Universidad Politécnica
de Madrid

Escuela Técnica Superior de Ingenieros Informáticos

Máster Universitario en Inteligencia Artificial

Trabajo Fin de Máster

Benchmarking de Modelos Fundacionales para la Anotación Celular en Datos scRNA-seq Pan-cancer

Autor(a): José Manuel Lamas Pérez
Tutor(a): Fátima Al-Shahrour Núñez (CNIO)
Tutor(a): Alfonso Rodríguez-Patón Aradas (UPM)

Madrid, Noviembre, 2024

Este Trabajo Fin de Máster se ha depositado en la ETSI Informáticos de la Universidad Politécnica de Madrid para su defensa.

Trabajo Fin de Máster
Máster Universitario en Inteligencia Artificial

Título: Benchmarking de Modelos Fundacionales para la Anotación Celular en Datos scRNA-seq Pan-cancer

Noviembre, 2024

Autor(a): José Manuel Lamas Pérez
Tutor(a): Fátima Al-Shahrour Núñez (CNIO)
Tutor(a): Alfonso Rodríguez-Patón Aradas (UPM)
Departamento de Inteligencia Artificial
ETSI Informáticos
Universidad Politécnica de Madrid

Resumen

El cáncer es una enfermedad caracterizada por una heterogeneidad compleja, donde las células tumorales pueden variar en su morfología, función y respuesta a tratamientos. Esta variabilidad inter e intratumoral hace que el análisis profundo de las células dentro de un tumor sea crucial para entender mejor la progresión de la enfermedad y las posibles terapias. El análisis single-cell, que permite estudiar células individuales en lugar de promediar la información de una población, se ha convertido en una herramienta fundamental para abordar esta heterogeneidad a nivel celular. La anotación celular es un paso crítico en el análisis single-cell, permitiendo a los investigadores identificar y caracterizar tipos de células individuales dentro de tejidos complejos. Este proceso es esencial para comprender la heterogeneidad celular, la biología del desarrollo y los mecanismos de la enfermedad a nivel granular. La anotación celular tradicional depende en gran medida del análisis manual y el conocimiento experto, involucrando la interpretación de patrones de expresión génica y el uso de conjuntos de datos de referencia. Este proceso manual es tanto lento como laborioso, a menudo requiriendo un esfuerzo humano considerable para anotar conjuntos de datos grandes. Modelos fundacionales como scBERT y scGPT ofrecen un enfoque distinto para la anotación celular. Estos modelos son modelos de aprendizaje automático a gran escala pre-entrenados en conjuntos de datos extensos y diseñados para ser adaptables a varias tareas posteriores mediante fine-tuning. En este estudio, se utilizan scBERT y scGPT como modelos fundacionales para la anotación celular en el análisis single-cell. Primero se hace fine-tuning en estos modelos usando el Atlas Terapéutico de Células de Cáncer (TCCA), un conjunto de datos pan-cáncer, para evaluar su capacidad de anotar con precisión diversos tipos de células. Subsecuentemente, se vuelve a realizar fine-tuning en exclusivamente células tumorales del TCCA para evaluar su capacidad en clasificar células según el tipo de tumor. Los resultados demuestran que scBERT y scGPT alcanzan alta precisión en ambas tareas de anotación de tipo celular y clasificación de tipo de tumor. Estos hallazgos indican que tales modelos fundacionales son capaces de generar incrustaciones celulares de alta calidad que representan las células, lo cual es beneficioso no solo para la anotación de tipo celular sino también para otras aplicaciones futuras. La capacidad de producir incrustaciones robustas abre posibilidades para análisis posteriores, tales como identificar nuevos estados celulares, comprender trayectorias celulares e integrar datos multi-ómicos. Esto subraya el potencial de scBERT y scGPT para avanzar en la investigación de single-cell proporcionando herramientas versátiles que mejoran nuestra capacidad de interpretar y utilizar datos biológicos complejos.

Abstract

Cancer is a disease characterized by complex heterogeneity, where tumor cells can vary in morphology, function, and response to treatments. This inter and intratumoral variability makes the in-depth analysis of cells within a tumor crucial for better understanding disease progression and potential therapies. Single-cell analysis, which allows the study of individual cells rather than averaging information from a population, has become a fundamental tool to address this heterogeneity at the cellular level. Cell annotation is a critical step in single-cell genomics, enabling researchers to identify and characterize individual cell types within complex tissues. This process is essential for understanding cellular heterogeneity, developmental biology, and disease mechanisms at a granular level. Traditional cell annotation relies heavily on manual analysis and expert knowledge, involving the interpretation of gene expression patterns and the use of reference datasets. This manual process is both time-consuming and labor-intensive, often requiring a considerable human effort to annotate large datasets. Foundational models like scBERT and scGPT offers a transformative approach to cell annotation. These models are large-scale machine learning models pre-trained on extensive datasets and designed to be adaptable to various downstream tasks through fine-tuning. In this study, we leverage scBERT and scGPT as foundational models for cell annotation in single-cell genomics. We first fine-tuned these models using the Therapeutic Cancer Cell Atlas (TCCA), a pan-cancer dataset, to evaluate their ability to accurately annotate diverse cell types. Subsequently, we fine-tuned the models exclusively on tumor cells from the TCCA to assess their capability in classifying cells according to tumor type. Our results demonstrate that scBERT and scGPT achieve high accuracy in both cell type annotation and tumor type classification tasks. These findings indicate that such foundational models are capable of generating high-quality cellular embeddings that effectively represent cells, which is beneficial not only for cell type annotation but also for other future applications. The ability to produce robust embeddings opens up possibilities for downstream analyses, such as identifying novel cell states, understanding cellular trajectories, and integrating multi-omics data. This underscores the potential of scBERT and scGPT to advance single-cell research by providing versatile tools that enhance our capacity to interpret and utilize complex biological data.

Tabla de contenidos

Capítulo 1

Introducción

1.1. El cáncer

Las células son estructuras altamente organizadas en las que sus componentes moleculares principales—como ácidos ribonucleicos, proteínas y lípidos—trabajan en perfecta sincronía para garantizar su correcto funcionamiento y, por extensión, el del organismo en su totalidad. Sin embargo, a lo largo de la vida de un organismo, esta armonía puede verse alterada por diversos factores internos y externos, dando lugar a modificaciones aberrantes. Estas alteraciones pueden provocar que una célula sana se transforme en una célula tumoral, marcando el inicio del desarrollo del cáncer.

El cáncer se da lugar en el momento en el que una célula dañada comienza a replicarse sin control, pudiendo llegar a formar lo que se conoce como tumores malignos [1]. Estos tumores se pueden formar en cualquier órgano del cuerpo, provocando su malfuncionamiento, y, en estados avanzados de la enfermedad, las células que forman el tumor pueden migrar a otras partes del cuerpo para seguir replicándose de forma incontrolable. A este proceso de migración de las células malignas se le denomina metástasis y es la principal causa de muerte por cáncer en el mundo [2].

Después de las enfermedades cardiovasculares, el cáncer es la segunda causa de muerte por enfermedad más común, y una de las enfermedades más diagnosticadas en el mundo. Se estima que en 2020 se diagnosticaron alrededor de 18 millones de casos nuevos de cáncer, y que esa cifra aumentará hasta los 28 millones para 2040 [3]. En la figura 1.1 se puede apreciar cuáles son los tipos de cáncer más comunes del año 2020, siendo estos el de mama, el de pulmón, el colorrectal y el de próstata, que se siguen manteniendo como los más frecuentes a lo largo de los años.

La adquisición de daño por parte de una célula puede deberse a varios factores, que pueden ser tanto internos como externos. Todas las células del cuerpo contienen los mismos genes, sin embargo, cada célula utiliza tan solo los necesarios para llevar a cabo sus funciones. Un gen se puede entender como la secuencia de ácido dexosirribonucleico (ADN) que porta las instrucciones necesarias para sintetizar una proteína, y cada proteína tiene una función específica. El cáncer es una enfermedad genética, esto es, que es provocada por cambios, llamados variantes o mutaciones genéticas, en los genes encargados de controlar la formación y replicación de las células [4]. El efecto de las mutaciones en el funcionamiento de la célula es variable, siendo la mayoría de ellas inocuas, mientras que otras pueden tener efectos menores, y solo un

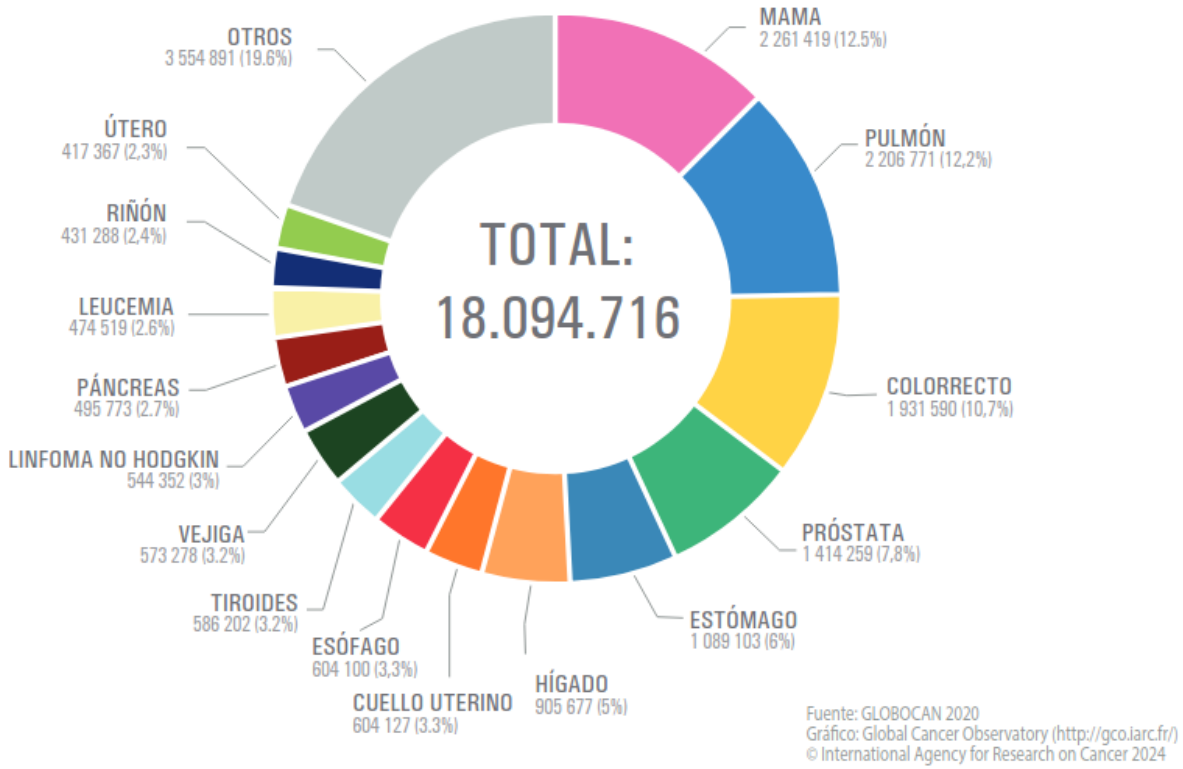

Figura 1.1: Número de diagnósticos por tipo de cáncer en 2020. [3]

pequeño porcentaje de las mutaciones que una célula acumula con el tiempo tienen efectos dañinos. La gran mayoría de los cánceres surgen como resultado de este proceso a lo largo del tiempo. Las mutaciones relacionadas con el cáncer normalmente son provocadas por los siguientes factores [5]:

- **Mutaciones aleatorias:** Errores espontáneos durante la replicación del ADN en la división celular pueden introducir cambios genéticos. Aunque las células cuentan con mecanismos de reparación que los corrigen, algunos pueden persistir y acumularse con el tiempo.
- **Mutaciones hereditarias:** Cambios genéticos que se transmiten de padres a hijos pueden predisponer a ciertos tipos de cáncer. Estos representan una minoría de los casos, pero son importantes en la evaluación del riesgo individual.
- **Mutaciones inducidas por carcinógenos:** La exposición a agentes carcinógenos, como sustancias químicas presentes en el humo del tabaco, radiación ultravioleta (UV) del sol o infecciones virales como el virus del papiloma humano (VPH), puede dañar el ADN y aumentar el riesgo de mutaciones [5].

Una de las mayores dificultades en el tratamiento del cáncer es lidiar con su heterogeneidad intratumoral (1.2). Como resultado de las interacciones y variaciones dentro del propio organismo, cada tumor es único y evoluciona de manera diferente, adquiriendo distintas mutaciones a medida que se desarrolla como resultado de esas interacciones con el ambiente que lo rodea [7]. Por ello, los tumores pueden desarrollar poblaciones de células con diferentes mutaciones a la que originó la enfermedad en primer lugar. De hecho, diversos autores achacan a la heterogeneidad intratumo-

ral el principal problema de la resistencia a fármacos o fallos de terapias en muchos pacientes [7][8]. Debido a esto, clasificar o definir protocolos y terapias para cada tipo de cáncer dependiendo únicamente del órgano de origen no es suficiente; cada tumor necesita ser estudiado y tratado de forma única.

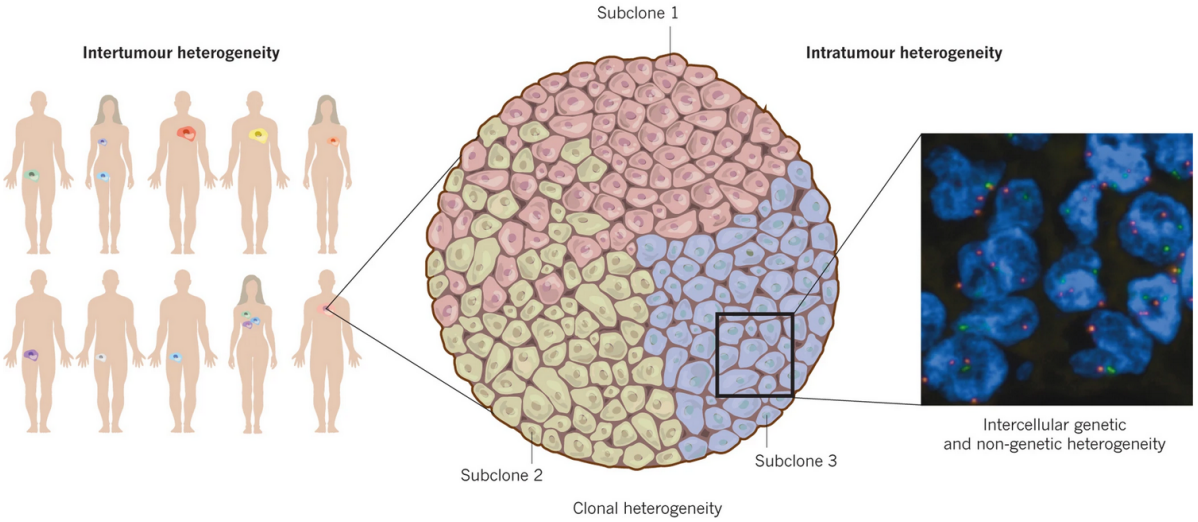

Figura 1.2: Diferentes subclones tumorales desarrollados debido a las variaciones que ocurren dentro de cada tumor. [6]

1.2. Medicina personalizada

La medicina personalizada o medicina de precisión es una práctica emergente de la medicina que utiliza el perfil genético de un individuo para guiar las decisiones tomadas en relación con la prevención, diagnóstico y tratamiento de la enfermedad. El conocimiento del perfil genético de un paciente puede ayudar a los médicos a seleccionar la medicina o la terapia adecuada, así como administrar la dosis o el régimen adecuados [9]. Esta práctica combina diferentes campos, como la medicina, la biotecnología o la informática, para conseguir un amplio ecosistema de datos distintos con el que ayudar a poder afrontar mejor la enfermedad. El avance en tecnologías de secuenciación de alto rendimiento ha sido fundamental para el desarrollo de la medicina personalizada. Mientras que la secuenciación del primer genoma humano en 2001 tuvo un coste aproximado de 95 millones de dólares, hoy en día es posible secuenciar un genoma completo por menos de 1000 dólares [10].

Una de las técnicas de secuenciación actuales más novedosa es la tecnología de secuenciación de single-cell (scRNA-seq). Esta técnica consiste en el análisis individual de cada célula para obtener su perfil transcriptómico, es decir, la información de la expresión genética de cada célula. Gracias a esto se puede obtener diversa información sobre las células que se están analizando, como anomalías en los genes, relaciones entre estos, o identificación y clasificación de tipos celulares. Desde la publicación del primer estudio de scRNA-seq publicado en 2009 [11], ha crecido el interés por continuar investigando las capacidades de esta tecnología, y es que, a pesar de que no es su único campo de aplicación, esta técnica resulta de especial utilizada para el estudio del cáncer, por sus propiedades de intraheterogeneidad descritas anteriormente. Para analizar el perfil transcriptómico recibido de una secuenciación

single-cell, se han desarrollado una serie de métodos y herramientas bioinformáticas con las que obtener o generar los datos de interés.

Un perfil transcriptómico se presenta a nivel informático como una matriz, en la que las filas representan las células, las columnas los genes y los valores son la expresión de cada gen en cada célula. El análisis de los datos generados por scRNA-seq implica importantes desafíos computacionales y bioinformáticos. Las matrices de expresión génica resultantes son de alta dimensionalidad y requieren técnicas avanzadas para su interpretación. Herramientas como Scanpy [12] en Python y Seurat [13] en R se han vuelto indispensables para el procesamiento y análisis de estos datos, permitiendo tareas como la normalización, reducción de dimensionalidad, clusterización y anotación celular [14].

La anotación celular es un paso fundamental en el análisis de datos de scRNA-seq. Consiste en asignar una identidad o tipo celular a cada célula individual, basándose en su perfil de expresión génica. Esto es crucial para comprender la composición celular de un tejido o tumor, y para identificar subpoblaciones celulares que puedan tener relevancia biológica o clínica.

Tradicionalmente, la anotación celular se ha realizado de forma manual, utilizando conocimiento experto y comparando la expresión de genes marcadores conocidos. Sin embargo, este proceso puede ser laborioso y subjetivo, especialmente cuando se trabaja con grandes volúmenes de datos o con células poco caracterizadas. En este contexto, la inteligencia artificial y el aprendizaje automático han emergido como herramientas poderosas para automatizar y mejorar las tareas de anotación celular.

1.3. Inteligencia artificial

La inteligencia artificial (IA) está revolucionando el mundo de una forma sin precedentes. En los últimos años, la inteligencia artificial se ha introducido en todos los ámbitos posibles, tanto sociables como laborables, sirviendo como una herramienta de ayuda para facilitar la realización de una gran variedad de tareas a las personas.

Una de las mayores revoluciones de la IA está ocurriendo en el campo de la salud y la biotecnología. Hoy en día, se generan enormes cantidades de datos, como imágenes clínicas de alta resolución y datos ómicos obtenidos mediante tecnologías de secuenciación, que superan la capacidad de análisis que los humanos pueden realizar por sí solos, por lo que es necesaria la ayuda de las máquinas para procesarlos y analizarlos. Al integrar estos datos clínicos con las capacidades de la IA como la predicción, detección y reconocimiento de patrones comunes se puede llegar a mejorar significativamente el diagnóstico e incluso el tratamiento de enfermedades [15].

Hasta ahora, métodos de aprendizaje automático más clásicos como *Random Forests* [16], *Support Vector Machines* [17] o regresiones logísticas han sido ampliamente utilizados en el campo de la biomedicina [18] [19] [20]. Sin embargo, los grandes volúmenes de datos que existen actualmente requieren sistemas más complejos. Modelos de aprendizaje profundo o redes de neuronas parecen ser capaces de lidiar con estas limitaciones, debido a su capacidad para aprender representaciones jerárquicas y capturar dependencias en los datos, incluso si estos contienen ruido o si presentan un gran número de dimensiones [21].

Uno de los principales usos del aprendizaje profundo en el campo de la salud es el

diagnóstico mediante imágenes, utilizando redes neuronales convolucionales (CNN) [22]. Estas redes son especialmente efectivas en el análisis de las modalidades de imágenes clínicas más comunes, como radiografías de rayos X, tomografías computarizadas (CT) e imágenes por resonancia magnética (MRI) [23]. Además del diagnóstico, estos modelos también son útiles como herramientas de apoyo, ya que se pueden emplear para la reconstrucción de imágenes de CT o MRI [24], o para mejorar la calidad de estas imágenes mediante técnicas de filtrado, armonización o eliminación de ruido [25], entre otros métodos. Esto no solo facilita una interpretación más clara por parte de los profesionales de la salud, sino que también puede conducir a diagnósticos más precisos y a una mejor planificación de los tratamientos.

