Genómica Computacional

Muscular-guardian

Objetivo General del Proyecto.

El proyecto busca desarrollar una técnica de PCR-multiplex predictiva para detectar deleciones en el gen DMD, utilizando una red neuronal. Este enfoque aprovecha la genómica computacional para superar limitaciones de métodos tradicionales, priorizando la simplicidad y eficiencia en la detección de mutaciones frecuentes en la DMD.

Requisitos/tareas:

1. Revisión Literaria:
    - Investigar la literatura científica relacionada con DMD y técnicas de diagnóstico molecular.
2. Recopilación de Datos:
    - Identificar y recopilar datasets genómicos relevantes para el gen DMD de fuentes como NCBI Gene, Ensembl, y otros.
3. Diseño e Implementación de la Red Neuronal:
    - Desarrollar una arquitectura de red neuronal para predecir deleciones en el gen DMD, implementarla y ajustar parámetros.
4. Entrenamiento y Evaluación del Modelo:
    - Entrenar la red neuronal, evaluar su rendimiento y comparar resultados con datos reales.
5. Documentación y Presentación:
    - Crear documentación técnica detallada y elaborar un informe final resumiendo hallazgos y contribuciones del proyecto.

Marco Teórico

El gen DMD es el gen humano más grande conocido, ubicado en el cromosoma X (Xp21.2), que codifica para la distrofina, una proteína esencial para la estructura y función de las fibras musculares.

Las mutaciones en el gen DMD causan las distrofinopatías, un grupo de **distrofias musculares** progresivas que incluyen la **distrofia muscular de Duchenne (DMD)** y la **distrofia muscular de Becker (DMB),** que se heredan de forma recesiva ligada al cromosoma X.

La **DMD** es la **forma más grave** y **frecuente de distrofinopatía**, que se caracteriza por una ausencia o deficiencia severa de distrofina (< 5%), que conduce a una degeneración y debilidad muscular generalizada, afectando también al **corazón** y al **cerebro**.

La DMD se manifiesta típicamente entre los 2 y 3 años, con **retraso en el desarrollo motor**, hipertrofia de las pantorrillas, marcha anserina, signo de Gowers positivo y elevación de la **creatina cinasa (CK)** en el suero. La progresión es rápida y los pacientes pierden la capacidad de caminar alrededor de los 10 años, desarrollando posteriormente **escoliosis**, **insuficiencia respiratoria** y **miocardiopatía**. La esperanza de **vida media** es de unos **26 años**.

La mayoría de los pacientes con **Duchenne (DMD)** (alrededor del 65%) presentan **grandes deleciones** o duplicaciones de uno o varios **exones** en el **gen DMD**, que alteran el marco de lectura y provocan la **ausencia** de **distrofina**. El resto de los casos se deben a **mutaciones puntuales**, siendo las más frecuentes las **sustituciones nucleotídicas** tipo **nonsense**, que introducen un **codón de parada prematuro** y también impiden la **síntesis de distrofina**.

Las deleciones en el **gen DMD** se distribuyen de **forma no aleatoria**, concentrándose en dos regiones denominadas "hot spots": la región proximal **(exones 2-20)** y la región distal **(exones 44-53)**. Estas regiones presentan una alta frecuencia de recombinación homóloga entre secuencias repetidas, lo que facilita la aparición de deleciones. La deleción **más común** es la del **exón 50**, que se observa en el 10% de los casos de DMD.

Las **deleciones** en el **gen DMD** pueden ser **detectadas** mediante técnicas de **PCR multiplex**, que amplifican simultáneamente varios exones del gen. Sin embargo, esta técnica **no** es capaz de **identificar** las deleciones que afectan a **exones no incluidos** en el panel de PCR, ni las duplicaciones o las **mutaciones puntuales**. Por ello, se recomienda **complementar** el estudio con otras técnicas, como el **análisis** de la expresión y la **función** de la **distrofina** en el tejido muscular, o el análisis de **secuenciación** del gen **DMD**.

