PHYLOLM: INFERRING THE PHYLOGENY OF LARGE LANGUAGE MODELS AND PREDICTING THEIR PERFORMANCES IN BENCHMARKS

Nicolas Yax
LNC2, INSERM, Paris, France
DEC, ENS, Paris, France
Inria, France
nicolas.yax@ens.psl.eu

Pierre-Yves Oudeyer*
Inria and University of Bordeaux, France

Stefano Palminteri*
LNC2, INSERM, Paris, France
DEC, ENS, Paris, France

* equal contribution

ABSTRACT

This paper introduces PhyloLM, a method adapting phylogenetic algorithms to Large Language Models (LLMs) to explore whether and how they relate to each other and to predict their performance characteristics. Our method calculates a phylogenetic distance metric based on the similarity of LLMs' output. The resulting metric is then used to construct dendrograms, which satisfactorily capture known relationships across a set of 111 open-source and 45 closed models. Furthermore, our phylogenetic distance predicts performance in standard benchmarks, thus demonstrating its functional validity and paving the way for a time and cost-effective estimation of LLM capabilities. To sum up, by translating population genetic concepts to machine learning, we propose and validate a tool to evaluate LLM development, relationships and capabilities, even in the absence of transparent training information.

1 INTRODUCTION

The Large Language Models (LLMs) landscape is vast and rapidly expanding, comprising both private and open-access models. Each day a few hundreds of new language models are created on the huggingface hub among which most will not be benchmarked, and a small minority are transparent about the training details. Evaluating these models presents challenges due to the sheer volume and the complexity of assessing their true capabilities. The evaluation methods used today mostly rely on a multitude of benchmarks, each focused on specific domains like reasoning or question-answering (Chollet, 2019; Hendrycks et al., 2021; Srivastava et al., 2023). However, tracking LLMs evolution and progress using benchmarks presents inherent limitations, including the fact that they are rather domain-specific, meaning that to get a full picture of a model's capabilities one has to run multiple costly tests that are prone to contamination (Chang et al., 2023; Deng et al., 2023; Liang et al., 2023). Moreover, the opacity of algorithmic and training data specifications in many models, adds further complexity and constraints to monitor progress in LLMs (Liao & Vaughan, 2023).

Our approach stems from the observation that most of the newly released models are not created ex-nihilo (from scratch). In fact, they rather inherit features from existing ones, such as training data or initial weights. We reasoned that we could therefore think about LLMs development as

code : https://github.com/Nicolas-Yax/PhyloLM
notebook : https://colab.research.google.com/drive/1GDbmEMmCVEOwhYk6-1AothdXeAlnqZ_j?usp=copy

an "evolutionary" process and therefore study their relationships and functional properties with conceptual and quantitative tools borrowed from genetics.

In the field of Phylogeny, algorithms have been developed that reconstruct evolutionary trees to understand evolutionary relationships among species (Takezaki & Nei, 1996). The idea of applying these methods, initially developed for biology, to cultural artefacts is not new. Previous studies yielded particularly useful insights into the evolution of popular tales, languages, or craft assemblages (Atkinson et al., 2008; Dawkins, 1976; d'Huy, 2013; Gray et al., 2010; Tehrani & d'Huy, 2017; Tehrani & Collard, 2009). We hypothesize here that LLMs, which are a new kind of cultural artefact (in the sense that they are productions of humans that convey information about the culture of their creators and users), may also be studied using similar tools.

Thus, we here apply a conceptually similar approach to LLMs and, by doing so, we make several contributions. In a **first contribution**, we introduce an algorithm, **PhyloLM**, inspired by a simplified phylogenetic model, but specifically tailored for Large Language Models (LLMs), which core idea is to consider that generated tokens are to contexts what alleles are to genes in genetics. This analogy makes it possible to apply algorithms from the genetics framework to LLMs and to generate distance matrices and dendrograms. In addition to presenting the underlying theory, we also explore the hyperparameters of our algorithm to strike a balance between precision and computational efficiency.

In our **second contribution**, we analyze the resulting phylogenetic trees ("dendrograms") and confirm that **PhyloLM** is capable of correctly retrieving known relationships between LLMs and overall correctly capturing models families and sub-families. Our analysis primarily focuses on open-access model families (Llama (Touvron et al., 2023a;b), Mistral (Jiang et al., 2023), Bloom (BigScienceWorkshop et al., 2023), Pythia (Biderman et al., 2023), Falcon (Almazrouei et al., 2023), OPT (Zhang et al., 2022), Qwen (Bai et al., 2023) and Gemma (Team et al., 2024b) families), where ground truth information is available, but also provides insights into fine-tuning relationships for proprietary models (GPT-3 (Brown et al., 2020), 3.5 (Ouyang et al., 2022), 4 (OpenAI et al., 2023), Claude , Palm (Chowdhery et al., 2022) and Gemini models (Team et al., 2024a)). Finally, in our **third contribution**, we examine whether phylogenetic distance can also be used to predict performance in several benchmarks, thus showing that the utility of **PhyloLM** extends to the assessment of functional properties of LLMs.

