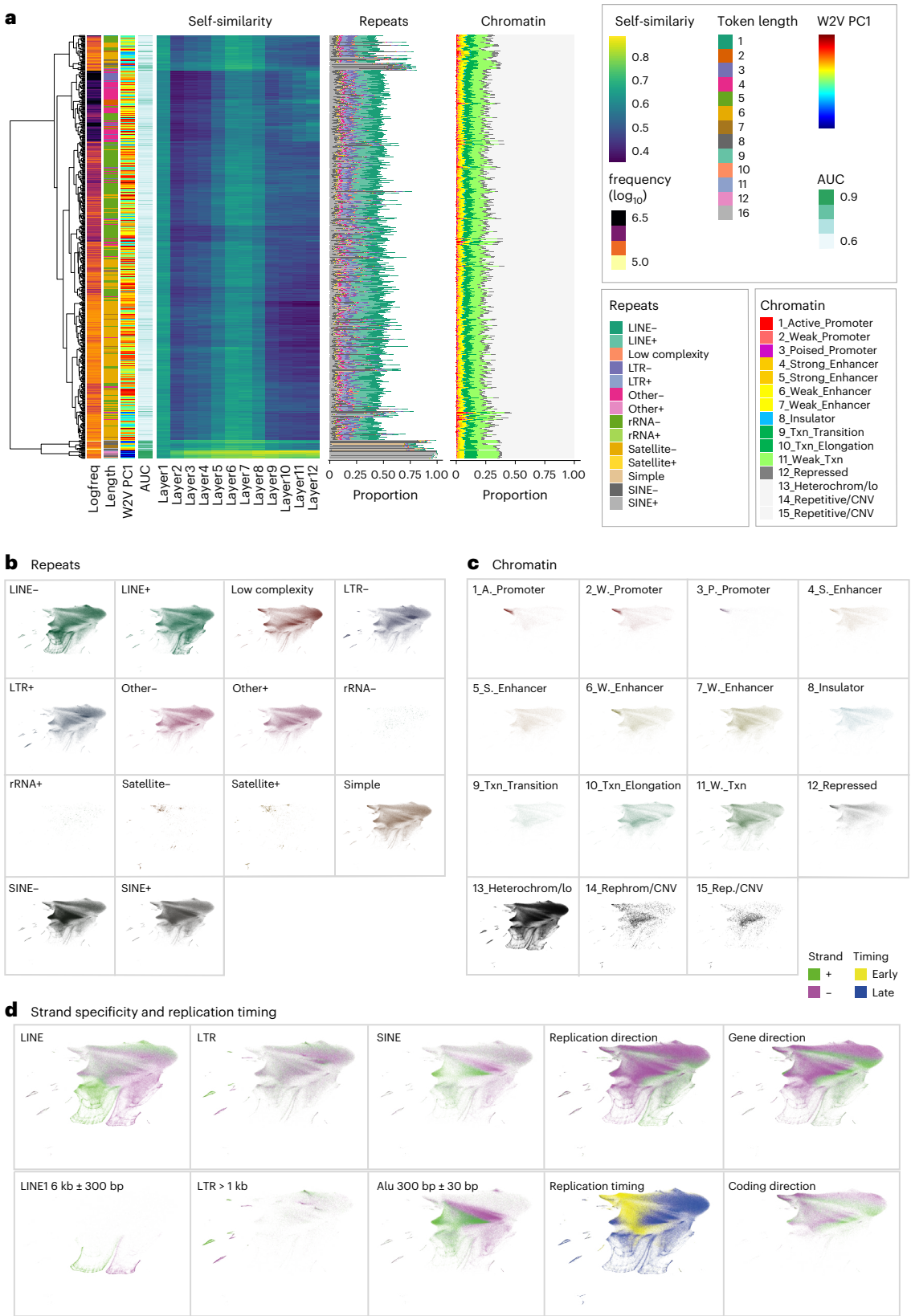


Fig. 5 | GROVER learns token context and genome annotation. a, Self-similarity per token sequence as extracted by cosine similarity of the same token in different contexts throughout the 12 transformer layers. Self-similarity is clustered with hierarchical clustering by Euclidean distance. Cluster annotation with token characteristics: that is, token frequencies in the genome, token length, performance (AUC), W2V embedding PC1, proportion of tokens falling into chromatin colours of ChromHMM and repeats. **b**–**d**, UMAP for the summarized embedding of regions in the genome, 510 tokens in size. Regions are annotated relative to repeats (**b**) and chromatin colours (**c**) and shown if they overlap with the feature of interest. Comparisons dependent on feature direction relative to the tokenization direction (+ strand) and annotation relative to early or late-replicating DNA as measured by OK-Seq (**d**). Bins with a unique assignment of direction or annotation are shown. A., active; W., weak; P., poised; S., strong; Rep., repetitive.