El **gen DMD**, extenso y **propenso a mutaciones**, puede sufrir distintos cambios, siendo las **deleciones** las más frecuentes (60-70% de los casos). Otros tipos incluyen **duplicaciones**, **inserciones** o **mutaciones puntuales**.

El **diagnóstico** molecular de la **DMD (Duchenne)** se realiza mediante técnicas de biología molecular. La **PCR-multiplex**, a pesar de su simplicidad, tiene limitaciones, mientras que la **MLPA**, más avanzada, permite analizar la dosis génica de los 79 exones del gen DMD, detectando deleciones y duplicaciones con mayor precisión.

Recopilación de Datos

Una entrada de datos en uno de estos conjuntos de datos debería contener la información genética (Secuencia ADN) necesaria para identificar la mutación en el **gen DMD**. El formato ideal para que podamos utilizar el conjunto de datos en general es FASTA, para apoyarnos del uso de *BioPython*, el conjunto de datos debe ser estructurado, completo, consistente y válido.

Para el posible entrenamiento de una red neuronal, que nos ayudara a encontrar marcadores genéticos deberíamos buscar un conjunto de datos robusto. Un ejemplo de formato estructurado ideal, debería contener los datos de las mutaciones en el gen DMD en un archivo Excel o CSV, con columnas similares a estas:

- Patient ID: un código numérico que identifica al paciente de forma anónima.
- Exon number: el número de exón o el rango de exones afectados por la mutación.
- Phenotype: el fenotipo clínico asociado a la mutación, como DMD, BMD o IMD.
- References: las referencias bibliográficas que reportan la mutación, si las hay.
- PK: niveles de Piruvato Quinasa.
- CK: niveles de Creatina Quinasa.
- Age: edad del paciente.
- Entre otros…

Diseño e Implementación de la Solución

Para nuestra solución vamos a buscar el número de ORF's (Open Reading Frames), para determinar la ausencia de producción de distrofina. Para llevar a cabo este análisis, se utilizó la biblioteca BioPython para leer y manipular secuencias de ADN.

Para comenzar, se obtuvieron secuencias de ADN del gen DMD. Estas secuencias fueron procesadas utilizando *BioPython* para realizar una comparación de ORF. El objetivo principal fue identificar cualquier discrepancia o diferencia en las secuencias de ADN que pudieran indicar una interrupción en la producción de distrofina.

Mediante *BioPython*, se implementó un algoritmo que permitió la comparación de las secuencias de ADN de las muestras con una secuencia de referencia conocida del **gen DMD**. Esto ayudó a identificar cualquier cambio o mutación en las regiones codificantes del gen, donde se encuentran los ORF responsables de la producción de distrofina.

Finalmente compararemos los resultados del algoritmo usando la herramienta del NCBI, ORFFinder.

Por otro lado, buscando tener un diagnóstico lo más acertado posible, se plantea la construcción de una red neuronal que, se propusieron los diferentes enfoques siguientes.

El enfoque inicial para identificar las deleciones más frecuentes en el gen DMD, y seleccionar los exones que se van a amplificar mediante la PCR-multiplex, deseamos utilizar un conjunto de datos público que contenga información sobre el tipo, la localización, la frecuencia y el fenotipo de las mutaciones en el gen DMD.

Otro enfoque que se pudo haber tomado pudo ser el de estudiar imágenes de tejidos musculares, donde se pueda apreciar la ausencia de distrofina, este enfoque se descartó debido a lo pesado que es entrenar modelos con imágenes, y del mismo modo, la dificultad de encontrar conjuntos de datos suficientes para el análisis.

Para el proyecto final, en vista de la dificultad de encontrar datos de pacientes, en donde se enuncien los exones para un análisis más directo, decidimos optar por usar la concentración creatina quinasa (CK) y piruvato quinasa (PK), como marcadores genéticos donde nuestra red neuronal va a detectar consistentemente si los pacientes tienen mayor tendencia de producir la mutación del gen DMD.