To sum up, our study illustrates the potential of leveraging methods from genetics to understand how models evolve, shedding light on their relationships and functional capabilities in a relatively cost-efficient manner, even in the absence of transparent training information and also without direct access to the model.

$$S(P_1, P_2) = \frac{\sum_{g \in G} \sum_{a \in A_g} P_1(a|g) P_2(a|g)}{\sqrt{(\sum_{g \in G} \sum_{a \in A_g} P_1(a|g)^2)(\sum_{g \in G} \sum_{a \in A_g} P_2(a|g)^2)}} \tag{1}$$

Equation 1: **Similarity computation** with P_1 and P_2 two populations seen as probability distribution of alleles a given a gene g estimated empirically in the selected populations. G is the set of genes considered and A_G the set of possible alleles for this gene and matrix S is the similarity matrix (bounded in [0,1]). In genetics people tend to use a distance matrix D to plot dendrograms derived from the similarity matrix with this formula $D(P_1, P_2) = -\log(S(P_1, P_2))$ (Takezaki & Nei, 1996). Seen from the autoregressive LLM framework, 'populations' are LLMs, 'genes' are contexts and 'alleles' are the different tokens in the vocabulary : $P(a|g) = LLM(t|c)$

2 METHODS

2.1 TRANSLATING PHYLOGENETIC ALGORITHMS TO LLMS

In the current landscape, LLMs predominantly operate on an autoregressive basis, wherein they learn the conditional probability denoted as $LLM(t|c)$. Here, LLM represents the probability learned by the language model, t signifies a token, and c denotes the context in which to sample token t. Transposing genetic methods to LLMs involves establishing analogies for the elements of the phylogenetic analysis, namely genes, alleles, and populations. Drawing a parallel with the notation for

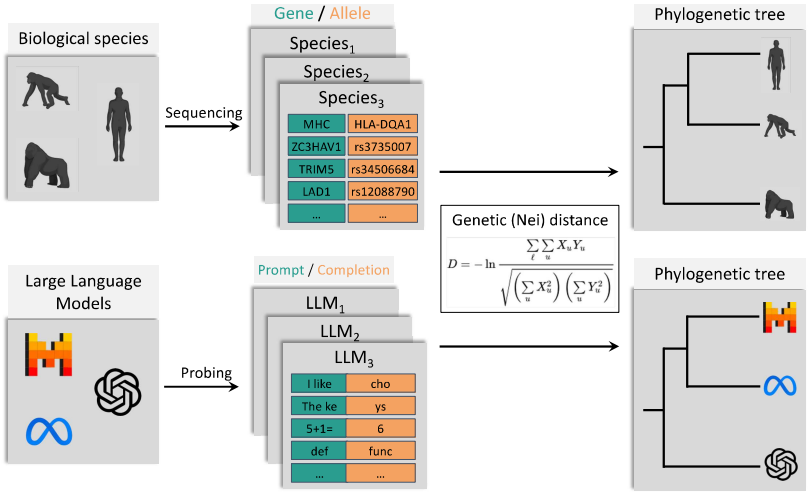


Figure 1: **Analogy between running human genetic studies and LLMs genetic studies.** The first stage consists in selecting genes (for both humans and LLMs). Then alleles are collected for each individual in the population and will be used to compare the populations (either populations of humans or LLMs seen as populations). Finally these data go through the Nei distance computation (Takezaki & Nei, 1996) that returns a distance matrix that can then be turned into dendrograms using the NJ algorithm (Saitou & Nei, 1987) in the same way for both humans and LLMs.

populations in the Nei genetic distance (see Equation 1) (Takezaki & Nei, 1996), $P_1(a|g)$ with a an allele and g a gene, we propose that LLMs play the role of populations (i.e., the set of the individuals belonging to a given population); contexts (or "prompts") are aligned to genes (i.e., portions of DNA); finally, tokens align with alleles (i.e., variants in the DNA sequence).

To substantiate this analogy, consider that, in the realm of genetics, populations are conceptualized as probability distributions of DNA, represented by $P(a|g)$, where a stands for specific alleles at gene locations. Gene-specific alleles are then considered to be probabilistically drawn from the abstract statistical construct that is the population, akin to context-specific tokens are probabilistically generated from Large Language Models, expressed as $LLM(t|c)$ (t being a token likely to follow text c). The generated text can therefore be seen as a thread of DNA, comprised of tokens (alleles) sampled in contexts (genes) according to a probability distribution defined by the LLM.

To elucidate this crucial point, consider a tokenized text sequence: 'I' '_like' '_choco' 'late'. This sequence can be analogous to a DNA thread represented as 'I_like_chocolate'. Breaking it down, the allele I corresponds to the gene ϵ (empty text), _like aligns with the gene I, _choco associates with the gene I_like, and late is linked to the gene I_like_choco. Now, consider another individual represented by 'I' '_prefer' '_ice' '_cream'. These two individuals share exactly two genes: ϵ, for which they possess the same allele I, and the gene I, for which they have distinct alleles (_like and _prefer). They do not share any further genes, as their prefixes diverge beyond this point.

The algorithm is illustrated in Figure 1. The initial step involves collecting model outputs to contexts (genes). Given a set of LLMs, a set of 'genes', and the specified number of individuals in each population (i.e., the number of times the model is queried on each gene refered to as the number of probes) as N, the models are queried for a single token N times. This process generates the matrix P, which serves as an approximation of $P(a|g)$, the proportion of the population with allele a to gene g. Subsequently, based on this approximation, the similarity matrix S is computed using the Nei genetic distance formula (Takezaki & Nei, 1996) depicted in Equation 1. The pseudo code of PhyloLM can be found in Algorithm 1 in Appendix C.

