Estudio y Aplicación del Modelo HyenaDNA en la Detección de Ancestría Genética

Gonzalo Cameto

Maestría en Ingeniería Matemática

Directores: María Inés Fariello, Federico Lecumberry

Director Académico: Marcelo Fiori

Facultad de Ingeniería — Universidad de la República

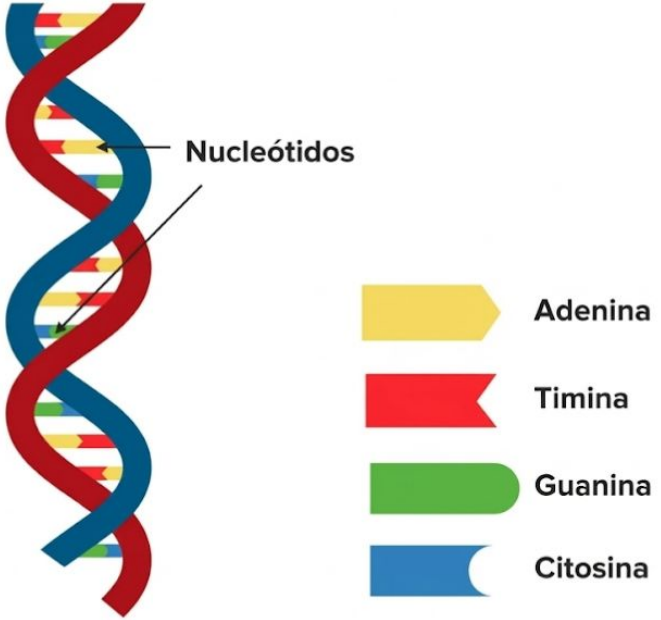
Workshop LLMs 2025

ADN: El Texto Biológico más Largo

El desafío genómico

- 3 mil millones de nucleótidos
- Contexto extremadamente largo
- Dependencias distantes (+100k nucleótidos)
- Vocabulario mínimo: {A, C, G, T}

```
...GGGGCGATAGAGTGAGACTCCCTCTCTGCCGGGCGCAGTGGCTCAGG
CCTGTAATCCCAGCACTTTGGGAGGCCAAGGCGGGCGGATCACAAGGTCA
TGAGATCGAGACCATCCTGGCTAACACGGTGAAACCCCGTCTCTAATAAA
AATATAAAAAGTTAGCCGGGCGTGTTGGCGGGCGCCTGTAGTCCCAGCTA
CTCGGGAGGCTGAGGCAGGAGAATGGCGTGAACCCCGGAGGCGGAGC...
```

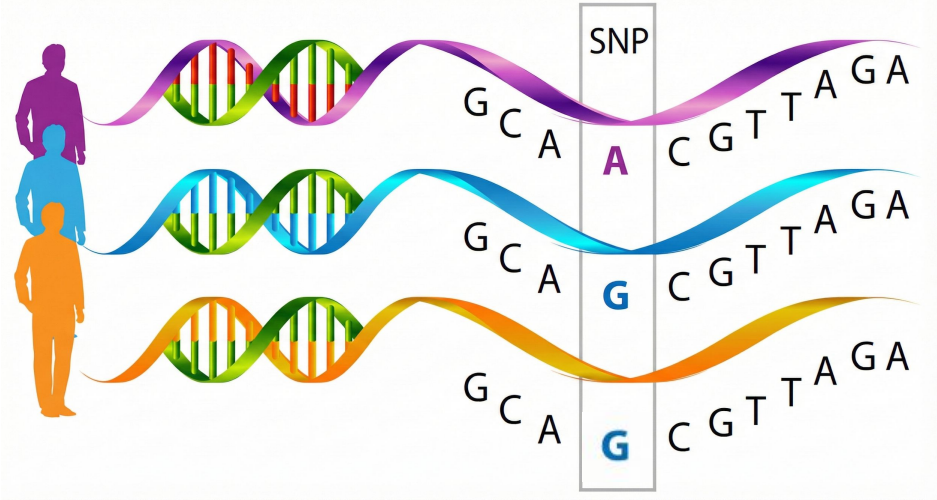


SNPs: La Señal Dispersa

SNP (Single Nucleotide Polymorphism)
- Variación en un solo nucleótido
- Frecuencia: ~1 cada 700 nucleótidos
- Solo 0.15% del genoma
- ~4.5 millones de SNPs por persona

Por qué importan
- Marcadores de diferenciación poblacional
- Señal de ancestría concentrada
- El resto del genoma es casi idéntico

Tarea: Detección de Ancestría Continental



- África (AFR)
- Asia Oriental (EAS)
- Europa (EUR)

