TRABAJO FIN DE GRADO

HERRAMIENTA PARA EL ESTUDIO DEL ALINEAMIENTO DE SECUENCIAS DE ADN MEDIANTE DEEPLEARNING

Universidad Católica San Antonio de Murcia
ESCUELA UNIVERSITARIA POLITECNICA
Grado en Ingeniería Informática

Autor: Juan Francisco Illán Sánchez
Directores: Francisco Arcas Túnez,
Jesús Soto Espinosa

Murcia, Abril de 2021

Identificación del Proyecto

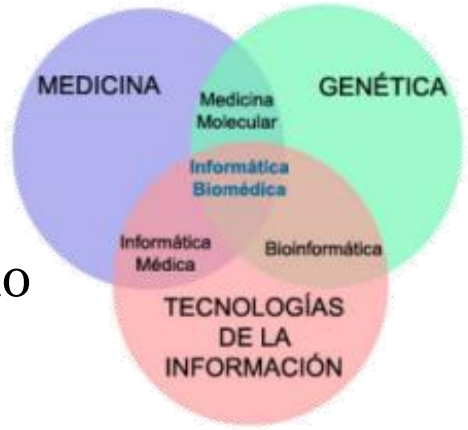


Imagen tomada de: Arias, Bahón, & Rodríguez (2006), Desarrollo de una plataforma de análisis de datos en Bioinformatica basada en Matlab, Universidad Complutense. Madrid

- El proyecto pretende una suite de laboratorio web para el alineamiento de secuencias genómicas.
- En bioinformatica el alineamiento pretende identificar similitudes entre una cadena genomica de consulta y una colección de cadenas en una base de datos.
- Palabras clave: Bioinformatica, alineamiento de secuencias, algoritmos heurísticos, redes neuronales, machine learning, clasificador bayes, identificación LSTM, python.

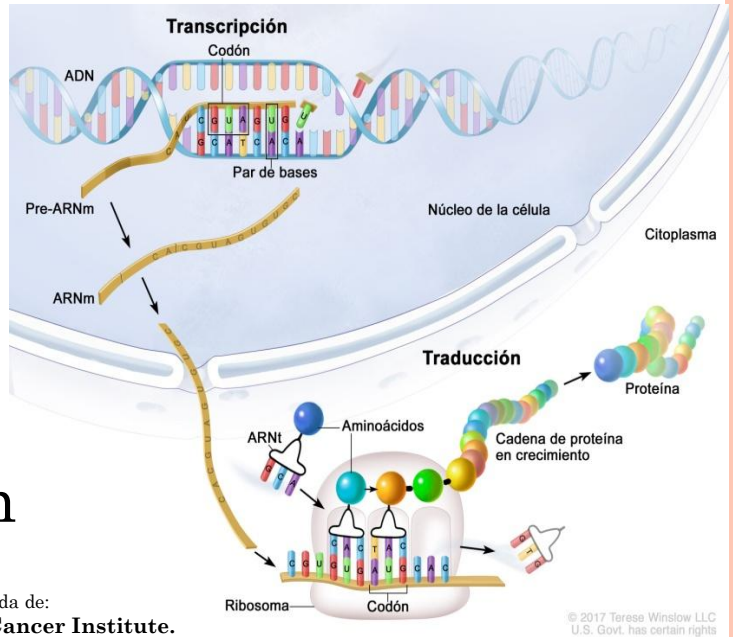


Imagen tomada de:
National Cancer Institute.
https://www.cancer.gov/espanol/publicaciones/diccionarios/diccionario-cancer/def/transcripcion

Características del Proyecto

- Desarrollado en:
  - python 3.7
  - Flask, Html, boostrap, css, js,

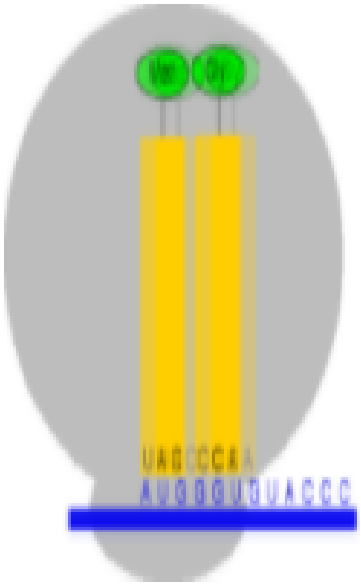
- Librerías especificas ciencia de datos / machine learning:
  - Numpy, Pandas
  - Matplotlib
  - sklearn.feature_extraction.text.CountVectorizer
  - sklearn.naive_bayes .MultinomialNB
  - Tensorflow
  - keras.models import Model, Sequential
  - keras.layers import Input, Dense, Embedding, LSTM

