Una aplicación de Modelos Ocultos de Markov y Maquinas de Soporte Vectorial para la Clasificación de Conotoxinas

Miguel Angel Peñaloza Pérez

Centro de Investigación Científica y de Educación Superior de Ensenada, Baja California

Abstract. Una conotoxina es un peptido generado en el organismo de caracoles marinos de la especie cono, esta sustancia es útil para paralizar a sus presas y obtener su alimento. Actualmente la comunidad cientifica en el area de la farmacología y las neurociencias estan trabajando intensamente en la secuenciacion de conotoxinas desconocidas debido a su potencial para el desarrollo de nuevos farmacos como potenciales fuentes de tratamientos terapeuticos. Sin embargo, el proceso de secuenciación es lento y costoso debido a las necesidades de personal altamente capacitado y instrumentos especializados. Sin embargo los métodos desarrollados en el área de inteligencia artificial ofrecen una alternativa eficiente para el descubrimiento de nuevas conotoxinas. En este proyecto se propone el desarrollo de un modelo oculto de markov que sus resultados van a ser entrada de una maquina de soporte vecotrial para la clasificación de conotoxinas desconocidas. El modelo fue entrenado a traves de solo los peptidos maduros reportados en las secuencias de ConoServer. Los resultados encontrados muestran que los modelos ocultos de markov siguen siendo competitivos a la par de métodos contemporaneos como las modelos de lenguaje especializados en biología. A su vez se identifica una maximización de las capacidades de los modelos ocultos de markov cuando se combinan con métodos como las maquinas de soporte vectorial.

Keywords: Conotoxinas · Modelos Ocultos de Markov · Maquinas de Soporte Vectorial.

1 Introducción

Las conotoxinas son peptidos neurotóxicos que son producidos en el organismo del caracol marino de la specie cono. Adicionalmente se ha identificado que estas conotoxinas tienen una alta posibilidad de ser la base para el desarrollo de nuevos medicamentos para tratamientos terapeúticos. (Li et al., 2025, Carrillo et al., 2025)

Actualmente las comunidades científicas se han dedicado a almacenar secuencias de conotoxinas conformando un reporsitorio con la mayoria de conotoxinas secuenciadas hasta ahora: ConoServer. (Kaas et al., 2008 & 2012). Dentro de

las clasificaciones de conotoxinas se han identificado hasta ahora 16 familias de conotoxinas

Sin embargo, se estima que aun existen alrededor de un millón de conotoxinas desconocidas y solo se han secuenciado un pequeño porcentaje de todas ellas. (Laht et al., 2012) Lo anterior debido a que el proceso de secuenciación es lento y costoso ya que se necesita personal altamente capacitado, instrumentos y tecnología especializada.

Actualmente los grandes modelos de lenguaje especializados ofrecen una alternativa eficiente y muy competitiva para la clasificación automática de conotoxinas desconocidas. (Zhao et al., 2025; Xion et al., 2025). Adicionalmente los métodos tradicionales siguen ofreciendo una opción como es el caso de los Modelos Ocultos de Markov. En este proyecto se entreno un modelo oculto de Markov para la clasificación de conotoxinas desconocidas. (Carrillo et al., 2025)

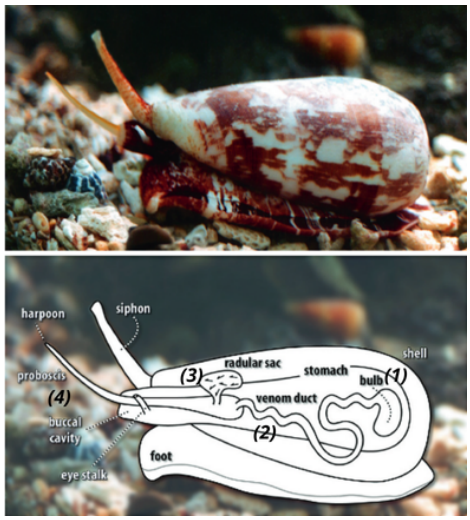

Fig. 1. Conus Striatus cazando, (1) bulbo venenoso que ayuda a impulsar el veneno,(2) ducto venoso donde se sintetizan las conotoxinas, (3) saco donde se almacenan los arpones para inmovilizar a su comida, (4) por donde consume los animales de la casería. (Fusetami & Kem, 2009)

2 Metodología

2.1 Los datos

Las secuencias para este estudio se descargarón de la base de datos Conoserver, una base de datos especializada en la secuencia de peptidos expresados por caracoles marinos carnivoros. Se descargarón los archivos de las secuencias de proteinas y posteriormente se eliminarión las secuencias sinteticas y se obtuvo un

total de secuencias. Posterior a este procesos de limpieza se siguio el procedimiento de la figura 1 . Los datos del conoserver para el entrenamiento del modelo oculto de Markov fueron solo la sección de peptidos maduros. El entrenamiento para el algoritmo de maquina de soporte vectorial fue a traves del etiquetado de las secuencias completas del conoserver siguiendo la siguiente rubrica: si el header de la secuencia no incluia la sección de precursor se asume que esa es el peptido maduro en otro caso se etqueto como cero. Por ultimo para realizar las predicciones se realizo una traducción de un transcriptoma a un proteoma a traves de transdecoder.

