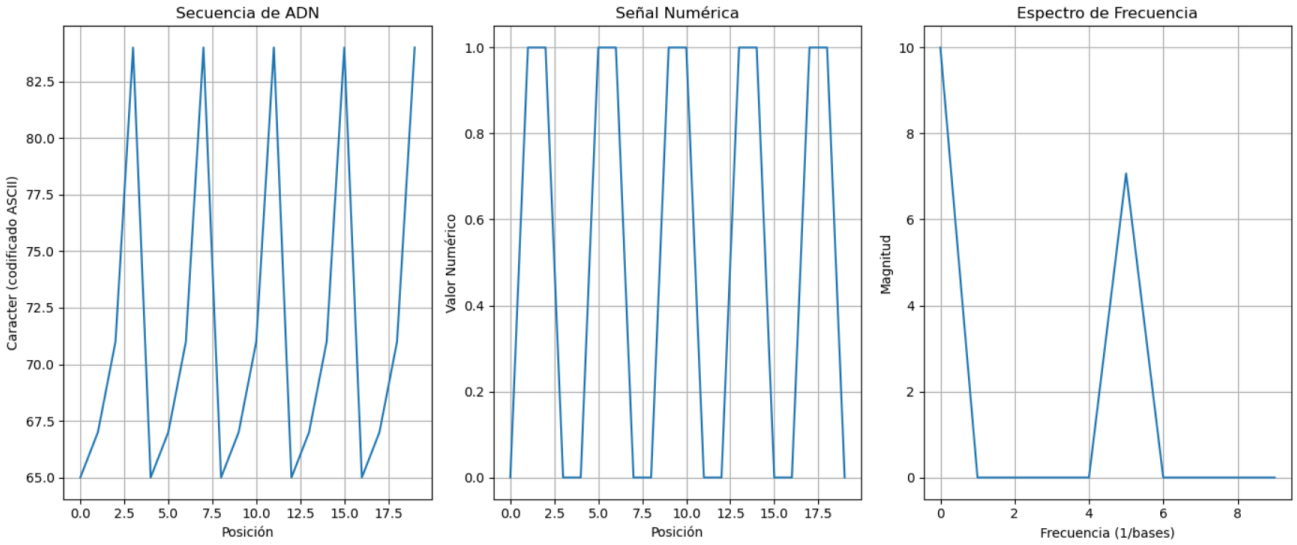
Tratamiento de Señales Genómicas

Codificación y Decodificación

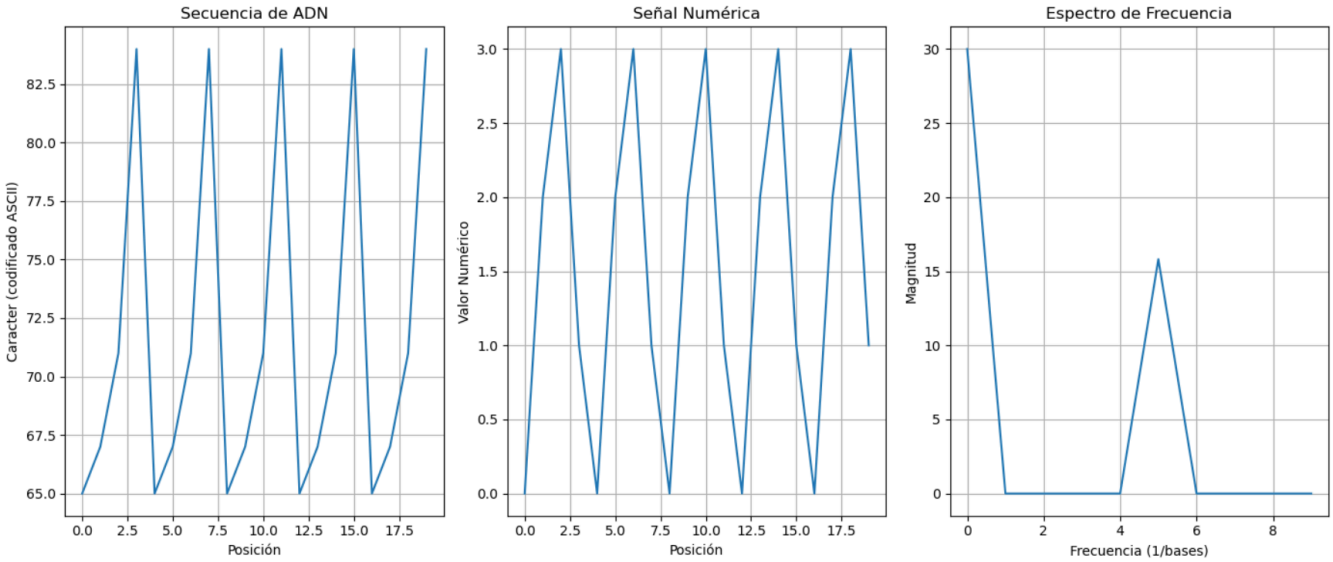
Partimos de un CoDec básico que permite codificar la señal de ADN de 3 maneras diferentes, y posteriormente decodificarlas de la misma manera que se codificó. Esto permite unas precisiones de las siguientes magnitudes:

Tipo	Precisión	Codificación
Complementario	50%	('A': 0, 'T': 0, 'C': 1, 'G': 1)
Único	100%	('A': 0, 'T': 1, 'C': 2, 'G': 3)
EIIP	100%	('A': 0.1260, 'C': 0.1340, 'G': 0.0806, 'T': 0.1335)

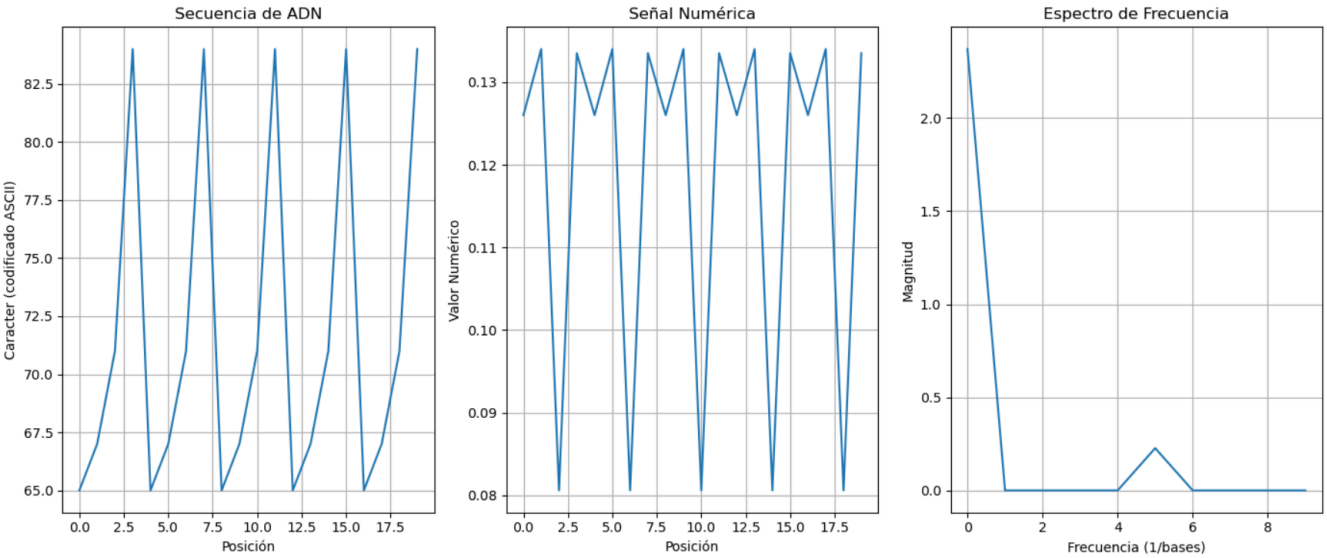
Además permite analizar la periodicidad de las señales, dando lugar a resultados como:



Tipo 'Complementario'



Tipo 'Único'



Tipo 'EIIP'

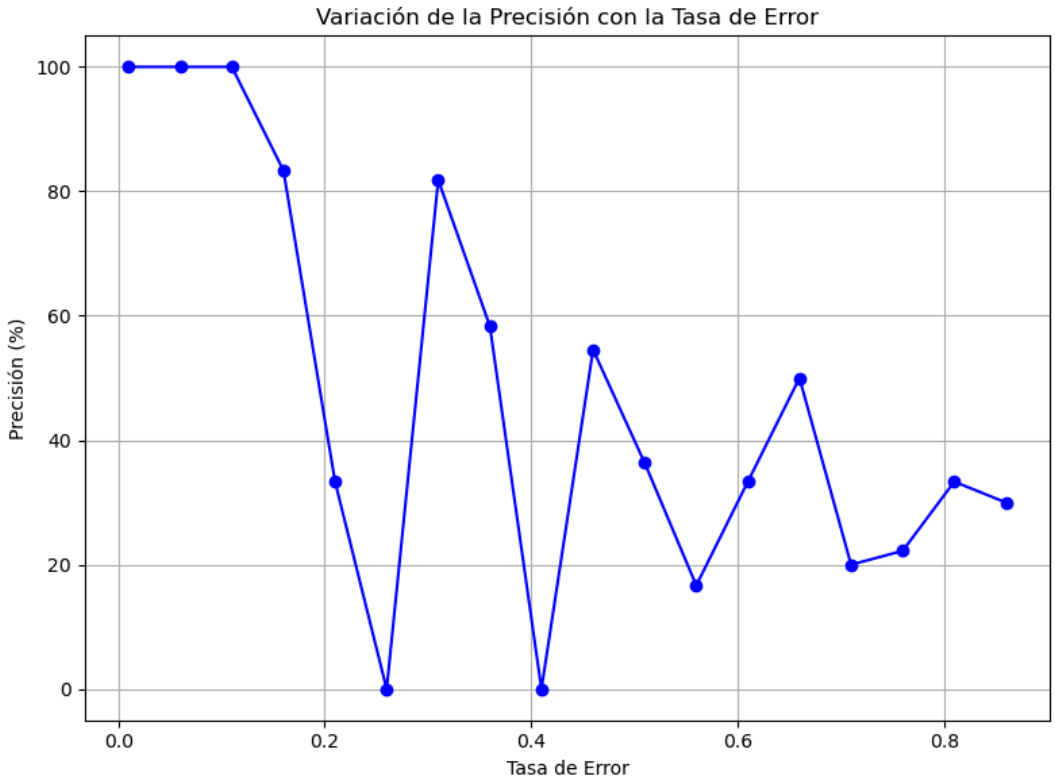
Como podemos ver para los tres tipos de codificación distinta, la secuencia codificada permanece igual, pero la gráfica de la señal numérica y el espectro de la frecuencia, varía entre ellos.