https://doi.org/10.1038/s42256-024-00872-0

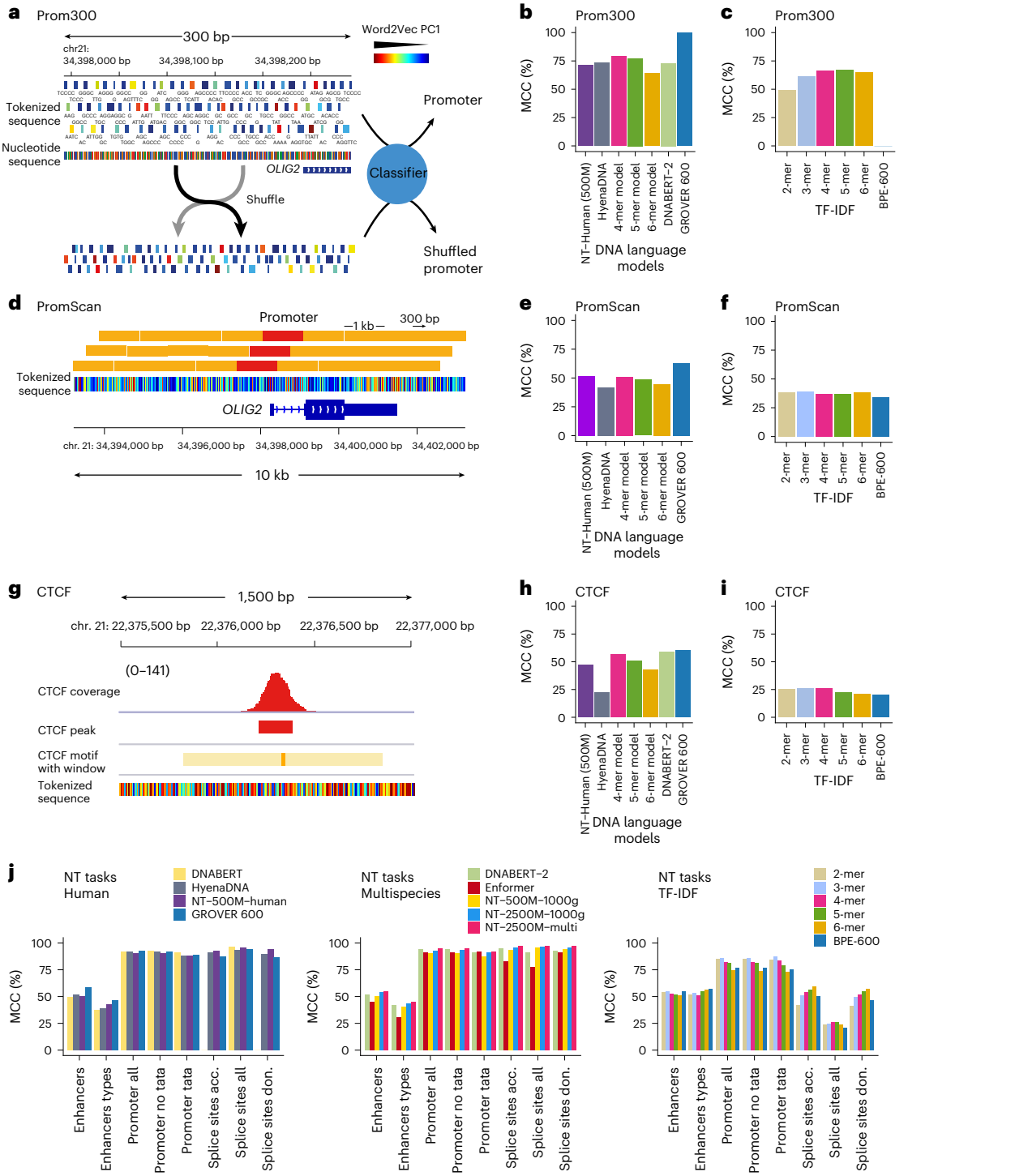


Fig. 6 | GROVER outperforms other models for biological fine-tuning tasks. GROVER tokens (BPE-600) are coloured by their rank in PC1 of W2V embedding. **a**, Prom300 promoter classification, defined as sequences around TSS −249/+50 nt. Promoters are classified into real promoters versus promoters with shuffled BPE-600 tokens. The *OLIG2* promoter is shown in the context of the nucleotide sequence with A in green, C in blue, G in yellow and T in red. **b**, MCC for the Prom300 task of GROVER, human NT, HyenaDNA, DNABERT-2 and models of fixed-size *k*-mers. **c**, Comparison of MCC for the Prom300 task to TF-IDF random forest classifiers with two- to six-nucleotide-long *k*-mers and BPE-600. **d**, For PromScan promoter assignment, windows are defined in regions of 1,001 bp around the TSS +/− 10 kb with an offset of 300 bp. PromScan classifies overlap with the TSS. Shown is the *OLIG2* promoter with BPE-600. **e**, MCC for the PromScan task of GROVER versus human NT, HyenaDNA and models of fixed-size *k*-mers. DNABERT-2 is not included due to problems with memory with this task. **f**, Comparison of MCC for the PromScan task to TF-IDF random forest classifiers with two- to six-nucleotide-long *k*-mers and BPE-600. **g**, CTCF binding prediction task, where regions are defined as CTCF binding motifs ± 500 bp and GROVER is trained for CTCF binding, determined by ChIP–seq in HepG2 cells. Depicted is a random region in the genome with a prominent CTCF binding peak on a CTCF binding motif, in the context of BPE-600. **h**, MCC for the CTCF-motif-binding task of GROVER versus human NT, HyenaDNA, DNABERT-2 and models of fixed-size *k*-mers. **i**, Comparison of MCC for the CTCF-motif-binding task to TF-IDF random forest classifiers with two- to six-nucleotide-long *k*-mers and BPE-600. **j**, MCC for the tasks of the NT study[8], for which human data are available. MCCs were obtained from the study and complemented with performance metrics for GROVER and TF-IDF random forest classifiers with two- to six-nucleotide-long *k*-mers and BPE-600. acc., acceptor; chr., chromosome; don., donor.

learning sequence context. It is still an open question how protein–DNA binding is generally encoded in the DNA beyond direct binding motifs. The larger context for this and other questions on genome function can now be addressed through extracting the learned representations from GROVER.

Discussion

We have built GROVER, a foundation DLM with an optimized vocabulary for the human genome, selected using next-*k*-mer prediction, a fine-tuning task that is independent of the structure of the foundation model and thus can handle different vocabulary sizes and tokenization strategies without directly selecting models for biological tasks. GROVER can grasp DNA language structure by learning both characteristics of the tokens and larger sequence contexts. It outperforms similar models both for next-*k*-mer prediction and fine-tuning tasks that address promoter identification and DNA–protein binding. Thus, we have identified the vocabulary that well defines the information of the human genome as it can be extracted by a BERT model. However, we have also revealed that standard tasks to interrogate genome biology show good predictability from token frequencies alone, which indicates that there is a need to develop further tasks to interrogate learning independent of token frequency that target learning of biological sequence context.