2.2 Modelos Ocultos de Markov

Los Modelos Ocultos de Markov son herramientas estadísticas que permiten modelar secuencias de observaciones, asumiendo que estas son generadas por un proceso oculto. En el caso de la identificación de conotoxinas, se utilizan para detectar patrones característicos en secuencias de proteínas, distinguiendo posiciones conservadas de regiones variables como inserciones o eliminaciones.

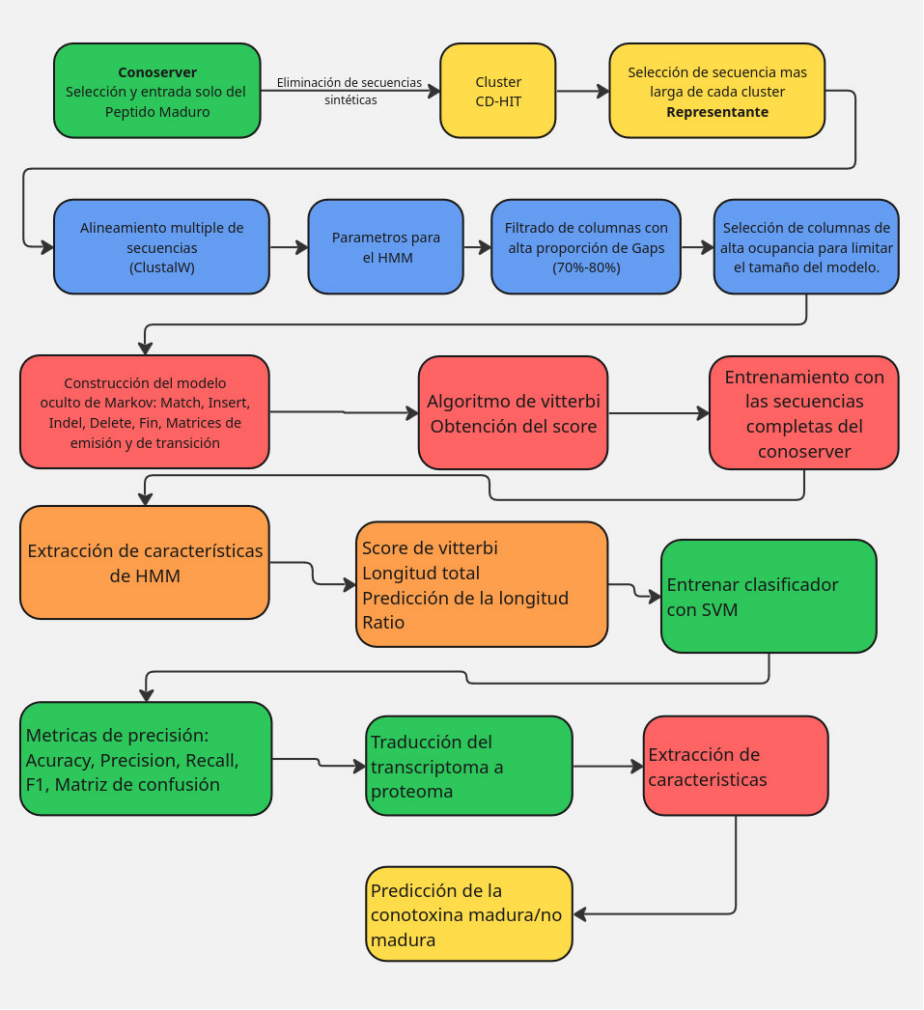

Fig. 2. Proceso para la construcción del algoritmo de clasificación de conotoxinas basado en modelos ocultos de Markov (HMM).

2.3 Estructura del Modelo

Un HMM se define a través de los siguientes componentes:

- **Estados ocultos:** Sea $S = \{S_1, S_2, \ldots, S_N\}$, donde cada estado representa, por ejemplo, una posición *Match* (M), *Insert* (I) o *Delete* (D). Adicionalmente se incluyen los estados especiales *Start* (S) y *End* (E).
- **Matriz de transición:** $T = \{a_{ij}\}$, con

$$a_{ij} = P(q_{t+1} = S_j \mid q_t = S_i)$$

que indica la probabilidad de pasar del estado S_i al estado S_j.

– **Matriz de emisión:** $E = \{b_i(o)\}$, donde

$$b_i(o) = P(o_t = o \mid q_t = S_i)$$

define la probabilidad de emitir el símbolo o en el estado S_i.