Dentro del ámbito de los datos ómicos, el aprendizaje profundo también tiene múltiples aplicaciones, aunque quizá ésats no están vinculadas de manera tan directa con la práctica clínica como en el caso de las imágenes, ya que estos estudios están principalmente enfocados hacia la investigación. La capacidad de ciertos modelos de aprendizaje profundo para aprender un espacio latente, es decir, una representación simplificada de los datos (llamadas incrustaciones o *embeddings*), los hace ideales para manejar las características no lineales de los datos *single-cell*. Esto permite capturar de manera efectiva las características celulares más importantes [26]. Un ejemplo de ellas es la predicción de perturbaciones o respuesta a fármacos. Modelos como el CPA [27] o ScGen [28] utilizan las representaciones aprendidas de datos de *single-cell* junto con características de distintos fármacos, interacciones con proteinas u otras posibles perturbaciones que puedan afectar a las células, para generar perfiles transcriptómicos ante nuevas perturbaciones no vistas antes.

Como se ha comentado anteriormente, la IA también está emergiendo dentro del campo del análisis genómico, buscando automatizar los procesos más repetitivos o laboriosos sobretodo con grandes volúmenes de datos a través de modelos fundacionales, modelos entrenados con grandes volúmenes de datos de manera general, en este caso, para que aprendan representaciones de genes o células, y que se pueden re-entrenar con un dataset más pequeño para una tarea específica en concreto, como puede ser la anotación celular [29]. En este ámbito destacan modelos como ScGPT [30], ScBERT [31] o ScFoundation [32].

1.3.1. Redes de Neuronas Artificiales

Inspiradas en el comportamiento de un cerebro humano, la arquitectura de una red neuronal artificial consiste en una serie de capas conectadas entre sí, siendo su principal componente la neurona o perceptrón. En estas redes, por norma general, las neuronas de cada capa reciben como datos de entrada el resultado de la capa anterior (Figura 1.3).

Las redes de neuronas artificiales son consideradas como un método de aprendizaje de representaciones ya que permiten a una máquina recibir datos sin procesar y, de forma automática, descubrir las representaciones necesarias para realizar tareas de detección o clasificación [34]. Estas representaciones son funciones matemáticas contenidas en cada neurona, que en su forma vectorizada se pueden expresar de la siguiente forma:

$$Z = \sigma(XW + b) \tag{1.1}$$

Donde Z es la salida de la neurona; σ, la función de activación; X, los datos de entrada; W, el vector de pesos y b, el sesgo de la neurona.

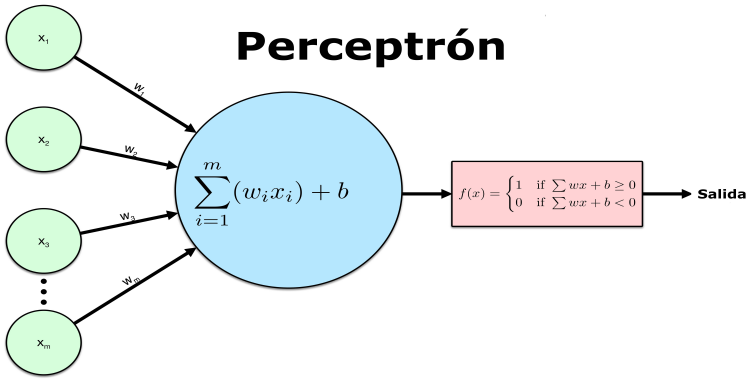

Figura 1.3: Neurona o Perceptrón. [33]

La función de activación es la encargada de introducir la no-linealidad dentro de las redes de neuronas, permitiendo al modelo encontrar las relaciones complejas entre las características de los datos. Algunos ejemplos de funciones de activación más populares pueden ser la *sigmoid*, *Tanh*, *ReLU* y la *Leaky ReLU* (Figura 1.4). La *sigmoid* y *Tanh* se utilizaron sobretodo en los comienzos de las redes neuronales; las dos tienen funciones parecidas, diferenciándose en que la *sigmoid* convierte los valores de la salida en $[0, 1]$ mientras que *Tanh* los convierte en $[-1, 1]$. Por otro lado, la función *ReLU* (*Rectifier Linear Unit*), la más utilizada en la práctica por su simplicidad y eficiencia, convierte en 0 los valores negativos y se comporta como la función identidad paralos valores positivos. *Leaky ReLU* es una variante de *ReLU* que transforma los valores negativos en valores muy pequeños para que no desaparezcan por completo [35].

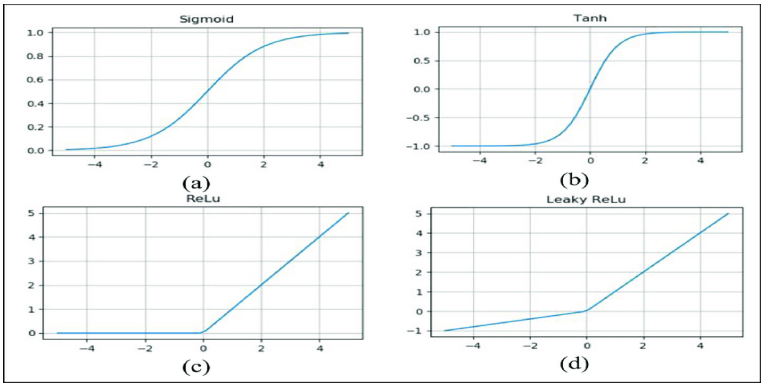

Figura 1.4: Ejemplos de funciones de activación.

El entrenamiento de una red neuronal consiste en encontrar los valores de las matrices W y b (los parámetros de la red) que provoquen el comportamiento deseado para la red. Esto se hace a través de la función de coste y la retropropagación o *backpropagation*.

El cálculo de la pérdida según la función de coste es esencial en el entrenamiento de una red neuronal, ya que cuantifica la diferencia entre las predicciones de la red y los valores reales esperados. Su principal objetivo es proporcionar un valor numérico que indique qué tan bien está funcionando la red neuronal en una tarea específica, como clasificación o regresión. La obtención de los pesos y sesgos ideales para el modelo se obtiene minimizando este valor. La función de coste depende de la tarea en

la que se quiera entrenar el modelo y la naturaleza de los datos. Algunas de las más utilizadas son [36]:

- **Error cuadrático medio (MSE)**: Para la mayoría de tareas de regresión logística.

$$C = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{1.2}$$

- **Entropía Cruzada**: Para clasificación binaria.

$$C = -\frac{1}{n}\sum_{i=1}^{n}[y_i \ln(\hat{y}_i) + (1 - y_i)\ln(1 - \hat{y}_i)] \tag{1.3}$$

- **Entropía Cruzada Categórica**: Para clasificación multiclase.

$$C = -\frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K} y_{i,k} \ln(\hat{y}_{i,k}) \tag{1.4}$$

Donde C es el coste total, n es el número de muestras, y_i es el valor real para la muestra i, $\hat{y}_i$ es la predicción de la red para la muestra i y K es el número de clases.

La retropropagación es el algoritmo utilizado para calcular el gradiente de la función de coste con respecto a los pesos y sesgos de la red neuronal. Comenzando desde la capa de salida, se calcula el gradiente de la función de coste con respecto a las salidas de cada neurona y se propaga hacia atrás. Esto implica calcular las derivadas parciales (gradientes) de la función de coste con respecto a cada peso y sesgo. De esta forma, los pesos y sesgos se actualizarán según el algoritmo de optimización escogido, siendo el Descenso de Gradiente Estocástico (SGD, por sus siglas en inglés) o la Estimación del Momento Adaptativa (Adam, por sus siglas en inglés). Las fórmulas de actualización de los pesos es:

$$W = W - \alpha \nabla W \tag{1.5}$$

$$b = b - \alpha \nabla b \tag{1.6}$$

Donde ∇ es el gradiente, α es la tasa de aprendizaje y W y b son las matrices de pesos.

Existen diversos tipos de redes neuronales (Figura 1.5) que pueden utilizarse dependiendo de los datos y la tarea que se desea realizar. Algunas de las más comunes son las redes *feedforward* (FFN) [38], las redes recurrentes (RNN), los autocodificadores (AE) y los *Transformers* [39], el tipo de arquitectura utilizada en este trabajo. Las FFN son el tipo más básico, donde la información fluye en una sola dirección desde la entrada hasta la salida. Las RNN están diseñadas para procesar datos secuenciales o temporales, permitiendo a la red mantener información de estados previos. Los autocodificadores son redes utilizadas para aprendizaje no supervisado, enfocadas en aprender representaciones eficientes de los datos para tareas como reducción de dimensionalidad. Por último, los *Transformers* son modelos basados en mecanismos de atención que comenzaron en el procesamiento del lenguaje natural y a lo largo de los últimos años han demostrado tener una gran eficacia en el resto de campos. Estos permiten capturar relaciones a largo plazo en los datos sin necesidad de procesarlos de forma secuencial.

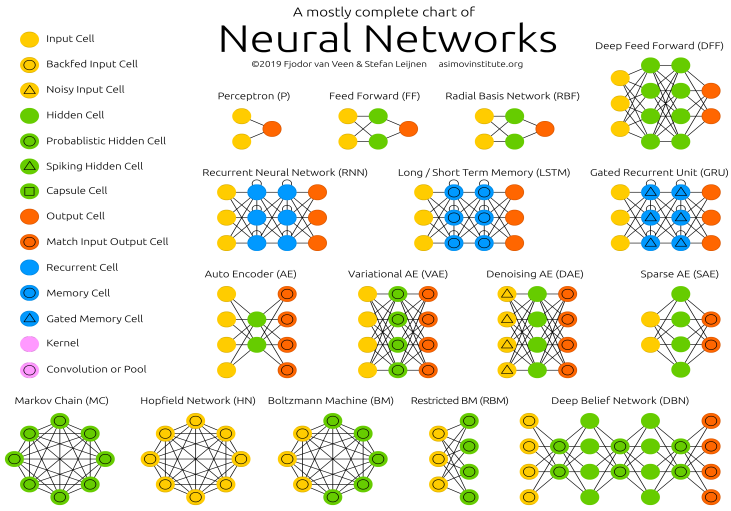

Figura 1.5: Tipos de redes neuronales. [37]

1.3.2. Transformers

Este tipo de modelos consisten en una arquitectura compuesta por redes neuronales llamadas mecanismos de atención. A pesar de que fueron diseñados con el propósito de ser utilizados en el procesamiento de lenguaje natural [39], su uso se ha extendido gracias a sus grandes resultados especialmente en campos como visión por computador y reconocimiento de voz [40].

En la Figura 1.6 se muestra gráficamente la arquitectura básica de un *Transformer*, que sigue un modelo de codificador-decodificador. El bloque Nx representa la repetición de capas del codificador y del decodificador, donde N es el número de veces que se repite cada bloque. En el caso de utilizar múltiples capas de codificador y decodificador, la arquitectura del *Transformer* se vería como en la Figura 1.7. Aquí, la salida de cada capa del codificador sirve como entrada para la siguiente capa de codificador. Una vez que la información ha pasado por todas las capas del codificador, la representación resultante se utiliza como entrada para cada una de las capas del decodificador. A su vez, la salida de cada capa del decodificador es la entrada para la siguiente capa.

Las figuras muestran cómo ocurre el flujo de los datos por la arquitectura de un *Transformer*. En estos modelos, la secuencia de entrada se divide en unidades llamadas *tokens*. Cada token es convertido en un vector de *embeddings* de dimensiones fijas. Para incorporar información sobre el orden de los tokens en la secuencia, se añade a estos *embeddings* una codificación posicional, que puede ser aprendida o fija. Esto es crucial, ya que a diferencia de las redes recurrentes o convolucionales, los Transformers no tienen un carácter de secuencialidad o posición por sí mismos, por lo que las codificaciones posicionales permiten al modelo tener en cuenta el orden de los tokens.

Los codificadores están compuestos por dos subcapas principales: un mecanismo de auto-atención multi-cabeza y una red neuronal de alimentación directa (*feed-forward*) completamente conectada, ambas seguidas de una conexión residual junto con una capa de normalización. Las redes muy profundas, como las que se encuen-

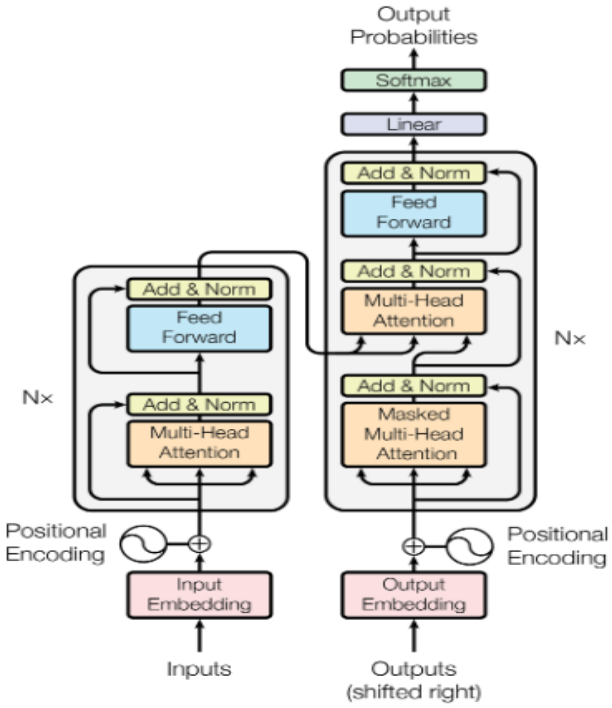

Figura 1.6: Arquitectura de un Transformer. [39]

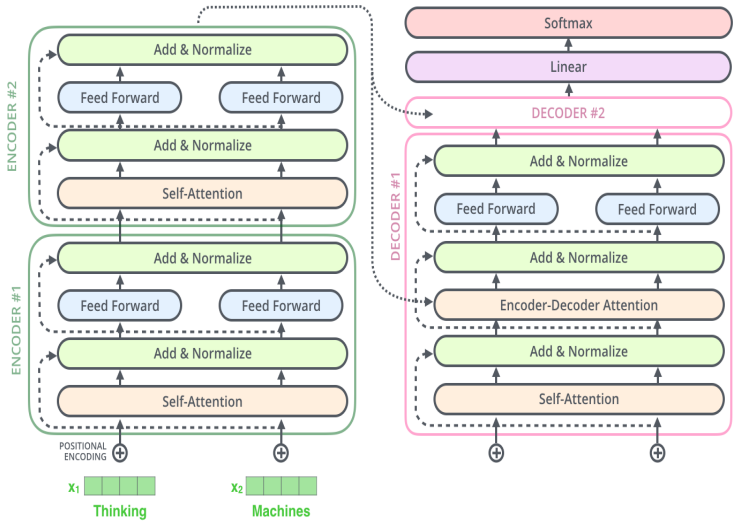

Figura 1.7: Ejemplo de Transformer con dos codificadores y dos decodificadores. [41]

tran en los *Transformers*, pueden sufrir el problema de los gradientes desvanecientes, es decir, que estos se vuelvan infinitamente pequeños tras el paso por un gran número de capas. La conexión residual resuelve este problema al sumar a la salida de cada red la propia entrada, antes de la normalización, mejorando el flujo de los datos por la red.

El mecanismo de auto-atención es el paso clave de esta arquitectura. Primero, se proyectan tres matrices de pesos entrenables durante el entrenamiento, que se utilizan

para calcular con cada token en la secuencia de entrada, tres vectores, que reciben los nombres de consulta (Q), clave (K) y valor (V).

$$Q = XW^Q, K = XW^K, V = XW^V \tag{1.7}$$

Donde X son los *embeddings* de entrada y las W corresponden a las matrices de consultas, claves y valores.

Con estos vectores, se puede calcular la atención entre cada uno de los tokens de entrada de la siguiente forma:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{1.8}$$

Donde d_k es la dimensión de las claves y consultas, que coincide con las dimensiones de los *embeddings* de entrada. El factor de escalado $\frac{1}{\sqrt{d_k}}$ se utiliza para evitar que los valores de la multiplicación de Q y K sean demasiado grandes, lo que podría llevar a la saturación de la función *softmax* y dificultar el aprendizaje. La función *softmax* se aplica a estas puntuaciones escaladas para convertirlas en una distribución de probabilidad. Esto normaliza los valores y asigna pesos positivos que suman 1 a todas las posibles interacciones entre tokens. De esta manera, la función *softmax* determina la importancia relativa de cada token en relación con los demás, permitiendo al modelo enfocar su atención en los tokens más relevantes al generar la representación final.

Esta fórmula utiliza el producto escalar para calcular cómo de relacionados están los tokens entre sí. Se calcula para cada token consigo mismo y con los demás, utilizando la consulta de uno y las claves de los otros. La multiplicación de la consulta con la clave mide la similitud entre los tokens. Estos valores, tras ser escalados y pasar por la función *softmax*, se convierten en pesos de atención positivos que determinan cuánto influirá el valor de cada token en la representación final. Si los tokens guardan una relación estrecha, el peso asignado será mayor; si la relación es menor, el peso será un valor muy pequeño tras pasar por la función *softmax*.

El proceso explicado hasta ahora se corresponde con un único cálculo de atención para una secuencia de entrada. En realidad, un *Transformer* cuenta con múltiples cabezas de atención, cada una con sus tres matrices de pesos, que realizan este cálculo en paralelo para cada secuencia de entrada. Cada una de las cabezas puede encontrar diferentes características, relaciones o representaciones de los tokens de entrada. El resultado de las múltiples cabezas de atención se concatena de la siguiente forma:

$$\text{Multihead}(Q, K, V) = \text{Concat}(head_1, head_2, \ldots, head_n)W^O$$
$$\text{donde} \quad head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$$

Por otra parte, la subcapa neuronal completamente conectada consiste en dos transformaciones lineales con una función de activación no lineal (*ReLU*) entre ambas. La primera transformación expande el vector de entrada, mientras que la segunda lo vuelve a reducir a su dimensión original.

El decodificador genera tokens de manera secuencial, basándose en los que ya ha generado y en las representaciones (embeddings) obtenidas del codificador. Una diferencia importante es la inclusión de una primera capa de atención enmascarada

(masked attention), que evita que el decodificador calcule atención sobre tokens posteriores a la posición que está generando, lo cual es especialmente útil durante el entrenamiento. A partir de este punto, el decodificador opera de forma similar al codificador. Las salidas del codificador se integran en la segunda capa de atención del decodificador, lo que le permite acceder al contexto de la secuencia de entrada. A continuación, se utiliza una red completamente conectada, igual que en el codificador. Finalmente, se aplica una transformación lineal seguida de una función softmax, que convierte las representaciones de salida del decodificador en probabilidades, utilizadas para seleccionar el siguiente token.

1.3.3. Transformers en *single-cell*

Como se mencionó anteriormente, la inteligencia artificial en biología está avanzando hacia el uso de modelos fundacionales que, por lo general, utilizan como arquitectura base los Transformers. Estos son modelos entrenados con grandes cantidades de datos y que pueden adaptarse fácilmente a tareas específicas. Este tipo de modelos, al igual que los *Transformers*, han sido muy efectivos en campos como la visión por computador, el análisis de series temporales y el modelado de datos genómicos y proteómicos. La capacidad de los *Transformers* para manejar grandes volúmenes de datos, su flexibilidad y su habilidad para adaptarse a diferentes tareas los convierten en opciones muy prometedoras para el análisis de datos de *single-cell* (Figura 1.8). Sin embargo, los datos *single-cell* no tienen una estructura secuencial, lo que representa un reto para estos modelos [42].

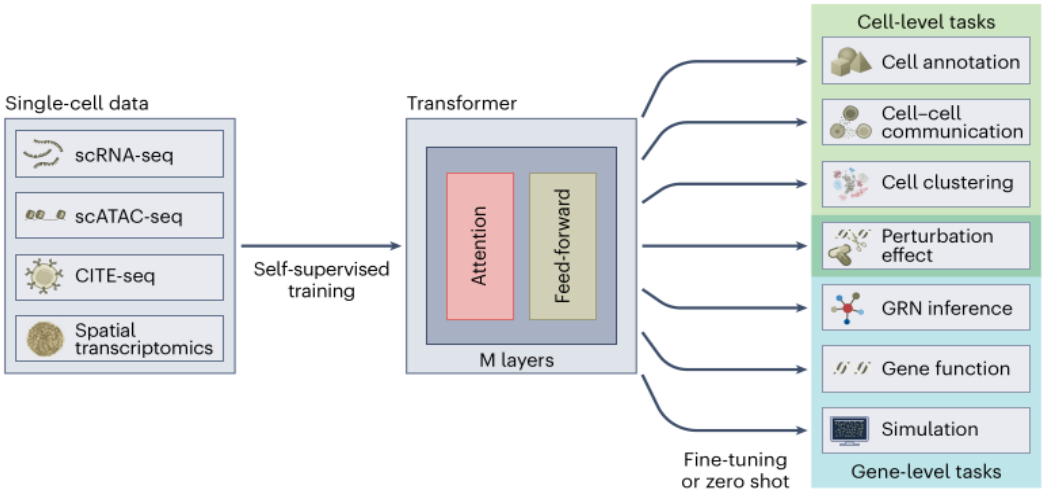

Figura 1.8: Transformers en *single-cell*. [42].

Para solucionar el problema de la no-secuencialidad se han desarrollado varios enfoques, siendo los más comunes el ordenamiento, la categorizacion de valores y la proyección de valores.