2.2 Choice of the set of genes

The implementation of phylogenetic algorithms requires selecting specific genes that show enough evolutionary changes among the species studied to differentiate them, while still retaining enough

similarity to trace relationships between closely related species (Grünwald et al., 2017). If these genes mutate too quickly and are completely altered between similar species, they will not provide useful information about their evolution. Conversely, if they are too stable and show no changes across the species being considered, they are also not informative. These genes must strike a balance between stability and variation among the species studied.

That is why we need to carefully select genes (i.e., prompt contexts) that could show a moderate variance between LLMs. Recent LLM development focused a lot on instruction tuning, reasoning and coding (Brown et al., 2020; Chiang et al., 2023; OpenAI et al., 2023; Taori et al., 2023). Selecting contexts on these topics might show a relevant variance between generations of language models as well as finetuning refinements that improved these models on these specific topics.

Furthermore, contexts ('genes') which are very likely to belong to the training data of these LLMs can suffer from contamination issues and generate very low variance[1]. To obviate this issue, we used contexts (or a 'gene' set) taken from recent test benchmarks because, in principle, LLMs shouldn't be trained on this data. To further assess the robustness of our approach and study the impact of the choice of the set of 'genes', we took our contexts from two different test sets: open-web-math (Paster et al., 2023) and MBXP (Athiwaratkun et al., 2023). They address different capabilities of LLMs: reasoning and coding, respectively, which are very relevant in recent LLM-related research and are therefore likely to deliver useful results.

The exact selection of contexts from the benchmarks consisted of randomly and uniformly selecting lines from the solution column in the datasets and truncating the text to leave it open for LLMs to complete the sentence. To decide the length of the contexts we need once more to think about making 'genes' show a moderate completion variance. If the context is only a few tokens long it may not be informative enough for LLMs to understand the topic of the context (that is relevant for the recent evolution of language models as discussed above) but also to follow the logics of the text that would constrain the generation. On the other hand, making it hundreds of tokens long will induce additional costs without necessarily improving the variability balance. That is why we decided to truncate randomly and uniformly between the 20th and 100th characters in each text (5 to 30 tokens approximately). 'Gene' examples are shown in Appendix A. More details about the impact of the gene length can be found in appendix K.1.

2.3 Selection of the hyper-parameters of the distance matrices

We devised two complementary analyses to estimate the right hyperparameters to run PhyloLM. The hyperparameters are the 'gene' set, the number of probes and sampling parameters from the LLM (see Appendix B). Testing the gene set is more difficult as testing thousands of different combinations of genes would come at a very expensive cost. Thus we limited ourselves at 2 parameters of the gene set : the topic (math and code in this paper) and the size of the gene set G. In this section we will investigate the impact of G and N in the math gene set, the results for the code gene set are in Appendix D.

First we investigate how G and N affect the variability of the distance matrix, namely how much the similarity matrix changes between different estimations. We focus on similarity matrices (the matrix S in Equation 1) instead of distance matrices at this point as they are bounded in [0,1] making them a lot easier to plot and compare. Then, once the variance is controlled, what combination of G and N approximate reasonably well a very high G' and N' distance matrix.

To assess the impact of the number of contexts ('genes') G and the number of probes/individuals N for each dataset, the algorithm was executed across a range of gene set sizes G (varying between 16 and 256 genes per run) and individuals N (ranging from 8 to 128) building similarity matrices. This optimization process, aimed at testing the best values for the algorithm hyperparameters, is particularly computationally expensive. Therefore it was only run on the 5 smallest OPENAI models (ada,babbage,text-ada-001,text-babbage-001 and babbage-002), in order to minimize the costs. Thus similarity matrices in this section are 5×5 making it an estimate of what could be a larger distance matrix at a very low cost.

[1]To understand this point, imagine using "*May the force be with*" as context. All models will complete this sentence with "*you*", thus making impossible establishing distance matrices between them

To investigate the variability of PhyloLM for different combination of hyperparameters, we composed 8 sets of genes of size G, each with different genes. Each set of gene is probed N times to build a similarity matrix $S_{G,N,i}$, $i \in [|0, 7\|]$ representing the independant set of genes of size G used to generate the matrix with N probes. A variance computation over this set of matrices is finally performed yielding a matrix V containing the variance of each distance between 2 models : ${V_{G,N}}^2 = \frac{1}{8}\sum_i \left(S_{G,N,i} - \left(\frac{1}{8}\sum_j S_{G,N,j}\right)\right)^2$. The square operator is applied coefficient by coefficient. The final variability score is the mean value of the coefficients in the matrix $v_{G,N} = \mu\left(\sqrt{{V_{G,N}}^2}\right)$.

Then we investigated the impact of these hyperparameters when trying to approximate a high precision matrix. For this purpose, we compute the variance around a very expensive distance matrix $S_{G',N'}$ with $G' = 2048$ and $N' = 128$. The gene set for the high precision matrix is independent from the lower size set of genes used to estimate it. The formula to compute this variance around the high precision matrix is ${V'_{G,N}}^2 = \frac{1}{8}\sum_i \left(S_{G,N,i} - S_{G',N'}\right)^2$. The final metric is the mean value in the matrix $v'_{G,N} = \mu\left(\sqrt{{V'_{G,N}}^2}\right)$.