Evaluación de Robustez

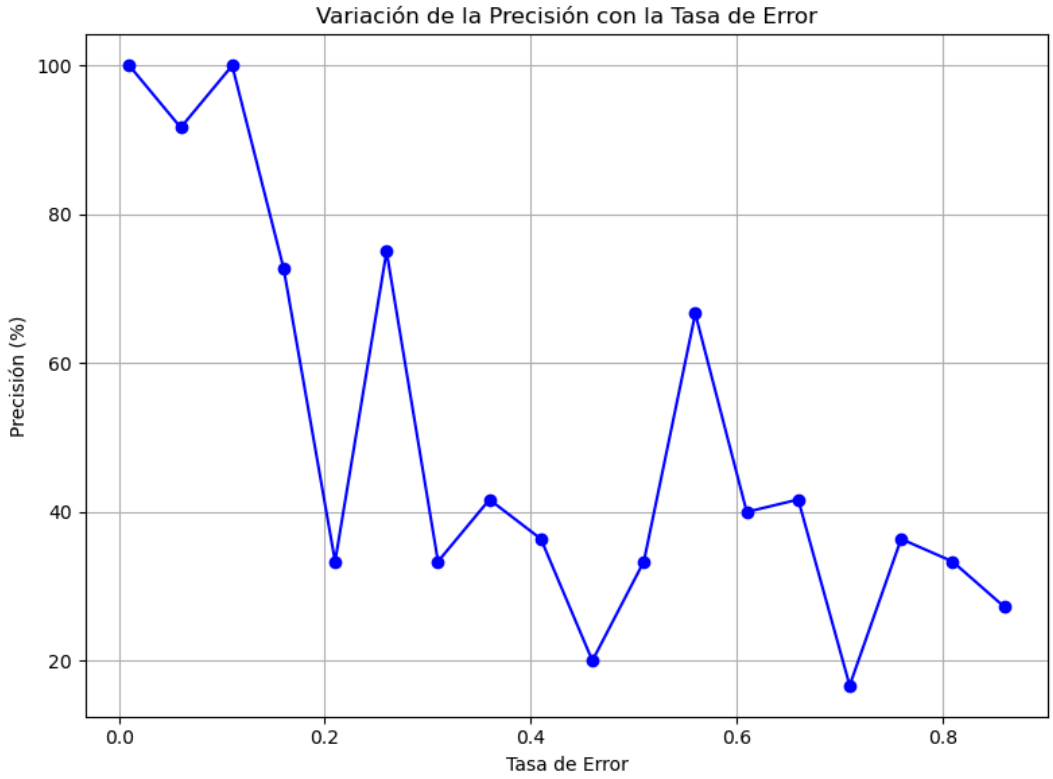
Para hacer los resultados más realistas, se han añadido distintos tipos de ruido a la secuencia recibida:

- Sustitución: cambiar una base por otra aleatoria.
- Inserción: añadir una base aleatoria extra.
- Borrado: eliminar una base aleatoria.

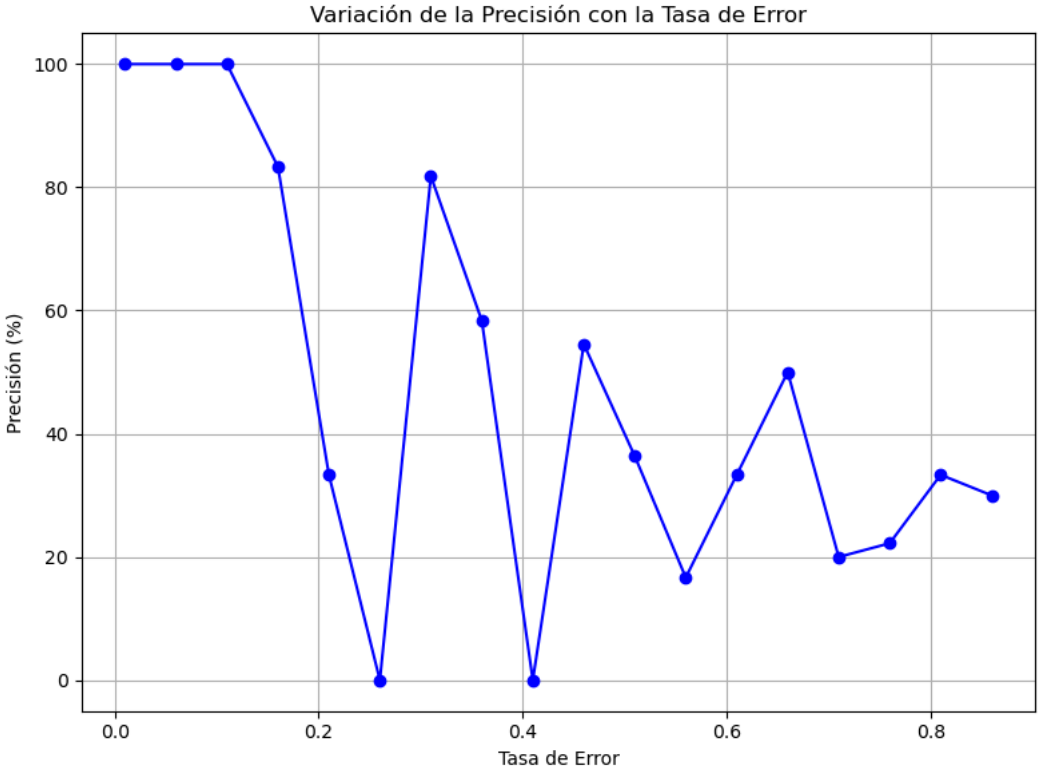
De este modo, conseguimos aumentar el porcentaje de error entre la secuencia original y la decodificada, asemejando el sistema a uno más prácticamente realizable. Además, también se ha configurado una tasa de error por defecto con el mismo objetivo. Variando esta tasa, se han llegado a los siguientes resultados:



Tipo 'Complementario'



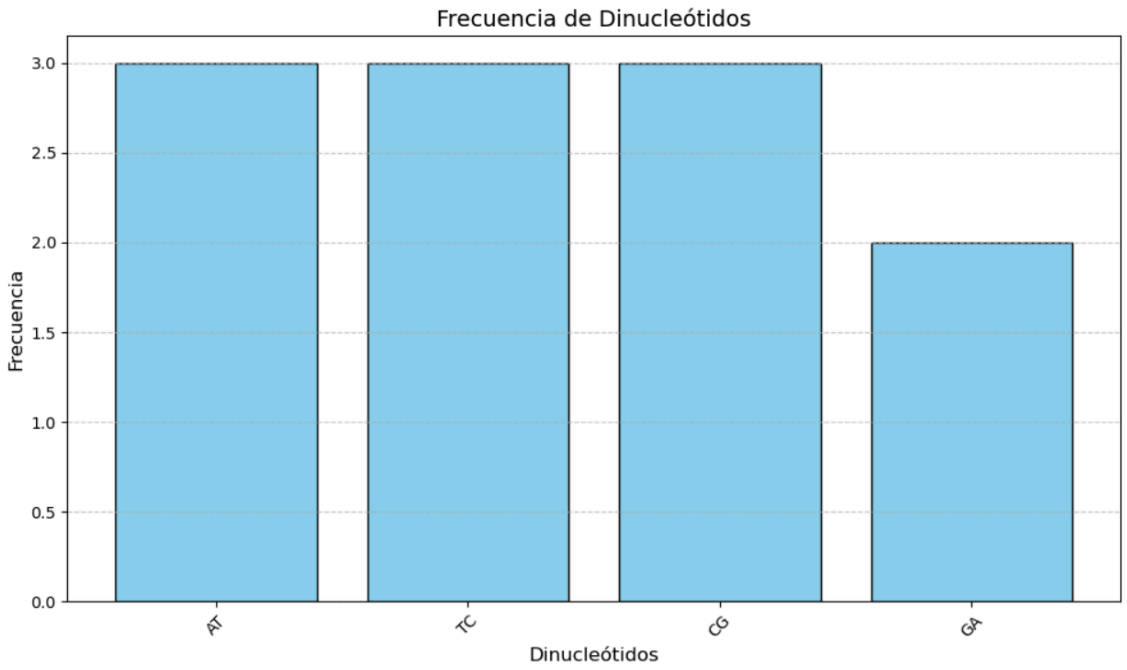
Tipo 'Único'



Tipo 'EIIP'

Representación de Dinucleótidos y Entropía

Para la secuencia de ADN "ATCGATCGATCG", se ha habilitado una función que puede contar el total de cada combinación de bases de nucleótidos, y otra para medir su frecuencia relativa. Para este ejemplo en concreto, obtenemos que:




```
-------------------------------------------
Frecuencia relativa de dinucleótidos:
   -  'AT': 0.27
   -  'TC': 0.27
   -  'G': 0.27
   -  'GA': 0.18181818181818182
-------------------------------------------
```

Por ejemplo, para el caso de 'AT', 0.27 quiere decir que la secuencia de ADN está formada por un 27% de dinucleótidos 'AT'.