- **Ordenamiento**: Los datos se representan como una secuencia, similar al procesamiento de lenguaje natural. El orden puede ser de cualquier tipo, como por ejemplo, por nivel de expresión. Este método reduce la resolución de los datos, lo que resulta en pérdida de información.

- **Categorización de valores**: Cada gen es representado con un embedding, mientras que los niveles de expresión son discretizados, creando categorías de valores. Cada categoría es representada por un *embedding*, de forma que el token de entrada al *Transformer* es la suma del *embedding* que representa el gen y de su *embedding* de valor. Con este método también se pierde cierta resolución de los datos.
- **Proyección de valor**: En este método, los tokens son la suma de una proyección de los valores de expresión y un *embedding* de posición o de representación del gen, donde la proyección es normalmente una transformación lineal. Este método no pierde resolución, sin embargo, los *embeddings* son valores continuos, lo que difiere de los transformers tan eficientes hechos para procesamiento de lenguaje natural, y el impacto de este tipo de *embeddings* en la eficiencia del modelo no es del todo claro.

Normalmente, los codificadores y decodificadores de los *Transformers* se utilizan independientemente. Por una parte, los modelos solo-codificador se utilizan son comunmente utilizados para generar *embeddings* que capturan las propiedades de los genes, así como sus niveles de expresión y su contexto. Por otro lado, los solo-decodificadores o codificadores-decodificadores se utilizan para generación de secuencias, permitiendo simular perfiles transcriptómicos a partir de matrices ya existentes.

Capítulo 2

Objetivos

Este trabajo de fin de máster se ha realizado en colaboración con la Unidad de Bioinformática (UB) del Centro Nacional de Investigaciones Oncológicas (CNIO). Esta unidad está compuesta por un equipo de bioinformáticos con una formación multidisciplinaria que emplea métodos computacionales para abordar preguntas científicas. Sus objetivos incluyen el desarrollo de nuevas metodologías y herramientas de bioinformática que faciliten la integración de datos biológicos y clínicos, así como el análisis de genomas de pacientes con cáncer para identificar biomarcadores y mecanismos de respuesta a tratamientos. Además, la UB ofrece apoyo en el análisis e interpretación de datos mediante métodos computacionales y estadísticos, mantiene las infraestructuras de computación científica del CNIO y proporciona formación en el uso de herramientas y métodos de bioinformática.

El principal objetivo de este trabajo es la exploración y análisis de herramientas basadas en inteligencia artificial para datos genómicos. A su vez, este objetivo se puede dividir en varios sub-objetivos más específicos, que son:

- Explorar el estado del arte sobre modelos fundacionales en datos ómicos.
- Evaluar el rendimiento de los modelos fundacionales escogidos en la tarea de anotación celular utilizando un dataset single cell pan-cancer, analizando su precisión y capacidad para clasificar correctamente diferentes tipos de células..
- Evaluar métodos de anotación célular clásicos y comparar su desempeño con las herramientas basadas en inteligencia artificial.
- Análisis de resultados para identificar los beneficios y limitaciones de cada modelo en el contexto de la anotación.

Capítulo 3

Materiales y Métodos

3.1. Dataset

El conjunto de datos utilizado en este trabajo es el Therapeutic Cancer Cell Atlas (TCCA), una extensa recopilación de 36 estudios de datos *single-cell* que han sido curados y anotados manualmente. Este dataset está compuesto por más de 1.800.000 células, recogidas de 538 pacientes. De estos pacientes, se han obtenido un total de 853 muestras, abarcando 41 tipos diferentes de tumores (Figura 3.1).

Los estudios que conforman el TCCA aportan tanto células tumorales como células del microambiente tumoral, es decir, células sanas que se encuentran en el entorno del tumor. Esto permite un análisis más completo y detallado del cáncer, ya que facilita el estudio de las interacciones entre las células malignas y las células sanas que rodean el tumor. En total, el dataset incluye aproximadamente 1 millón de células tumorales y 800.000 células sanas.

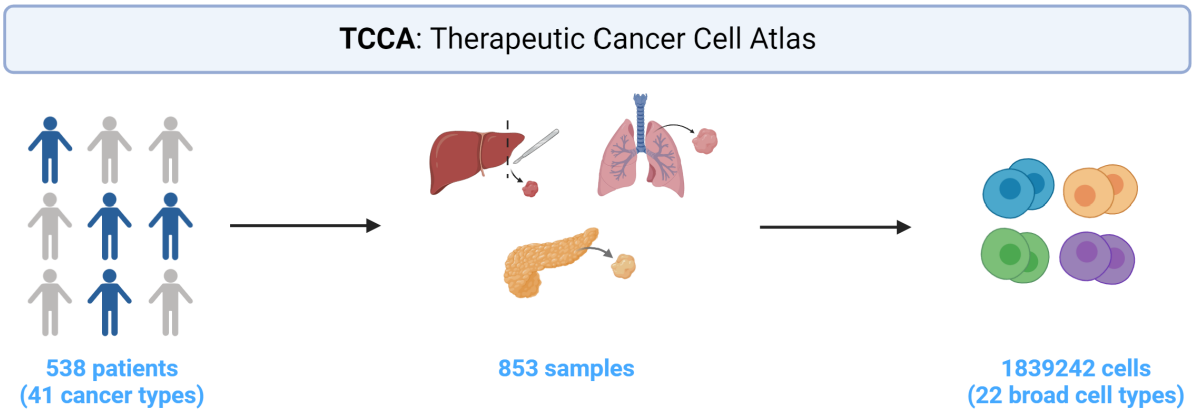

Figura 3.1: The Therapeutic Cancer Cell Atlas.

El TCCA contiene 22 tipos celulares distintos, cuya distribución se puede observar en la figura 3.2. Sin embargo, es importante destacar que este dataset presenta un gran desbalanceo de clases, ya que alrededor de un 60 % de las células pertenecen a la clase *Malignant*. En cuanto al resto de los tipos celulares, la distribución es más equilibrada excepto por el caso de los tipos celulares *Granulocyte*, *Innate lymphoid cells* y *Platelet*, que cuentan con menos de 100 células cada uno. Debido al escaso

número de muestras, se considera que estos tipos celulares no proporcionan suficiente información para que los modelos puedan aprender sus características. Por esta razón, se decidió eliminar estos tipos celulares del dataset antes de entrenar los modelos. Además, también se excluyeron las células cuya clase era desconocida, las cuales sumaban aproximadamente 2.000 células. Esta limpieza del dataset garantiza que los modelos se entrenen con datos de calidad y representativos.

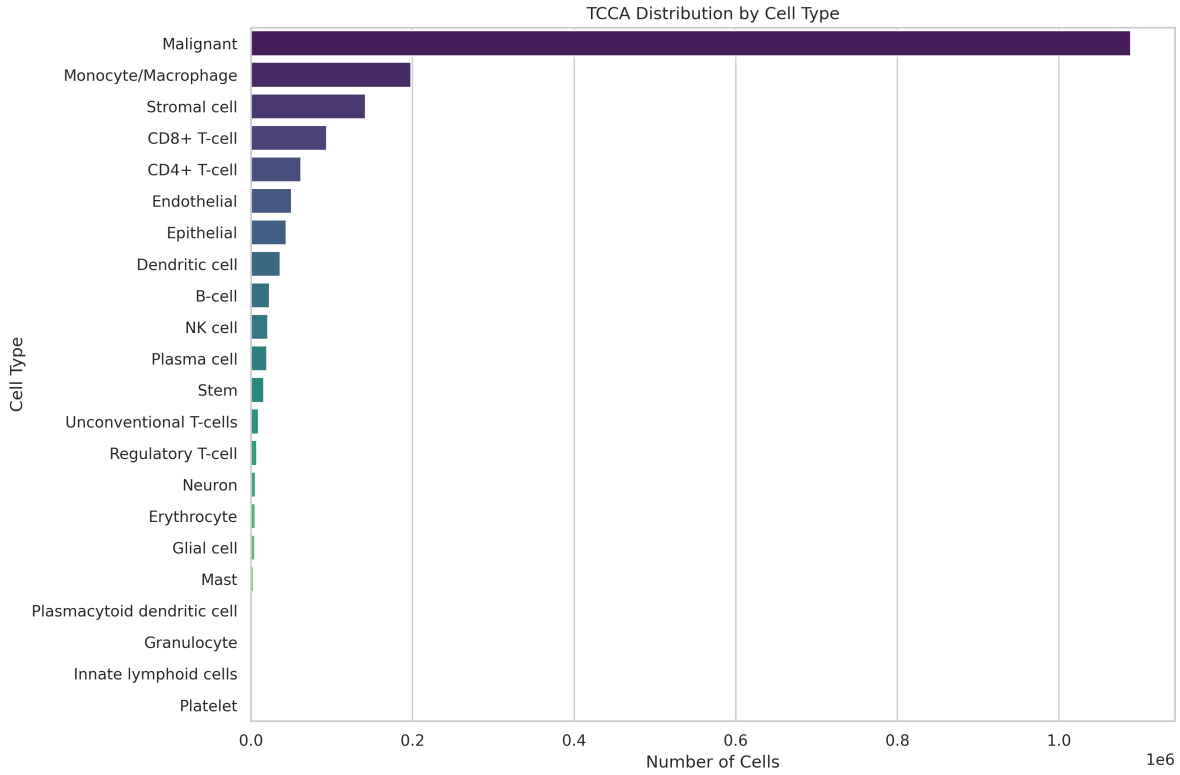

Figura 3.2: Distribución de tipos celulares.

Con el fin de tener un dataset de entrenamiento y uno de test en el que comprobar el funcionamiento del modelo ante datos no vistos, el TCCA se ha dividido en dos datasets: *Train* y *Test*. Para hacer esta división se han seleccionado los x pacientes con más células para el dataset de *Train*, de forma que el *test* del modelo se hace con células de pacientes que no ha visto antes. De esta forma, es posible evaluar el modelo en un escenario más realista, simulando su aplicación en datos de nuevos pacientes. En las figuras 3.3, 3.4 se puede ver la distribución de los datasets de *train* y *test*, respectivamente, mientras que en el cuadro 3.1 se muestra el número de células y pacientes de cada división.

	Num. Celulas	**Pacientes**	**Num. Tipos celulares**
TCCA	1.839.242	538	22
Train	1.478.992	243	19
Test	355.781	295	19

Tabla 3.1: División del TCCA en los datasets *Train* y *Test*.

También se ha modificado el TCCA para tener un dataset sin células malignas y

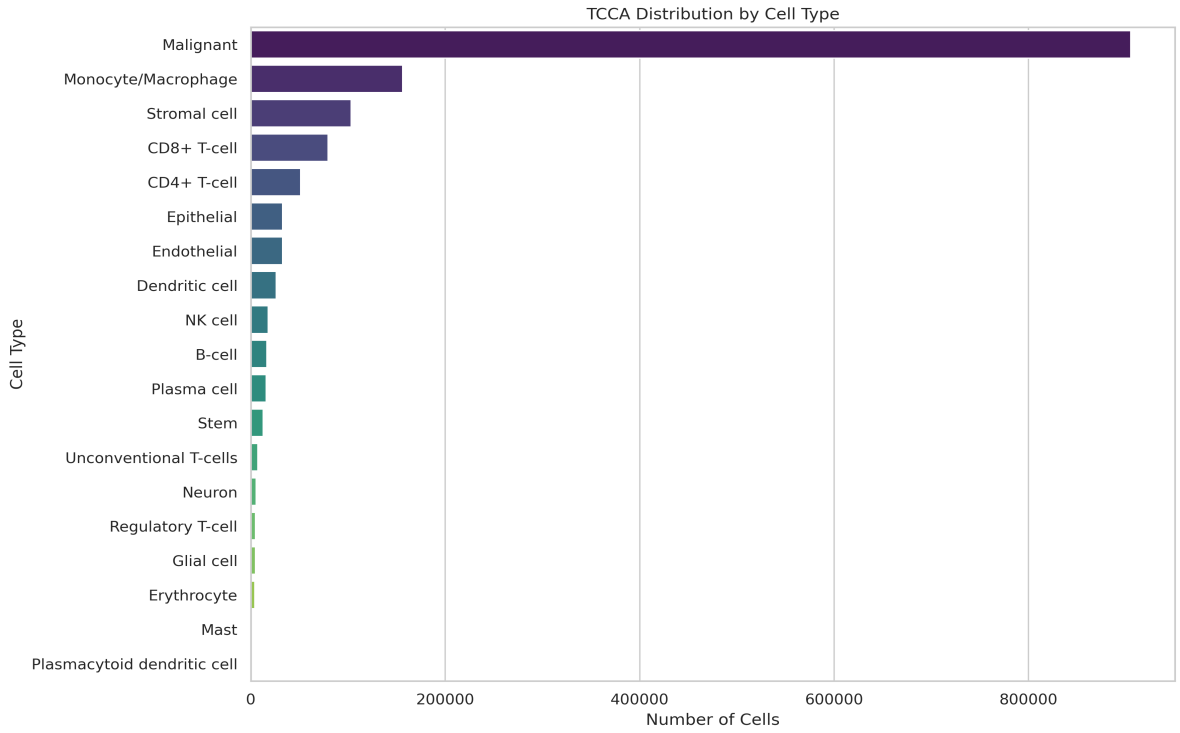

Figura 3.3: Distribución de tipos celulares del dataset de *Train*.

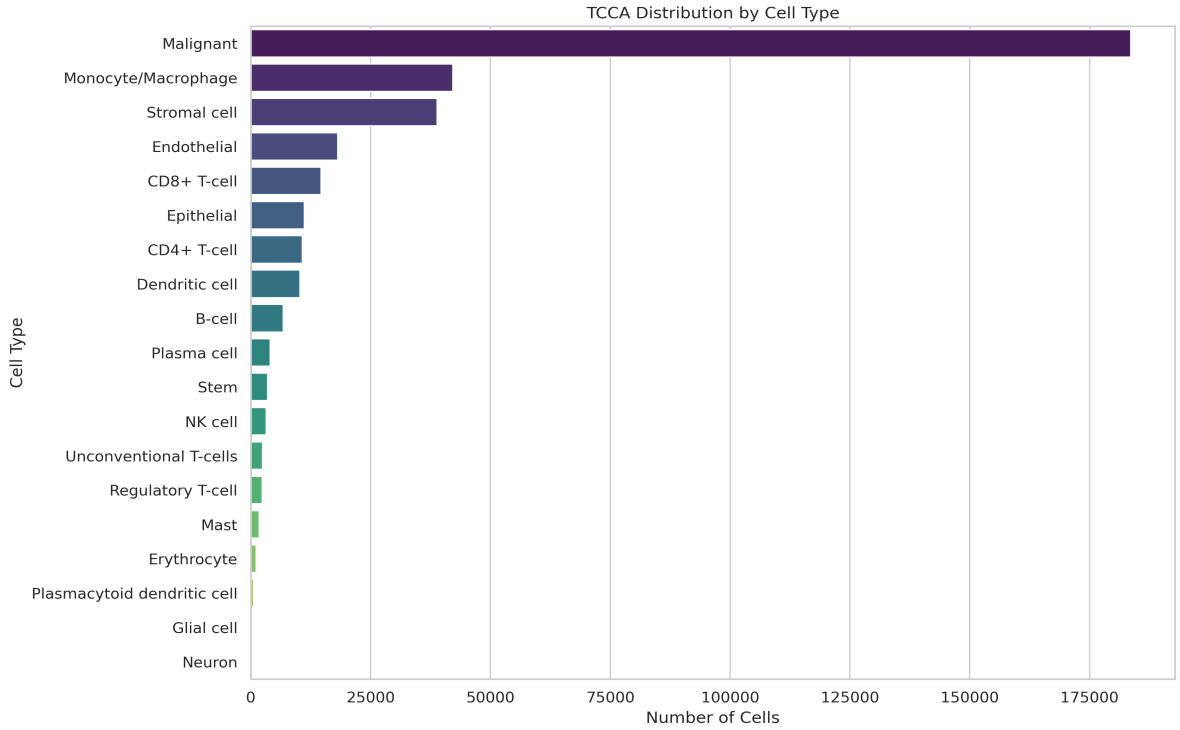

Figura 3.4: Distribución de tipos celulares del dataset de *Test*.

otro con únicamente células malignas. Normalmente las células malignas presentan

mutaciones genéticas o cambios drásticos en la expresión génica que las hacen destacar en este tipo de modelos, por lo que la clase 'Malignant' es relativamente fácil de predecir, ya que es muy genérica. Las células sanas presentan un desafío mayor en su identificación y clasificación debido a que muchas de ellas comparten perfiles de expresión génica, por lo que la heterogeneidad celular en tejidos sanos es sutil y a menudo requiere técnicas avanzadas y sensibles para distinguir entre diferentes subtipos celulares. La diversidad funcional de las células sanas implica que pequeñas diferencias en la expresión génica pueden tener significados biológicos significativos, pero son más difíciles de detectar y clasificar. Por estos motivos es interesante comprobar si los modelos escogidos son capaces de obtener buenos resultados tan solo con las células sanas. Para obtener el dataset sin células malignas tan solo se ha eliminado la clase 'Malignant' de Train y Test.

El dataset que contiene únicamente células malignas es útil para ver si los modelos son capaces de distinguir el tipo tumoral del que provienen estas células. Analizar diferencias entre tipos tumorales puede aportar información muy valiosa de cara al estudio del cáncer y así comprender cómo puede responder un tumor ante un tratamiento o cómo será su progresión. Para preparar este *dataset* y establecer una división adecuada entre los conjuntos de entrenamiento y prueba, se han eliminado las células del TCCA que correspondían a líneas celulares, siendo estas 50,000 células aproximadamente. De esta forma, quedan únicamente las células provenientes de pacientes reales. El principal motivo para eliminar estas células es que introducían una gran cantidad de tumores únicos en el conjunto de datos, lo que hacía prácticamente imposible realizar una división correcta entre Train y Test. Esto podría provocar que el modelo no generalizase bien a nuevos datos, afectando su rendimiento y capacidad predictiva. Además, las células de líneas celulares crecen en condiciones de laboratorio, fuera del entorno natural del cuerpo humano. Por esta razón, no se comportan de la misma manera que las células extraídas directamente de pacientes, lo que podría introducir sesgos y reducir la validez de los resultados obtenidos. Teniendo en cuenta que estas células representaban una proporción pequeña en comparación con el total de aproximadamente un millón de células, su eliminación no afectaba significativamente al rendimiento de los modelos. Por todas estas razones, se optó por excluirlas.

3.2. Modelos

Tras una revisión a través del estado del arte sobre *Transformers* y datos ómicos en la que se examinaron modelos como ScFoundation, Geneformer, ScBERT o ScGPT, se decidió utilizar estos dos últimos. La elección de ScBERT se basa en que es el primer modelo de Transformer adaptado específicamente para datos de *single-cell*, siendo pionero en el campo. Por otro lado, se ha seleccionado ScGPT debido a que actúa como una contraparte al modelo BERT tradicional y ha demostrado un éxito notable en distintas evaluaciones.

3.2.1. ScGPT

ScGPT [30] es un modelo fundacional pre-entrenado con datos de *single-cell* que puede ser adaptado para realizar diversas tareas para estos datos. Entre estas se encuentran clusterización, corrección batch, anotación celular, inferencia de red de

genes, integración multiómica y predicción de perturbaciones (Figura 3.7).

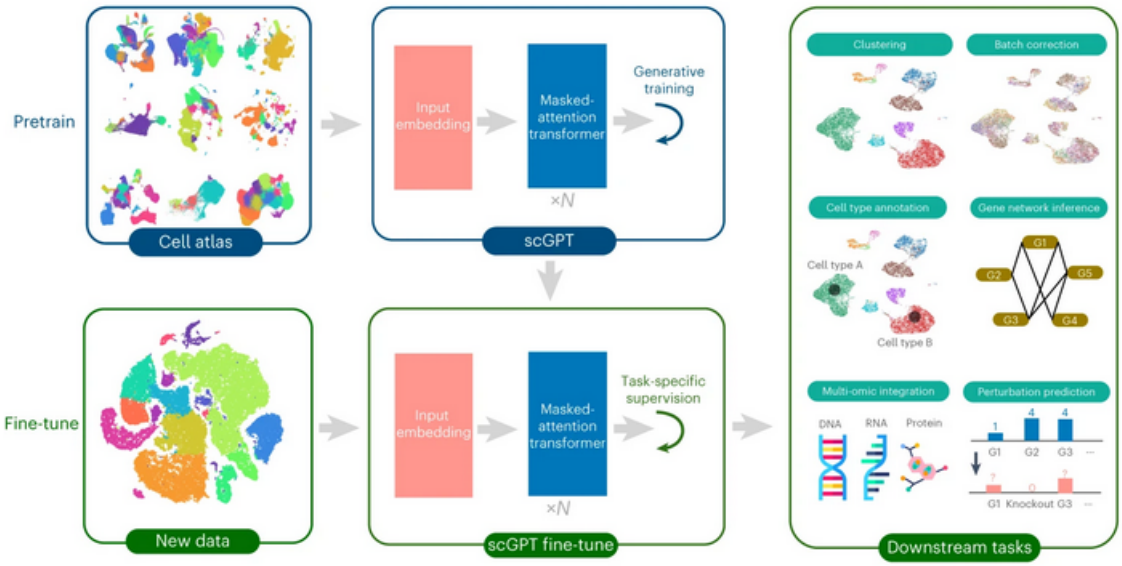

Figura 3.5: *Workflow* de ScGPT. [30]

El modelo pre-entrenado cuenta con 12 bloques de tipo *Transformer* con 8 cabezas de atención cada uno, y cuenta con un espacio latente de tamaño 512, al igual que la capa neuronal completamente conectada. Se ha entrenado utilizando Adam como optimizador, con un mini-batch size de 32 y una tasa de aprendizaje de 0.0001 durante 6 épocas. Hay disponibles varios modelos pre-entrenados, de entre los cuales para este trabajo se ha elegido el modelo pan-cancer, entrenado en 5.7 millones de células procedentes de varios tipos de cáncer.