2.4 Alignment of the results across different tokenization

In situations where models do not share the same tokenizer, comparing only the first alleles generated can pose challenges. For instance, if the context is "*The president of the US is Joe*," and one model could complete with "*Biden*" in one token while another could complete with "*Bi*" "*den*" in two tokens they would be considered as different alleles while both LLM meant the same completion.

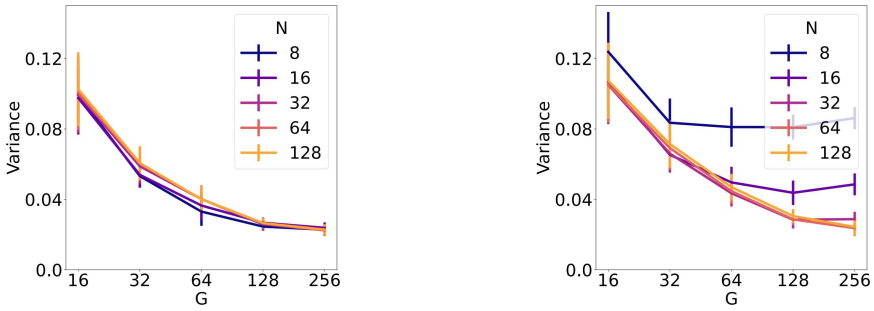
To mitigate this issue of tokenizer alignment, a proxy approach was employed by only using the first 4 characters of the generated text instead of the first token. Practically, each model was instructed to generate at least 4 tokens (tokens are at least 1 character long) and the comparison focused on the first 4 characters in the concatenation of these tokens. In the previous example, the word "*Biden*" generated in one token or in two ("*Bi*" and "*den*") would have been considered as the same response, because the first 4 characters ("*Bide*") constitute the same 'allele', despite having being tokenized differently. In other words, we are retokenizing the text using a tokenizer with a vocabulary of words that are 4 characters long, and then comparing the first token of the generated text with this new tokenization scheme. An example of the results of such a proxy approach is presented in Appendix A. Further details about why 4 characters is efficient are discussed in Appendix K.2.

2.5 Visualization of the results

From a distance matrix obtained by the phylogenetic algorithm it is usual to plot dendrograms representing a possible evolution between the entities in the distance matrix. For this purpose many different algorithms exist and we chose the Neighbour Joining (NJ) technique (Saitou & Nei, 1987) for its simplicity, efficiency and being a common choice in genetics. We plotted unrooted trees as they are easier to make figures that fit in a paper and are more adapted to LLM evolution than rooted ones. The analysis of the resulting dendrograms also allowed us to validate the capability of our algorithm to predict actual relationship between LLMs in cases where the ground truth is known.

2.6 Predict benchmark scores from genetic distance

We explored whether genetic distance can predict model performance by using logistic regression to estimate benchmark scores of large language models based on their similarity to other models. Due to the high dimensionality of the similarity matrix, we reduced the input dimensions to 15 using Independent Component Analysis (ICA), resulting in 15 parameters to learn from approximately 100 data points per fit. We then applied a sigmoid function to the output to scale the predictions between 0 and 1, corresponding to benchmark scores ranging from 0% to 100%. Since benchmark scores can be highly correlated within a family of models, we employed a leave-one-family-out method (see Figure 5 (a)). This involved training the regressor on all families but one and testing it on the excluded family. A Mean Squared Error loss was used with an Adam optimizer (learning rate of 10^{-3}).



(a) Variability of distance matrices

(b) Distance to the high precision matrix

Figure 2: **Hyperparameters impact on distance matrices in the math set of genes** (a) shows the variability of distance matrices for different number of genes G and number of probes N in the math benchmark. Each set of genes of specified size contains different and independent genes from the other matrices for a total of 8 distance matrix for each data point in the figure. (b) shows the distance to the high precision matrix made of 2048 genes and N=128 in the math benchmark. Errorbars represent the standard error of the mean.

We tested the benchmarks available on the hugging face open llm leaderboard which includes MMLU, ARC, Hellaswag, TruthfulQA, Winogrande and GSM8k (HuggingFaceH4) and only included open access models for which the scores are available on the leaderboard. Thus we didn't include proprietary models in this study as, as explained in later sections, distance computation is slightly biased for these models and benchmark scores are not obtained in the same conditions as in the leaderboard (number of shots, CoT, ...). The benchmarks used for the 'gene' set were distinct from these benchmarks to avoid any type of contamination between the 'alleles' used to generate genetic distances and the performance of the models in the considered benchmark tasks.

3 Experiments and Results

3.1 What is the impact of hyperparameters on the distance matrix?

We first ran the hyperparameters' optimization process explained in Methods2.3 and plotted the results in figure 2a left side. This graph shows a clear decrease in the variability as the number of 'genes', G grows with almost no effect from N. This is interesting : it seems that having different sets of 'genes' doesn't appear to change the similarity matrix as long as there are enough of them (at least in the open-web-math and mbxp dataset - see Appendix D for the results on the code set of 'genes').

However this method doesn't make it possible to find a good N, indeed, the probability for two models to generate the same token in the same context in only one try is quite low. Therefore, a very low N will make all models appear particularly different making the similarity matrix look like the identity matrix yielding unsatisfactory results despite having a low variance. Thus having a N high enough is required to get a useful similarity matrix and we need to find a better metric but how to choose it ?