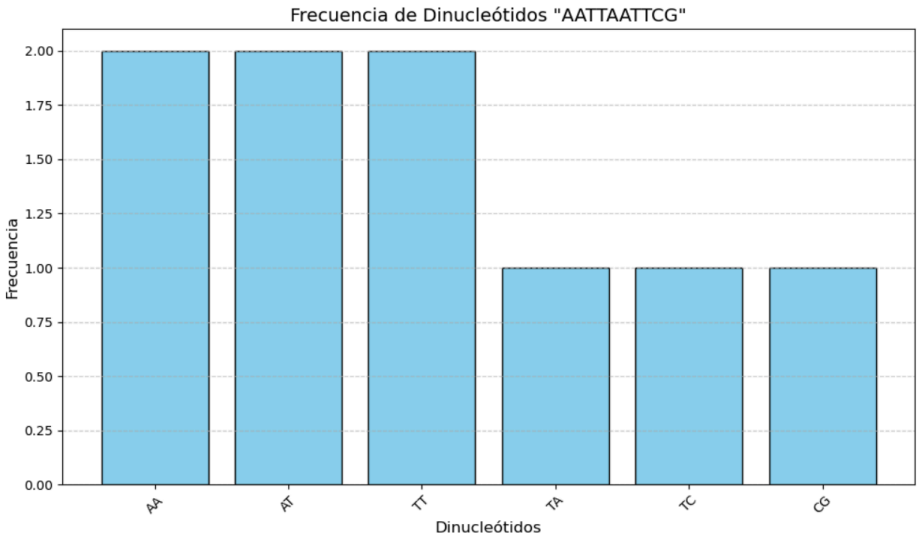
También se ha añadido un cálculo de la entropía para los nucleótidos y dinucleótidos, llegando a los siguientes resultados:

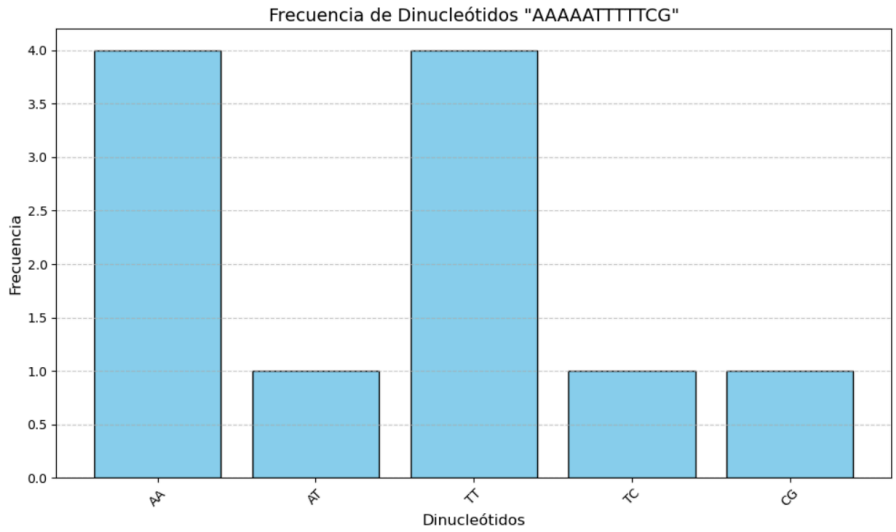
```
---------------------------------------
Entropía de la secuencia:  2.0
Entropía de los dinucleótidos: 1.43
---------------------------------------
```

Esto es porque, debido a que la cadena original de ADN tiene el mismo número de A, C, T y G, alcanza la máxima entropía posible; ya que sus elementos muestran la misma distribución (tienen la misma probabilidad). Sin embargo, al agruparlos en dinucleótidos, esta entropía cambia, ya que las probabilidades de las agrupaciones son distintas.

Para comprobar esto, podemos utilizar otras cadenas y observar los nuevos valores calculados:

Secuencia	Entropía de Nucleótidos	Entropía de Dinucleótidos
AATTAATTCG	1.72	2.88
AAAAATTTTTCG	1.65	1.91





La entropía puede ser útil para comparar regiones codificantes y no codificantes:

- Regiones codificantes: menos diversidad (restricciones funcionales).
- Regiones no codificantes: mayor diversidad (menor organización).

Clasificador de Regiones Codificantes y No Codificantes

Ahora, vamos a añadir un clasificador capaz de identificar las categorías (regiones codificantes y regiones no codificantes) de señales de adn generadas artificialmente; para estudiar qué tan preciso es, su exhaustividad, su promedio armónico, y sus métricas globales (promedio macro, promedio ponderado y precisión global).
Para entender los resultados, es necesario conocer la interpretación de estos:

Categorías:
- 0: regiones no codificantes.
- 1: regiones codificantes.

Precisión:
Porcentaje de predicciones correctas para esta clase respecto al total de veces que el modelo predijo dicha clase.

$$\text{Precision} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Positives (FP)}}$$

Recall (exhaustividad):
Porcentaje de predicciones correctas para esta clase respecto al total de casos reales que pertenecen a esta clase.

$$\text{Recall} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negatives (FN)}}$$

F1-Score:
Métrica que combina precisión y recall en un único valor (promedio armónico). Es útil cuando las clases están desbalanceadas.

$$\text{F1-Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

Support:
Cantidad de ejemplos reales en la clase. Por ejemplo:

- Para tipo 0, support = 10 quiere decir que hay 10 ejemplos para esta clase.
- Para tipo 1, support = 12 quiere decir que hay 12 ejemplos para esta clase.

Macro avg (Promedio macro):
Promedio simple de las métricas por clase.