Como *embeddings* de entrada, ScGPT utiliza *tokens* de gen, valores de expresión y *tokens* de condición. Los genes se consideran como la unidad de información más pequeña en ScGPT, representando cada nombre de gen con un identificador $id(g_j)$ único para cada gen, por lo que el vocabulario de *tokens* de ScGPT está formado por estos identificadores. Esto aporta una gran flexibilidad, ya que permite la integración de múltiples estudios en los que no todos los genes son comunes. Los *tokens* de gen de entrada de la célula i se representan mediante un vector $t_g^{(i)}$ mediante:

$$t_g^{(i)} = [id(g_1^{(i)}), id(g_2^{(i)}), ..., id(g_M^{(i)})] \tag{3.1}$$

donde M es la longitud de entrada máxima pre-establecida.

Los valores de expresión requieren de un procesamiento antes de servir como datos de entrada al modelo, debido a que estos pueden adoptar magnitudes muy distintas debido a los distintos protocolos de secuenciación. Estas diferencias no se mitigan fácilmente con técnicas de procesado como normalización o transformaciones, por eso, se recurre a la categorización de valores para convertir todas las cuentas de expresión en valores relativos. Para dividir los valores en categorías se crean B intervalos consecutivos $[b_k, b_{k+1}...b_B]$ donde $k \in [1, 2, ...B]$ de igual proporción de genes $(1/B)$ para cada célula. El valor de expresión categorizado para la célula i puede representarse como:

$$x_j^{(i)} = \begin{cases} k, & \text{si } X_{i,j} > 0 \text{ y } X_{i,j} \in [b_k, b_{k+1}] \\ 0, & \text{si } X_{i,j} = 0 \end{cases}$$

Los *tokens* de condición representan meta-información asociada con cada gen de forma individual, que se representa mediante un vector de la misma longitud que el vector de genes, de la forma:

$$t_c^{(i)} = [t_{c,1}^{(i)}, t_{c,2}^{(i)}, ..., t_{c,M}^{(i)}] \tag{3.2}$$

donde $t_{c,j}^{(i)}$ representa un *integer* que corresponde a una condición.

Una vez asociados los *tokens* y categorizados los valores de expresión, se pasan por las capas de *embeddings*, para los tokens de gen y de condición se utilizan las capas convencionales de *embedding* que proporciona PyTorch, mientras que para los valores de expresión se utiliza una capa completamente conectada para mejorar la expresividad. La expresión del *embedding* final $h^{(i)} \in R^{MxD}$ para la célula i se define como:

$$h^{(i)} = emb_g(t_g^{(i)}) + emb_x(x^{(i)}) + emb_c(t_c^{(i)}) \tag{3.3}$$

Finalmente, $h^{(i)}$ es el *embedding* que entra en la secuencia de bloques *transformer*, obteniendo a su salida una representación $h_n^{(i)}$ del mismo tamaño que el *embedding* de entrada. Para tareas que necesiten representación a nivel celular, como por ejemplo, la anotación, se utiliza un *token* especial llamado 'cls'. Este *token* se introduce en el comienzo del vector de *tokens* de genes, de forma que la representación final de la célula i, $h_c^{(i)} \in R^D$, corresponde al vector en la posición del *token* 'cls' de la salida de los bloques *transformer*:

$$h_c^{(i)} = h_n^{(i)}[cls\_index] \tag{3.4}$$

Durante el pre-entrenamiento, se utiliza un enmascaramiento en el mecanismo de atención para tratar con el problema de la no secuencialidad en los genes:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d}} + A_{mask})V \tag{3.5}$$

Donde $A_{mask} \in \{0, -inf\}$ es la máscara de atención que controla qué genes generan atención entre sí según:

$$a_{i,j} = \begin{cases} 0, & \text{si } j \notin \text{genes desconocidos,} \\ 0, & \text{si } i = j \text{ y } j \in \text{genes desconocidos,} \\ -\infty, & \text{si } i \neq j \text{ y } j \in \text{genes desconocidos.} \end{cases}$$

Añadiendo -inf a un elemento en la posición (i,j) de la matriz, el peso de atención correspondiente se reduce a cero tras aplicar la función softmax. Esto impide que haya atención entre la consulta número i y la clave número j. Por el contrario, si suma el valor 0, los pesos de atención no se modifican y permanecen iguales. Este método de enmascaramiento de atención permite que el modelo ponga el enfoque en distintos elementos según el contexto. En cada iteración, ScGPT predice la expresión génica de un nuevo set de genes, los cuales se vuelven genes conocidos en el siguiente cálculo de atención. De esta forma se permite emular la técnica de enmascaramiento causal utilizada en los decodificadores de transformadores tradicionales para predecir el siguiente token. Lo logra al realizar predicciones de manera secuencial en datos de células individuales que no siguen una secuencia por sí mismos.

Como se muestra en la Figura 3.6A, durante el entrenamiento, se asignan aleatoriamente unos genes como desconocidos, de modo que sus valores de expresión se omiten en la entrada. Como se ha explicado anteriormente, la atención solo se aplica entre los genes conocidos y el propio gen desconocido que se está consultando, pero no hacia las posiciones de otros genes desconocidos. Es decir, el gen que se va a predecir en la posición j solo tiene atención con el *embedding* de la célula, los genes conocidos y consigo mismo, pero no con otros genes desconocidos, como se muestra en la última fila de la máscara de atención.

Durante la inferencia para la generación con *cell-prompt*, ScGPT genera toda la expresión génica a nivel del genoma condicionada a tipos celulares específicos. Se introduce un *embedding* celular en la primera posición, representando la condición del tipo celular.

El proceso completo de generación de los valores de expresión génica se realiza en K pasos iterativos (Figura 3.6B). En una iteración $i \in 1, 2, \ldots, K$, el mecanismo de enmascaramiento de atención permite atención con todos los genes predichos desde las iteraciones previas de 0 a $i-1$. En cada iteración, ScGPT selecciona el mejor 1/K de los genes del conjunto desconocido con la mayor confianza de predicción para incluirlos como genes conocidos en la siguiente iteración $i+1$.

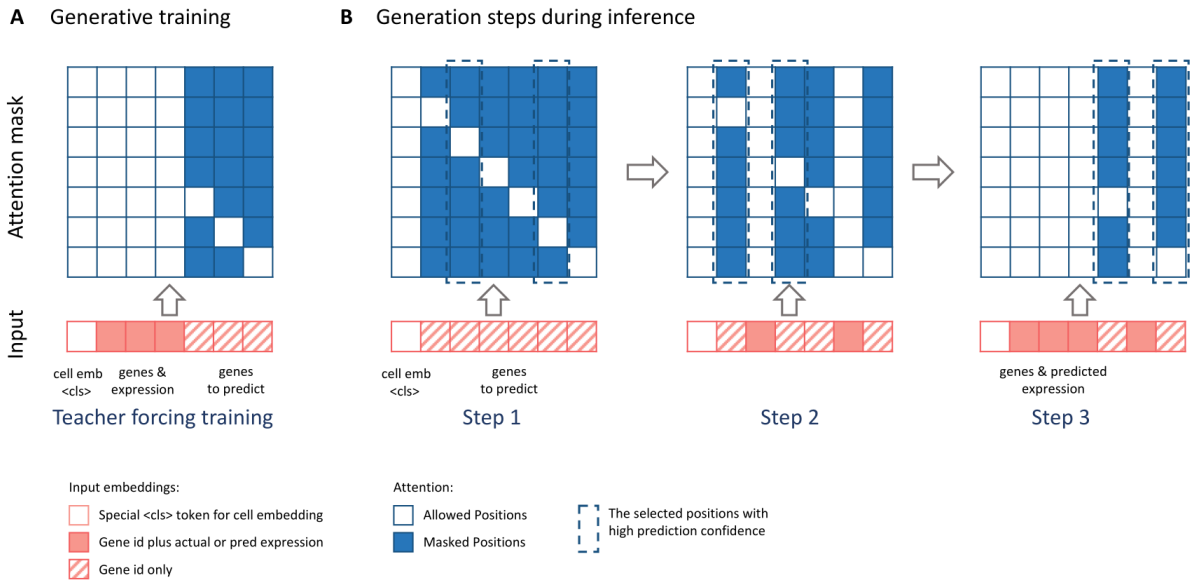

Figura 3.6: Pre-entrenamiento e inferencia de ScGPT. [30]

3.2.1.1. Fine-tuning para clasificación celular

Para la anotación celular, se utilizan los *embeddings* de células como entradas para un clasificador perceptrón multicapa (MLP) con el fin de que clasifique tipos celulares a partir de sus representaciones $h_c^{(i)}$. Además de su correspondiente categorización, los valores de expresión génica se normalizan y transforman. El *fine-tuning* se ha realizado utilizando las etiquetas de los *datasets* de entrenamiento y de test del TCCA como *ground truths*, optimizando el entrenamiento con el algoritmo *Adam* y una función de pérdida de entropía cruzada categórica.

La estructura del modelo clasificador es casi idéntica a la del pre-entrenamiento, excepto por el clasificador perceptrón multicapa que se añade. Este cuenta con tres capas lineales, cada una de ellas seguidas de una capa de activación *ReLU* y una de activación. A la hora de hacer el *fine-tuning*, se descongelan todos los parámetros, es decir, se vuelve a entrenar el modelo pre-entrenado a la vez que el MLP.

Las gráficas de pérdida de entrenamiento y validación para este proceso de *fine-tuning* se muestran en el Anexo A.

3.2.2. ScBERT

Single-cell BERT (ScBERT) [31], es un modelo de BERT adaptado para la anotación celular en datos de scRNA-seq. Siguiendo el enfoque de pre-entrenamiento y *fine-tuning*, este modelo ha sido pre-entrenado mediante aprendizaje autosupervisado con alrededor de un millón de células de diferentes estudios, sin anotaciones ni etiquetas, tan solo utilizando los valores de expresión génica.

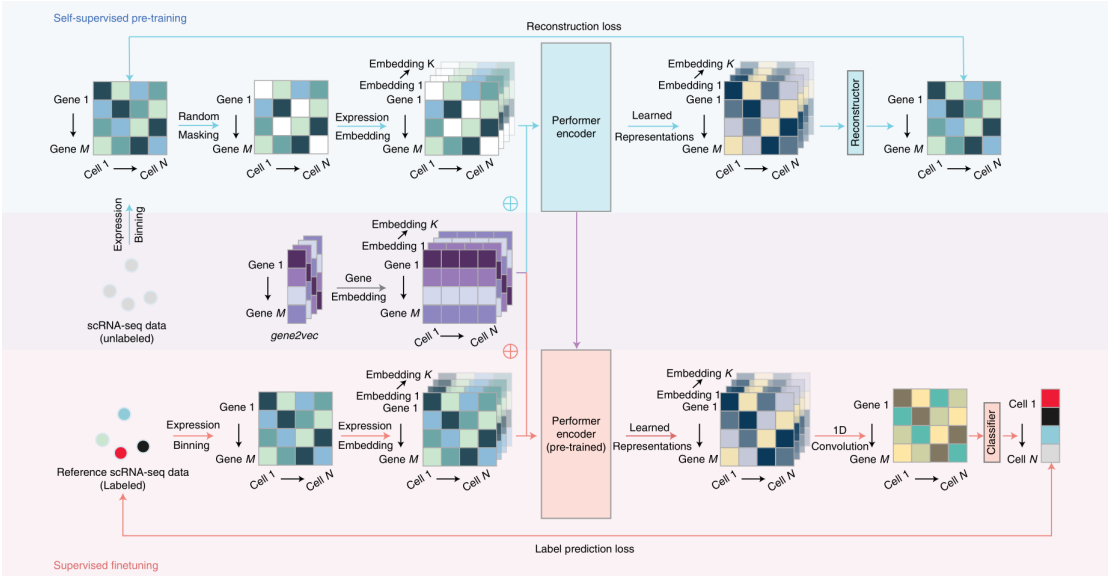

Figura 3.7: *Workflow* de ScBERT. [31]

El enfoque que sigue ScBERT se basa en dos etapas clave: el pre-entrenamiento y el *fine-tuning*. En la fase de pre-entrenamiento, el modelo se entrena de manera autosupervisada utilizando alrededor de un millón de células provenientes de diversos estudios. En este paso, no se utilizan anotaciones ni etiquetas; el modelo aprende directamente de los valores de expresión génica. Esto le permite captar patrones generales y relaciones entre genes a partir de una amplia variedad de datos no etiquetados.

Una vez que el modelo ha sido pre-entrenado, se procede a la etapa de *fine-tuning*. Aquí, ScBERT incorpora un clasificador y ajusta sus parámetros utilizando conjuntos de datos de referencia que sí están etiquetados. Este proceso permite que el modelo adapte el conocimiento general adquirido durante el pre-entrenamiento a tareas específicas, como la asignación precisa de tipos celulares. Al igual que ocurre con ScGPT, el pre-entrenamiento ayuda al modelo a aprender la "sintaxis"general de las interacciones entre genes, lo cual es útil para eliminar efectos de lote entre distintos

conjuntos de datos y mejorar su capacidad de generalización. El *fine-tuning* garantiza que los *embeddings* de salida para cada gen codifiquen información contextual relevante para los perfiles de expresión del conjunto de datos de referencia utilizado.

Un desafío importante al trabajar con datos de scRNA-seq es la gran cantidad de genes involucrados. Los modelos *Transformer* tradicionales están limitados a procesar secuencias de entrada de hasta 512 tokens, pero los datos de scRNA-seq suelen incluir más de 10,000 genes. Para superar esta limitación, ScBERT reemplaza las capas codificadoras tipo *Transformer* por capas tipo *Performer* [43], una variante diseñada para mejorar la eficiencia y escalabilidad en el procesamiento de secuencias largas. Gracias a esta modificación, ScBERT puede manejar toda la información a nivel de genes sin necesidad de reducir la cantidad de datos o enfocarse solo en ciertos genes. Esto permite que los genes más relevantes y sus interacciones se destaquen de forma natural, facilitando el descubrimiento de patrones de expresión y dependencias importantes para la anotación celular. En concreto, ScBERT está compuesto por 6 capas de tipo *Performer*, cada una con 10 cabezas de atención.

Al igual que en ScGPT, los valores continuos se discretizan en categorías para simplificar el procesamiento. Después de este paso, algunos valores de expresión se enmascaran aleatoriamente antes de introducirse al modelo, en lugar de hacerlo mediante una máscara de atención como en ScGPT. Este enmascaramiento aleatorio ayuda al modelo a aprender a predecir los valores ocultos basándose en el contexto de los demás genes, mejorando su comprensión de las interacciones entre los genes.

Para generar los *embeddings* de los genes, ScBERT utiliza gene2Vec [44], un método de aprendizaje automático que asigna a cada gen humano un vector en un espacio de 200 dimensiones. Esto significa que cada gen tiene un embedding único que captura similitudes semánticas basadas en la coexpresión. Los genes que suelen expresarse juntos tienden a tener representaciones vectoriales más cercanas entre sí. Estas representaciones distribuidas han demostrado ser útiles para capturar interacciones y relaciones funcionales entre genes. Una vez obtenidas estas representaciones, son sumadas a los valores de expresión génica discretizados y enmascarados para introducirse en el modelo.

3.2.2.1. Fine-tuning para clasificación celular

La salida de ScBERT es una característica de 200 dimensiones para cada gen. Para extraer información más abstracta de estas características, se aplica una convolución unidimensional a las representaciones de los genes. Esta convolución ayuda a reducir la dimensionalidad de las características para quedarse con las más importantes. Al igual que en ScGPT, se utiliza una red neuronal de tres capas para la clasificación y la entropía cruzada categórica como función de pérdida con las etiquetas del TCCA.

De igual forma que para ScGPT, en el Anexo A se muestran las gráficas de pérdida de entrenamiento y validación para este modelo.

3.3. Métricas de validación

Estos modelos utilizan como métricas de validación las técnicas clásicas del aprendizaje automático, siendo estas la exactitud (*accuracy*), la precisión (*precision*), la sensibilidad (*recall*) y el valor F1.

La exactitud mide la proporción de predicciones correctas sobre el total de casos evaluados, ofreciendo una visión general de cuántas predicciones son correctas en total. Se calcula sumando los verdaderos positivos (VP) y los verdaderos negativos (VN), y dividiendo el resultado entre el número total de muestras:

$$\text{Exactitud} = \frac{\text{VP} + \text{VN}}{\text{VP} + \text{VN} + \text{FP} + \text{FN}} \tag{3.6}$$

Donde FP son los falsos positivos (casos negativos clasificados incorrectamente como positivos) mientras que FN son los falsos negativos (casos positivos clasificados incorrectamente como negativos). VP son Verdaderos Positivos, es decir, número de casos positivos correctamente clasificados y VN Verdaderos Negativos, número de casos negativos correctamente clasificados.

La precisión evalúa la exactitud de las predicciones positivas realizadas por el modelo. Indica la proporción de casos predichos como positivos que realmente son positivos:

$$\text{Precisión} = \frac{\text{VP}}{\text{VP} + \text{FP}} \tag{3.7}$$

La sensibilidad, mide la capacidad del modelo para identificar todos los casos positivos reales. Representa la proporción de casos positivos que el modelo logra detectar correctamente:

$$\text{Sensibilidad} = \frac{\text{VP}}{\text{VP} + \text{FN}} \tag{3.8}$$

El valor F1 es la media armónica de la precisión y la sensibilidad. Combina ambas métricas en una sola medida para evaluar el equilibrio entre la exactitud de las predicciones positivas y la capacidad de capturar todos los casos positivos. Esta métrica es especialmente útil cuando existe un desequilibrio en las clases o cuando es importante equilibrar la precisión y la sensibilidad:

$$\text{Valor F1} = 2 \times \frac{\text{Precisión} \times \text{Sensibilidad}}{\text{Precisión} + \text{Sensibilidad}} \tag{3.9}$$

Capítulo 4

Resultados

4.1. Clasificación

Utilizando los conjuntos de datos descritos anteriormente, se evaluaron los modelos en tres escenarios distintos para analizar su rendimiento en diferentes contextos. En el primer escenario, se utilizaron todas las células del conjunto de datos para obtener una visión global de la capacidad de los modelos para identificar los tipos celulares. En el segundo escenario, se excluyeron las células tumorales con el fin de evaluar la precisión en la clasificación de células no tumorales. Finalmente, en el tercer escenario, se trabajó exclusivamente con células tumorales para determinar si los modelos son capaces de diferenciar entre células según el tipo de tumor al que pertenecen.

Los resultados obtenidos por estos modelos también se presentan en tres gráficas: la primera es la matriz de confusión, para representar la precisión obtenida en cada clase. La segunda es un UMAP que agrupa las predicciones por tipos celulares, además también se proporciona un UMAP de las etiquetas originales. Por último, la tercera gráfica representa los aciertos en verde y los errores en rojo, para facilitar la comparación entre el UMAP original y el UMAP de predicciones. Todos los resultados se han obtenido con sus respectivos *datasets* de test.

4.1.1. Clasificación celular del TCCA

En el primer escenario, donde se entrenaron los modelos con todas las clases del TCCA, ScGPT mostró un rendimiento superior en comparación con ScBERT. En la tabla 4.1 se pueden ver los resultados numéricos. Concretamente, ScGPT alcanzó una exactitud de 0.9014, superando ligeramente a ScBERT, que obtuvo 0.8893. En términos de precisión, ScGPT logró 0.7697, mientras que ScBERT un 0.7345. La sensibilidad también fue mayor para ScGPT (0.7567) en comparación con scBERT (0.6886). El F1-Score de ScGPT fue de 0.7525, superando el 0.6894 de ScBERT.

Las figuras 4.1-4.4 representan los resultados obtenidos por scGPT y las figuras 4.5-4.8 los resultados de SCBERT representando en ambos casos el UMAP de las etiquetas reales, el UMAP de las etiquetas, el UMAP de aciertos y errores y la matriz de confusión de cada modelo.