ÓRF's, Entrenamiento y Evaluación del Modelo

Al comienzo de la implementación vamos a aplicar la solución de PCR que desarrollamos durante el curso, para dada la secuencia del **Gen DMD**, determinar sus ORF's, apoyándonos del ORfinder de NCBI.

```python
from Bio.SeqRecord import SeqRecord
from Bio import SeqIO

# Leer las secuencias del archivo
dmd_records = list(SeqIO.parse('DMD_GENE.fna', 'fasta'))

# Si solo quieres trabajar con la primera secuencia del archivo, puedes hacerlo así:
dmd = dmd_records[0]

# Mostrar los primeros 1000 nucleótidos en el gen DMD
dmd_DNA = dmd.seq
print(dmd_DNA[:1000])


CodiumAI: Options | Test this function
def nt_search(seq, start_codon, stop_codon):
    orfs = []
    start_positions = [i for i in range(len(seq)) if seq.startswith(start_codon, i)]

    for start in start_positions:
        for i in range(start, len(seq), 3):
            if seq[i:i+3] == stop_codon:
                orfs.append(seq[start:i+3])
                break

    return orfs

# Número de ORFs (Open Reading Frames) viables
orfs_all = nt_search(str(dmd_DNA), 'ATG', 'TGA') # TGA es el codón de paro para DMD (Distrofina),
                                                 # pero puede variar para otras secuencias

# Número de ORFs no espurios con probabilidad < 0.05
orfs_05 = [orf for orf in orfs_all if len(orf) >= 150 and len(orf) % 3 == 0]

# Número de ORFs no espurios con probabilidad < 0.01
orfs_01 = [orf for orf in orfs_all if len(orf) >= 180 and len(orf) % 3 == 0]

# Imprimir resultados
print(f"a) Número de ORFs viables: {len(orfs_all)}")
print(f"b) Número de ORFs no espurios con probabilidad < 0.05: {len(orfs_05)}")
print(f"c) Número de ORFs no espurios con probabilidad < 0.01: {len(orfs_01)}")

# Código para obtener el número de ORFs reales y compara con los resultados anteriores
# Utilizando ORFfinder:
# - Guarda la secuencia en un archivo FASTA
with open('sequence_dmd.fasta', 'w') as fasta_file:
    SeqIO.write(dmd, fasta_file, 'fasta')

# - Utilizando ORFfinder en línea: https://www.ncbi.nlm.nih.gov/orffinder/
# - archivo __all.fa contiene los resultados de ORFfinder
orfs_reales = []
with open('dmd__all.fa', 'r') as file:
    lines = file.readlines()
    for line in lines:
        if line.startswith(">"):
            orfs_reales.append(line)
print(f"e) Número de ORFs reales: {len(orfs_reales)}")
```

Donde tenemos los siguientes resultados.

```
GGCAGTAATAGAATGCTTTCAGGAAGATGACAGAATCAGGAGAAAGATGCTGTTTTGCACTATCTTGATTTGTTACAGCAGCCAACTTATTGGCA
a) Número de ORFs viables: 44921
b) Número de ORFs no espurios con probabilidad < 0.05: 16680
c) Número de ORFs no espurios con probabilidad < 0.01: 13661
e) Número de ORFs reales: 77
```

Los resultados indican que se encontraron 44921 ORF totales, 16680 ORF no espurios con una probabilidad de error menor que 0.05, 13661 ORF no espurios con una probabilidad de error menor que 0.01 y 77 ORF reales.

A partir de estos resultados, se puede concluir lo siguiente:

La secuencia de ADN analizada tiene alta diversidad de ORF, lo que sugiere que puede contener muchos genes potenciales. Sin embargo, no todos los ORF son realmente traducidos en proteínas, por lo que se necesita más evidencia para confirmar la presencia de genes.