Weighted avg (Promedio ponderado):
Promedio ponderado por el número de ejemplos en cada clase.

Accuracy (Precisión global):
Proporción de predicciones correctas sobre el total de ejemplos.

Una vez sabemos lo que significa cada cosa, podemos seguir adelante.

Para el script básico, en el que se genera un dataset con nucleótidos elegidos al azar, todos los clasificadores dan lugar a los mismos resultados:

```
------------------------------------------------------------
Reporte de Clasificación:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        19
           1       1.00      1.00      1.00        21

    accuracy                           1.00        40
   macro avg       1.00      1.00      1.00        40
weighted avg       1.00      1.00      1.00        40
------------------------------------------------------------
```

Esto se debe a que en la generación de datasets, no se produce ningún error ni simulamos ningún ruido, dando lugar a una clasificación perfecta. Sin embargo, si añadimos ruido (se ha establecido una norma de producir un ruido para el 5% de las etiquetas generadas, las cuales son fundamentales para saber si las secuencias son ‘codificantes’ o ‘no codificantes’) y un peso diferente para cada nucleótido a la hora de generarlos; se le complica considerablemente el trabajo al clasificador. Con ello, obtenemos:

- **Decision Tree Classifier:**

```
------------------------------------------------------------
Reporte de Clasificación:  clf
              precision    recall  f1-score   support

           0       0.48      0.55      0.51        20
           1       0.47      0.40      0.43        20

    accuracy                           0.48        40
   macro avg       0.47      0.48      0.47        40
weighted avg       0.47      0.47      0.47        40
------------------------------------------------------------
```

- **Random Forest Classifier:**

```
------------------------------------------------------------
Reporte de Clasificación:  rf
              precision    recall  f1-score   support

           0       0.48      0.50      0.49        20
           1       0.47      0.45      0.46        20

    accuracy                           0.48        40
   macro avg       0.47      0.47      0.47        40
weighted avg       0.47      0.47      0.47        40
------------------------------------------------------------
```

- **K Nearest Neighbors (con k = 3):**

```
------------------------------------------------------------
Reporte de Clasificación:  knn
              precision    recall  f1-score   support

           0       0.63      0.63      0.63        19
           1       0.67      0.67      0.67        21

    accuracy                           0.65        40
   macro avg       0.65      0.65      0.65        40
weighted avg       0.65      0.65      0.65        40
------------------------------------------------------------
```

- **Support Vector Machine (kernel lineal):**

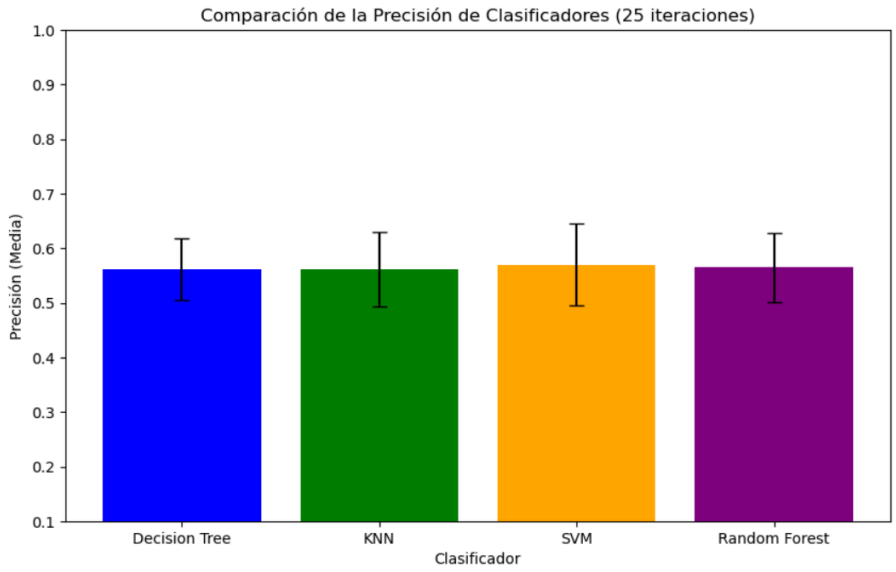
```
------------------------------------------------------------
Reporte de Clasificación:  svm
              precision    recall  f1-score   support

           0       0.44      0.94      0.60        17
           1       0.75      0.13      0.22        23

    accuracy                           0.48        40
   macro avg       0.60      0.54      0.41        40
weighted avg       0.62      0.47      0.38        40
------------------------------------------------------------
```