Como era de esperar observando las métricas, la comparación de UMAPs entre las

Modelo	Exactitud	Precisión	Sensibilidad	F1-Score
scGPT	0.9014	0.7697	0.7567	0.7525
scBERT	0.8893	0.7345	0.6886	0.6894

Tabla 4.1: Comparación de métricas de rendimiento entre scGPT y scBERT para el TCCA completo.

etiquetas verdaderas y las predichas son muy similares; por este motivo, para facilitar la comprensión, se han añadido los UMAPs de aciertos y errores. Analizando estas figuras se aprecia que ambos modelos siguen el mismo patrón de aciertos y errores, a pesar de que estos últimos esten esparcidos alrededor de todas las clases, ScBERT muestra un color más intenso en ellos que ScGPT, confirmando lo que se ve en las métricas. Respecto a las matrices de confusión, ambos modelos muestran un bajo rendimiento en la clase 'Stem', probablemente debido a que este tipo de células son células pluripotentes capaces de convertirse en cualquier célula, por lo que no tiene un tipo definido. También es destacable que ScBERT tiene un rendimiento menor en clases parecidas, como pueden ser los distintos tipos de células T('CD8+ T-cell', 'CD4+ T-cell', 'Unconventional T-cells') y las células que pertenecen al cerebro/sistema nervioso ('Glial', 'Neuron').

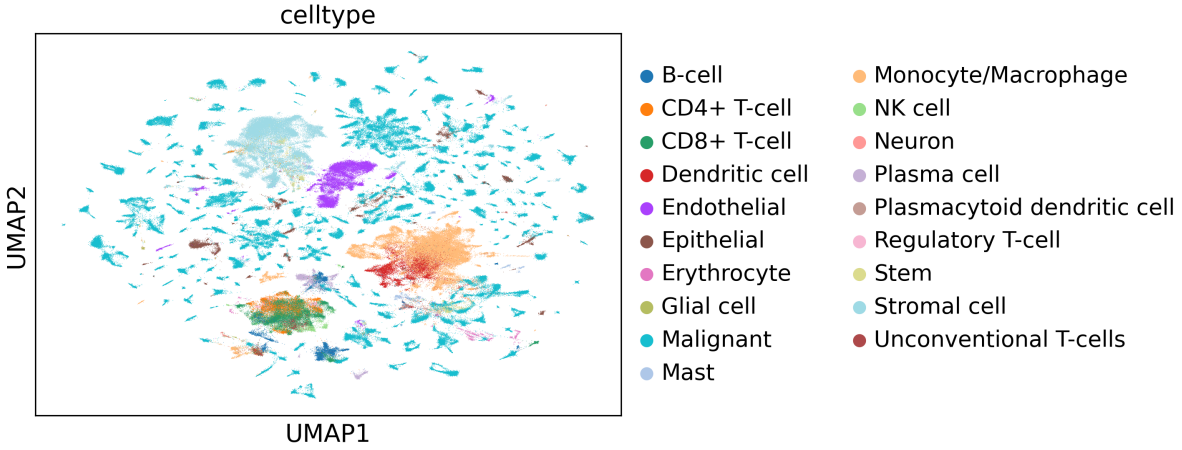

Figura 4.1: UMAP de ScGPT con las etiquetas verdaderas del TCCA.

4.1.2. Clasificación celular sin células malignas

En este caso, en el que se han eliminado las células malignas para comprobar la eficacia del modelo en células sanas, los resultados se pueden ver en la tabla 4.2 y en las figuras A.6, 4.10, 4.11 y 4.12 para ScGPT y en las figuras 4.13, 4.14, 4.15 y 4.16 para ScBERT.

Nuevamente, ScGPT mostró un rendimiento superior. La exactitud de ScGPT fue de 0.8795, comparada con 0.8496 de ScBERT. La precisión y sensibilidad de ScGPT fueron 0.7900 y 0.7975 respectivamente, mientras que ScBERT obtuvo 0.7582 en precisión y 0.7304 en sensibilidad. El F1-Score de ScGPT fue de 0.7825, superando el 0.7172 de ScBERT.

Siendo destacable que los modelos conserven buenas métricas a través de las células del ambiente tumoral, las gráficas de los modelos para este caso dan prácticamen-

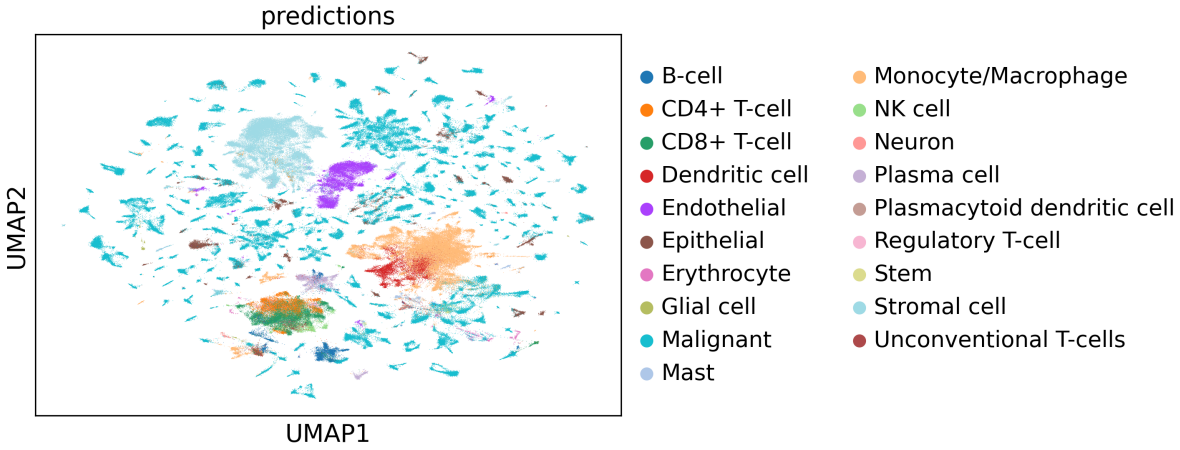

Figura 4.2: UMAP de ScGPT con las etiquetas predichas para el TCCA.

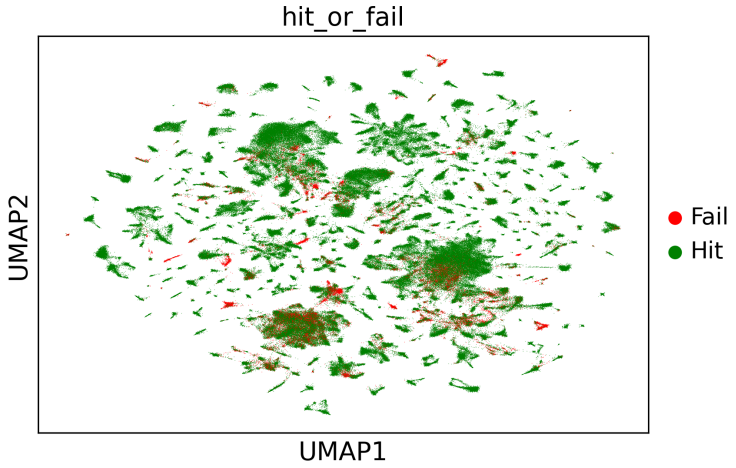

Figura 4.3: Errores y aciertos de ScGPT con el TCCA.

te la misma información que para el primer caso, siendo los UMAPs de aciertos y errores son muy parecidos entre los dos modelos. Las matrices de confusión también recuerdan a las del primer escenario, con un rendimiento ligeramente superior en ambos modelos, aunque ScGPT es algo más consistente a través de todas las clases mientras que ScBERT sigue teniendo más problemas identificando clases similares.

Modelo	**Exactitud**	**Precisión**	**Sensibilidad**	**F1-Score**
scGPT	0.8795	0.7900	0.7975	0.7825
scBERT	0.8496	0.7582	0.7304	0.7172

Tabla 4.2: Comparación de métricas de rendimiento entre scGPT y scBERT sin células malignas.

4.1.3. Clasificación de células por tipo tumoral

En el tercer escenario, los modelos se entrenaron para clasificar únicamente las células malignas según su tipo tumoral. Ambos modelos mostraron altos niveles de

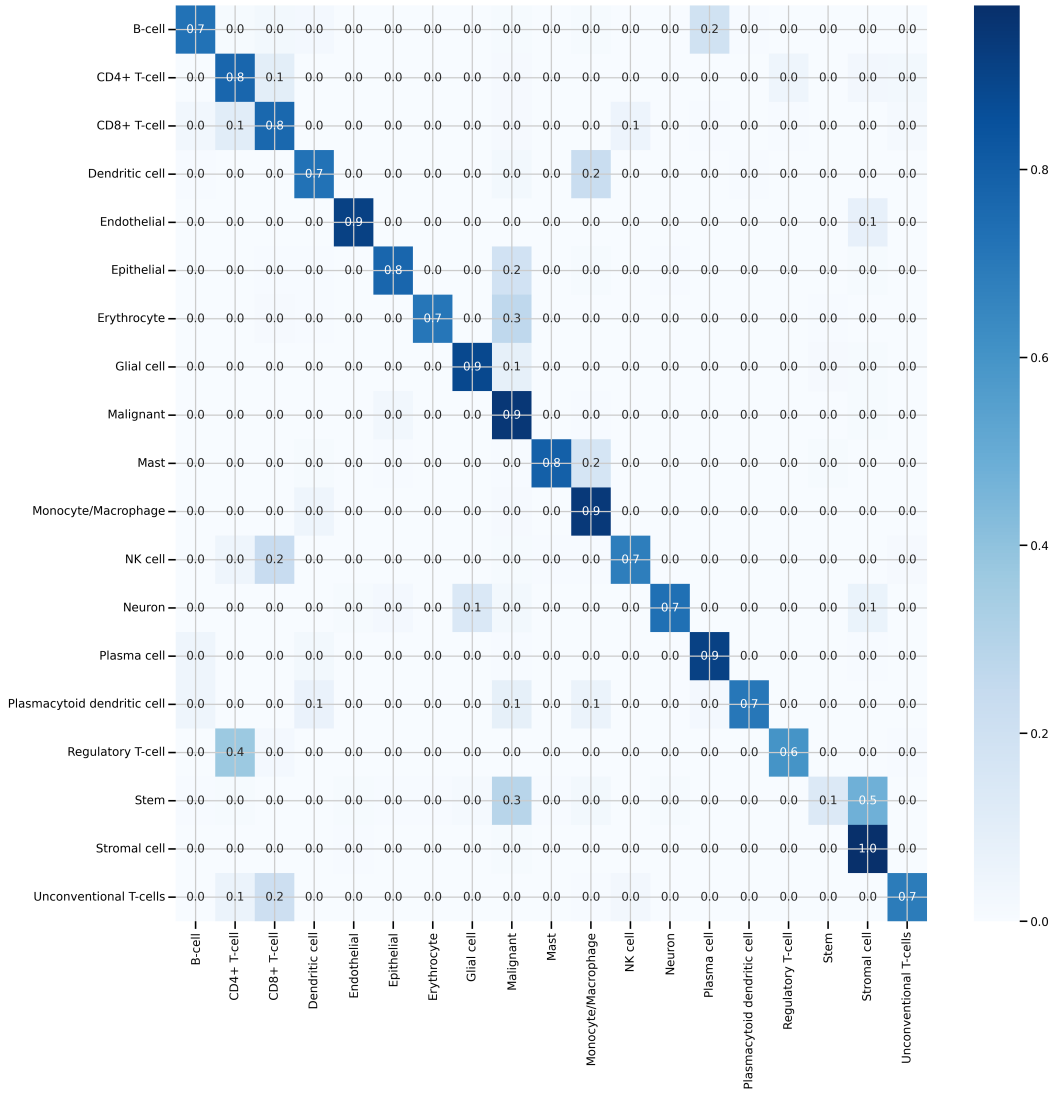

Figura 4.4: Matriz de confusión de ScGPT para el TCCA.

rendimiento, de hecho, este caso es en el que los modelos tuvieron un rendimiento más parecido, aunque ScGPT siguió obteniendo unos valores ligeramente más altos. En la tabla 4.3 se ve como la exactitud de ScGPT fue de 0.9351, mientras que ScBERT obtuvo 0.9337. La precisión y sensibilidad de scGPT fueron 0.8733 y 0.8767 respectivamente, en comparación con 0.8500 y 0.8462 de ScBERT. El F1-Score también fue superior para scGPT, con 0.8577 frente a 0.8086 de scBERT.

Modelo	**Exactitud**	**Precisión**	**Sensibilidad**	**F1-Score**
ScGPT	0.9351	0.8733	0.8767	0.8577
ScBERT	0.9337	0.8500	0.8462	0.8086

Tabla 4.3: Comparación de métricas de rendimiento entre scGPT y scBERT para clasificación por tipo tumoral.

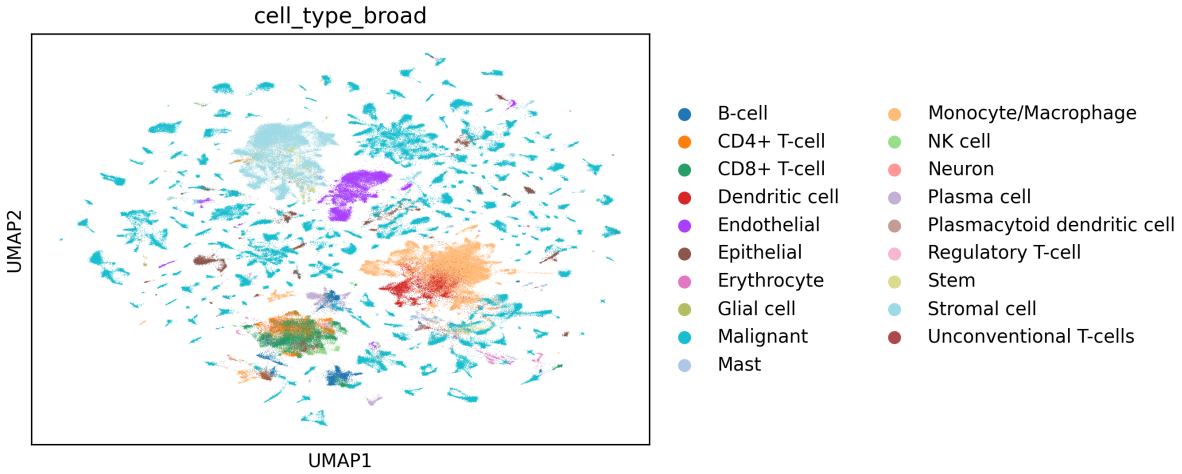

Figura 4.5: UMAP de ScBERT con las etiquetas verdaderas del TCCA.

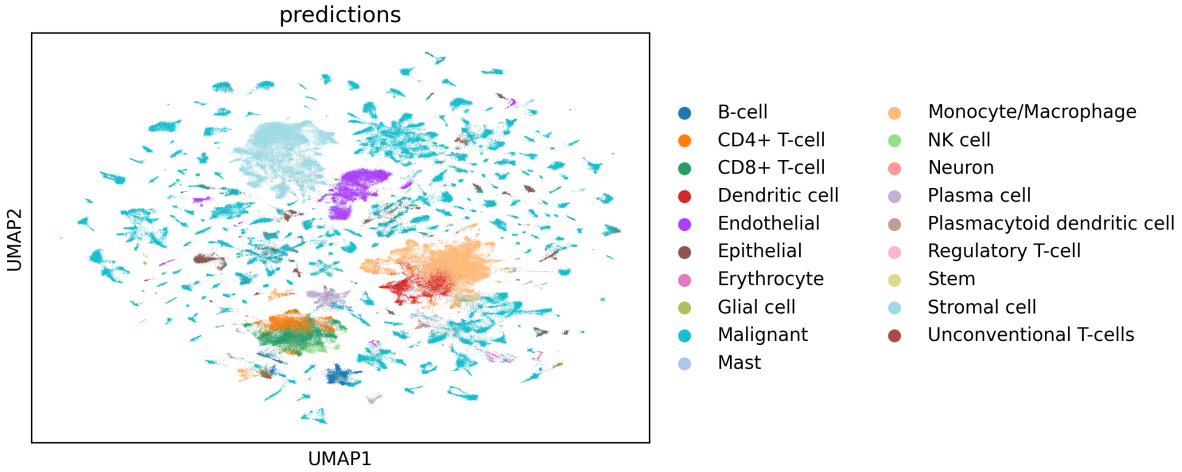

Figura 4.6: UMAP de ScBERT con las etiquetas predichas con el TCCA.

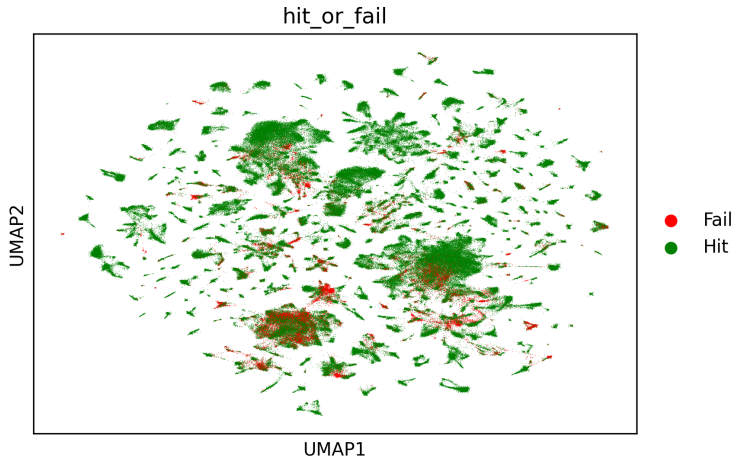

Figura 4.7: Errores y aciertos de ScBERT con el TCCA.

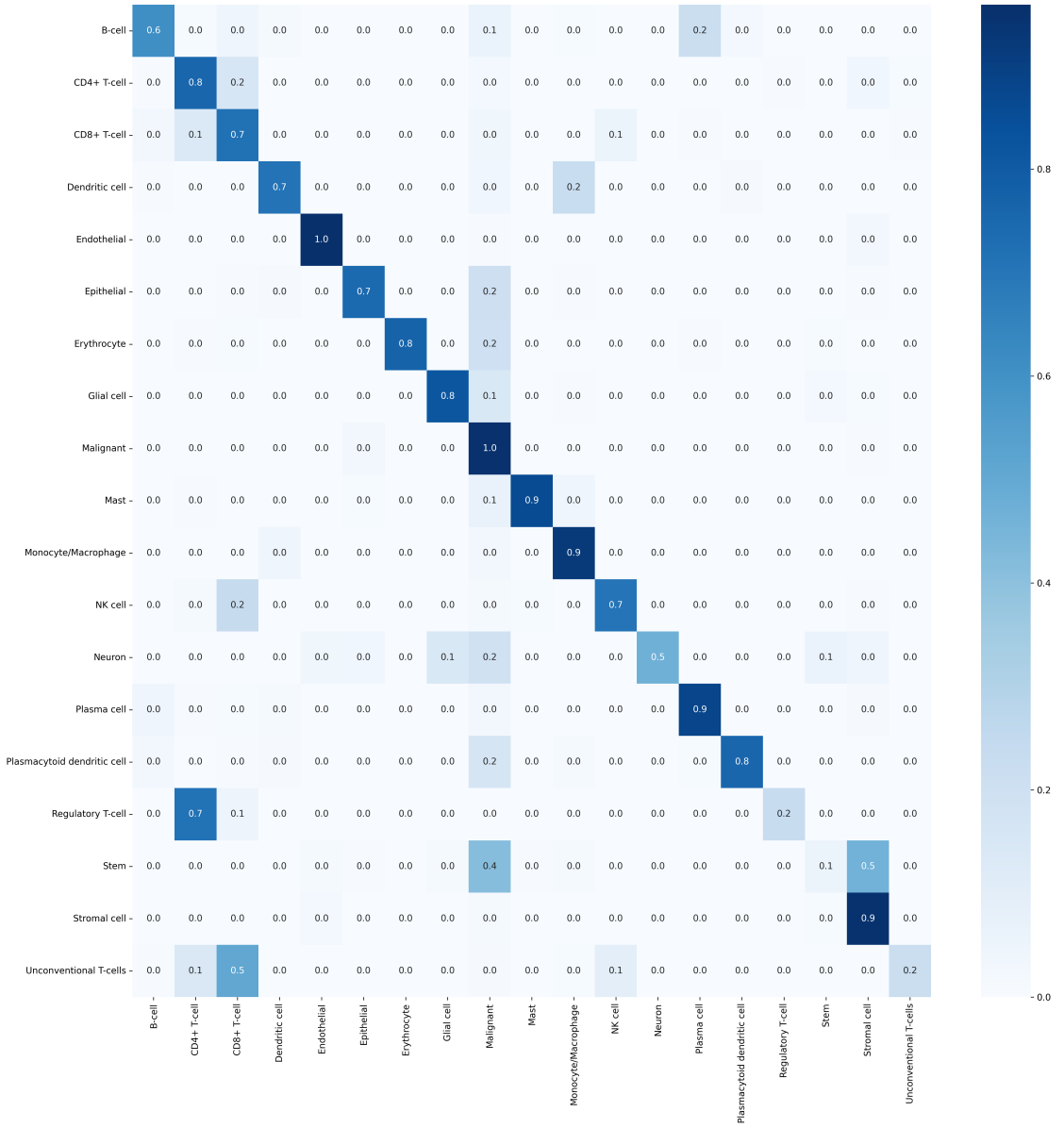

Figura 4.8: Matriz de confusión de ScBERT para el TCCA.