GROVER can be a basis to extract the information content of the genome by learning its grammatical and general language structures via analysing trained token embeddings, learning gradients or through extracting attention from the foundation model. These can also be obtained from specific fine-tuning tasks to interrogate specific genome biology. Such tasks could be genome annotation with functional data, genotype-phenotype predictions or technical tasks such as data augmentation.

GROVER uses the human genome exclusively. For DNABERT-2 (ref. 13) and NT[8], different genomes are combined in one model. Although using one genome limits the training data, the models serve different purposes given that different genomes follow different language rules. For example, the human genome contains about 12% primate-specific Alu retrotransposons. The coding genome is relatively conserved over species; the non-coding genome is more unique. It is these language rules we aim to learn with GROVER in a transparent way, specifically for biomedical questions. Despite this one-species strategy, we are not compromising in regards to performance for biological fine-tuning tasks. However, this approach may be limited for smaller genomes.

Different layers of the genetic code can now be approached through these models, and it can be extracted how DNA is coding for protein and transcripts, for gene regulation, self-propagation and stability. In there lies not only the key to genotype-to-phenotype prediction and the information of what in DNA makes us human but also information about predisposition to disease and treatment responses, which is to a large extent encoded in the patients' general and somatic genomes. DLMs like GROVER therefore have the potential to substantially push progress in personalized medicine.

Methods

Unless otherwise specified as being written in R (v.4.2.1), all code is written in Python (v.3.12.2) with Scikit-learn (v.1.4.2).

Data

We used the *Homo sapiens* (human) genome assembly GRCh37 (*hg19*) and only take into account the sequences that contain A, C, G and T. Tokenization was performed as described below. Each chromosome is split into windows varying between 20 and 510 tokens in size. Specifically, with a 50% probability, the size of a window is 510. With another 50% probability, its size is a random integer between 20 and 510. Eighty percent of the windows are taken as a training set and the remaining windows as a test set.

Byte-pair tokenization

BPE or byte-pair tokenization was originally developed as a text compression algorithm[14] and is now widely adapted as a tokenization strategy for transformer models on natural languages like GPT-3 (ref. 4). We adapted the tokenizer from ref. 24 for genome sequence and used up to 5,000 cycles of tokenization.

Tokens are visualized in a Word cloud with the R package Wordcloud2 (https://github.com/Lchiffon/wordcloud2).

The optimal vocabulary was selected through performance assessment with next-*k*-mer prediction with GROVER.