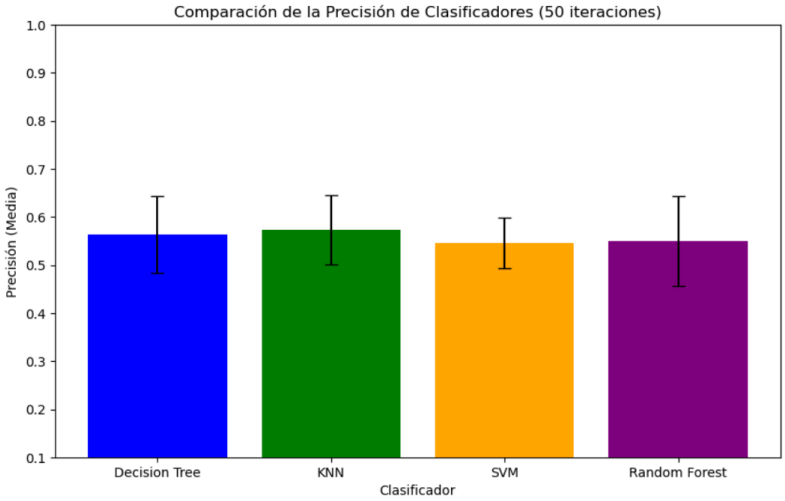
Teniendo en cuenta que para cada ejecución, se obtiene un resultado distinto para cada clasificador, se ha propuesto un cálculo de medias y desviaciones (a partir de un número de 'n' ejecuciones) para poder aproximarse a una precisión global de detección. Con ello, se ha obtenido lo siguiente:

- **n = 25:**



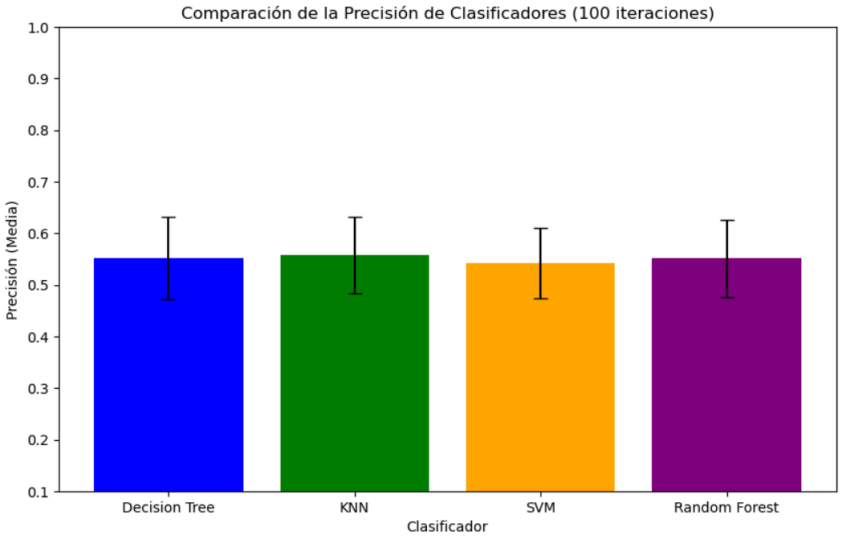

```
------------------------------------------------------------------
Decision Tree-Media de precisión: 0.56, Desviación estándar: 0.06
KNN-Media de precisión: 0.56, Desviación estándar: 0.07
SVM-Media de precisión: 0.57, Desviación estándar: 0.07
Random Forest-Media de precisión: 0.56, Desviación estándar: 0.06
------------------------------------------------------------------
```

- **n = 50:**



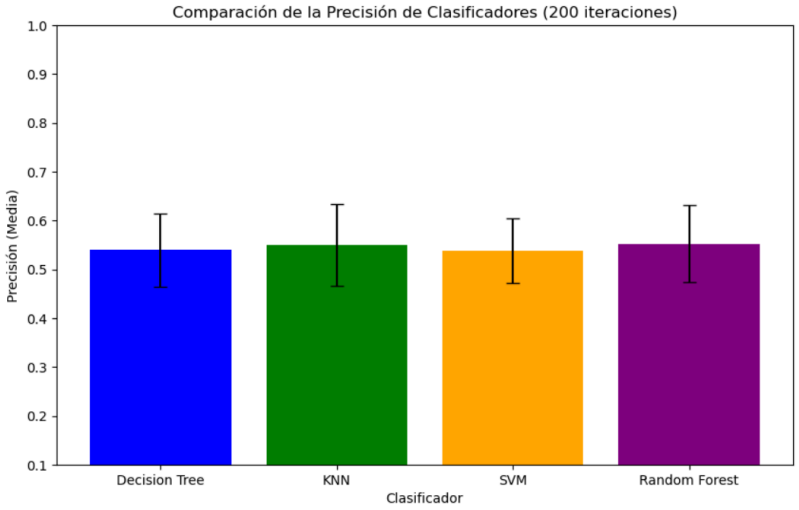

```
------------------------------------------------------------------
Decision Tree-Media de precisión: 0.56, Desviación estándar: 0.08
KNN - Media de precisión: 0.57, Desviación estándar: 0.07
SVM - Media de precisión: 0.55, Desviación estándar: 0.05
Random Forest-Media de precisión: 0.55, Desviación estándar: 0.09
------------------------------------------------------------------
```

- **n = 100:**




```
------------------------------------------------------------------
Decision Tree-Media de precisión: 0.55, Desviación estándar: 0.08
KNN-Media de precisión: 0.56, Desviación estándar: 0.07
SVM-Media de precisión: 0.54, Desviación estándar: 0.07
Random Forest-Media de precisión: 0.55, Desviación estándar: 0.07
------------------------------------------------------------------
```