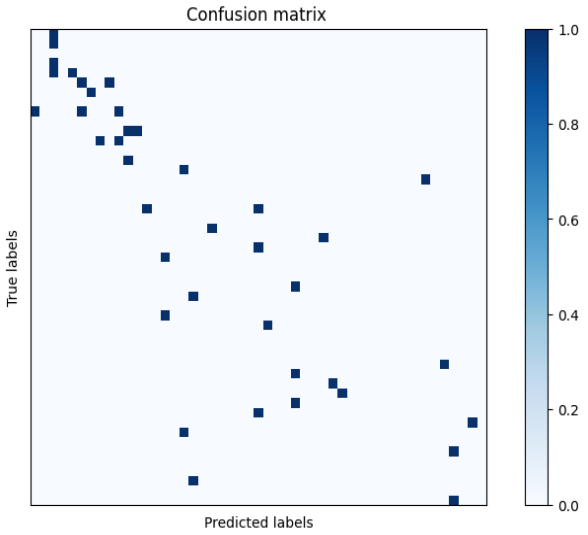
El número de ORF no espurios disminuye a medida que se reduce la probabilidad de error, lo que implica que hay menos falsos positivos. Esto significa que se puede ajustar el nivel de probabilidad de error para filtrar los ORF más confiables y relevantes para el análisis biológico.

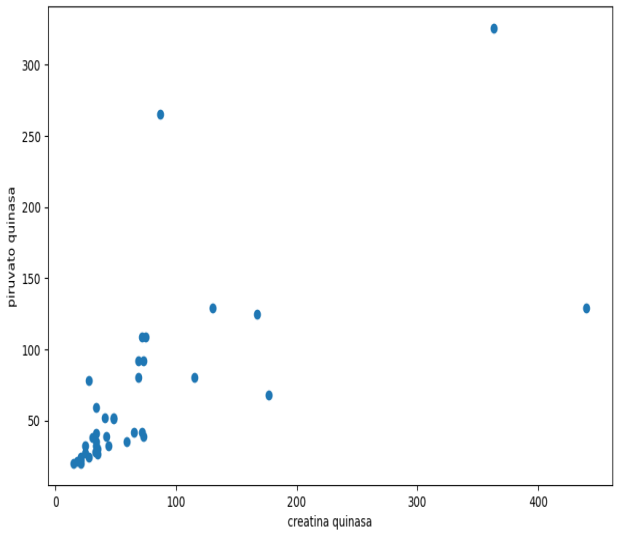
El número de ORF reales es muy bajo en comparación con el número total de ORF, lo que indica que la mayoría de los ORF encontrados son artefactos o ruido. Esto podría mejorar usando una estructura secundaria además de los codones de inicio y parada, pero debido a la naturalidad de nuestro proyecto fue complejo encontrar muestras que mostrasen una comparación útil para el análisis.

Posteriormente ejecutaremos el modelo de red neuronal, que va a clasificar e identificar ADN del **gen DMD**. Entendiendo que las mutaciones en este gen causan la distrofia muscular de Duchenne y de Becker. Nuestro código utiliza las bibliotecas de Python *numpy*, *pandas*, *matplotlib*, *sklearn y LabelEncoder* para realizar las siguientes tareas:

- Leer un archivo de datos que contiene información sobre el ADN del gen DMD y sus etiquetas pk y ck (Piruvato Quinasa y Creatina Quinasa).
- Limpiar los datos eliminando las filas con valores faltantes.
- Separar los datos en características (X) y etiquetas (y).
- Convertir los datos categóricos a numéricos usando la clase LabelEncoder.
- Dividir los datos en conjuntos de entrenamiento y prueba usando la función train_test_split.
- Crear un modelo de red neuronal usando la clase MLPClassifier, que tiene tres capas ocultas de 100 neuronas cada una, un número máximo de iteraciones de 5000, un parámetro de regularización de 0.0001, un algoritmo de optimización llamado adam, una semilla aleatoria de 21, una tolerancia de 0.000000001, un tamaño de lote de 64 y una tasa de aprendizaje inicial de 0.001.
- Entrenar el modelo con los datos de entrenamiento usando el método fit.
- Predecir las etiquetas de los datos de prueba usando el método predict.
- Calcular la precisión del modelo usando la función accuracy_score, que compara las etiquetas reales con las predichas.
- Crear una matriz de confusión usando la función confusion_matrix, que muestra el número de aciertos y errores del modelo por cada clase.
-

```python
# Red neuronal para clasificar e identificar ADN del gen DMD
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import confusion_matrix, accuracy_score
from sklearn.preprocessing import LabelEncoder

# Leer el archivo de datos
df = pd.read_csv('dmd.csv')

# Eliminar las filas con valores faltantes
df.dropna(inplace=True)

# Separar las características de las etiquetas
X = df.drop('pk', axis=1)
y = df['ck']

# Convertir datos categoricos a numéricos
le = LabelEncoder()
X = X.apply(le.fit_transform)

# Separar los datos en conjuntos de entrenamiento y prueba
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)

# Crear el modelo de red neuronal
model = MLPClassifier(hidden_layer_sizes=(100, 100, 100), max_iter=5000, alpha=0.0001, solver='adam',
                      random_state=21, tol=0.000000001, batch_size=64, learning_rate_init=0.001)

# Entrenar el modelo
model.fit(X_train, y_train)

# Predecir las etiquetas de los datos de prueba
y_pred = model.predict(X_test)

# Calcular la precisión del modelo
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy}")

# Crear la matriz de confusión
cm = confusion_matrix(y_test, y_pred)
print(cm)
```





A partir de la gráfica de la derecha, se puede concluir lo siguiente:

- El modelo tiene una precisión moderada, ya que hay una tendencia general de que los puntos se acerquen a la diagonal, pero también hay una gran dispersión y variabilidad de los datos.
- La gráfica muestra que hay una correlación positiva entre los niveles de ck y pk, es decir, que a mayor ck, mayor pk. Esto puede deberse a que ambas enzimas están involucradas en el mismo proceso metabólico y se regulan de forma coordinada.
- Y de forma inversamente proporcional, dado que nuestro conjunto de datos se trata de pacientes ya diagnosticados, podemos abstraer que los bajos índices en estas enzimas indican que efectivamente, los datos extraídos son de personas donde las enzimas no tienen una fuerte presencia, indicando problemas en los tejidos musculares.

Y a partir de la gráfica de la izquierda, se puede concluir que el modelo tiene bastante precisión, esto tiene sentido, dado que estamos analizando datos de personas ya diagnosticadas.

Documentación y Presentación:

Anexamos el código fuente para el proyecto, en forma de un jupyter notebook.

Además de una presentación en formato pdf.

La secuencia de DMD en formato FASTA.

Un pequeño dataset con los marcadores genéticos en formato csv.

Referencias

Distrofia muscular de Duchenne | Anales de Pediatría Continuada - Elsevier. https://www.elsevier.es/es-revista-anales-pediatria-continuada-51-articulo-distrofia-muscular-duchenne-S1696281814701684.

Distrofia muscular de Duchenne - Genetic and Rare Diseases Information .... https://rarediseases.info.nih.gov/espanol/13375/distrofia-muscular-de-duchenne/.

Implementación de la Prueba del Multiplex PCR para el Gen DMD en .... https://www.redalyc.org/articulo.oa?id=371637126002.

Detección de mutaciones causantes de distrofia muscular de ... - SciELO. http://www.scielo.org.pe/scielo.php?script=sci_arttext&pid=S1726-46342019000300475.

Dystrophin (DMD) sequence variations. https://www.dmd.nl/database.html.

The UMD-DMD French Knowledgebase. http://www.umd.be/DMD.

MSD Manuals. (s. f.). Distrofia muscular de Duchenne y distrofia muscular de Becker. Recuperado el 11 de diciembre de 2023, de https://www.msdmanuals.com/es/professional/pediatría/trastornos-musculares-hereditarios/distrofia-muscular-de-duchenne-y-distrofia-muscular-de-becker.