The DLM GROVER

GROVER adapts the transformer encoder BERT architecture[9]. It takes as input tokenized sequences of up to 510 tokens in length. In addition to the vocabulary generated from the genome, GROVER takes five special tokens: CLS, PAD, UNK, SEP and MASK. These special tokens are commonly used in language models: CLS represents the classification token, PAD is used for padding the right side of the sequence in case it is shorter than the maximum input length of the model, UNK is for sequences of nucleotides that do not belong to the vocabulary, SEP is used to indicate the end of a sequence, and MASK represents the masked tokens. The model is trained for masked token prediction.

In a given input sequence, 2.2% of the tokens are selected, of which 80% are substituted with a special mask [MASK] token; 10% of tokens are randomly replaced with standard tokens (that is, any token different from the class [CLS], pad [PAD] or mask [MASK] token).

To pretrain the model, we gather more than 5 million samples from the genome. Training was carried out on clusters of A100 graphics processing units, on batches of sizes 64 with an Adam optimizer and a learning rate of 4^{-4}, epsilon 10^{-6}, beta 0.99, maximum input length of 50, dropout probability of 0.5, and 0.022 probability of masking.

Next-k-mer prediction

For model validation and selection of the optimal vocabulary, we used a fine-tuning task of next-*k*-mer prediction that we previously developed[10]. It allows us to compare different foundation models that rely on context learning independent of how their vocabulary was generated, the size of the vocabulary or the learning parameters. The task is not dependent on a specific biological question. The principle of next-*k*-mer prediction is to take the pretrained language model and fine-tune it to predict the next *k*-mer, where *k* is 2, 3, 4, 5 and 6.

Chromosome 21 is split into sequences of 510 nucleotides, where we keep the first 56 nucleotides of each sequence. We randomly select 500,000 sequences, 80% for training and 20% for testing.

The samples are defined as the first 50 nucleotides of each sequence. For the labels, we take the *k* nucleotides that follow the 50 nucleotides. The next-*k*-mer model has 4^k different classes: that is, 16, 64, 256, 1,024 and 4,096, respectively, which are all the permutations of *k* nucleotides. We use an Adam optimizer with learning rate 10^{-6}, epsilon 10^{-8}, beta 0.99, maximum input length of 50, dropout probability of 0.5 and batch_size 64.

For NT, HyenaDNA and DNABERT-2, the models are fine-tuned with the pretrained models provided by the authors.

From the models, we extract performance metrics and use accuracy of token prediction for the decision of the optimal vocabulary and comparison to other tokenization strategies. For comparison of byte-pair tokenization cycles and visualization with ggplot2 (v.3.4.1) in R (v.4.2.1), we use a loess fit with a 95% confidence interval.

k-mer models

For the *k*-mer models, we use the same parameters and samples as GROVER, differing only by the vocabulary and tokenization. Tokenization is performed with non-overlapping *k*-mers four, five and six nucleotides in length. The vocabularies consist of all the permutations

of k consecutive nucleotides (that is 256, 1,024 and 4,096, respectively) as well as the five special tokens described above.

W2V

For comparison of token embeddings, we use W2V as a static word embedding tool[20] that maps each word to a single vector. In general, this mapping function does not account for lexical ambiguity: that identical letter sequences can have multiple interpretations or different grammatical roles. We use W2V with a continuous bag-of-words approach for learning representations of words. Continuous bag-of-words predicts the middle word from surrounding words in a sentence with a small number of words as context. The order of the words is not taken into consideration. To generate the W2V embeddings, 300,000 sequences are randomly chosen from the training set. We use the W2V module of Gensim (https://radimrehurek.com/gensim/models/word2vec.html), with the following parameters: min_count = 1, vector_size = 768, window = 5.

Model embedding

We obtain a contextualized word representation that is the token embedding of the BERT model. To obtain the trained token embeddings of the model, we extract the weights of the layer 'word_embeddings'. Where needed, we derive either the embedding of all 12 transformer layers, a summarized version for each token sequence or a summarized version for sequences of 510 token lengths with the classify token (CLS).

Dimensionality reduction

We obtain dimensionality reduction from average token embeddings that are represented as vectors with length 768. PCA and UMAP were performed in R (v.4.2.1) with the packages 'stats' (v.4.2.1) and 'UMAP' (v.0.2.10.0), respectively, with default parameters. MEV as a measure for context learning[21] was extracted as the variance explained by PC1.

Clustering of the token embeddings and self-similarity was performed with hierarchical clustering of Euclidean distance with pheatmap (v.1.0.12).

Dimensionality reduction for extracted embeddings of genome windows was performed in Python. The whole genome was split in non-overlapping bins of 510 tokens. Then these sequences were used as input to the model to obtain the CLS token embedding of the last layer. The 768 dimensions of the embedding matrix were reduced to two dimensions using UMAP. UMAP was configured with standard parameters, including 15 nearest neighbours and a minimum distance of 0.1.