We have just seen that G monitors the variability of the matrix (variability parameter), thus a similarity matrix with a very high G should be particularly stable across different sets of genes. We then compared modestly parametrized similarity matrices to study how hyperparameters G and N influence the difference between a lower precision matrices to a high precision matrix on average (see Methods 2.3 for the computational details). This new metric should penalize having a low N leading to similarity matrices close to the identity matrix and may yield more satisfying results.

As shown in Figure 2b, while increasing the number of genes still seems to approximate better high precision matrix, this time, the number of probes is also very important. Indeed, for each value of N, the performance saturates from some G value making less and less improvement when G increases. Thus, this figure gives an optimal G for a given N in order to approximate the high precision matrix efficiently with a low cost. The total cost of the algorithm in tokens being proportional to $G \times N$, we found a good tradeoff between variance and precision around $G = 128$ and $N = 32$.

The estimated cost to run the algorithm per model is therefore 128 genes $\times$ 32 probes $= 4096$ queries of ≈ 20 tokens. As a point of reference, conducting the MMLU benchmark requires around 14,000 queries on significantly longer prompts (≈ 70 tokens each), making PhyloLM approximately 10 times less expensive in terms of the number of tokens required.

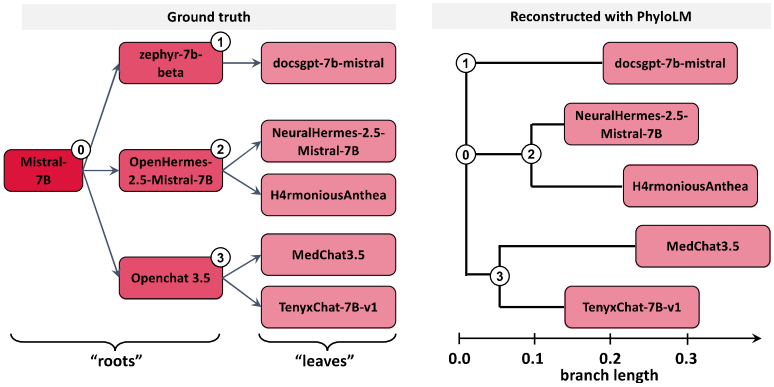


Figure 3: **Phylogenetic tree reconstruction**. On the left it is shown the ground truth concerning the relation of some LLMs of the Mistral family. Right is the reconstruction from the phylogenetic algorithm on the 'math' set of genes for the five latest models of this family ("leaves" of the phylogenetic tree) on which we run PhyloLM. On the right, it is shown the reconstructed phylogenetic tree PhyloLM on the 5 "leafs" models. The numerical labels (0:3) map the true common ancestors (on the right, "ground truth") to the inferred ones (on the left, "reconstructed"). It can be seen that the true and the reconstructed trees are topologically equivalent

3.2 Can we trace back the genealogy of LLMs using tools from genetics?

We first examine the results of PhyloLM by analyzing the resulting phylogenic trees (materialized as dendrograms). However, before dwelling into the results, an important point to understand is that, in genetics, branches in the tree show probable speciation events that occured in the past, when from an extinct common ancestor, two (or more) current species (leaves of the tree) emerged. When looking at LLMs, 'common ancestors' are not extinct, but rather among the studied 'populations'. Take for instance Mistral 7B that is the common ancestor of OpenChat3.5 and Zephyr 7B Alpha, but still included in our analysis. Oblivious of this difference, the dendrogram plotting method will put all models at the 'leaves' of the tree, while, in fact, some of them (such as Mistral 7B) should be at a speciation node. As such, without additional information about which model is at a node, it is difficult to interpret them in the same way as in genetics. Without this important phylogenetic assumption, one has to bear in mind that what matters (and should be compared with the ground truth) is their relative distance and position when evaluating the dendrograms resulting from the phylogenetic analysis of LLMs. Indeed the distance between two models is represented by the distance from their respective leaves in the dendrogram.

To investigate the capabilities of PhyloLM, let's first start by respecting this assumption by looking at a set of 9 models from the Mistral family whose relationships are known because transparently disclosed by their creators. Out of these 9 models, 5 are leaves in the ground truth dendrogram (Arc53, 2023; mlabonne, 2023; Tenyx, 2024; Ullah, 2024; Vallego, 2024). Running PhyloLM on these 5 models getting the distance matrix between them and finally plotting the NJTree we perfectly get back the ground truth phylogenetic tree (see Fig 3) validating the method. These rooted trees are not necessarily very stable as the NJ algorithm makes an unrooted tree of the evolution but then has to choose the root. In Appendix D we show that, on the code genome, the root has been mistakenly attributed to model 3 while the structure of the tree is right. That is why we prefer to plot unrooted trees in the rest of this paper.