Orphanet. (s. f.). Distrofia muscular de Duchenne y Becker. Recuperado el 11 de diciembre de 2023, de https://www.orpha.net/consor/cgi-bin/OC_Exp.php?lng=ES&Expert=262.

Natera-de Benito, D., Nascimento, A., Abreu-González, L., Jiménez-Mallebrera, C., Jou, C., Rodríguez-Sánchez, J. M., … & García-Romero, M. (2016). Espectro mutacional de la distrofia muscular de Duchenne en España: estudio de 284 casos. Neurología, 31(8), 522-530. https://www.elsevier.es/es-revista-neurologia-295-articulo-espectro-mutacional-distrofia-muscular-duchenne-S0213485316000219.

García, S., & Canto, P. (2007). Biología molecular de la distrofia muscular de Duchenne. Boletín médico del Hospital Infantil de México, 64(2), 221-222. http://anmm.org.mx/bgmm/1864_2007/1996-132-2-221-222.pdf.

Pérez, M. L., Rodríguez, H., González, L. M., & Pérez, M. (2016). muscular de Diagnóstico molecular de distrofia Duchenne mediante reacción en cadena de la polimerasa multiplex. MEDISAN, 20(5), 581-589. http://scielo.sld.cu/pdf/ms/v16n5/ms11516.pdf.

Watson, J. D., Baker, T. A., Bell, S. P., Gann, A., Levine, M., & Losick, R. (2014). Biología molecular del gen (7a ed.). Wolters Kluwer. https://booksmedicos.org/biologia-molecular-del-gen-7a-edicion/.

PTC Campus. (s. f.). Distrofia muscular de Duchenne: qué es. Recuperado el 11 de diciembre de 2023, de https://ptccampus.es/dmd.

Duchenne.com. (s. f.). ¿Qué es la distrofina? Recuperado el 11 de diciembre de 2023, de 1

Duchenne Spain. (s. f.). La distrofina. Recuperado el 11 de diciembre de 2023, de 2

GARD. (2019, 18 de julio). Distrofia muscular de Duchenne. Recuperado el 11 de diciembre de 2023, de 3

Duchenne Spain. (s. f.). Distrofina. Recuperado el 11 de diciembre de 2023, de 4

Rodríguez, M. A., & Natera, J. (2014). Distrofia muscular de Duchenne. Anales de Pediatría Continuada, 12(6), 287-292.

NCBI. (s. f.). DMD dystrophin [Homo sapiens (human)]. Recuperado el 11 de diciembre de 2023, de https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE199659

NCBI. (s. f.). GSE199659: RNA-seq of human skeletal muscle from Duchenne muscular dystrophy patients and healthy controls. Recuperado el 11 de diciembre de 2023, de https://www.ncbi.nlm.nih.gov/gene/1756

Duchenne.com. (s. f.). ¿Qué es la distrofina? Recuperado el 11 de diciembre de 2023, de 1

Duchenne Spain. (s. f.). La distrofina. Recuperado el 11 de diciembre de 2023, de 2

GARD. (2019, 18 de julio). Distrofia muscular de Duchenne. Recuperado el 11 de diciembre de 2023, de 3

Duchenne Spain. (s. f.). Distrofina. Recuperado el 11 de diciembre de 2023, de 4

Rodríguez, M. A., & Natera, J. (2014). Distrofia muscular de Duchenne. Anales de Pediatría Continuada, 12(6), 287-292.

NCBI. (s. f.). DMD dystrophin [Homo sapiens (human)]. Recuperado el 11 de diciembre de 2023, de https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE199659

NCBI. (s. f.). GSE199659: RNA-seq of human skeletal muscle from Duchenne muscular dystrophy patients and healthy controls. Recuperado el 11 de diciembre de 2023, de https://www.ncbi.nlm.nih.gov/gene/1756