Self-similarity

Self-similarity was assessed as the cosine similarity of different embeddings from the same token sequence, separately for all 12 transformer layers of the BERT architecture.

Five thousand embeddings per token were gathered from the test set, and pairwise cosine similarity for each token was computed in every layer.

Genome annotation

Tokens and sequence windows were annotated to token characteristics and functional genomics data in R (v.4.2.1) with the GenomicRanges (v.1.50.2) package, for genome information BSgenome.Hsapiens.UCSC.hg19 (v.1.4.3) and TxDb.Hsapiens.UCSC.hg19.knownGene (v.3.2.2). Sequence was derived with Biostrings (v.2.66.0) to obtain GC, AC, AG and nucleotide content as well as k-mer frequencies. Gene element annotation was performed with ChIPSeeker (v.1.34.1). Regression of GC content was performed using the residuals of a loess regression from the stat (v.4.2.1) package. Strand annotation relative to transcription was obtained from TxDb.Hsapiens.UCSC.hg19.knownGene (v.3.2.2). ChromHMM annotation was used from the GM12878 lymphoblastoid cell line from ENCODE, downloaded via the UCSC genome browser (https://www.genome.ucsc.edu/), where the repeat masker was also obtained for annotating repeats. Repeat classes were differentiated into the displayed categories, which were pooled with the respective '?' category. Categories that are not displayed separately were pooled as 'other'. Replication strand and timing were obtained from OK-Seq data from K562 chronic myeloid leukaemia cells from the reanalysis by ref. 25 downloaded via GEO (GSE131417). Tokens were annotated by determining the proportion of tokens overlapping with a specific annotation. Genome regions were annotated by determining overlaps. For differential assignment of annotation by strand, only bins are visualized that have overlaps with the annotation in a unique direction. Replication timing is visualized only for early- and late-replicating DNA for clearer visualization than taking all four categories. Of the total ranges, 995 points of the 1,378,385 total ranges (0.07%) were regarded as outliers and not visualized. Their inclusion did not add information but impacted visibility.

Fine-tuning task promoter identification, Prom300

Promoter sequences were obtained from the EPD database (https://epd.epfl.ch/) and lifted over to the hg19 assembly. Promoters were defined as the TSS −249/+50 bp. The corresponding byte-pair sequences were retrieved from the byte-pair-tokenized genome or converted into k-mer tokenized sequences with $k = [4, 5, 6]$. For generating negative class samples, each tokenized sequence was split into eight chunks, and the tokens of six randomly selected chunks were shuffled among those six chunks. For the classification task for correct or shuffled sequence, the dataset was split 80%–10%–10% for training, validation and test. For NT, HyenaDNA and DNABERT-2, the models are fine-tuned with the pretrained models provided by the authors.

Fine-tuning task promoter scanning, PromScan

The same human promoter annotations as used in the Prom300 task were taken in 10 kb windows around the TSS. The sequence was divided into overlapping 1,001 bp windows with a step size of 300 bp. Training was performed for classification of the presence of a TSS in a respective window. Only the central TSS was considered, even in the presence of more than one TSS. The dataset was split 80%–10%–10% for training, validation and test. The training was done with the 'Trainer' method of the 'Transformers' library, with the hyperparameters learning_rate = 1×10^{-6}, batch size 8, warmup steps 50 and weight decay 0.01. For NT and HyenaDNA, the models are fine-tuned with the pretrained models provided by the authors. DNABERT-2 struggles with the memory required for the larger ranges and sample sizes of this task and was therefore excluded.

CTCF motif binding

CTCF ChIP–seq peaks in HepG2 cells were derived from the ENCODE project (https://www.encodeproject.org/experiments/ENCSR000BIE/). The human CTCF motif was retrieved from the JASPAR database (https://jaspar.genereg.net/matrix/MA0139.1/), and motif occurrences in the hg19 genome were derived with FIMO from the MEME suite using the default background model and $P < 10^{-4}$. Motifs that intersect with a CTCF peak were considered to be CTCF bound. Classification was performed for bound and unbound CTCF motifs for a 1 kb window around the binding motif with a split 80%–10%–10% for training, validation and test. Training was done with the 'Trainer' method of the 'Transformers' library, with the hyperparameters learning_rate = 1×10^{-6}, batch size 16, warmup steps 50 and weight decay 0.01. For NT, HyenaDNA and DNABERT-2, the models are fine-tuned with the pretrained models provided by the authors.

Fine-tuning task GUE

Datasets were obtained from https://github.com/MAGICS-LAB/DNABERT_2. We selected only the tasks with human data: promoter detection, core promoter detection, transcription factor binding site prediction and splice site prediction. Training was done with the

'Trainer' method of the 'Transformers' library, with the hyperparameters warmup_steps = 50, weight_decay = 0.01, learning_rate = $1 \times 10-4$ and adamw_torch as optimizer.