- Base de datos: Ficheros sobre disco en formato FASTA

Funcionalidad del Proyecto

- El proyecto implementa 3 funcionalidades resueltas utilizando diferentes técnicas:
  - Smith-Waterman – Heuristic Algorithm
  - Multinomial Naive Bayes Classifier – Machine Learning
  - LSTM Classifier – Machine Learning

Alignament Nucleotide
(Heuristic algorithm Smith-Waterman)



Nucleotide to Protein
(Multinomial Bayes Classifier)

Secuence Identified as Pathogen
(LSTM Classifier)

Smith-Waterman – Heuristic Algorithm

- El Algoritmo BLAST puede describirse en tres etapas: (wikipedia.org, BLAST, 2021)
  - *Localización de semillas (HSSP) de un tamaño k, con una alta puntuación sin huecos. High-Scoring Segment Pair.*
    - Problema ¿donde empezar a contar? Todas las combinaciones de longitud k (k-mers) seran candidatas a semilla. (Clelia DI Serio, Pietro Liò, Alessandro Nonis, Roberto Tagliaferri, Computational Intelligence Methods for Bioinformatics and Biostatistics, Cambridge, 2014)
    - *Heurística:*
      - *Selección solo de mejores semillas para cada cadena del la base de datos.*
  - *Extender las mejores semillas con huecos.*
    - *Matriz **Smith-Waterman**: para cada semilla seleccionada, consiste en ir generar una matriz con la mejor puntuación de alineamiento posible a cada celda extendiendo con huecos la puntuación inicial de la semilla sobre cada posición de la matriz calculada.*
    - *Finalizada la matriz, se reconstruye la ruta solución desde la celda con la puntuación mas alta para esa semilla*
    - *Heurística:*
      - *Acotar calculo de la Matriz Smith-Waterman a la "frontera" próxima a la diagonal optima.*
  - *Evaluación y ordenación de las mejores soluciones (alineamientos)*