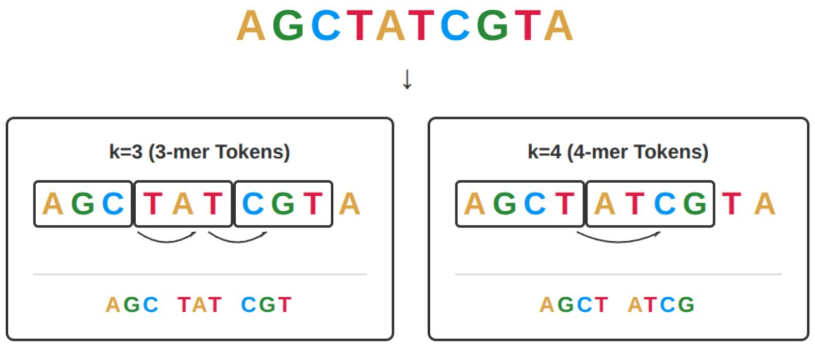
El Problema: Complejidad Cuadrática

Mecanismo de Atención

- Cada posición interactúa con todas las demás
- Matriz de atención: $L \times L$
- Complejidad: $O(L^2)$ en tiempo y memoria

Fórmula de Atención

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$



$4^3 = 64$ tokens

$4^4 = 256$ tokens

DNABERT (Ji et al., 2021)

- Limitado a ~4.000 tokens
- Usa k-mers (solapamiento de k bases)

La Pregunta de Investigación

¿Existen operadores subcuadráticos que puedan igualar la calidad de la atención a escala?

Hyena Hierarchy (Poli et al., 2023):

"Un operador definido por una recurrencia de dos primitivas subcuadráticas eficientes:"

1. Convolución larga
2. Compuerta multiplicativa elemento a elemento

Hyena Hierarchy:
Towards Larger Convolutional Language Models

Michael Poli[*,1], Stefano Massaroli[*,2], Eric Nguyen[1,*],
Daniel Y. Fu[1], Tri Dao[1], Stephen Baccus[1],
Yoshua Bengio[2], Stefano Ermon[1,†], Christopher Ré[1,†]

Version: submitted draft, Last Compiled: April 21, 2023

Abstract

Recent advances in deep learning have relied heavily on the use of large Transformers due to their ability to learn at scale. However, the core building block of Transformers, the attention operator, exhibits quadratic cost in sequence length, limiting the amount of context accessible. Existing subquadratic methods based on low-rank and sparse approximations need to be combined with dense attention layers to match Transformers, indicating a gap in capability. In this work, we propose **Hyena**, a subquadratic drop-in replacement for attention constructed by interleaving implicitly parametrized **long convolutions** and **data-controlled gating**. In recall and reasoning tasks on sequences of thousands to hundreds of thousands of tokens, Hyena improves accuracy by more than 50 points over operators relying on state-spaces and other implicit and explicit methods, matching attention-based models. We set a new state-of-the-art for dense-attention-free architectures on language modeling in standard datasets (WikiText103 and The Pile), reaching Transformer quality with a 20% reduction in training compute required at sequence length 2K. Hyena operators are twice as fast as highly optimized attention at sequence length 8K, and 100× faster at sequence length 64K.

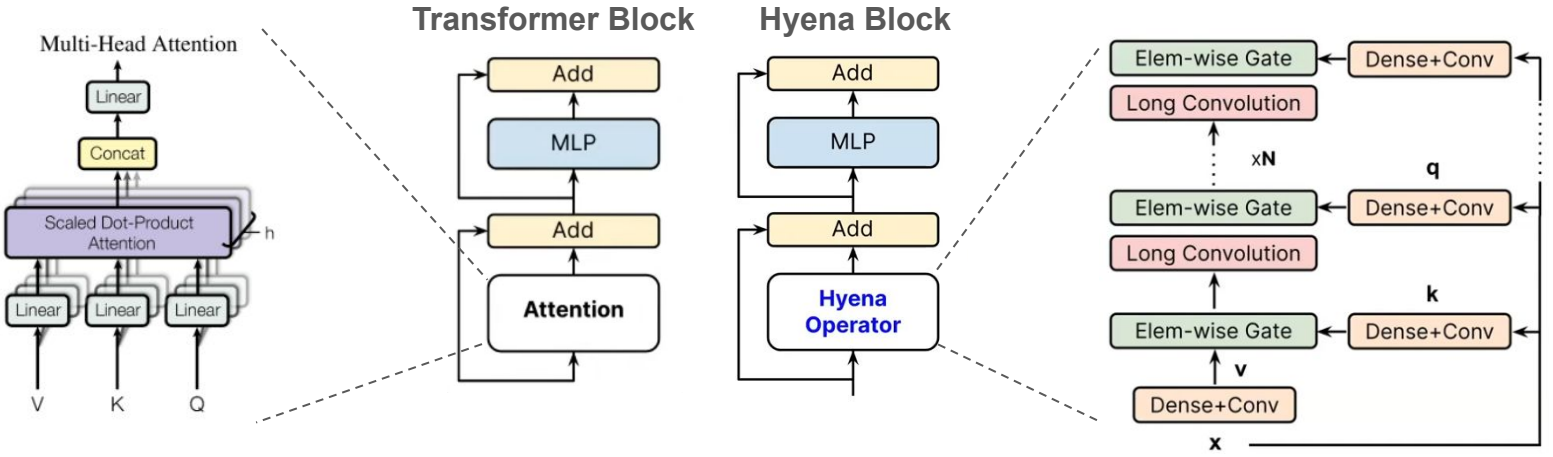
De Atención a Convoluciones + Compuertas