Fine-tuning task NT

Datasets were obtained from https://huggingface.co/datasets/InstaDeepAI/nucleotide_transformer_downstream_tasks, with the 'load_dataset' method and InstaDeepAI/nucleotide_transformer_downstream_tasks as parameter. The test split was used to report the results, and 5% of the training split was used for validation. We selected only the tasks with human data: promoter sequence prediction (promoter_all, promoter_no_tata, promoter_tata), enhancer sequence prediction (enhancers, enhancers_types) and splice site prediction (splice_sites_acceptors, splice_sites_all, splice_sites_donors). Training was done with the 'Trainer' method of the 'Transformers' library, with the hyperparameters learning_rate = 1×10^{-5} and batch size 8.

TF-IDF

Initially the sequences are tokenized with non-overlapping *k*-mers (2, 3, 4, 5, 6) and with the BPE of GROVER. We use 'TfidfVectorizer' from scikit (v.1.5) with default parameters to extract the features of each sequence of the training set. From these features, we train a random forest classifier and choose the best number of estimators between 100 and 2,000. This model is purely trained on token frequencies without any sense of grammar, syntax or overall 'language' context between tokens. It thus serves as a negative control for tasks that benefit from application of a language model.

Reporting summary

Further information on research design is available in the Nature Portfolio Reporting Summary linked to this article.

Data availability

Pretrained GROVER[26] with 600 cycles of BPE and the respective tokenised genome are available at https://huggingface.co/PoetschLab/GROVER. The model can be directly implemented with Python for any suitable fine-tuning task. The vocabulary for the tokenized hg19 genome (600 cycles)[27] is available as a data resource for fine-tuning models based on GROVER and can also be used to train different model architectures or for different purposes. The full datasets are available via Zenodo at https://doi.org/10.5281/zenodo.8373202 (ref. 28). Source data are provided with this paper.

Code availability

The full code in R and Python needed to reproduce the findings of this study is available via Zenodo at https://doi.org/10.5281/zenodo.8373202 (ref. 28). A tutorial on how to use GROVER as a foundation model is available via Zenodo at https://doi.org/10.5281/zenodo.8373158 (ref. 29). The tutorial includes full instructions for configuring GROVER for a fine-tuning task using the example of CTCF binding prediction. It is written as a Jupyter notebook with Python.

References

1. Lander, E. S. et al. Initial sequencing and analysis of the human genome. *Nature* **409**, 860–921 (2001).
2. Crick, F. H., Barnett, L., Brenner, S. & Watts-Tobin, R. J. General nature of the genetic code for proteins. *Nature* **192**, 1227–1232 (1961).
3. Vaswani, A. et al. Attention is all you need. In *Proc. Advances in Neural Information Processing Systems* (eds Guyon, I. et al.) (IEEE, 2017); https://proceedings.neurips.cc/paper/7181-attention-is-all
4. Brown, T. et al. Language models are few-shot learners. In *Proc. Advances in Neural Information Processing Systems* (eds Larochelle, H. et al.) 1877–1901 (IEEE, 2020).
5. Avsec, Ž. et al. Effective gene expression prediction from sequence by integrating long-range interactions. *Nat. Methods* **18**, 1196–1203 (2021).
6. Yang, M. et al. Integrating convolution and self-attention improves language model of human genome for interpreting non-coding regions at base-resolution. *Nucleic Acids Res.* **50**, e81 (2022).
7. Ji, Y., Zhou, Z., Liu, H. & Davuluri, R. V. DNABERT: pre-trained bidirectional encoder representations from transformers model for DNA-language in genome. *Bioinformatics* **37**, 2112–2120 (2021).
8. Dalla-Torre, H. et al. The Nucleotide Transformer: building and evaluating robust foundation models for human genomics. Preprint at *bioRxiv* https://www.biorxiv.org/content/10.1101/2023.01.11.523679.abstract (2023).
9. Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. Preprint at https://doi.org/10.48550/arXiv.1810.04805 (2018).
10. Sanabria, M., Hirsch, J. & Poetsch, A. R. Distinguishing word identity and sequence context in DNA language models. Preprint at *bioRxiv* https://www.biorxiv.org/content/10.1101/2023.07.11.548593 (2023).
11. Mo, S. et al. Multi-modal self-supervised pre-training for large-scale genome data. Poster at NeurIPS 2021 AI for Science Workshop. *OpenReview.net* https://openreview.net/forum?id=fdV-GZ4LPfn (2021).
12. Nguyen, E. et al. Hyenadna: long-range genomic sequence modeling at single nucleotide resolution. Preprint at https://arxiv.org/pdf/2306.15794 (2023).
13. Zhou, Z. et al. Dnabert-2: efficient foundation model and benchmark for multi-species genome. Preprint at https://arxiv.org/pdf/2306.15006 (2023).
14. Ziv, J. & Lempel, A. A universal algorithm for sequential data compression. *IEEE Trans. Inf. Theory* **23**, 337–343 (1977).
15. Cooper, D. N. & Youssoufian, H. The CpG dinucleotide and human genetic disease. *Hum. Genet.* **78**, 151–155 (1988).
16. Sinsheimer, R. L. The action of pancreatic desoxyribonuclease. I. Isolation of mono- and dinucleotides. *J. Biol. Chem.* **208**, 445–459 (1954).
17. Holliday, R. & Pugh, J. E. DNA modification mechanisms and gene activity during development. *Science* **187**, 226–232 (1975).
18. Poetsch, A. R. & Plass, C. Transcriptional regulation by DNA methylation. *Cancer Treat. Rev.* **37**, S8–S12 (2011).
19. Yoder, J. A., Walsh, C. P. & Bestor, T. H. Cytosine methylation and the ecology of intragenomic parasites. *Trends Genet.* **13**, 335–340 (1997).
20. Mikolov, T., Chen, K., Corrado, G. & Dean, J. Efficient estimation of word representations in vector space. Preprint at https://arxiv.org/pdf/1301.3781.pdf (2013).
21. Ethayarajh, K. How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings. Preprint at https://arxiv.org/pdf/1909.00512 (2019).
22. Sultana, T. et al. The landscape of L1 retrotransposons in the human genome is shaped by pre-insertion sequence biases and post-insertion selection. *Mol. Cell* **74**, 555–570.e7 (2019).
23. The ENCODE Project Consortium. An integrated encyclopedia of DNA elements in the human genome. *Nature* **489**, 57–74 (2012).
24. Sennrich, R., Haddow, B. & Birch, A. Neural machine translation of rare words with subword units. Preprint at https://arxiv.org/pdf/1508.07909.pdf (2015).
25. Pongor, L. S. et al. BAMscale: quantification of next-generation sequencing peaks and generation of scaled coverage tracks. *Epigenetics Chromatin* **13**, 21 (2020).
26. Sanabria, M., Hirsch, J. & Poetsch, A. R. GROVER pretrained DNA language model of the human genome. *Zenodo* https://doi.org/10.5281/zenodo.8373117 (2023).