En este caso los resultados se corresponden a las figuras 4.17, 4.18, 4.19 y 4.20 para ScGPT y 4.21, 4.22, 4.23 y 4.24 para ScBERT. En las figuras de aciertos y errores vuelve a suceder que los dos modelos muestran un patrón parecido. Para este caso, las matrices de confusión son casi idénticas, donde los errores de los dos modelos destacan en tipos tumorales similares, confundiendo sobretodo la clase 'NSCLC' con 'LUAD' (ambos de cáncer de pulmón), y la clase 'OGD' con 'GBM' (dos cánceres de cerebro).

4.2. Comparación con técnicas clásicas

Resulta interesante estudiar qué aportan estos modelos Transformers frente a los métodos automáticos ya existentes que funcionan por mapeo por referencia, de for-

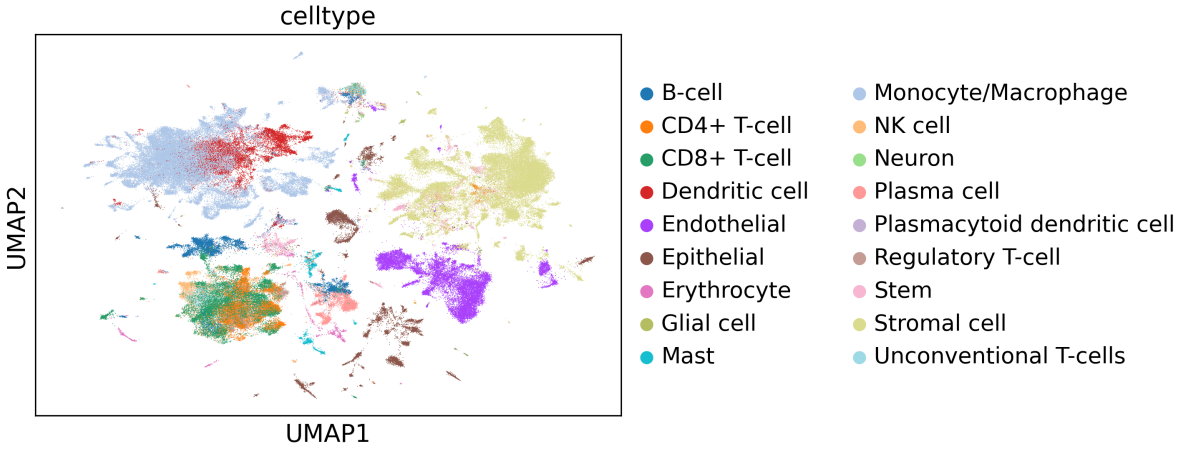

Figura 4.9: UMAP de ScGPT con las etiquetas verdaderas sin células malignas.

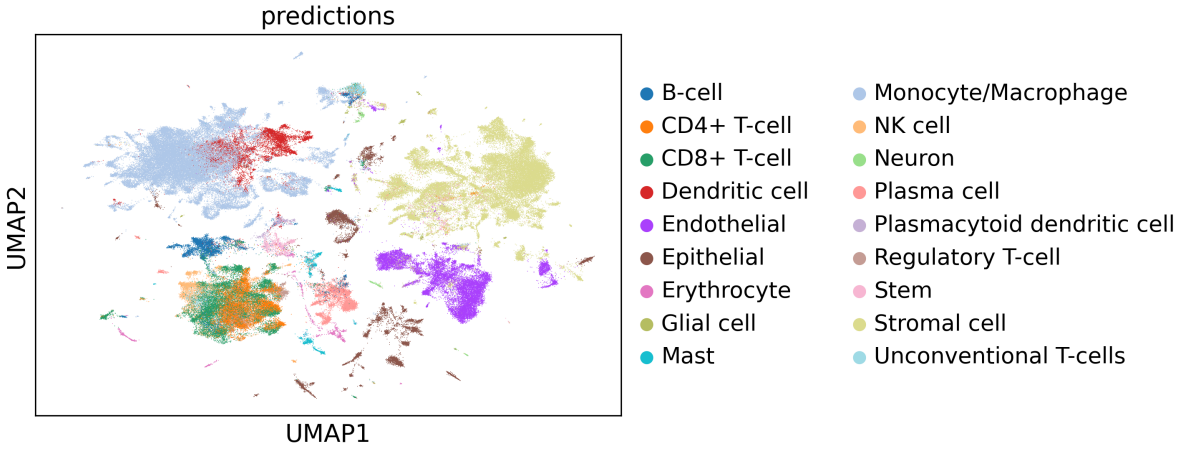

Figura 4.10: UMAP de ScGPT con las etiquetas predichas sin células malignas.

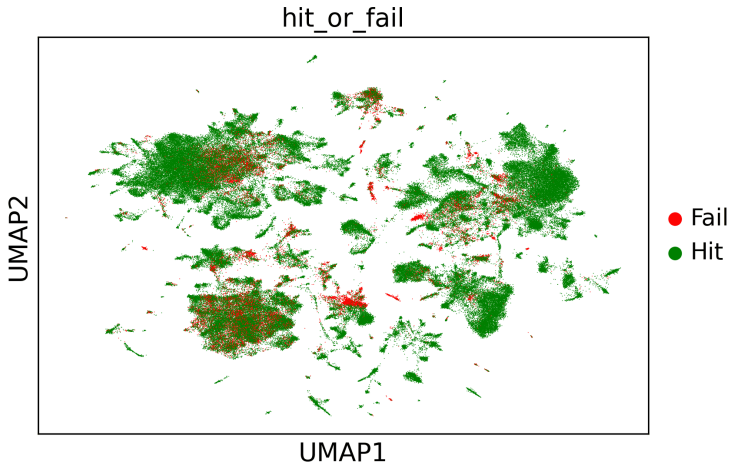

Figura 4.11: Errores y aciertos de ScGPT sin células malignas.

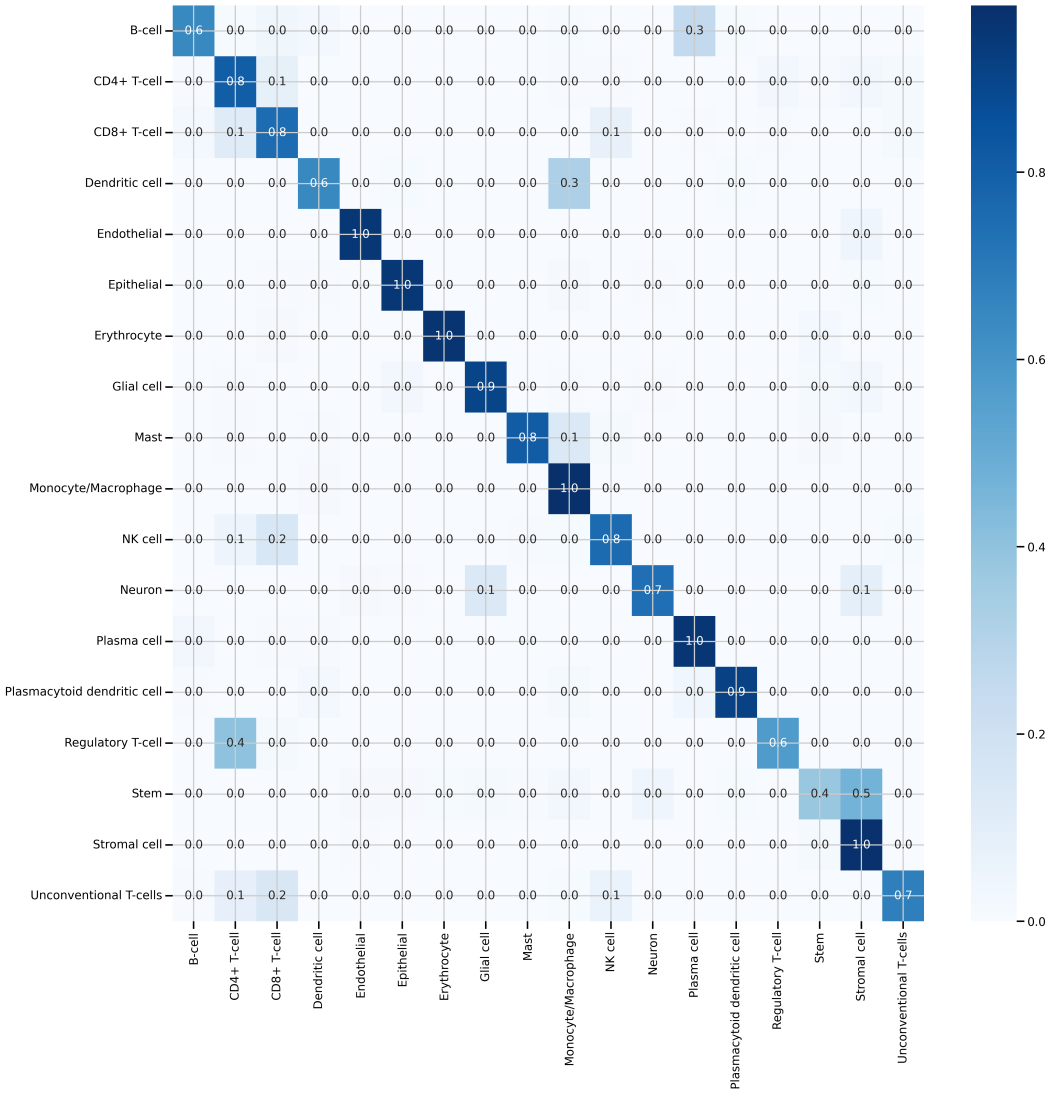

Figura 4.12: Matriz de confusión de ScGPT sin células malignas.

ma que mapean las células a un tipo ya aprendido que se encuentre la referencia utilizada.

Para realizar esta comparativa de forma equivalente, se consideraron las células no malignas del TCCA que corresponden al ambiente de los tipos tumorales de pulmón, ya que es el tipo tumoral más grande que contiene las anotaciones originales de los autores de los estudios, anotadas manualmente. Se hizo una división para el entrenamiento y validación similar a la realizada para los dos primeros casos de clasificación celular (con el 50 % de los pacientes con más células para el entrenamiento y el 50 % restante para validación), dejando 122.898 células para el entrenamiento y 18.621 para la validación. Por otro lado, el método de anotación automática escogido es Azimuth [45], con el que se utilizó la referencia por defecto de pulmón utilizada para anotar parte del TCCA. Azimuth es una herramienta basada en aprendizaje automá-

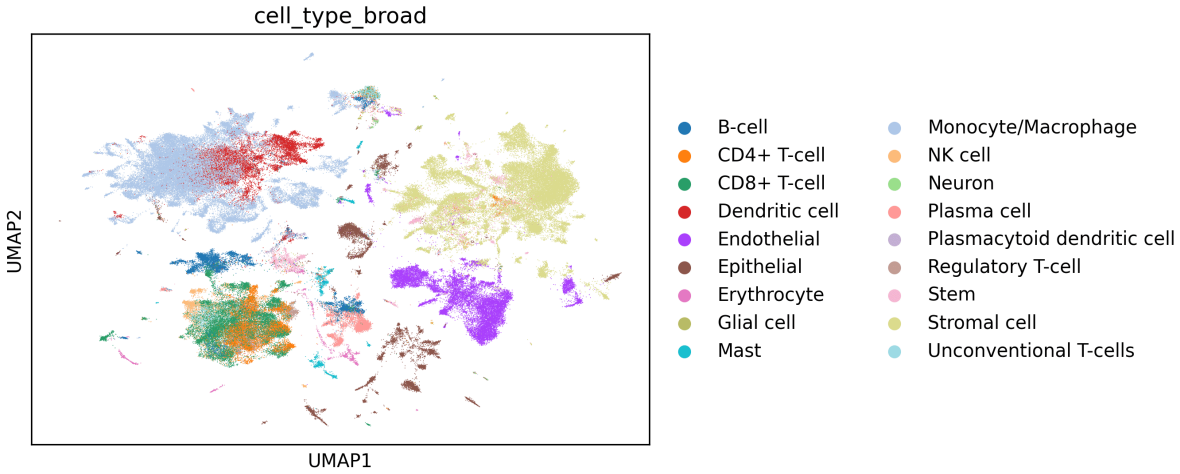

Figura 4.13: UMAP de ScBERT con las etiquetas verdaderas sin células malignas.

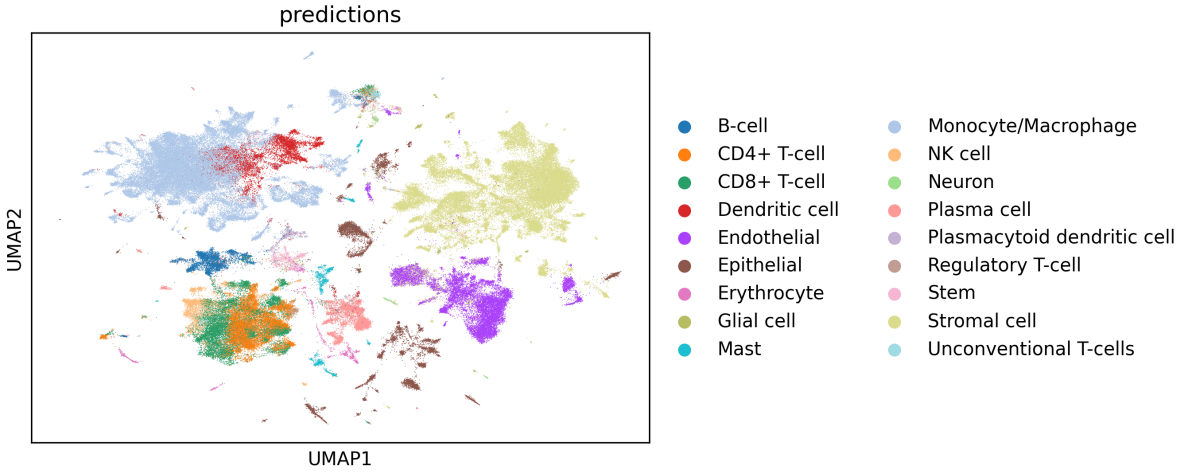

Figura 4.14: UMAP de ScBERT con las etiquetas predichas sin células malignas.

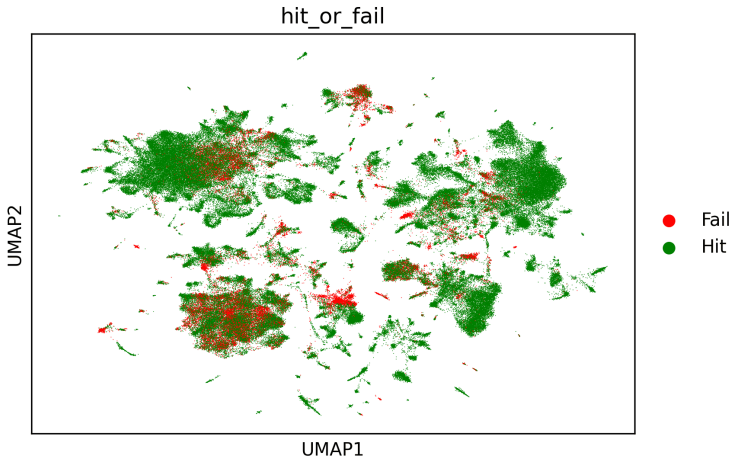

Figura 4.15: Errores y aciertos de ScBERT sin células malignas.

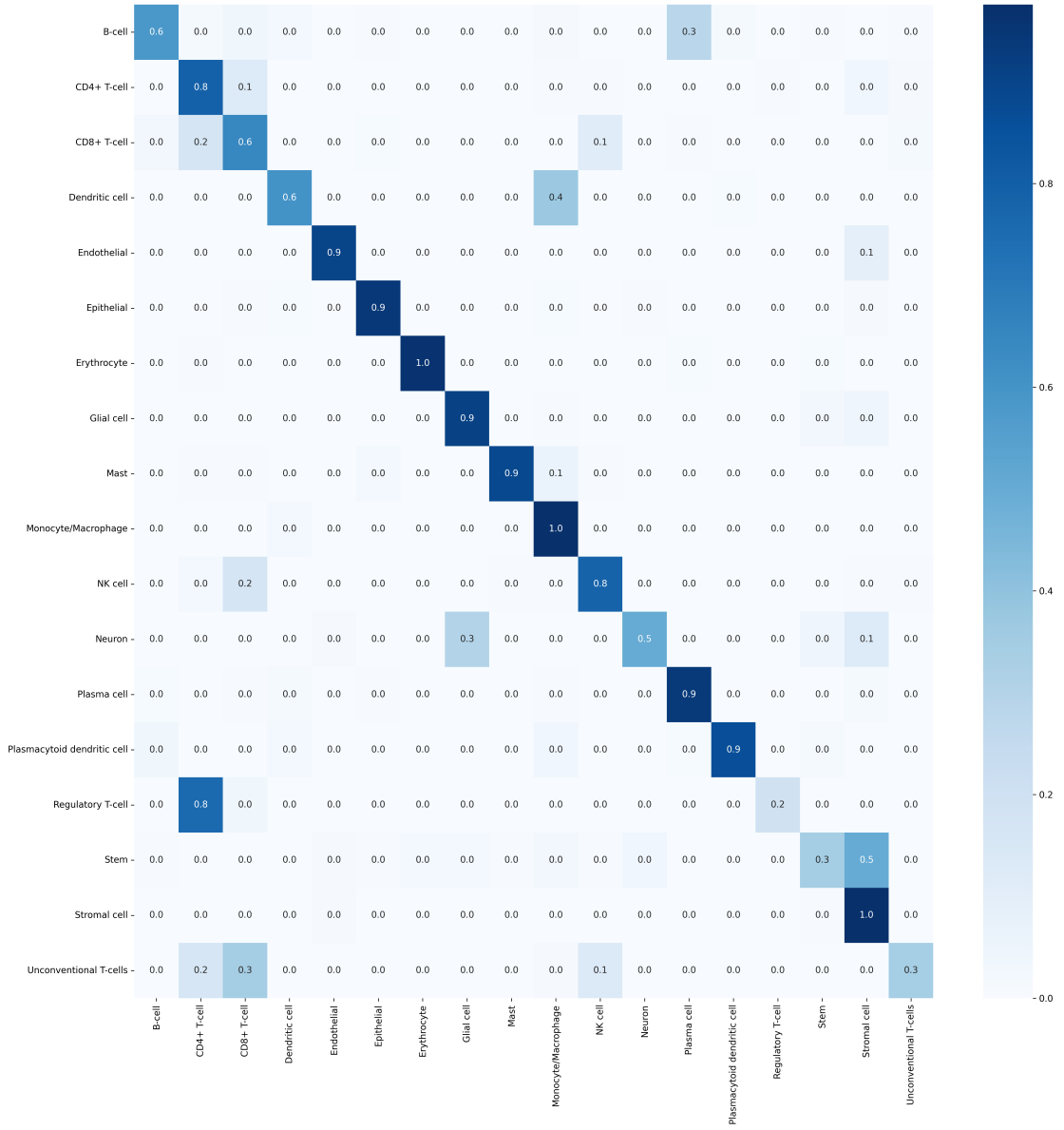

Figura 4.16: Matriz de confusión de ScBERT sin células malignas.

tico para la anotación automática de tipos celulares en datos de single-cell RNA-seq. Utiliza un modelo de referencia preentrenado con datos de células anotadas, permitiendo predecir tipos celulares en nuevos conjuntos de datos. Esta herramienta facilita una anotación eficiente y precisa, siendo muy utilizada en análisis *single-cell*. Para que la comparativa fuese justa, Azimuth se utilizó en las células del conjunto de datos de validación. Los resultados de esta comparativa se pueden ver en la tabla 4.4.