3.2.1 Global dendrogram

LLMs: open-source vs private, completion vs chat Now let's drop the assumption of not having 'common ancestors' in the set of LLMs. The LLMs we are investigating here include 111 open

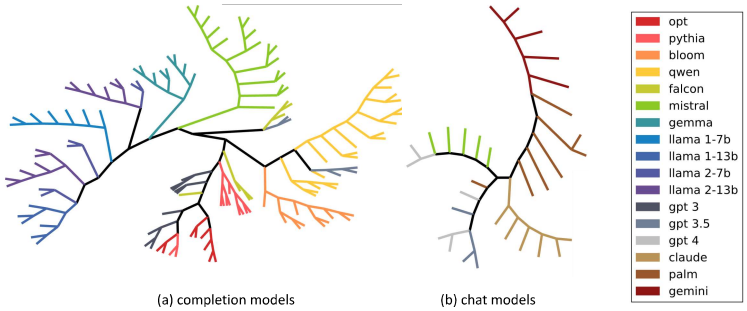


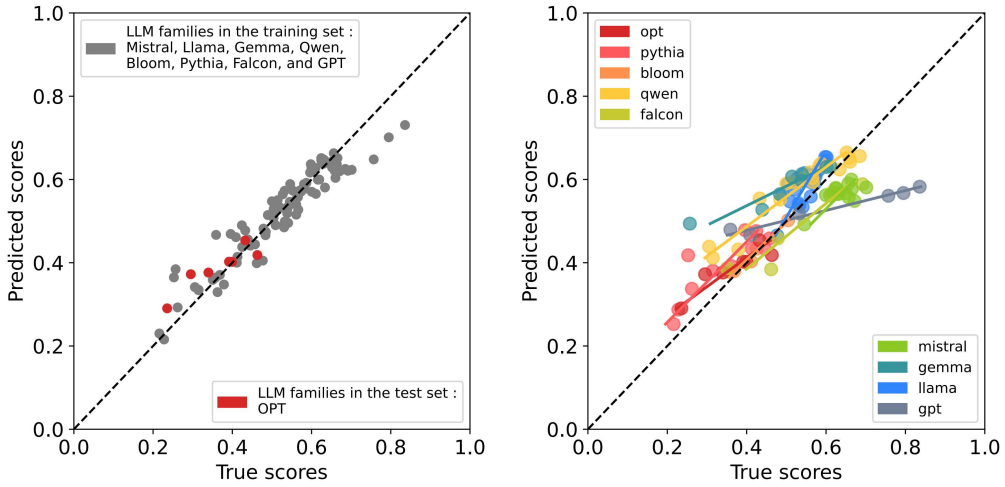
Figure 4: **Inferred phylogenetic tree of LLMs on the 'math' set of genes**. (a) completion models inlcude all open source models included in our study and the 14 openai completion models (b) chat models include additional proprietary models. Completion and chat models were separated because they are not comparable due to additional prompting from the API. Llama models have been split by version of the pretrained model and the number of parameters.

access models spanning from 70M to 176B parameters and 45 closed LLMs. Most modern LLMs are only accessible through a chat API which naturally adds new tokens to the prompt such as chat messages markers biasing the completion of the given 'gene'. This can strongly influence PhyloLM as the algorithm will compare 'alleles' that do not correspond to the same 'gene'. As such we call *completion models* LLMs that were accessed in a way that can generate a completion to a very specific sequence of tokens without adding more tokens. All the 111 open access models we included in this study were accessed in this completion setting but among the 45 proprieraty models we only considered 14 of them to be completion models (see Appendix B for more details). That is why we split the LLMs and investigated them in 2 groups: completion models (to show the capabilities of PhyloLM when run in good conditions) and the others on which we suspect additional prompting manipulation. In both classes of models we found that our algorithm was largely capable of clustering LLMs into their original families, with only a few specificities discussed below. Dendrograms for both model classes are in Figure 4.

In the completion group of models we notice very clear Llama clusters separating the family from other families but also on a more fine grained level, subfamilies of llama linked to the version of the models and their respective sizes. Similarly clear cluster appear for Mistral, Qwen and Bloom. The other families such as Falcon, OPT, Pythia and GPT 3 are more mixed with each other and indeed we know that OPT, Pythia and Falcon-RW-1B (the one the closest to OPT in the tree) were trained each on their own version of the Common Crawl dataset and thus share a similar training set. Lastly, some GPT-3 models (ada, babbage and curie) appear to be close to this OPT,Pythia and Falcon-RW cluster showing they may have been trained on a version of the CommonCrawl as well. On the other hand, GPT-3.5 completion models including text-davinci-002 and text-davinci-003 seem to share more with Falcon than other models while davinci-002, babbage-002 and gpt-3.5-turbo-instruct look more related to Qwen and more precisely its CausalLM finetuning. It is important to understand that dendrograms in LLMs are just a visualisation tool, much more details can be found in the similarity matrix shown in Figure 9 Appendix I shows dendrograms with model names of the models (see Figure 23).

In the chat models group, we also find a lot of structure : Palm and Gemini models are on the same branch, Gemini seems to be a further improvement on Palm as it is further on the branch (and indeed they are both from Google showing maybe a sharing of their training data) while claude has its own branch and Mistral / GPT-3.5 and GPT-4 models show some similarities. Dendrograms with model names are provided in Figure 23 in Appendix I.

Additional figure are available and discussed in Appendix: similarity matrices are in Figure 9 (Appendix E). Code results are in Appendix D, with the dendrogram in Figure 8 and the similarity matrix in Figure 10. Additional mixed class figures are in Appendix G: Figure 18 (math), Figure 19 (code), and global similarity matrices in Figures 16 and 17.