Atención

- Cada posición interactúa con todas
- Expresivo pero costoso: $O(L^2)$

Convolución + Compuertas

- Interacción mediante filtros aprendidos
- Compuertas controlan flujo de información
- Eficiente: $O(L \log L)$



Convoluciones vía Transformada Rápida de Fourier (FFT) + Filtros Implícitos

Teorema de Convolución

$$\mathcal{F}\{x * h\} = \mathcal{F}\{x\} \odot \mathcal{F}\{h\}$$

Algoritmo eficiente → O(L log L)

$$y = \mathrm{IFFT}(\mathrm{FFT}(x) \odot \mathrm{FFT}(h))$$

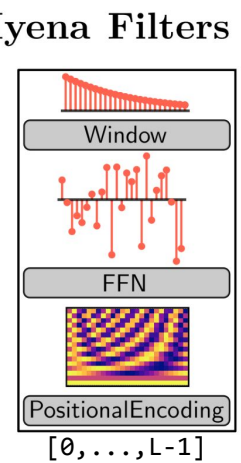
- FFT: *O(L log L)* vs *O(L²)* directo
- Permite filtros de largo L

Filtros Implícitos

Parametrizar con red neuronal pequeña:

$$h_t = \mathrm{Window}(t) \cdot (\mathrm{FFN} \circ \mathrm{PosEnc})(t)$$

- M parámetros → filtros de cualquier L
- M << L
- Aprende forma óptima del filtro



Compuertas Multiplicativas

Matrices Diagonales D_x

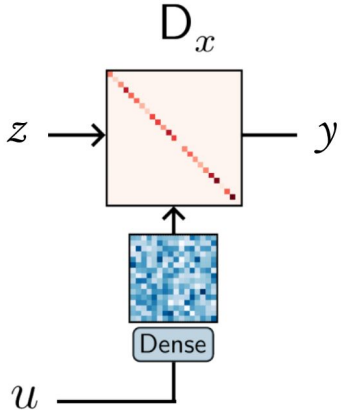
- Selección dinámica de información
- Valores dependen de la entrada
- Modulan amplitud por posición

Operación

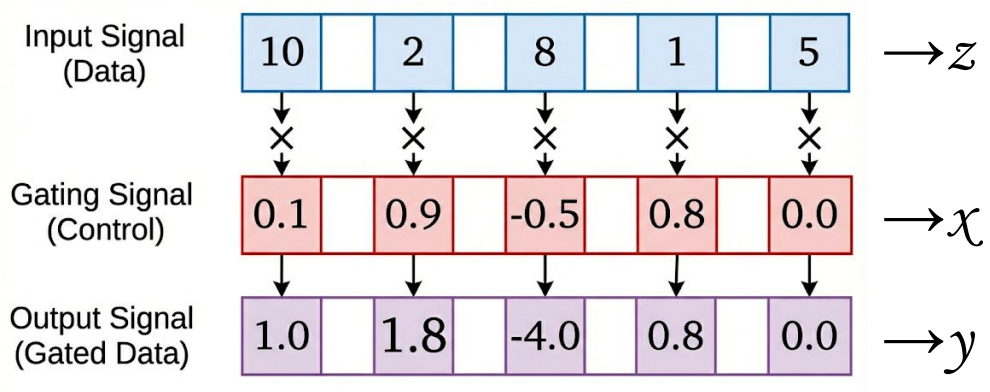
- $y = D_x z$ (elemento a elemento)
- $D_x = \text{diag}(x_1, x_2, ..., x_n)$