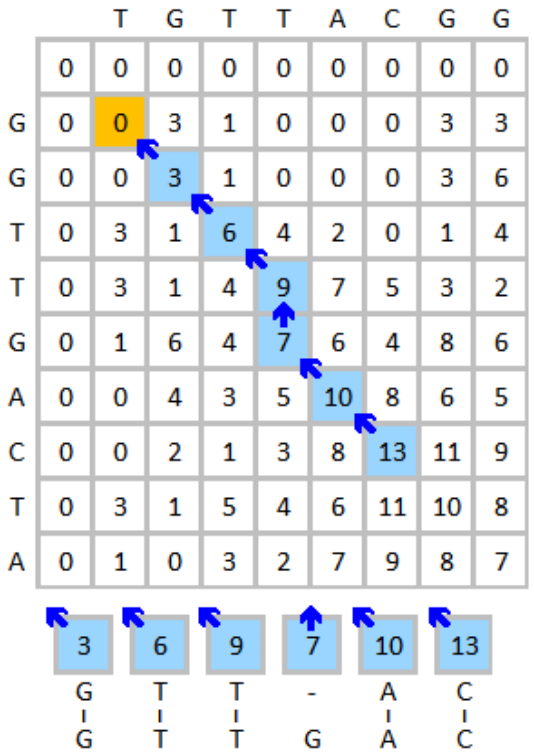
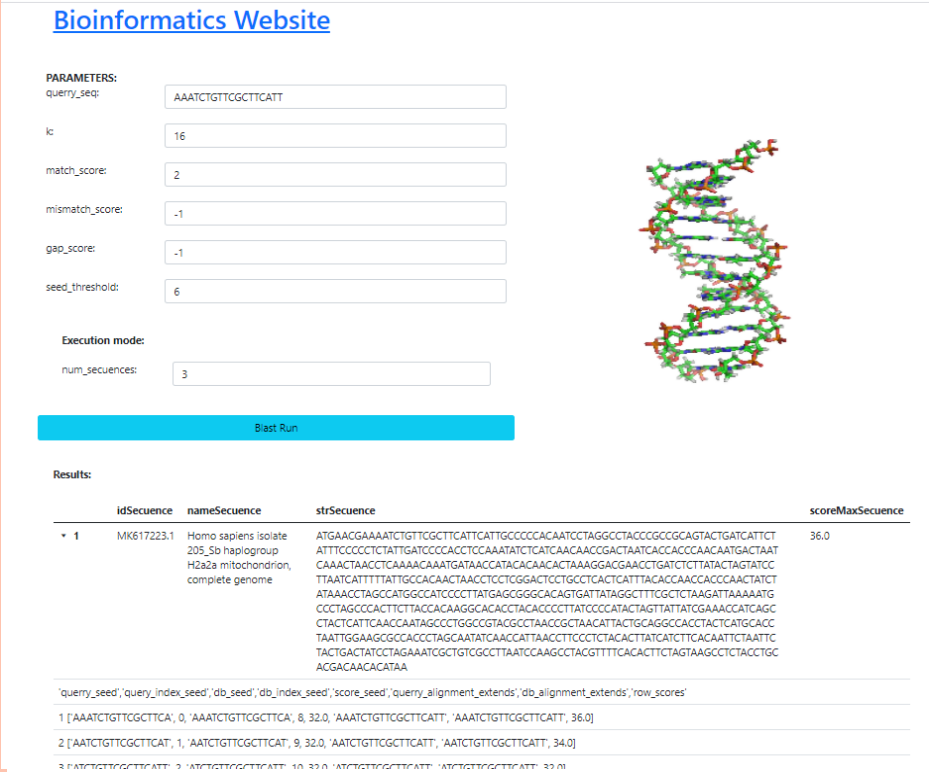
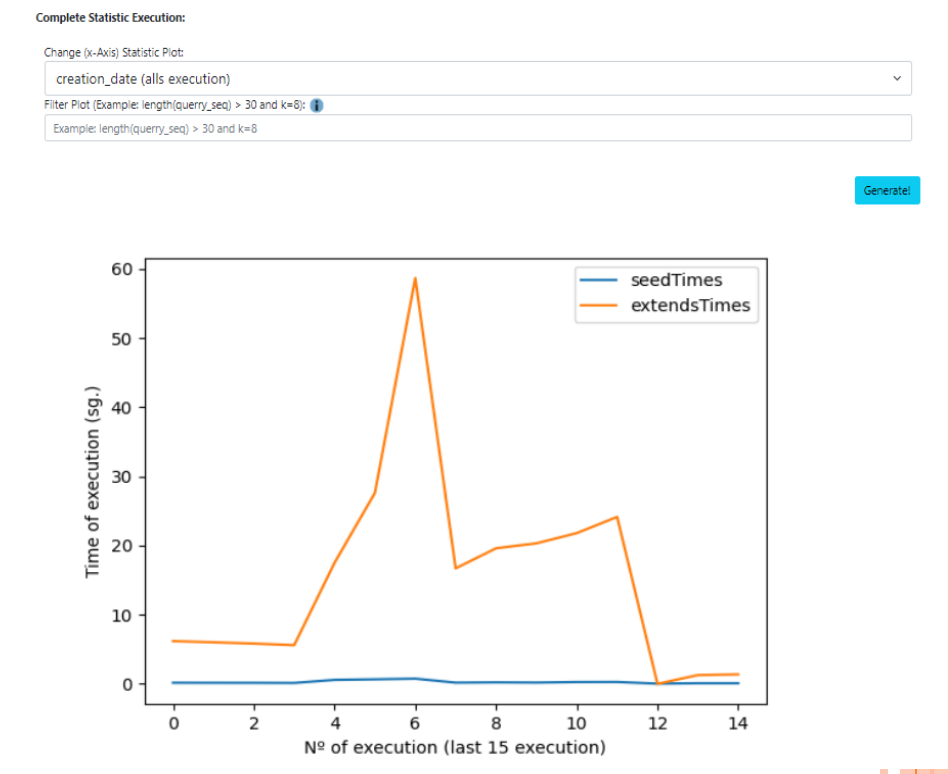


Imagen tomada de: wikipedia.org, Smith–Waterman algorithm, 2021

Smith-Waterman – Heuristic Algorithm

- Se crea una implementación funcional para la consulta y utilización del sistema.
  - Se calcula y se muestran los mejores alineamientos en base a los parámetros de entrada
  - Se implementa un sistema de almacenamiento de estadísticas de ejecución, para valorar como afecta variar cada parámetro.





Multinomial Naive Bayes Classifier Machine Learning

- Es un clasificador probabilístico basado en el teorema de Bayes.
- El teorema de Bayes vincula la probabilidad de un evento A dándose B con la probabilidad de B dándose A.

$$P(A_i|B) = \frac{P(B|A_i)P(A_i)}{P(B)}$$

donde:

- $P(A_i)$ son las probabilidades a priori,
- $P(B|A_i)$ es la probabilidad de B en la hipótesis A_i,
- $P(A_i|B)$ son las probabilidades a posteriori.

- Es muy eficiente al trabajar con un árbol de decisión numérico completo, de 0 a 1.
- Se utiliza en detección de correos spam, en toma de decisiones, ...
- **OBJETIVO**: Pretendemos dada una cadena de consulta identificar con que esta relacionado. Por ejemplo que proteína sintetiza esa cadena.