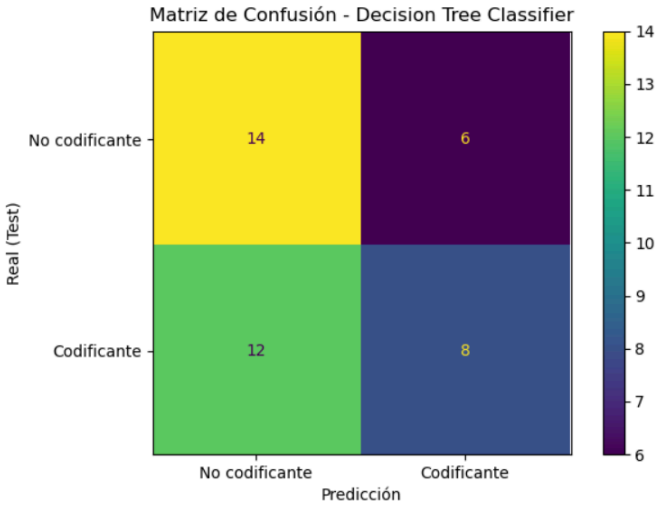
- **n = 200:**




```
------------------------------------------------------------------
Decision Tree-Media de precisión: 0.54, Desviación estándar: 0.08
KNN-Media de precisión: 0.55, Desviación estándar: 0.08
SVM-Media de precisión: 0.54, Desviación estándar: 0.07
Random Forest-Media de precisión: 0.55, Desviación estándar: 0.08
------------------------------------------------------------------
```

Comparando las ejecuciones para todos los distintos valores de n (25, 50, 100 y 200), parece seguro concluir que todos los clasificadores realizan la tarea con precisiones más o menos similares, lo que significa que de ahora en adelante, mientras los datasets se mantengan iguales o cuya modificación sea ligera y prácticamente invariante, se podrá utilizar el que prefiramos, sin miedo a creer que es el incorrecto.

Por ejemplo, se ha decidido continuar con el clasificador Decision Tree Classifier. Gracias a la librería de sklearn.metrics, somos capaces de representar una matriz de confusión, para ver qué falsos positivos y qué falsos negativos hemos obtenido:



Gracias a la matriz de confusión, se pueden interpretar fácilmente los resultados obtenidos. Del dataset generado, se han obtenido 20 regiones 'codificantes' y 20 'no codificantes'. Para el clasificador que hemos escogido, se han detectado 26 regiones 'no codificantes', de las cuales 12 son falsos positivos; y 14 regiones 'codificantes', de las cuales 6 son falsos positivos. Esto quiere decir que ha tenido una precisión del 55 %, tal y como nos muestra nuestro script:

```
----------------------------------------
 Precision Decision Tree Classifier:  0.55
----------------------------------------
```

¿Y qué ocurriría si modificamos el porcentaje de ruido en las etiquetas de los datasets? La precisión se reduciría de forma inversamente proporcional a esa tasa. Veámoslo gráficamente:
Para un número de ejecuciones igual a 100 (estableciendo una media para cada tasa para contrarrestar la aleatoriedad en cierta medida), obtenemos las siguientes precisiones:

Como se puede observar, desde una tasa de 0.05 hasta 0.5, la precisión se reduce bastante, debido a que las etiquetas erróneas se producen con una mayor frecuencia, lo que confunde al modelo durante el entrenamiento. Sin embargo, desde una tasa de 0.6 en adelante, podemos ver un efecto de 'rebote' en la precisión, formando una especie de 'U' a lo largo de toda la gráfica. Este fenómeno se produce ya que según se alcanzan mayores valores de tasas de ruido, las etiquetas generadas se acercan a ser completamente aleatorias, y como lo que debe hacer el clasificador es decidir si es 'codificante' o 'no codificante', esa probabilidad ronda el 50%.

Conclusión

En este trabajo, se ha desarrollado un enfoque computacional para la clasificación de regiones codificantes y no codificantes en secuencias de ADN, combinando conceptos de procesamiento digital de señales (DSP) con técnicas de aprendizaje automático. A través de la representación de las secuencias en términos de entropía de nucleótidos y dinucleótidos, se han generado características que capturan patrones informativos de las regiones analizadas. La robustez del sistema se ha evaluado en escenarios con distintos niveles de ruido en las etiquetas, mostrando cómo este afecta el rendimiento de varios clasificadores. Los resultados obtenidos reflejan el impacto del ruido y destacan la importancia de técnicas como los bosques aleatorios para mantener un desempeño estable en condiciones adversas.

Además, se han llevado a cabo análisis complementarios, como la interpretación de las matrices de confusión y la evaluación de la precisión media en múltiples ejecuciones, para garantizar la fiabilidad de las conclusiones. Este proyecto no solo contribuye a demostrar el potencial de herramientas como la entropía y el aprendizaje supervisado en la genómica computacional, sino que también establece un marco flexible que puede extenderse a estudios más complejos o aplicarse a datos biológicos reales. En resumen, este trabajo representa un paso inicial hacia la integración de técnicas avanzadas de procesamiento de datos en el análisis y clasificación de información genética, con posibles aplicaciones en biología molecular y bioinformática.