Capacidades

- Atenuar ($|x_i| < 1$)
- Amplificar ($|x_i| > 1$)
- Invertir polaridad ($x_i < 0$)



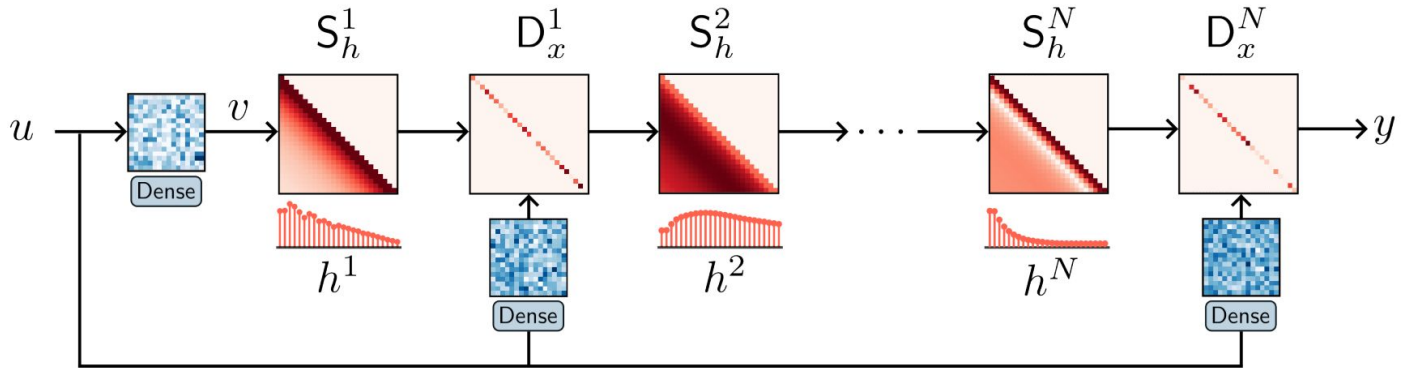
Control contextual del flujo de información



El Operador Hyena

Combina ambas primitivas:

- S□: Matrices Toeplitz (convoluciones largas)
- D_x: Matrices diagonales (gating)



Formulación:

$$\mathbf{y} = \mathrm{H}(\mathbf{u})\mathbf{v} = \mathrm{D}_x^N \mathrm{S}_h^N \cdots \mathrm{D}_x^2 \mathrm{S}_h^2 \mathrm{D}_x^1 \mathrm{S}_h^1 \mathbf{v}$$

- Proyecciones de entrada u ⟶ v, x_1, x_2, …, x_N
- Alternancia: conv ⟶ gate ⟶ conv ⟶ gate
- Orden típico: N = 2
- Complejidad total: *O(L log L)*

HyenaDNA: Modelo Hyena aplicado a ADN

Paper principal de la tesis (Nguyen et al., 2023)

- Pre-entrenados en un genoma humano completo
- Next Token Prediction
- SotA en múltiples benchmarks

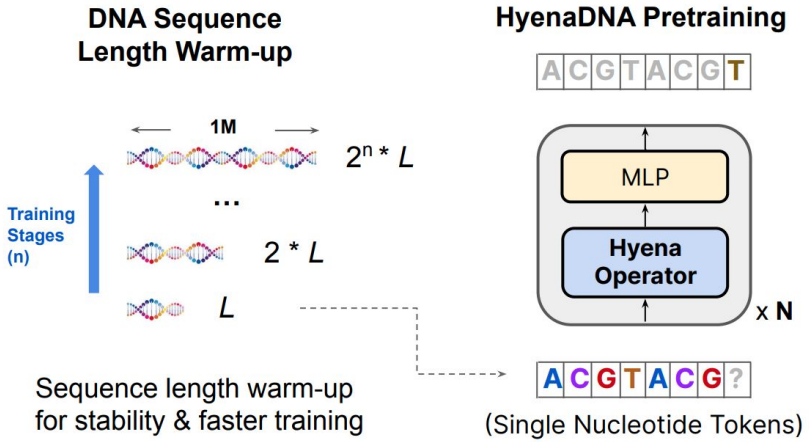
Contexto máximo probado

- HyenaDNA soporta hasta 1M tokens (A100-80)
- Útil para: predicción genómica, variantes

Sequence Warm-up

- Para modelos largos (+200k tokens)
- Entrenamiento progresivo de contexto
- Mayor estabilidad en convergencia

Modelo	Contexto	Layers
tiny	1k	2
small	32k	4
medium	160k	8
large	450k	8
xlarge	1m	8