27. Sanabria, M., Hirsch, J. & Poetsch, A. R. GROVER tokenized Human Genome hg19 data set. *Zenodo* https://doi.org/10.5281/zenodo.8373053 (2023).
28. Sanabria, M., Hirsch, J., Joubert, P. & Poetsch, A. R. The human genome's vocabulary as proposed by the DNA language model GROVER - the code to the paper. *Zenodo* https://doi.org/10.5281/zenodo.8373202 (2023).
29. Sanabria, M., Hirsch, J. & Poetsch, A. R. GROVER DNA language model tutorial. *Zenodo* https://doi.org/10.5281/zenodo.8373158 (2023).

Acknowledgements

This work was supported by the Center for Scalable data analytics and artificial intelligence (Scads.AI) Dresden-Leipzig. This work was partially funded by the Center for Advanced Systems Understanding (CASUS) which is financed by Germany's Federal Ministry of Education and Research (BMBF) and by the Saxon Ministry for Science, Culture, and Tourism (SMWK) with tax funds on the basis of the budget approved by the Saxon State Parliament. A.R.P. was supported by the Mildred Scheel Early Career Center Dresden P2, funded by the German Cancer Aid. M.S. was supported by a TU Dresden Junior Fellowship and a Maria Reiche Postdoctoral fellowship of TU Dresden. J.H. was supported by the TU Dresden programme 'FOSTER – Funds for Student Research'.

Author contributions

A.R.P. has conceptualized the study. M.S., P.M.J. and J.H. applied the models and implemented the fine-tuning tasks. All authors designed the fine-tuning tasks and analysed the data. A.R.P. wrote the manuscript with input from M.S. and J.H.

Funding

Open access funding provided by Deutsches Krebsforschungszentrum (DKFZ).

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s42256-024-00872-0.

Correspondence and requests for materials should be addressed to Anna R. Poetsch.

Peer review information *Nature Machine Intelligence* thanks Ramana Davuluri and the other, anonymous, reviewer(s) for their contribution to the peer review of this work.

nature portfolio

Corresponding author(s): Anna R. Poetsch

Last updated by author(s): Jun 17, 2024

Reporting Summary

Nature Portfolio wishes to improve the reproducibility of the work that we publish. This form provides structure for consistency and transparency in reporting. For further information on Nature Portfolio policies, see our Editorial Policies and the Editorial Policy Checklist.