Multinomial Naive Bayes Classifier Machine Learning

- En las cadenas genómicas, la codificación biológica se interpreta en las oraciones y párrafos, no en letras aisladas.
- ¿Porque?
  - Sabemos que la traducción de ADN/ARN a aminoácidos es cada 3 nucleótidos, genera 1 aminoácido en los ribosomas.
  - Que hay un lenguaje de 20 aminoácidos para formar las proteínas como cadenas de cientos de aminoácidos.
  - La importancia a nivel de frase o párrafo (de 20 - 50 nucleótidos) para la finalidad de encontrar una relación o identificación biológica.
- ¿Cómo interpretarlo?

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	MOLECULE_TYPE	ID	PROTEIN_ID	PROTEIN_NAME	SEQ_ID	NAME	SEQUENCE										
2	dna	4382	4	Synthase	D38113.1	Pan troglodytes mitochondrial dr	ATGAACGAAAATCTATTCGCTTCATTCGCTGCCCCCACAATCCTAGGCTTACCCGCCGCAGTACTAATCATTCTATTCCCCCCTCTA										
3	dna	4381	4	Synthase	KX211955.1	Pan troglodytes verus isolate Dor	ATGCCCCAACTAAATACCGCCGTATGACCCACCATAATTACCCCCATACTCCTGACACTATTTCTCGTCACCCAACTAAAAATATT										
4	dna	2	4	Synthase	MK617223.1	Homo sapiens isolate 205_Sb hap	ATGAACGAAAATCTGTTCGCTTCATTCATTGCCCCCACAATCCTAGGCCTACCCGCCGCAGTACTGATCATTCTATTTCCCCCTCTA										
5	dna	1	4	Synthase	MW389273.1	Homo sapiens haplogroup U8b1b	ATGCCCCAACTAAATACTACCGTATGGCCCACCATAATTACCCCCATACTCCTTACACTATTCCTCATCACCCAACTAAAAATATTA										
6	dna	5	3	Synthetases	NM_001178075	Homo sapiens asparagine synthe	ATGCAACAGCATTTTGAATTTGAATACCAGACCAAAGTGGATGGTGAGATAATCCTTCATCTTTATGACAAAGGAGGAATTGAG										
7	dna	31	6	Transcription Facto	NM_001323042	Homo sapiens putative homeodc	ATGGCCTCAAATGAGAGAGATGCTATATCGTGGTACCAAAAGAAGATTGGAGCCTACGATCAGCAGATATGGGAAAAGTCAAT										
8	dna	3	3	Synthetases	NM_001352496	Homo sapiens asparagine synthe	ATGTGTGGCATTTGGGCGCTGTTTGGCAGTGATGATTGCCTTTCTGTTCAGTGTCTGAGTGCTATGAAGATTGCACACAGAGGTCC										
9	dna	6	3	Synthetases	NM_001352496	Homo sapiens asparagine synthe	ATGTGTGGCATTTGGGCGCTGTTTGGCAGTGATGATTGCCTTTCTGTTCAGTGTCTGAGTGCTATGAAGATTGCACACAGAGGTCC										
10	dna	10	3	Synthetases	NM_001352496	Homo sapiens asparagine synthe	ATGTGTGGCATTTGGGCGCTGTTTGGCAGTGATGATTGCCTTTCTGTTCAGTGTCTGAGTGCTATGAAGATTGCACACAGAGGTCC										
11	dna	43	0	g protein coupled r	NM_003272.4	Homo sapiens G protein-coupled	ATGAGGCCCGAGCGTCCCCGGCCGCGCGGCAGCGCCCCCGGCCCGATGGAGACCCCGCCGTGGGACCCAGCCCGCAACGACTC										
12	dna	44	0	g protein coupled r	NM_003272.4	Homo sapiens G protein-coupled	ATGAGGCCCGAGCGTCCCCGGCCGCGCGGCAGCGCCCCCGGCCCGATGGAGACCCCGCCGTGGGACCCAGCCCGCAACGACTC										
13	dna	167	1	Tirosina kynase	NM_004560.4	Homo sapiens receptor tyrosine	ATGAAGACCATTACCGCCACTGGCGTCCTGTTTGTGCGGCTGGGTCCAACGCACAGCCCAAATCATAACTTTCAGGATGATTACC										
14	dna	24	2	Tirosina Phosfate	NM_007039.4	Homo sapiens protein tyrosine p	ATGCCACTGCCATTTGGGTTGAAACTGAAACGCACCCGGCGCTACACGGTGTCCAGCAAGAGTTGCCTGGTTGCCCGGATCCAAC										
15	dna	25	2	Tirosina Phosfate	NM_021625.5	Homo sapiens transient receptor	GCCACCACAGGCCTGAAGATGAAGCACCTCCTTACTGGGCAAGAGAGGACCGTCTGGCACCTCCAATACACAGACTGGCCTGAA										
16	dna	17	5	ion channel	NM_021625.5	Homo sapiens transient receptor	ATGGCGGATTCCAGCGAAGGCCCCCGCGCGGGGCCCGGGGAGGTGGCTGAGCTCCCCGGGGATGAGAGTGGCACCCCAGGTGG										
17	dna	18	5	ion channel	NM_021625.5	Homo sapiens transient receptor	ATGGCGGATTCCAGCGAAGGCCCCCGCGCGGGGCCCGGGGAGGTGGCTGAGCTCCCCGGGGATGAGAGTGGCACCCCAGGTGG										