Pipeline de Datos

Dataset: 1000 Genomes Project

- 590 individuos totales:
  - Europa: 240 (UK, Italia, España, Finlandia)
  - África: 160 (Gambia, Kenia, Sierra Leona)
  - Asia: 190 (China, Japón, Vietnam)

- ~4.5 millones SNPs por individuo
- 22 cromosomas autosómicos

Almacenamiento HDF5

- Estructura jerárquica: individuo→cromosoma
- ~12 MB por archivo (individuo)
- Total: ~7 GB de datos procesados

Ventajas de HDF5

- Acceso aleatorio eficiente
- Compresión transparente
- Carga parcial por cromosoma
- Compatible con PyTorch DataLoader

Muestreo de entrenamiento

1. Sorteo aleatorio de población
2. Sorteo de individuo
3. Sorteo de cromosoma
4. Sorteo de posición inicial
5. Extracción de ventana de SNPs y posiciones

Probando HyenaDNA De Especies a Ancestría Humana

Progresión de experimentos

#	Experimento	Accuracy	Insight
1	Clasificación de especies (5 mamíferos)	92%	✓ Funciona (diferencias grandes)
2	Ancestría con secuencia completa	~ 33%	✗ Random (señal diluida en 99.85%)
3	Ancestría solo con SNPs	< 50%	✗ Sin contexto posicional
4	Ancestría con SNPs + posición	93 - 99%	✓ Contexto posicional es crítico

¿Qué aprendimos?

- Diversidad inter-especie >> intra-humana
- Secuencia completa: señal diluida (0.15% SNPs)
  - ~1.5 SNPs en secuencias de 1k
  - ~49 SNPs en secuencias de 32k
- Solo SNPs: Mejora, pero falta contexto posicional
- SNPs + posición: información completa

Hallazgo principal

La combinación de:

1. Filtrar SNPs (concentrar señal)
2. Agregar posición (contexto)

Motivación: evaluar codificaciones posicionales

Evaluación de Codificaciones Posicionales

El problema

- Cromosoma más largo: ~250M nucleótidos
- SNPs heredan posiciones cromosómicas
- Positional Encoding (PE) no dio buenos resultados
- Positional Embedding: 250M × 128 ⟶ ~60 GB

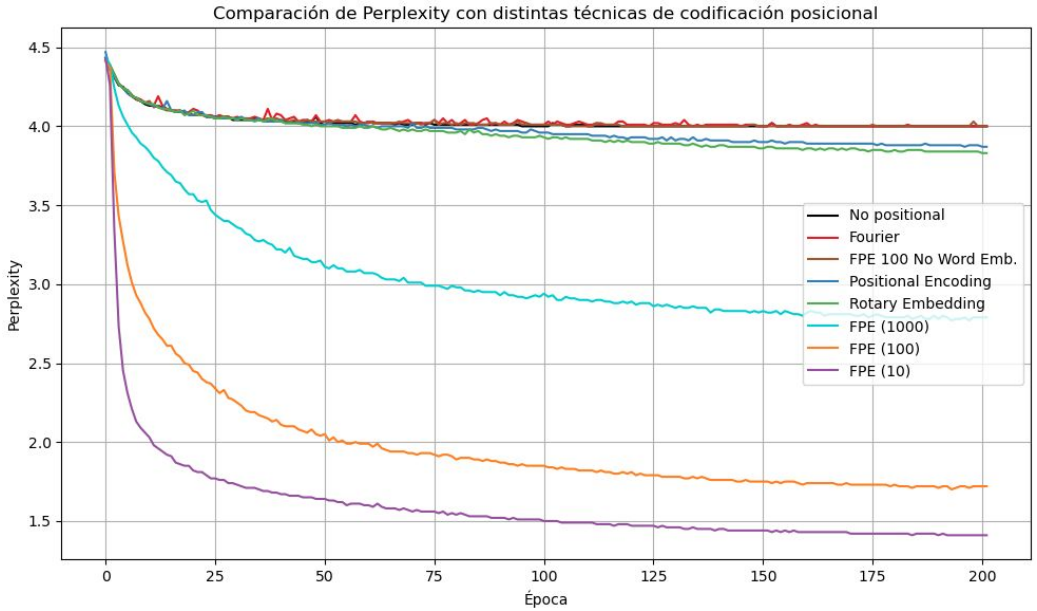
Técnicas evaluadas

- PE sinusoidal (Vaswani, 2017)
- Rotary Positional Embeddings (Su, 2021)
- Embeddings de Fourier
- Positional Embedding
- FPE: factorización de bajo rango

Resultado clave

- PE, RoPE, Fourier: ~3.8-4.0 (marginales)
- FPE: 1.41 - 2.79 (significativas)