(a) Fitting scatter plot

(b) Predictions for all families

Figure 5: **Predictions from the logistic regression compared to ground truth for every model (leave one family out method) on ARC benchmark**. (a) Scatter plot showing the fitting of the logistic regression on all models but the OPT family (in grey) and the prediction of OPT performance by the regression (in red). (b) Predictions from the logistic regression for each family. To predict a family, the regressor fits on all the other families to finally predict the score of the models from the remaining family (leave one family out method - see (a)).

3.3 Can we infer model capabilities from the genetic distance?

We then investigated whether the genetic distance metric can be used to predict the abilities of language models. As such we used the benchmark scores from the Huggingface open LLM leaderboard. The results indicate that the prediction correlates with the true score of the models (Figures 5 (b) and 15 (a)). Indeed, we found that the Pearson's correlation coefficients (r) of the correlation between the true scores and the predicted ones was positive and significant for all benchmarks and regardless of the set of "genes" used to make the prediction (mean$\pm$sem: 0.68$\pm$0.04; Student's t-test again zero: t(11)=16.0, p<0.001; see Figure 15 (b) in Appendix F). In other terms, within benchmarks and across families, the phylogenetic distance metric allowed us to predict on average 48.2$\pm$0.03% of the variance of the between-model benchmark performance. In a control analysis, we also verified that significant correlation was also achieved within families, thus eliminating the possibility that significant prediction in the previous analysis was driven by our metric simply capturing the fact that different families have different levels of performance on average. To do so, we calculated the Pearson correlation between the true and the predicted scores per benchmark and within each family separately. The results indicate that, even though for some combinations of families and benchmarks, we obtained small or negative correlation coefficients (which is unsurprising, since these correlations were sometimes calculated across very few data points), also in this case, the results were in average positive and significant difference from zero (0.64$\pm$0.05; t(107)= 20.7, p<0.001; see Figure 15 (c) in Appendix F). Within families, the variance explained by our method amounted to 52.2$\pm$0.03% on average, thus indicating that our metric achieved good predictive power even when drastically increasing the level of granularity. Individual plots for each benchmark are shown in Appendix F

4 Discussion

Here we show that an algorithm, inspired by those used in phylogeny, is successful in reconstructing important aspects of the genesis of LLMs, based solely on their outputs to diverse short queries. By leveraging the genetic distance matrix, it becomes feasible to robustly trace the relationships and evolution of models over time. This is particularly evident in the constructed dendrograms, where clear clusters align with distinct families of LLMs, offering a visual representation of their evolutionary trajectories or at least their training similarity. It is important to also emphasize the

applicability of these methods to proprietary models. Understanding the fine-tuning relationships and performance characteristics of private models is often challenging due to limited access to training details and data. PhyloLM offers a valuable tool for gaining insights into these aspects, by providing to the research community a more transparent image of how proprietary models evolve.

We also show that the utility of the "genetic" distance, derived from our algorithm, was not limited to capturing the training relationships, but could be used to infer the performances of models on various benchmarks. The observation that a logistic regression trained on the genetic distance matrix can accurately predict benchmark accuracy has the potential to accelerate the evaluation of new LLMs capabilities in a very computationally efficient manner. Overall, our method provides a robust and insightful analysis of the history, relationships, and performance of Large Language Models, even in cases where detailed training information is not publicly available.

Despite these promising results, it is important to acknowledge the inherent limitations of applying the genetic metaphor to LLMs. Phylogenetic algorithms, traditionally designed for biological analysis where common ancestors are not included among the tested species, face challenges when applied to LLMs, where common ancestors are present among the studied models. Furthermore, chat interfaces complicate the acquisition of reliable genetic material. Nonetheless, this work lays the foundation for further studies aimed at refining these algorithms to better fit the LLM framework and chat models. Our study did not explore the effect of temperature, and while our results were consistent across two sets of genes (and more in Appendix J), examining an even broader range of genes could provide additional insights. Additionally, while the predictive results for benchmark scores are promising (roughly 50% of the variance explained) and could be practically applied to estimate the capabilities of new models, it remains room for improvement (a possible venue being using multiple sets of genes in the evaluation).

Lastly, similarity matrices serve as versatile tools with numerous applications in the study and optimization of large language models (LLMs). For instance, in our investigation of model quantization, we discovered that as the size of the model increases, the quantized version more closely approximates the original model (see Appendix H). Additional fields in which PhyloLM could provide very good insights could also include model merging (Goddard et al., 2024) and scaling laws but we leave it for further research.

5 Acknowledgments

This work was granted access to the HPC/IA resources of [IDRIS HPE Jean Zay A100] under the allocation 2023- [AD011013693R1] made by GENCI. SP is supported by the European Research Council under the European Union's Horizon 2020 research and innovation program (ERC) (RaReMem: 101043804), the Agence National de la Recherche (CogFinAgent: ANR-21-CE23-0002-02; RELATIVE: ANR-21-CE37-0008- 01; RANGE: ANR-21-CE28-0024-01), the Alexander Von Humboldt foundation and a Google unrestricted gift. PYO is supported by ANR AI individual chair ANR-19-CHIA-0004.

References

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, Daniele Mazzotta, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. The falcon series of open language models, 2023.

Arc53. Arc53/docsgpt-7b-mistral, December 2023. URL `https://huggingface.co/Arc53/docsgpt-7b-mistral`.

Ben Athiwaratkun et al. Multi-lingual evaluation of code generation models, 2023.