Multinomial Naive Bayes Classifier Machine Learning

- La idea la podemos extrapolar de la detección de correos spam.
  - Si aprendemos que cuando aparecen las palabras: win/won, movie, downloads : es spam, ya tenemos algo que mas o menos funciona…
  - Pero … y si el mensaje pone :

    "Hey, I won, I choose!

    This afternoon we are going to the cinema to see the movie

    Here is the link to downloads the tickets"

    → *No parece que no sea SPAM, pero tiene palabras asociadas a mensajes de SPAM…*
  - Solución: el sistema debe aportar contexto a las palabras. Para ello contamos si aparecen las palabras en un contexto y no de forma aislada.
    - "Free dowloads movie"
    - "game free to play, now avalaible dowloads"
    - "Your PC has a virus, download antivirus"

Multinomial Naive Bayes Classifier Machine Learning

Esta idea se desarrollada para el alineamiento genético por: (*Thomas Nelson, Kaggle.com - Working with DNA sequence data for ML part 2, 2019*), consiste en estas ideas principales:

- Expandir cada cadena a la combinación de k-mers que da lugar. (Utilizamos K = 6).
- Utilizamos un CountVectorizer (CV). Utilizaremos el CV definiendo como rango de palabras para contar frases de (N=3 palabras) de 6 nucleotidos, mayor contexto e interelacion entre bloques contiguos (6 aminoacidos) .
- El CV procesa las cadenas de base de datos [fit_transform()] aprendiendo entonces el diccionario de vocabulario de 3-Words y nos devuelve para cada cadena un vector de lenguaje (y cada fila como un array de conteo de oapraiciones del vocabulario N-Words identificadas del conjunto de datos.

SEQUENCE: ATGAACGAAA	PROTEIN_ID: 4			
atgaac	tgaacg	gaacga		
	tgaacg	gaacga	aacgaa	
		gaacga	aacgaa	acgaaa

```
cv = CountVectorizer(ngram_range=(3,3))
X = cv.fit_transform(seq_texts)
```

X.shape → (6061, 65570)

Donde:
6061 secuencias
#65570 frases de 3 palabras de 6 nucleótidos

- Una vez generado el vector X e Y, dividimos un conjunto para entrenamiento y otro para test. Realizar el entrenamiento del clasificador el test.
- Creamos el clasificador MNB que se creara al entrenarlo con 65570 entradas, un contador/valor por cada 3-WORD identificada como lenguaje en los datos de entrenamiento.

```
classifier = MultinomialNB(alpha=0.1)
```

```
y_pred = classifier.predict(X_test)
```

Statistics:

Confusion matrix

Predicted	0	1	2	3	4	5	6
Actual							
0	164	0	0	0	1	0	2
1	0	144	0	0	0	0	6
2	0	0	105	0	0	0	4
3	1	1	0	164	1	0	0
4	0	0	0	0	195	0	2
5	0	0	0	0	0	71	1
6	0	0	0	2	0	0	349

accuracy = 98.26875515251443 %
precision = 98.30808426399021 %
recall = 98.26875515251443 %

- La etapa de validación → nos da un 98% de precisión!! **FUNCIONA**!

Multinomial Naive Bayes Classifier Machine Learning

- La etapa de clasificación en “real” consistirá en clasificar una cadena de entrada
- Convertirla en un array de contadores de 3-WORD sobre el lenguaje ya aprendido por el CV, de la forma (1, 65570)
- Una vez esto, ese array será la entrada al clasificados MNB.

```
def clasiffierMNB(cbp, cv, classifier):

    seq_texts = list(getKmers(cbp.querry_seq,6))
    seq_texts = ' '.join(seq_texts)

    seq = list()
    seq.append(seq_texts)

    # encode document
    X_seq = cv.transform(seq)

    print(X_seq.shape)
    # clasification
    y_pred = classifier.predict(X_seq)
    print("Predictions x_test: ", str(y_pred[0]))
```