⟶ No paramétricos no capturan posiciones genómicas



FPE: Trade-off Memoria vs Precisión

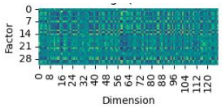
Factorized Positional Embeddings (FPE)

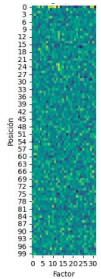
- $E = A \times B$ (factorización de bajo rango)
- $E \in \mathbb{R}^{P \times d}$, $A \in \mathbb{R}^{P \times k}$, $B \in \mathbb{R}^{k \times d}$
- k << d (típicamente k = 16-32 y d = 128)

Factor de escala α

- $P_{efectivo} = P_{cromosoma} / \alpha$

$$E = A \times B, \quad A \in \mathbb{R}^{P_{eff} \times k}, \quad B \in \mathbb{R}^{k \times d}$$





Trade-off α: Memoria vs Perplejidad

α	P_eff	Memoria	Perpl.
1	250M	59.6 GiB	—
10	25M	5.96 GiB	1.41
100	2.5M	0.6 GiB	1.72
1000	250k	0.04 GiB	2.79

→ α = 100 es el balance óptimo:

- Memoria manejable (0.6 GiB)
- Perplejidad competitiva (1.72)
- Permite contextos largos
- Memoria disponible para batch más grandes

Entrenamiento en Dos Etapas

Problema

- Pre-training en nucleótidos completos NO transfiere a secuencias de SNPs
- Entrenamiento desde cero para clasificación no muestra buenos resultados

Solución: entrenamiento específico

Etapa 1: Next-SNP Prediction

- Sin etiquetas de ancestría
- Entrenamiento autosupervisado
- Aprende dependencias entre SNPs
- Modelo de lenguaje genómico

Etapa 2: Fine-tuning Clasificación

- Inicializa con pesos de Etapa 1
- Agrega MLP clasificador
- Entrenamiento supervisado

Resultado:

Representaciones específicas para SNPs antes de la tarea supervisada

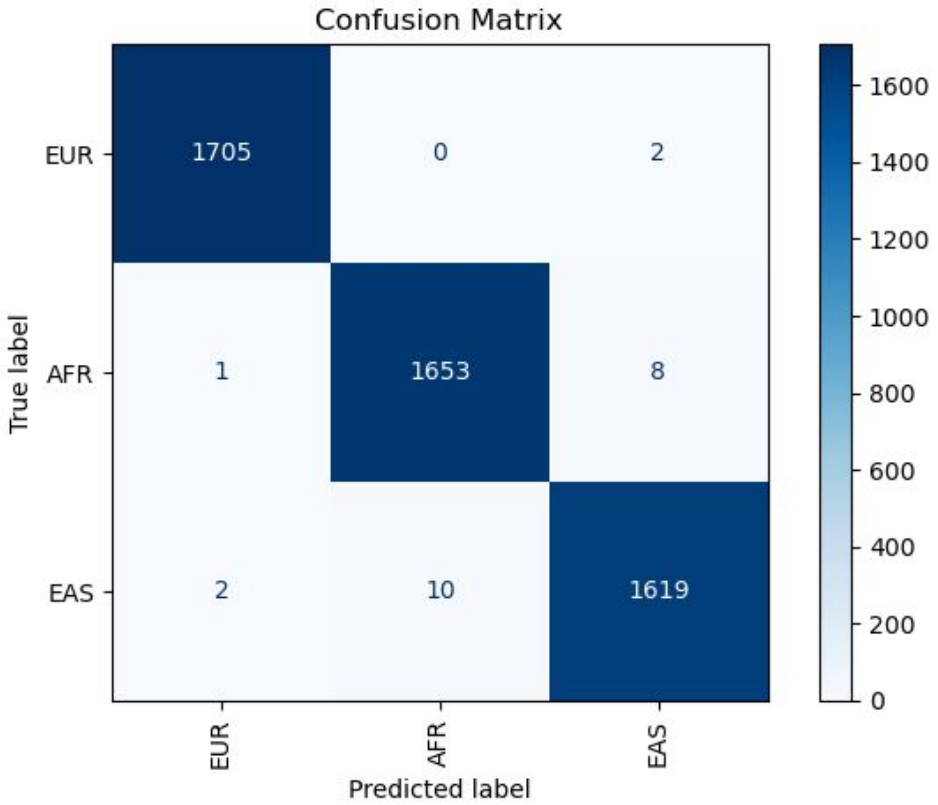
Accuracy vs Contexto y Factor de Escala

Accuracy de clasificación

SNPs	α=10	α=100
128	83%	77%
512	—	93%
1024	—	97.04%
2048	—	97.83%
4096	—	99.34%

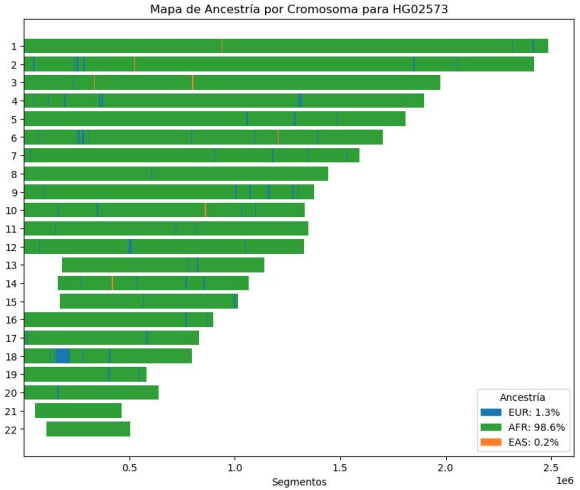
Observaciones:

- Más contexto → mejor accuracy
- α=10: mejor en contextos cortos (128)
- α=100: escala mejor a contextos largos
- Accuracy final: 99.34% (4096 SNPs)

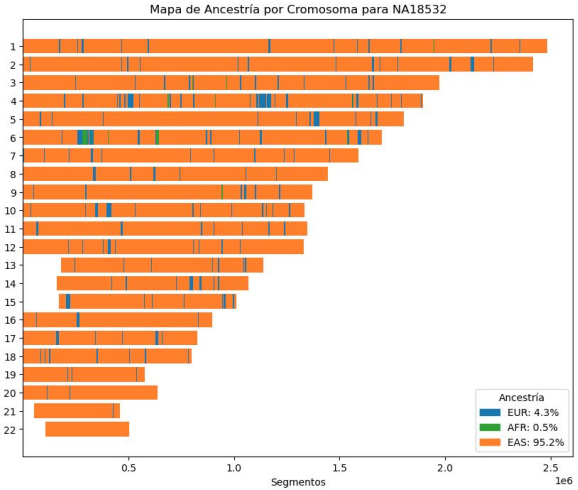


Clasificación de Ancestría: Visualización Geográfica

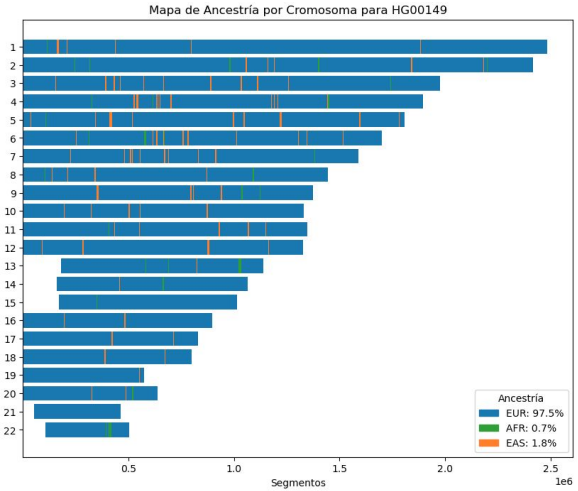
Inferencia en 3 individuos de ejemplo



África (AFR)



Asia Oriental (EAS)



Europa (EUR)

- Contexto: 1024, Step: 512
- Los errores se pueden suavizar con las predicciones locales

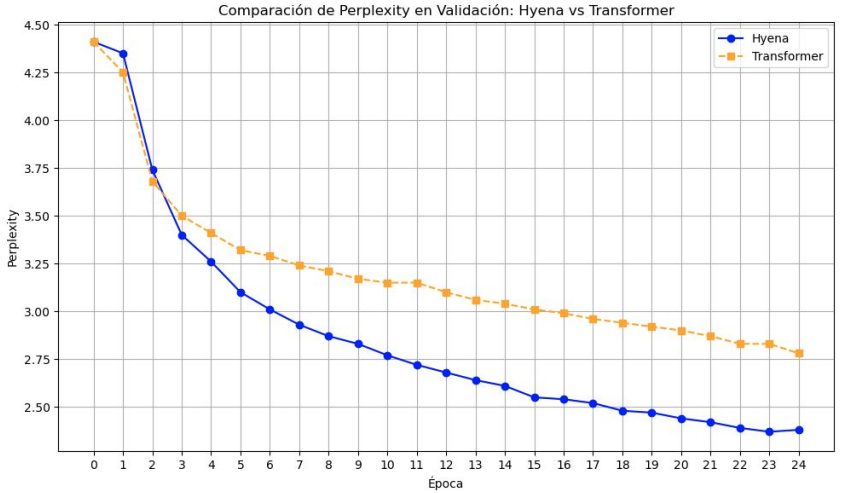
HyenaDNA vs Transformer

Configuración controlada

- ~41M parámetros en ambos
- Mismo dataset, 1024 tokens
- GPU: RTX 4060 Ti (16GB)