Observando los resultados de la tabla, se ve que ScGPT supera claramente a Azimuth en la clasificación celular usando datos de scRNA-seq. ScGPT consigue una exactitud de 0.8845, mientras que Azimuth solo alcanza 0.6043. La precisión y la sensibilidad de scGPT son casi el doble que las de Azimuth, lo que indica que funciona mucho mejor identificando correctamente las células tumorales y reduciendo errores en la

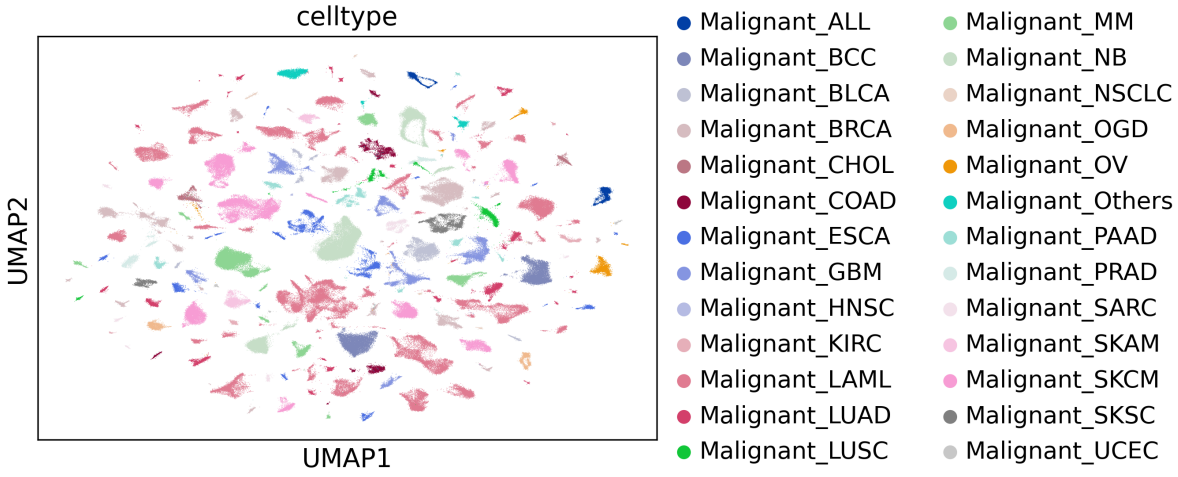

Figura 4.17: UMAP de ScGPT con las etiquetas verdaderas para clasificación por tipo tumoral.

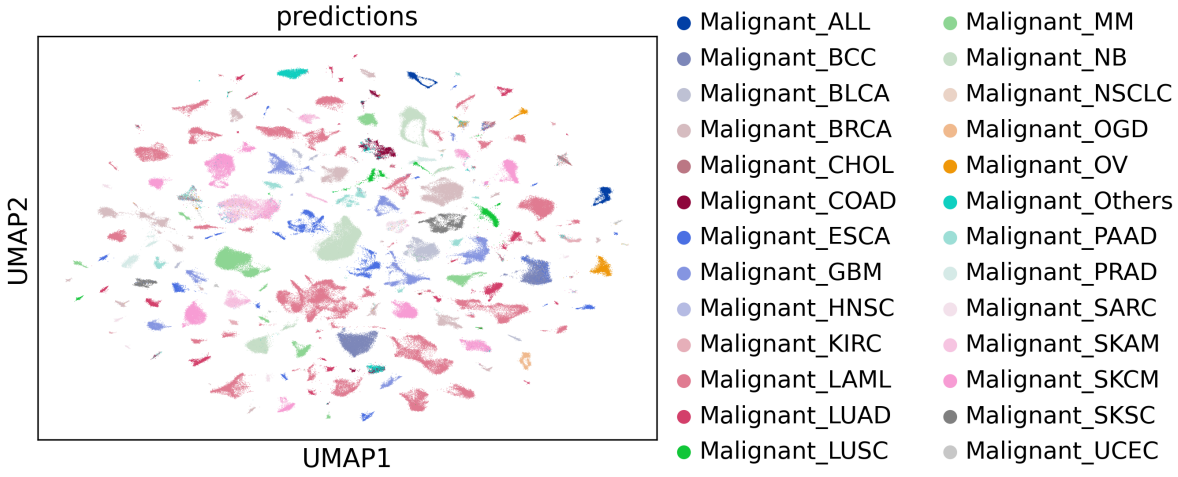

Figura 4.18: UMAP de ScGPT con las etiquetas predichas para clasificación por tipo tumoral.

Modelo	**Exactitud**	**Precisión**	**Sensibilidad**	**F1-Score**
scGPT	0.8845	0.5877	0.5967	0.5805
Azimuth	0.6043	0.2837	0.2904	0.2804

Tabla 4.4: Comparación de métricas de rendimiento entre scGPT y Azimuth para clasificación por tipo tumoral.

clasificación. El F1-Score también es más alto en ScGPT, mostrando un mejor equilibrio entre precisión y sensibilidad. Además, en términos de velocidad, ScGPT demostró ser significativamente más rápido; mientras Azimuth anota alrededor de 10,000 células por minuto, scGPT fue capaz de anotar todas las 18.621 células en 6.9 segundos, resaltando no solo la superioridad de scGPT en precisión, sino también su eficiencia en el procesamiento de datos a gran escala.

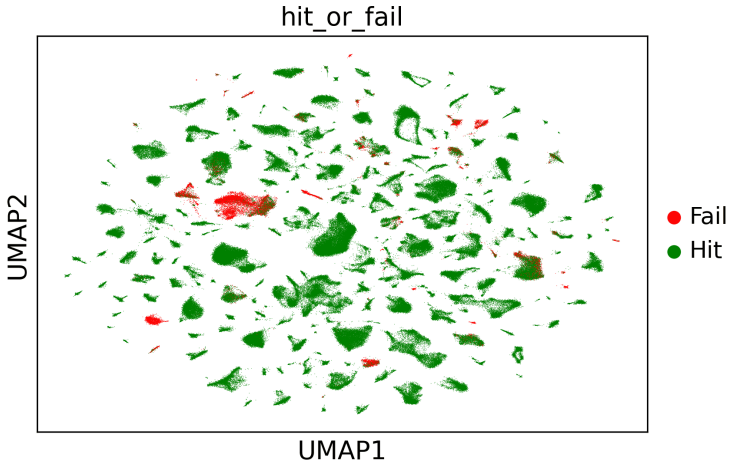

Figura 4.19: Errores y aciertos de ScGPT para clasificación por tipo tumoral.

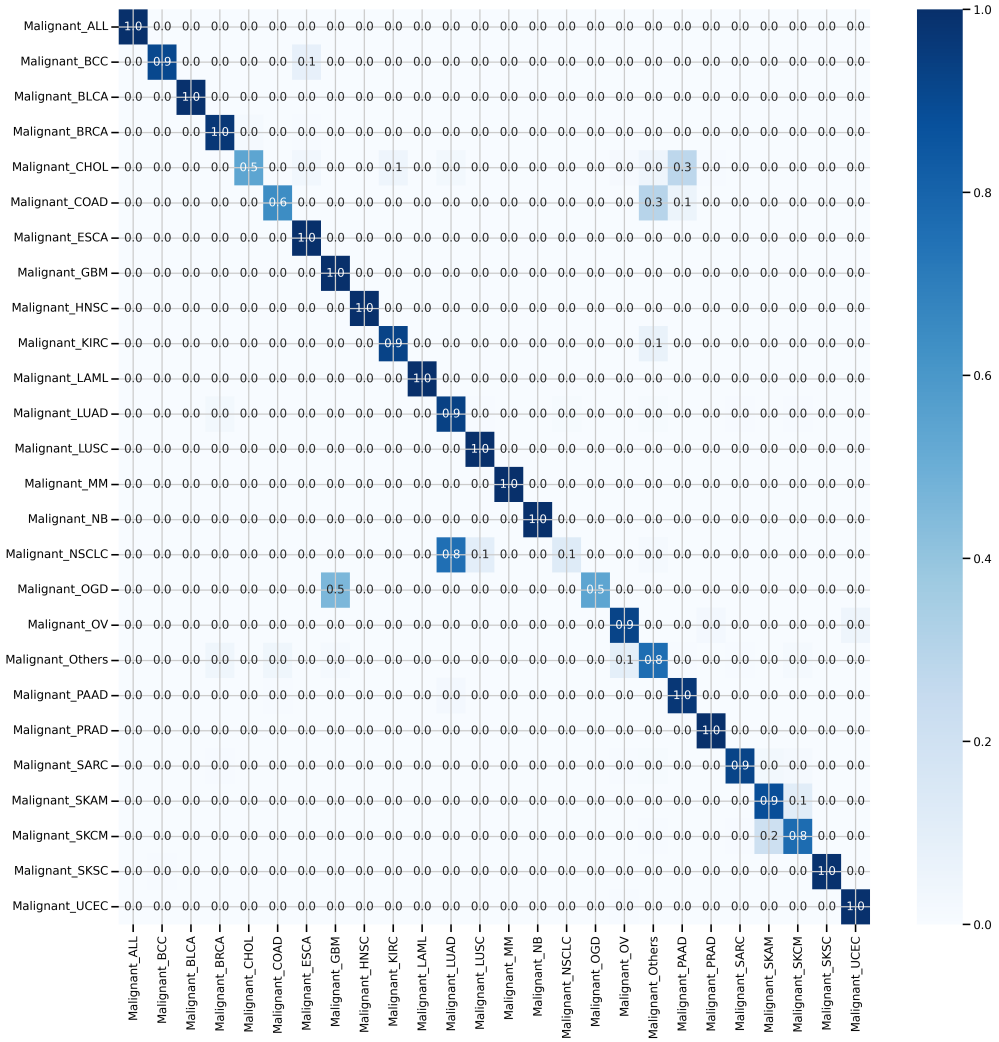

Figura 4.20: Matriz de confusión de ScGPT para clasificación por tipo tumoral.

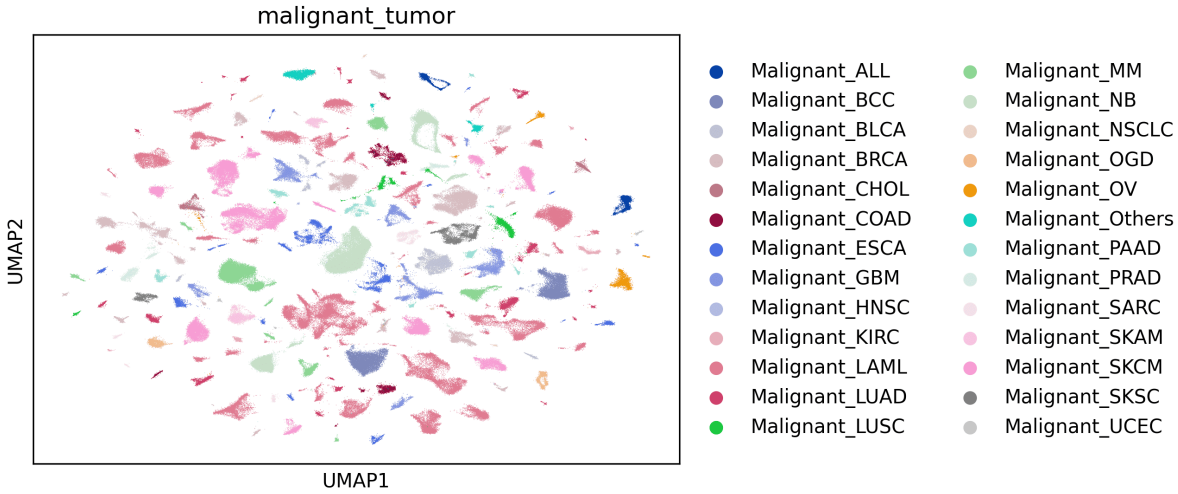

Figura 4.21: UMAP de ScBERT con las etiquetas verdaderas para clasificación por tipo tumoral.

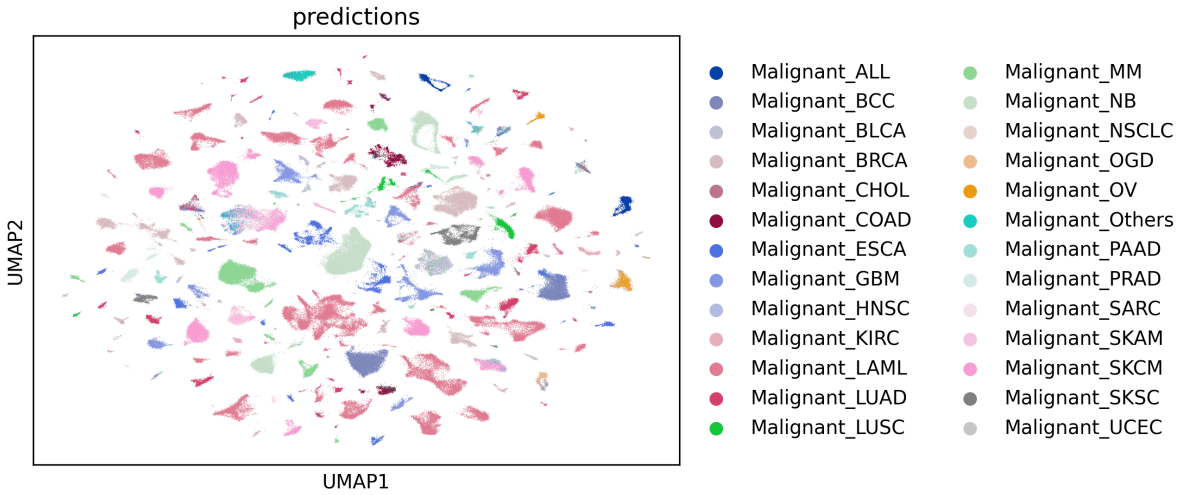

Figura 4.22: UMAP de ScBERT con las etiquetas predichas para clasificación por tipo tumoral.

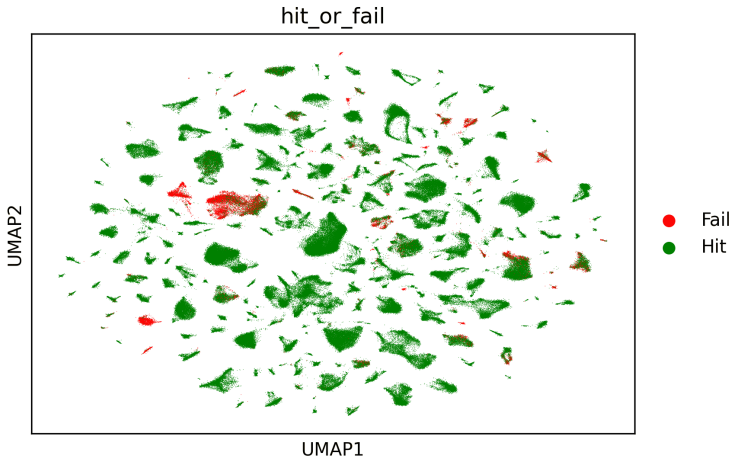

Figura 4.23: Errores y aciertos de ScBERT para clasificación por tipo tumoral.

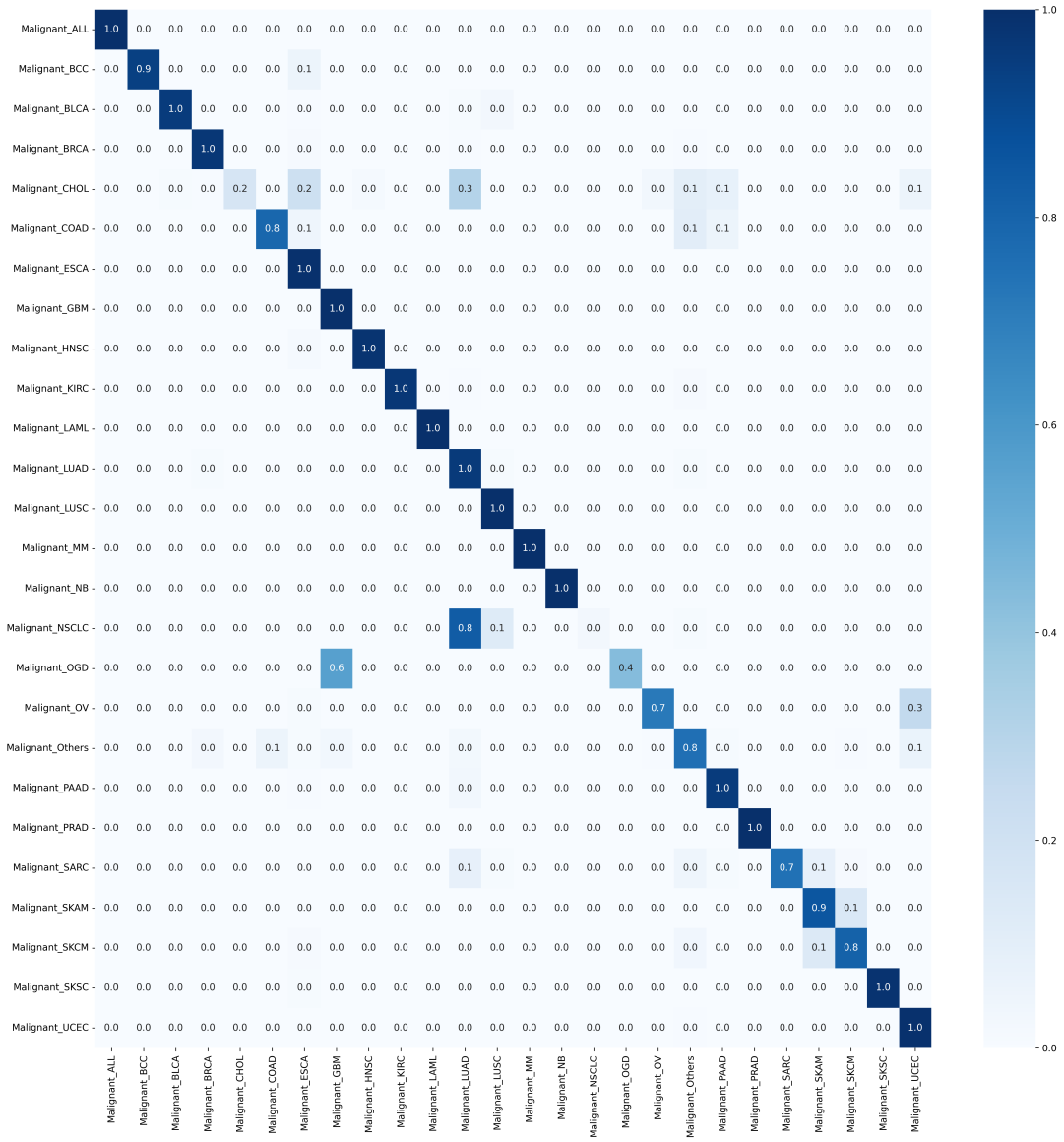

Figura 4.24: Matriz de confusión de ScBERT para clasificación por tipo tumoral.

Capítulo 5

Conclusiones y líneas futuras

5.1. Conclusiones

En este trabajo de fin de master se ha realizado un *benchmarking* de dos tipos de modelos Transformer de anotación celular para un dataset scRNA-seq pan-cancer. Este *benchmarking* se ha llevado a cabo evaluando los modelos en tres escenarios distintos: clasificando todos los tipos celulares del dataset, clasificando únicamente células sanas y por último clasificando únicamente células malignas por tipo tumoral. Además, se ha comparado también ScGPT, el que ha demostrado mejor rendimiento, con uno de los métodos automáticos ya existentes más conocidos para anotación celular.

Los resultados obtenidos muestran un gran rendimiento en los dos modelos, muy similar en las tres situaciones evaluados. A pesar de que ScGPT es ligeramente superior y más consistente en las tres, es interesante cómo ambos modelos clasifican de forma muy parecida, significando que las representaciones que forman de las células puede llegar a ser igual a pesar de las diferencias en los modelos.

La comparación de ScGPT con Azimuth muestra cómo estos modelos son superiores a las técnicas tradicionales de anotación. Estos resultados demuestran que los modelos Transformers consiguen mejores resultados en este tipo de tareas. Una razón es que pueden manejar datos muy complejos y con muchas variables, como los perfiles de expresión génica de células individuales. Los Transformers pueden detectar patrones sutiles y relaciones no lineales en los datos que los métodos tradicionales pueden pasar por alto. Además, tienen la capacidad de aprender automáticamente qué características son más importantes para la clasificación, sin necesidad de una intervención manual extensa.

Por último, el hecho de que las representaciones celulares aprendidas por este tipo de modelos sean tan útiles tanto para anotar tipos celulares como para clasificar células según su tipo tumoral sugiere que pueden ser de gran ayuda para entender mejor la biología y enfermedades complejas. Estas representaciones permiten capturar patrones y relaciones en los datos de expresión génica que podrían pasar desapercibidos con métodos tradicionales. Comprender y representar de esta forma los distintos tipos de células que existen significa que es posible que puedan aprender cómo interaccionan los genes y las células entre sí o entre fármacos o cualquier tipo de perturbación que puedan sufrir las células, abriendo oportunidades a explorar

nuevas vías terapéuticas y comprender mejor las comunicaciones celulares.

5.2. Líneas futuras de investigación

El trabajo realizado abre varias oportunidades para continuar explorando este tipo de modelos, tanto usando el mismo conjunto de datos como aplicándolo a nuevos. A partir de aquí, hay varias direcciones interesantes en las que se podría profundizar. Por ejemplo, se pueden seguir explorando las funcionalidades de los modelos, como la capacidad de integrar diferentes tipos de datos, o incluso predecir cómo las células responderán a perturbaciones. Esto no solo ampliaría el alcance de los modelos, sino que también podría mejorar la precisión y la robustez de las predicciones.

Otro camino que podría ser interesante es evaluar cómo se comportan los modelos al incorporar más capas de datos ómicos, como metilación del ADN, información proteómica o metabolómica. Además, podría ser útil integrar datos de otras modalidades, como imágenes médicas o incluso texto generado por expertos, para ver si los modelos logran una representación más completa de la biología subyacente. Esto podría abrir puertas a nuevas funcionalidades, como predecir respuestas terapéuticas basadas en un enfoque multi-modal.