- Resultados obtenidos:
  - Hardware domestico para entrenar un sistema con 6000 secuencias de longitud variable
  - 6 clases de identificación
  - 98 % de precisión
  - Rapidez y flexibilidad.
  - Time to fit = 14.602660179138184 sg.
  - Time to prediction: 0.005034446716308594 sg

querry_seq: ACGGTGCTACTCAAGGCCCGGGAAGGTGGCGGTGGAAATCGCA

Results:

The sequence is identified with the protein class: 6

Protein class table:

0 - G protein coupled receptors
1 - Tirosina kynase
2 - Tirosina Phosfate
3 - Synthetases
4 - Synthase
5 - Ion channel
6 - Transcription Factor

LSTM Classifier – Machine Learning

- Un tipo de red neuronal que permite dar contexto son las redes neuronales recurrentes. (*Christopher Olah, (2015) Understanding LSTM Networks. https://colah.github.io/posts/2015-08-Understanding-LSTMs/*)
- Concretamente usaremos las LSTM, utilizadas por ejemplo, para el procesamiento del lenguaje natural PNL.
- El enfoque de contexto nos lo genera la propia arquitectura de la red LSTM.
  - La red LSTM ira procesando palabra a palabra, pasando en cada ciclo su salida a la entrada del análisis de la siguiente palabra, generando a la salida final de la cadena un dato de salida.
- **OBJETIVO**: una red neuronal que a un fragmento de cadena de entrada nos identifique si es o no cierto patógeno conocido (Sars-CoV-2)

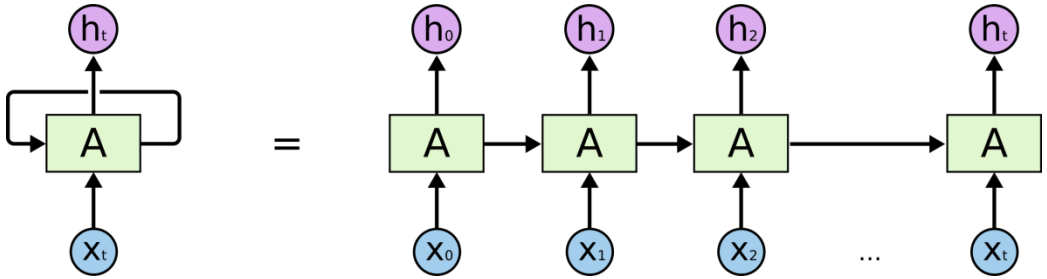


Imagen tomada de: (*Christopher Olah, (2015) Understanding LSTM Networks. https://colah.github.io/posts/2015-08-Understanding-LSTMs/*)

LSTM Classifier – Machine Learning

- Implementación de la solución:
  - Para cada secuencia se generara el array de kmers (K=6).
  - A cada k-mer único se le asignara un Id de palabra único.
  - Por tanto una cadena pasa a representarse como un array de números [12, 152, 32, 16, 34, 62, 22, 39]
  - Se procesaran todos los arrays de representación de la base de datos indicando a la red neuronal si es o no es el patógeno conocido.
  - Esta implementación esta basada en las ideas formuladas para la identificación de patrones de Neanderthal con LSTM por (*Nikolay Oskolkov, (2019), LSTM to Detect Neanderthal DNA. https://towardsdatascience.com/lstm-to-detect-neanderthal-dna-843df7e85743)*