Métrica	Transformer	HyenaDNA	Mejora
Memoria GPU	14.8 GB	3.9 GB	3.75×
Tiempo/época	1:40	0:39	2.56×
Perplejidad	2.78	2.38	mejor
Accuracy	86%	90%	+4%

→ Hyena: más eficiente y mejor rendimiento



Visualización de Embeddings

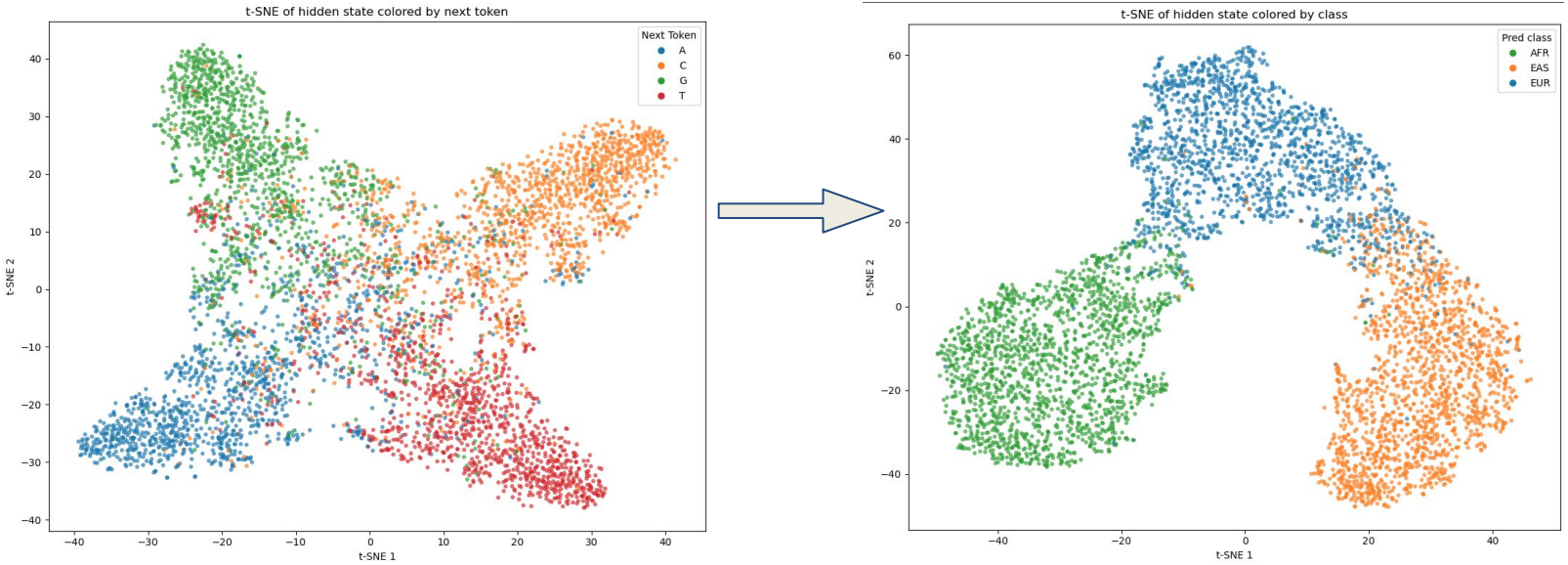
Antes: Next Token Prediction

- t-SNE de hidden states
- Coloreados por nucleótido objetivo
- Estructura en estrella (4 clusters)

Después: Fine-tuning Clasificación

- Coloreados por ancestría
- 3 clusters bien separados
- AFR (verde), EAS (naranja), EUR (azul)

Transformación del espacio de embeddings luego del fine-tuning



Takeaways

1. *O(L log L)* vs *O(L²)*
   - 3.75× menor memoria, 2.56× más rápido en secuencias de 1024

2. Codificación posicional domina el modelo
   - FPE: 96.3% de los parámetros (39.8M de 41M)

3. Pre-training específico sobre SNPs es necesario
   - Next-SNP Prediction + Fine-tuning = 99% accuracy

4. Pipeline de datos escalable
   - HDF5 + muestreo estratificado → 590 individuos, 7GB

¿Preguntas?