Statistics

For all statistical analyses, confirm that the following items are present in the figure legend, table legend, main text, or Methods section.

n/a	Confirmed	
☐	☒	The exact sample size (*n*) for each experimental group/condition, given as a discrete number and unit of measurement
☒	☐	A statement on whether measurements were taken from distinct samples or whether the same sample was measured repeatedly
☐	☒	The statistical test(s) used AND whether they are one- or two-sided *Only common tests should be described solely by name; describe more complex techniques in the Methods section.*
☒	☐	A description of all covariates tested
☒	☐	A description of any assumptions or corrections, such as tests of normality and adjustment for multiple comparisons
☐	☒	A full description of the statistical parameters including central tendency (e.g. means) or other basic estimates (e.g. regression coefficient) AND variation (e.g. standard deviation) or associated estimates of uncertainty (e.g. confidence intervals)
☐	☒	For null hypothesis testing, the test statistic (e.g. *F*, *t*, *r*) with confidence intervals, effect sizes, degrees of freedom and *P* value noted *Give P values as exact values whenever suitable.*
☒	☐	For Bayesian analysis, information on the choice of priors and Markov chain Monte Carlo settings
☒	☐	For hierarchical and complex designs, identification of the appropriate level for tests and full reporting of outcomes
☐	☒	Estimates of effect sizes (e.g. Cohen's *d*, Pearson's *r*), indicating how they were calculated

Our web collection on statistics for biologists contains articles on many of the points above.

Software and code

Policy information about availability of computer code

Data collection

The human genome version hg19, supporting public data from ENCODE and publications, which are cited in the methods section.
https://doi.org/10.5281/zenodo.8373117
https://doi.org/10.5281/zenodo.8373053

Data analysis

Custom code in python and R, which uses other data packages, which are cited in the methods section.
https://doi.org/10.5281/zenodo.8373202
https://doi.org/10.5281/zenodo.8373158
Homo sapiens (human) genome assembly GRCh37 (hg19)

R (4.2.1)
ggplot2 (3.4.1)
stats (4.2.1)
UMAP' (0.2.10.0)
pheatmap (1.0.12)
GenomicRanges (1.50.2)
BSgenome.Hsapiens.UCSC.hg19 (1.4.3)
TxDb.Hsapiens.UCSC.hg19.knownGene (3.2.2)
Biostrings (2.66.0)
ChIPSeeker (1.34.1)
https://radimrehurek.com/gensim/models/word2vec.html
https://github.com/Lchiffon/wordcloud2

Python (3.12.2)
Scikit (1.4.2)

For manuscripts utilizing custom algorithms or software that are central to the research but not yet described in published literature, software must be made available to editors and reviewers. We strongly encourage code deposition in a community repository (e.g. GitHub). See the Nature Portfolio guidelines for submitting code & software for further information.

Data

Policy information about availability of data

All manuscripts must include a data availability statement. This statement should provide the following information, where applicable:

- Accession codes, unique identifiers, or web links for publicly available datasets
- A description of any restrictions on data availability
- For clinical datasets or third party data, please ensure that the statement adheres to our policy

GSE131417
https://www.genome.ucsc.edu/
https://www.encodeproject.org/experiments/ENCSR000BIE/
https://jaspar.genereg.net/matrix/MA0139.1/
https://github.com/MAGICS-LAB/DNABERT_2
https://huggingface.co/datasets/InstaDeepAI/nucleotide_transformer_downstream_tasks

Human research participants

Policy information about studies involving human research participants and Sex and Gender in Research.

Reporting on sex and gender	NA
Population characteristics	NA
Recruitment	NA
Ethics oversight	NA

Note that full information on the approval of the study protocol must also be provided in the manuscript.

Field-specific reporting

Please select the one below that is the best fit for your research. If you are not sure, read the appropriate sections before making your selection.

☒ Life sciences ☐ Behavioural & social sciences ☐ Ecological, evolutionary & environmental sciences

For a reference copy of the document with all sections, see nature.com/documents/nr-reporting-summary-flat.pdf

Life sciences study design

All studies must disclose on these points even when the disclosure is negative.

Sample size	Sample sizes were determined arbitrarily
Data exclusions	remaining tokens for the nucleotides A, C, and T were excluded, because they were artefacts from tokenisation adjacent to Ns.
Replication	Replication is not applicable to this study, because there is no relevant variability to be expected from the model training.
Randomization	Randomization is not applicable to this study, because it is not the type of statistics that requires randomization
Blinding	There are no human subjects involved in this study

Reporting for specific materials, systems and methods

We require information from authors about some types of materials, experimental systems and methods used in many studies. Here, indicate whether each material, system or method listed is relevant to your study. If you are not sure if a list item applies to your research, read the appropriate section before selecting a response.

Materials & experimental systems

n/a	Involved in the study
☒	☐ Antibodies
☒	☐ Eukaryotic cell lines
☒	☐ Palaeontology and archaeology
☒	☐ Animals and other organisms
☒	☐ Clinical data
☒	☐ Dual use research of concern

Methods

n/a	Involved in the study
☒	☐ ChIP-seq
☒	☐ Flow cytometry
☒	☐ MRI-based neuroimaging