	A	B	C	D	E	F	G
1	MOLECULE_TYPE	ID	PATHOGEN	PATHOGEN_NAME	SEQ_ID	NAME_SECUENCE	SEQUENCE
2	dna	536	1	SARS-CoV-2	NC_045512.2	Severe acute respiratory syndrome	AAAAACAGTAAAGTACAAATAGGAGAGTACACCTTTGAAAAAGGTGACTATGGTGATGCTGTTGTTTACCGAGGT
3	dna	121	1	SARS-CoV-2	NC_045512.2	Severe acute respiratory syndrome	AAAAACAGTAAAGTACAAATAGGAGAGTACACCTTTGAAAAAGGTGACTATGGTGATGCTGTTGTTTACCGAGGT
4	dna	516	1	SARS-CoV-2	NC_045512.2	Severe acute respiratory syndrome	AAAAACAGTACAATTCTGTGATGCCATGCGAAATGCTGGTATTGTTGGTGTACTGACATTAGATAATCAAGATCT
5	dna	101	1	SARS-CoV-2	NC_045512.2	Severe acute respiratory syndrome	AAAAACAGTACAATTCTGTGATGCCATGCGAAATGCTGGTATTGTTGGTGTACTGACATTAGATAATCAAGATCTC
6	dna	976	0	OTHER	XX_XXXXX	Other Secuence (NO coronavirus 2)	AAAAAGGAAGCTGGCTATGCAGTATCACCGTATCCCACACAGTTCTCTTGAAATCAGCACCCTCGGGCTGGGAAC
7	dna	347	0	OTHER	XX_XXXXX	Other Secuence (NO coronavirus 2)	AAAAAGGAAGCTGGCTATGCAGTATCACCGTATCCCACACAGTTCTCTTGAAATCAGCACCCTCGGGCTGGGAAC
8	dna	574	1	SARS-CoV-2	NC_045512.2	Severe acute respiratory syndrome	AAAAATCTTAGGGAATTTGTGTTTAAGAATATTGATGGTTATTTTAAAATATATTCTAAGCACACGCCTATTAAT
9	dna	159	1	SARS-CoV-2	NC_045512.2	Severe acute respiratory syndrome	AAAAATCTTAGGGAATTTGTGTTTAAGAATATTGATGGTTATTTTAAAATATATTCTAAGCACACGCCTATTAATTT
10	dna	795	1	SARS-CoV-2	NC_045512.2	Severe acute respiratory syndrome	AAAACAAATGTGTCAATTTCAACTTCAATGGTTTAACAGGCACAGGTGTTCTTACTGAGTCTAACAAAAAGTTTC
11	dna	1170	0	OTHER	XX_XXXXX	Other Secuence (NO coronavirus 2)	AAAACGGCCCAGCAGAGCCAGATCGCGCCCGGTGTTAAACAGGCCGGCGTTGAAGAAATAGGGGCTCTTGCGCC
12	dna	440	1	SARS-CoV-2	NC_045512.2	Severe acute respiratory syndrome	AAAACTTACTGACAATGTATACATTAAAAATGCAGACATTGTGGAAGAAGCTAAAAAGGTAAAACCAACAGTGG
13	dna	25	1	SARS-CoV-2	NC_045512.2	Severe acute respiratory syndrome	AAAACTTACTGACAATGTATACATTAAAAATGCAGACATTGTGGAAGAAGCTAAAAAGGTAAAACCAACAGTGG
14	dna	690	1	SARS-CoV-2	NC_045512.2	Severe acute respiratory syndrome	AAAAGAATAACTTACCTTTTAAGTTGACATGTGCAACTACTAGACAAGTTGTTAATGTTGTAACAACAAAGATAG
15	dna	686	1	SARS-CoV-2	NC_045512.2	Severe acute respiratory syndrome	AAAATGTGAAGAATCATCTGCAAAATCAGCGTCTGTTTACTACAGTCAGCTTATGTGTCAACCTATACTGTTACT
16	dna	1025	0	OTHER	XX_XXXXX	Other Secuence (NO coronavirus 2)	AAACATTGTTTCATTTAACTTATCCCACTCTTCATGAAAAAGAAATACGCGGCCAGAGAAGTGGAAAGTTGATCA
17	dna	396	0	OTHER	XX_XXXXX	Other Secuence (NO coronavirus 2)	AAACATTGTTTCATTTAACTTATCCCACTCTTCATGAAAAAGAAATACGCGGCCAGAGAAGTGGAAAGTTGATCA(

LSTM Classifier – Machine Learning

```
tokenizer = Tokenizer()
tokenizer.fit_on_texts(seq_texts) # tokenizer the word in each secuence
encoded_docs = tokenizer.texts_to_sequences(seq_texts) # Transform unique each token in a integer value
max_length = max([len(s) for s in encoded_docs]) # 135 max langth of all secuences
X = pad_sequences(encoded_docs, maxlen = max_length, padding = 'post') # the context is determinate in less 100 nucleotid
```