Por último, también se podrían aplicar estos modelos a tareas más específicas, como la clasificación de subtipos de cáncer o la identificación de biomarcadores predictivos para la respuesta a tratamientos, lo que podría tener un gran impacto en la medicina personalizada.

Bibliografía

[1] Cáncer [Online]. Disponible en: https://www.cancer.gov/espanol/cancer/naturaleza/que-es

[2] Organización Mundial de la Salud (OMS). [Online]. Disponible en: https://www.who.int/es/news-room/fact-sheets/detail/cancer

[3] Cifras del cáncer [Online]. Disponible en: https://seom.org/images/publicaciones/informes-seom-de-evaluacion-de-farmacos/LAS_CIFRAS_2024.pdf

[4] Genes [Online]. Disponible en: https://www.cancer.org/es/cancer/entendimiento-del-cancer/genetica-y-cancer/cambios-geneticos-y-cancer.html

[5] Genética [Online]. Disponible en: https://www.cancer.gov/espanol/cancer/causas-prevencion/genetica

[6] Burrell, R., McGranahan, N., Bartek, J. et al. The causes and consequences of genetic heterogeneity in cancer evolution. *Nature 501, 338–345 (2013). https://doi.org/10.1038/nature12625*

[7] Ramón Y Cajal, S., Sesé, M., Capdevila, C., Aasen, T., De Mattos-Arruda, L., Diaz-Cano, S. J., Hernández-Losa, J., & Castellví, J. (2020). Clinical implications of intratumor heterogeneity: challenges and opportunities. *Journal of Molecular Medicine (Berlin, Germany)*, *98*(2), 161–177. https://doi.org/10.1007/s00109-020-01874-2

[8] Marusyk, A., Janiszewska, M., & Polyak, K. (2020). Intratumor Heterogeneity: The Rosetta Stone of Therapy Resistance. *Cancer Cell*, *37*(4), 471–484. https://doi.org/10.1016/j.ccell.2020.03.007

[9] Medicina personalizada [Online]. Disponible en: https://www.genome.gov/es/genetics-glossary/Medicina-personalizada

[10] Ramaswami, R., Bayer, R., & Galea, S. (2018). Precision Medicine from a Public Health Perspective. *Annual Review of Public Health*, *39*, 153–168. https://doi.org/10.1146/annurev-publhealth-040617-014158

[11] Tang, F., Barbacioru, C., Wang, Y., Nordman, E., Lee, C., Xu, N., Wang, X., Bodeau, J., Tuch, B. B., Siddiqui, A., Lao, K., & Surani, M. A. (2009). mRNA-Seq whole-transcriptome analysis of a single cell. *Nature Methods*, *6*(5), 377–382. https://doi.org/10.1038/nmeth.1315

[12] Wolf, F., Angerer, P., & Theis, F. (2018). SCANPY: large-scale single-cell gene expression data analysis. *Genome Biology*, *19*, 15. https://doi.org/10.1186/s13059-017-1382-0

[13] Hao, Y. *et al.* (2023). Dictionary learning for integrative, multimodal and scalable single-cell analysis. *Nature Biotechnology*. [Seurat V5]

[14] Heumos, L., Schaar, A. C., Lance, C., *et al.* (2023). Best practices for single-cell analysis across modalities. *Nature Reviews Genetics*, *24*, 550–572. https://doi.org/10.1038/s41576-023-00586-w

[15] Topol, E. J. (2019). High-performance medicine: the convergence of human and artificial intelligence. *Nature Medicine*, *25*, 44–56. https://doi.org/10.1038/s41591-018-0300-7

[16] Ho, T. K. (1995). Random decision forests. In *Proceedings of the 3rd International Conference on Document Analysis and Recognition* (pp. 278–282). Montreal, QC.

[17] Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, *20*, 273–297.

[18] Eid, F.-E., ElHefnawi, M., & Heath, L. S. (2015). DeNovo: virus–host sequence-based protein–protein interaction prediction. *Bioinformatics*, *32*(8), 1144–1150.

[19] Weis, C., *et al.* (2022). Direct antimicrobial resistance prediction from clinical MALDI-TOF mass spectra using machine learning. *Nature Medicine*, *28*, 164–174.

[20] Thomas, A. M., *et al.* (2019). Metagenomic analysis of colorectal cancer datasets identifies cross-cohort microbial diagnostic signatures and a link with choline degradation. *Nature Medicine*, *25*, 667–678.

[21] Sharma, A., Lysenko, A., Jia, S., Boroevich, K. A., & Tsunoda, T. (2024). Advances in AI and machine learning for predictive medicine. *Journal of Human Genetics*, *69*(10), 487–497. https://doi.org/10.1038/s10038-024-01231-y

[22] Fukushima, K. (1980). Neocognitron: a self organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, *36*, 193–202.

[23] Zhou, S. K., Greenspan, H., Davatzikos, C., Duncan, J. S., van Ginneken, B., Madabhushi, A., Prince, J. L., Rueckert, D., & Summers, R. M. (2021). A review of deep learning in medical imaging: Imaging traits, technology trends, case studies with progress highlights, and future promises. *Proceedings of the IEEE*, *109*(5), 820–838. https://doi.org/10.1109/JPROC.2021.3054390

[24] Wang, G., Zhang, Y., Ye, X., & Mou, X. (2019). *Machine learning for tomographic imaging*. IOP Publishing.

[25] Gaillochet, M., Tezcan, K. C., & Konukoglu, E. (2020). Joint reconstruction and bias field correction for undersampled MR imaging. In *International Conference on Medical Image Computing and Computer-Assisted Intervention* (pp. 44–52). Springer.

[26] Ji, Y., Lotfollahi, M., Wolf, F. A., & Theis, F. J. (2022). Machine learning for perturbational single-cell omics. *Cell Systems*, *12*(6), 522–537.

[27] Lotfollahi, M., Klimovskaia Susmelj, A., De Donno, C., Hetzel, L., Ji, Y., Ibarra, I. L., Srivatsan, S. R., Naghipourfar, M., Daza, R. M., Martin, B., Shendure, J., McFaline-Figueroa, J. L., Boyeau, P., Wolf, F. A., Yakubova, N., Günnemann, S., Trapnell, C., Lopez-Paz, D., & Theis, F. J. (2023). Predicting cellular responses to complex perturbations in high-throughput screens. *Molecular Systems Biology*, *19*(6), e11517. https://doi.org/10.15252/msb.202211517

[28] Lotfollahi, M., Wolf, F. A., & Theis, F. J. (2019). scGen predicts single-cell perturbation responses. *Nature Methods*, *16*, 715–721. https://doi.org/10.1038/s41592-019-0494-8

[29] Lobentanzer, S., Rodriguez-Mier, P., Bauer, S., & Saez-Rodriguez, J. (2024). Molecular causality in the advent of foundation models. *Molecular Systems Biology*, *20*(8), e848–858. https://doi.org/10.1038/s44320-024-00041-w

[30] Cui, H., Wang, C., Maan, H., *et al.* (2024). scGPT: toward building a foundation model for single-cell multi-omics using generative AI. *Nature Methods*, *21*, 1470–1480. https://doi.org/10.1038/s41592-024-02201-0

[31] Yang, F., Wang, W., Wang, F., *et al.* (2022). scBERT as a large-scale pretrained deep language model for cell type annotation of single-cell RNA-seq data. *Nature Machine Intelligence*, *4*, 852–866. https://doi.org/10.1038/s42256-022-00534-z

[32] Hao, M., Gong, J., Zeng, X., *et al.* (2024). Large-scale foundation model on single-cell transcriptomics. *Nature Methods*, *21*, 1481–1491. https://doi.org/10.1038/s41592-024-02305-7

[33] Neurona o perceptrón [Online]. Disponible en: `https://blog.josemarianoalvarez.com/2018/06/10/el-perceptron-como-neurona-artificial/`

[34] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*, 436–444. https://doi.org/10.1038/nature14539

[35] Rasamoelina, A. D., Adjailia, F., & Sinčák, P. (2020). A Review of Activation Function for Artificial Neural Network. In *2020 IEEE 18th World Symposium on Applied Machine Intelligence and Informatics (SAMI)* (pp. 281–286). Herlany, Slovakia. https://doi.org/10.1109/SAMI48414.2020.9108717

[36] Wang, Q., Ma, Y., Zhao, K., *et al.* (2022). A Comprehensive Survey of Loss Functions in Machine Learning. *Annals of Data Science*, *9*, 187–212. https://doi.org/10.1007/s40745-020-00253-5

[37] Tipos de redes neuronales [Online]. Disponible en: `http://www.asimovinstitute.org/neural-network-zoo/`

[38] Bebis, G., & Georgiopoulos, M. (1994). Feed-forward neural networks. *IEEE Potentials*, *13*(4), 27–31. https://doi.org/10.1109/45.329294

[39] Vaswani, A., *et al.* (2017). Attention is all you need. In *Advances in Neural Information Processing Systems*.

[40] Lin, T., Wang, Y., Liu, X., & Qiu, X. (2022). A survey of transformers. *AI Open*, *3*, 111–132. https://doi.org/10.1016/j.aiopen.2022.10.001

[41] Ejemplo de un Transformer Codificador-Decodificador [Online]. Disponible en: https://teksands.ai/blog/introduction-to-transformers

[42] Szałata, A., Hrovatin, K., Becker, S., *et al.* (2024). Transformers in single-cell omics: a review and new perspectives. *Nature Methods*, *21*, 1430–1443. https://doi.org/10.1038/s41592-024-02353-z

[43] Choromanski, K., *et al.* (2021). Rethinking attention with performers. In *International Conference on Learning Representations*.

[44] Du, J., *et al.* (2019). Gene2vec: distributed representation of genes based on co-expression. *BMC Genomics*, *20*, 82.

[45] Hao, Y., Hao, S., Andersen-Nissen, E., Mauck, W. M., 3rd, Zheng, S., Butler, A., Lee, M. J., Wilk, A. J., Darby, *et al.* ... Satija, R. (2021). Integrated analysis of multimodal single-cell data. *Cell, 184(13)*, 3573–3587.e29. https://doi.org/10.1016/j.cell.2021.04.048

Apéndice A

Gráficas de fine-tuning para entrenamiento y validación

En este anexo se presentan las gráficas que muestran las pérdidas de entrenamiento y validación de los modelos entrenados. Estas gráficas ayudan a visualizar cómo se ajustó el modelo con el tiempo a los datos para entender si está funcionando bien o no. Al observar cómo disminuye la pérdida durante el entrenamiento, se puede ver si el modelo está aprendiendo de los datos de manera efectiva. Además, al comparar la pérdida en los datos de entrenamiento con la pérdida en los datos de validación, se puede identificar si el modelo está sobreajustando—es decir, si está aprendiendo demasiado de los datos de entrenamiento y no generaliza bien a datos nuevos—o si aún necesita más entrenamiento.

Ambos modelos han sido entrenados en todos los escenarios con las mismas configuraciones; los dos modelos se han entrenado durante 10 épocas, ScGPT con un *mini batch size* de 128, un *dropout* de 0.2, tasa de aprendizaje de 0.0001 y 51 categorías de valores, mientras que en ScBERT se utilizó un *mini batch size* de 8, sin *dropout*, una tasa de aprendizaje de 0.0001 y 5 categorías de valores. Los dos modelos utilizan *Adam* como algoritmo de optimización y una función de pérdida de entropía cruzada categórica.

En ScGPT, las gráficas de pérdida de validación empiezan ya con valores muy bajos, dejando poco margen para que puedan seguir reduciéndose, sobre todo en la clasificación por tipo tumoral. En cambio, en ScBERT sí se ve un descenso más claro en las pérdidas tanto de entrenamiento como de validación, y este descenso ocurre bastante rápido, hacia la tercera época. En general, ambos modelos muestran cómo la pérdida de entrenamiento se reduce con cada época, destacando el caso de ScBERT, que refleja un aprendizaje más notable al inicio.

Gráficas de pérdida de entrenamiento y validación para todos los tipos celulares

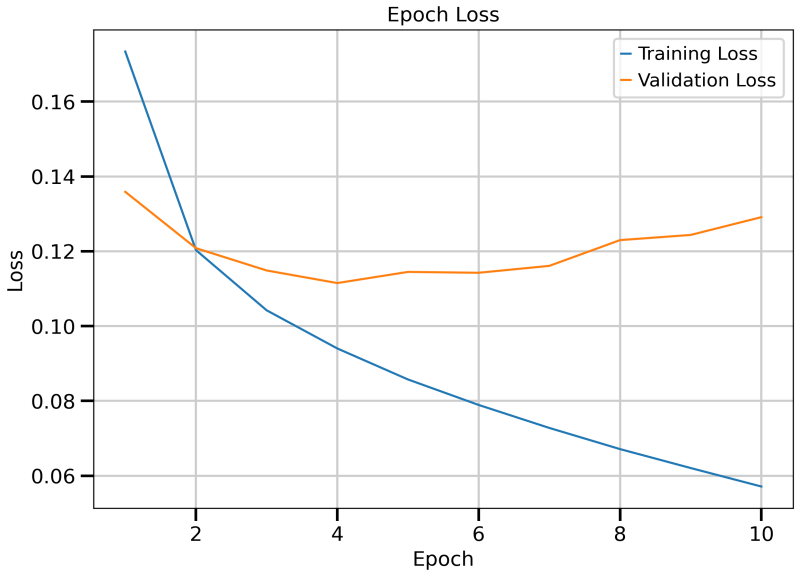

Figura A.1: Pérdidas de entrenamiento y validación de ScGPT en el primer escenario.

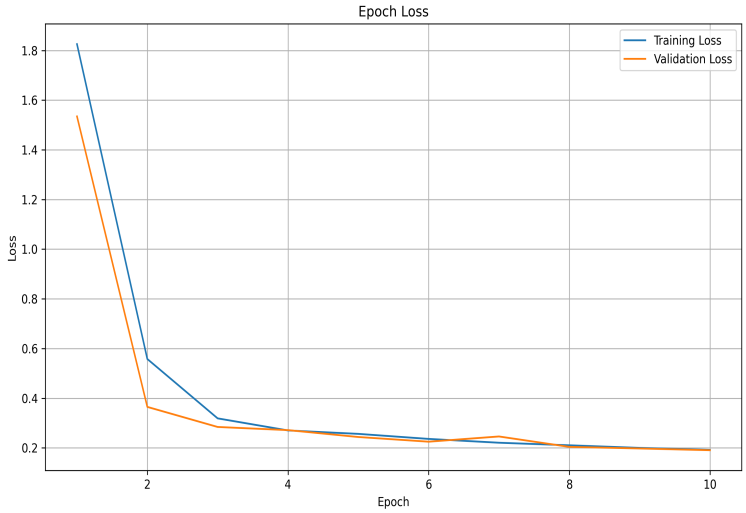

Figura A.2: Pérdidas de entrenamiento y validación de ScBERT en el primer escenario.

Gráficas de pérdida de entrenamiento y validación para el escenario en el que se eliminan las células malignas

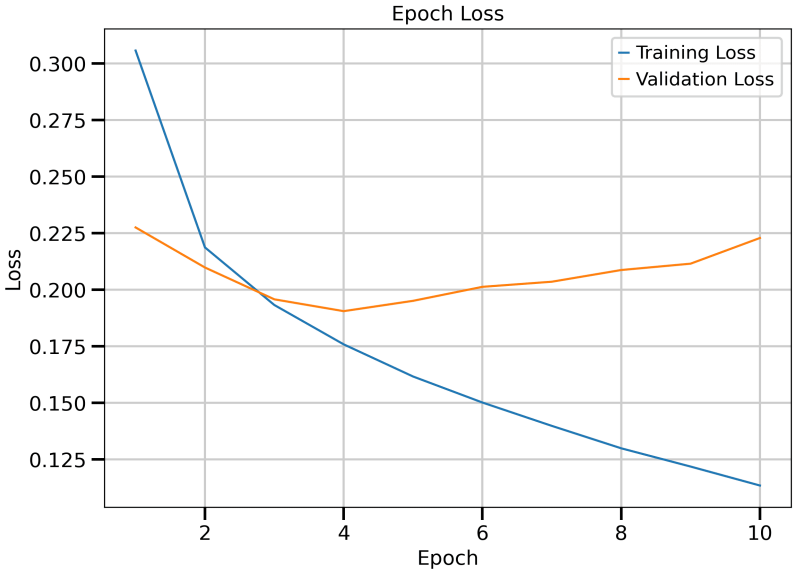

Figura A.3: Pérdidas de entrenamiento y validación de ScGPT en el segundo escenario.

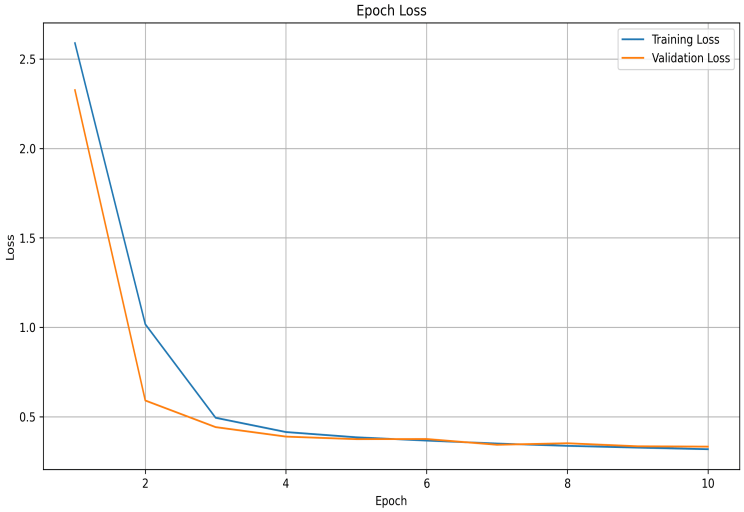

Figura A.4: Pérdidas de entrenamiento y validación de ScBERT en el segundo escenario.

Gráficas de pérdida de entrenamiento y validación para la clasificación por tipo tumoral

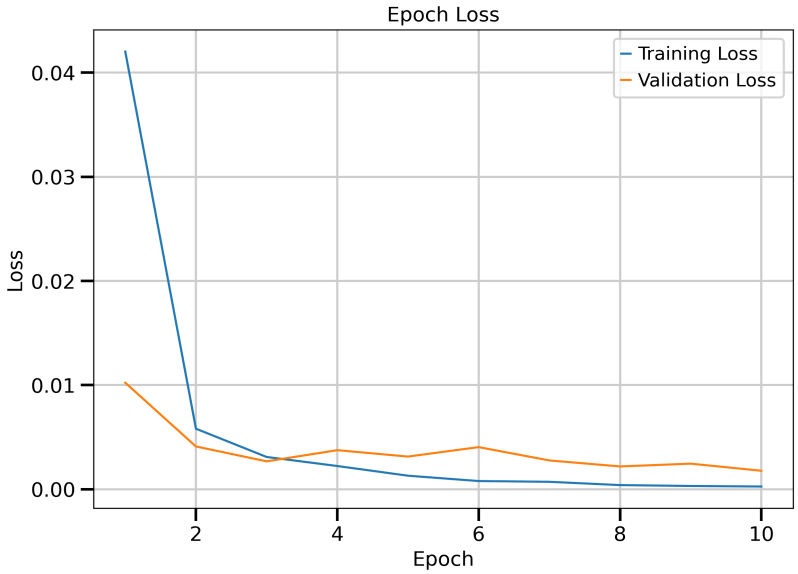

Figura A.5: Pérdidas de entrenamiento y validación de ScGPT en el tercer escenario.

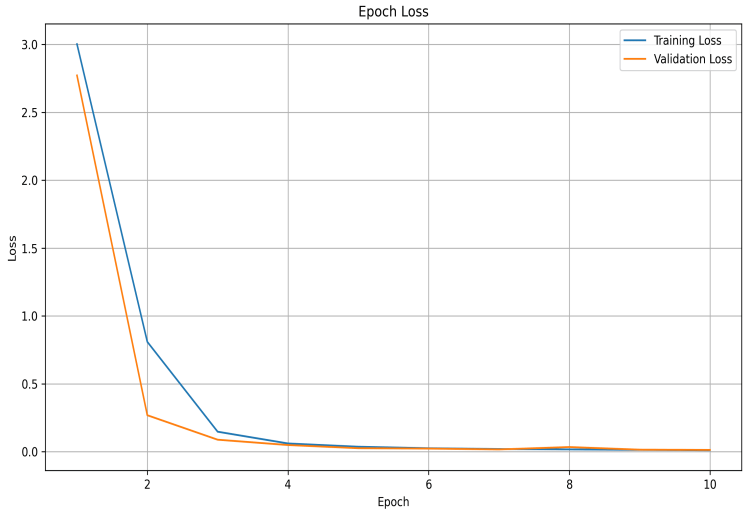

Figura A.6: Pérdidas de entrenamiento y validación de ScBERT en el tercer escenario.