Quentin D Atkinson, Andrew Meade, Chris Venditti, Simon J Greenhill, and Mark Pagel. Languages evolve in punctuational bursts. *Science*, 319(5863):588–588, 2008.

Jinze Bai et al. Qwen technical report, 2023.

Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. Pythia: A suite for analyzing large language models across training and scaling, 2023.

BigScienceWorkshop et al. Bloom: A 176b-parameter open-access multilingual language model, 2023.

Tom B. Brown et al. Language models are few-shot learners, 2020.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, Wei Ye, Yue Zhang, Yi Chang, Philip S. Yu, Qiang Yang, and Xing Xie. A survey on evaluation of large language models, 2023.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023. URL https://lmsys.org/blog/2023-03-30-vicuna/.

François Chollet. On the measure of intelligence, 2019.

Aakanksha Chowdhery et al. Palm: Scaling language modeling with pathways, 2022.

R Dawkins. *The Selfish Gene*. Oxford University Press, Oxford, UK, 1976.

Chunyuan Deng, Yilun Zhao, Xiangru Tang, Mark Gerstein, and Arman Cohan. Investigating data contamination in modern benchmarks for large language models, 2023.

Julien d'Huy. Polyphemus (aa. th. 1137): A phylogenetic reconstruction of a prehistoric tale. *Nouvelle Mythologie Comparée/New Comparative Mythology*, 1(1):http–nouvellemythologiecomparee, 2013.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. Gptq: Accurate post-training quantization for generative pre-trained transformers, 2023.

Charles Goddard, Shamane Siriwardhana, Malikeh Ehghaghi, Luke Meyers, Vlad Karpukhin, Brian Benedict, Mark McQuade, and Jacob Solawetz. Arcee's mergekit: A toolkit for merging large language models. *arXiv preprint arXiv:2403.13257*, 2024.

Russell D Gray, David Bryant, and Simon J Greenhill. On the shape and fabric of human history. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 365(1559):3923–3933, 2010.

Niklaus Grünwald, Sydney Everhart, Brian Knaus, and Zhian Kamvar. Best practices for population genetic analyses. *Phytopathology*, 107, 05 2017. doi: 10.1094/PHYTO-12-16-0425-RVW.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding, 2021.

HuggingFaceH4. open llm leaderboard. URL https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023.

Percy Liang et al. Holistic evaluation of language models, 2023.

Q. Vera Liao and Jennifer Wortman Vaughan. Ai transparency in the age of llms: A human-centered research roadmap, 2023.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. Awq: Activation-aware weight quantization for llm compression and acceleration, 2024.

mlabonne. mlabonne/neuralhermes-2.5-mistral-7b, November 2023. URL https://huggingface.co/mlabonne/NeuralHermes-2.5-Mistral-7B.

OpenAI et al. Gpt-4 technical report, 2023.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022.

Keiran Paster, Marco Dos Santos, Zhangir Azerbayev, and Jimmy Ba. Openwebmath: An open dataset of high-quality mathematical web text, 2023.

N Saitou and M Nei. The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Molecular Biology and Evolution*, 4(4):406–425, 07 1987. ISSN 0737-4038. doi: 10.1093/oxfordjournals.molbev.a040454. URL https://doi.org/10.1093/oxfordjournals.molbev.a040454.

Aarohi Srivastava et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models, 2023.

Naoko Takezaki and Masatoshi Nei. Genetic Distances and Reconstruction of Phylogenetic Trees From Microsatellite DNA. *Genetics*, 144(1):389–399, 09 1996. ISSN 1943-2631. doi: 10.1093/genetics/144.1.389. URL https://doi.org/10.1093/genetics/144.1.389.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca, 2023.

Gemini Team et al. Gemini: A family of highly capable multimodal models, 2024a.

Gemma Team et al. Gemma: Open models based on gemini research and technology, 2024b.

Jamie Tehrani and Julien d'Huy. *2017. Phylogenetics Meets Folklore: Bioinformatics Approaches to the Study of International Folktales*, pp. 91–114. 01 2017. ISBN 978-3-319-39443-5. doi: 10.1007/978-3-319-39445-9_6.

Jamshid J Tehrani and Mark Collard. On the relationship between interindividual cultural transmission and population-level cultural diversity: a case study of weaving in iranian tribal populations. *Evolution and Human Behavior*, 30(4):286–300, 2009.

Tenyx. Tenyxchat: Language model alignment using tenyx fine-tuning, January 2024. URL https://huggingface.co/tenyx/TenyxChat-7B-v1.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023a.

Hugo Touvron et al. Llama 2: Open foundation and fine-tuned chat models, 2023b.

Imran Ullah. Imran1/medchat3.5, January 2024. URL https://huggingface.co/Imran1/MedChat3.5.

Jorge Vallego. neovalle/h4rmoniousanthea, January 2024. URL https://huggingface.co/neovalle/H4rmoniousAnthea.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. Opt: Open pre-trained transformer language models, 2022.

Shen Zheng, Yuyu Zhang, Yijie Zhu, Chenguang Xi, Pengyang Gao, Xun Zhou, and Kevin Chen-Chuan Chang. GPT-Fathom: Benchmarking large language models to decipher the evolutionary path towards GPT-4 and beyond, 2023. URL https://arxiv.org/abs/2309.16583.