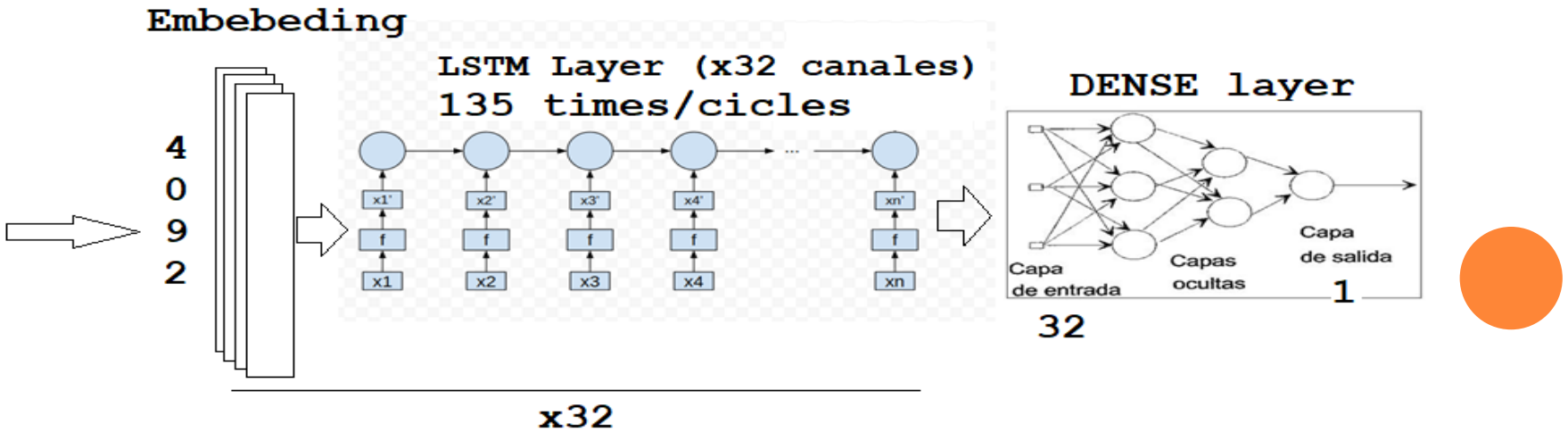
```
vocab_size = len(tokenizer.word_index) + 1
```

```
model = Sequential()
model.add(Embedding(vocab_size, 32))
model.add(LSTM(32))
model.add(Dense(1, activation='sigmoid'))
```

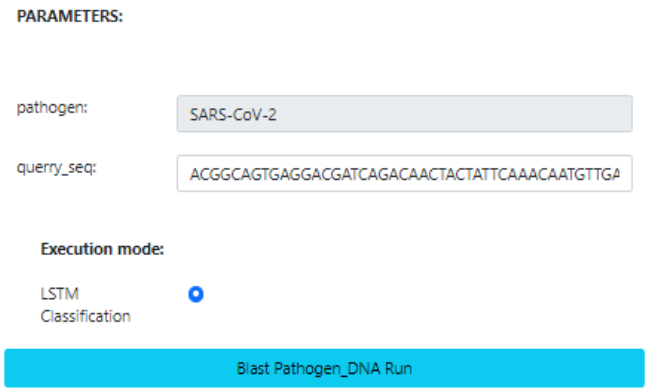
```
vocab_size = 4092
X_train.shape = (996, 135)
X_test.shape = (249, 135)

# Donde 135 es el máximo tamaño en
palabras  6-mer de la base de datos.
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding (Embedding)        (None, None, 32)          130944
_________________________________________________________________
lstm (LSTM)                  (None, 32)                8320
_________________________________________________________________
dense (Dense)                (None, 1)                 33
=================================================================
Total params: 139,297
Trainable params: 139,297
Non-trainable params: 0
_________________________________________________________________
```



LSTM Classifier – Machine Learning




```
def clasiffierLSTM(cbp, tokenizer, model):

    seq_texts = list(getKmers(cbp.querry_seq,6))
    seq_texts = ' '.join(seq_texts)

    seq = list()
    seq.append(seq_texts)

    # encode document
    X_seq = tokenizer.texts_to_sequences(seq)

    y_pred = model.predict_classes(X_seq)
    print("Predictions x_test: ", str(y_pred[0]))

    if y_pred[0]==1:
        return "Identificate pathogen: SARS-CoV-2
    else:
        return "No identificate pathogen SARS-CoV-2
```

- Resultados obtenidos:
  - Hardware domestico para entrenar un sistema con 6000 secuencias de longitud variable
  - Identificación de un patógeno mezclando una base de datos de un patógeno con cadenas de otro origen
  - 91 % de precisión en la etapa de validación
  - Rapidez y flexibilidad.
  - Time to fit = 10.1767761707 sg.
  - Time to prediction = 0.30618071556 sg.

Statistics:

Confusion matrix

	0	1
0	115	6
1	15	113

accuracy = 91.56626462936401 %

Results:

Identificate pathogen: SARS-CoV-2 | Severe acute respiratory syndrome coronavirus 2 isolate Wuhan-Hu-1(complete genome)

Conclusiones

- Los **OBJETIVOS** del proyecto se han cumplido.
  - El objetivo principal del proyecto era el estudio técnico e implementación de un sistema de alineamiento de secuencias genómicas
  - La implementación funcional de un sistema de identificación de patrones biológicos basado en arquitecturas de deep learning.

- La **VIAS FUTURAS**
  - Sistemas de laboratorio e investigación
  - Sistemas médicos comerciales: como la detección de cadenas en test de detección de virus, patologías o predisposición genética, compatibilidad genética …
  - Distintas arquitecturas y modelos que mejor se adapte a la naturaleza de cada problema

¡MUCHAS GRACIAS!