2.4 Probabilidad Total de una Secuencia

Para una secuencia de observaciones $O = (o_1, o_2, \ldots, o_T)$, la probabilidad de que el HMM genere dicha secuencia es:

$$P(O \mid \lambda) = \sum_{Q} P(O, Q \mid \lambda)$$

donde $Q = (q_1, q_2, \ldots, q_T)$ es la secuencia de estados ocultos y λ abarca todos los parámetros del modelo (matrices T y E, y las probabilidades iniciales).

2.5 Algoritmo Forward

El cálculo recursivo de la probabilidad acumulada hasta el tiempo t en el estado i se define como:

$$\alpha_t(i) = \left[\sum_{j=1}^{N} \alpha_{t-1}(j)\, a_{ji} \right] b_i(o_t)$$

con la condición inicial:

$$\alpha_1(i) = \pi_i\, b_i(o_1)$$

donde $\pi_i = P(q_1 = S_i)$ es la probabilidad de iniciar en el estado S_i.

2.6 Algoritmo Backward

De manera análoga, la función backward calcula la probabilidad de observar la secuencia restante desde el tiempo t en adelante, dado que el sistema se encuentra en el estado S_i:

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij}\, b_j(o_{t+1})\, \beta_{t+1}(j)$$

con la condición final:

$$\beta_T(i) = 1 \quad \forall i$$

2.7 Reestimación de Parámetros (Baum-Welch)

Para ajustar los parámetros del HMM se utilizan las siguientes variables:

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{P(O \mid \lambda)}$$

que representa la probabilidad de estar en el estado S_i en el tiempo t, y

$$\xi_t(i,j) = \frac{\alpha_t(i)\, a_{ij}\, b_j(o_{t+1})\, \beta_{t+1}(j)}{P(O \mid \lambda)}$$

que es la probabilidad conjunta de estar en S_i en t y en S_j en $t+1$.

Con estos, se actualizan las matrices:

$$a_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)}$$

$$b_i(o_k) = \frac{\sum_{\substack{t=1 \\ o_t=o_k}}^{T} \gamma_t(i)}{\sum_{t=1}^{T} \gamma_t(i)}$$

Para una consulta mas a profundidad sobre los modelos ocultos de Markov y su enfoque en bioinformática se sugiere la consulta de ().

Algorithm 1 Baum-Welch: Estimación de Parámetros de un HMM

Conjunto de secuencias codificadas S, número de estados N, número de símbolos M, iteraciones I π (prob. iniciales), A (matriz de transición), B (matriz de emisión)
Inicialización:

for $i = 1\ N$ **do** $\pi(i) \leftarrow \frac{1}{N}$ Inicializar A y B con valores aleatorios y normalizarlos en cada fila

for $iter = 1\ I$ **do** Reinicializar acumuladores $\pi_{\text{accum}}(i) \leftarrow 0$, $A_{\text{num}}(i,j) \leftarrow 0$, $A_{\text{den}}(i) \leftarrow 0$ $B_{\text{num}}(i,k) \leftarrow 0$, $B_{\text{den}}(i) \leftarrow 0$

for cada secuencia $O = (o_0, \ldots, o_{T-1}) \in S$ **do** Calcular paso Forward $(\alpha, c) \leftarrow \textsc{Forward}(O, A, B, \pi)$
Calcular paso Backward $\beta \leftarrow \textsc{Backward}(O, A, B, c)$
Calcular γ y ξ

for $t = 0\ T-1$ **do** $\gamma_t(i) \leftarrow \alpha_t(i) \cdot \beta_t(i)$ para todo i, y normalizar

for $t = 0\ T-2$ **do**

for $i = 1\ N$ **do**

for $j = 1\ N$ **do**

$$\xi_t(i,j) \leftarrow \frac{\alpha_t(i)\, A(i,j)\, B(j, o_{t+1})\, \beta_{t+1}(j)}{\sum_{p=1}^{N} \sum_{q=1}^{N} \alpha_t(p)\, A(p,q)\, B(q, o_{t+1})\, \beta_{t+1}(q)}$$

Acumular las probabilidades

for $i = 1\ N$ **do** $\pi_{\text{accum}}(i) \leftarrow \pi_{\text{accum}}(i) + \gamma_0(i)$

for $t = 0\ T-2$ **do**

for $i = 1\ N$ **do** $A_{\text{den}}(i) \leftarrow A_{\text{den}}(i) + \gamma_t(i)$

for $j = 1\ N$ **do** $A_{\text{num}}(i,j) \leftarrow A_{\text{num}}(i,j) + \xi_t(i,j)$

for $t = 0\ T-1$ **do**

for $i = 1\ N$ **do** $B_{\text{num}}(i, o_t) \leftarrow B_{\text{num}}(i, o_t) + \gamma_t(i)$ $B_{\text{den}}(i) \leftarrow B_{\text{den}}(i) + \gamma_t(i)$

Actualizar los parámetros del modelo

for $i = 1\ N$ **do** $\pi(i) \leftarrow \pi_{\text{accum}}(i)/|S|$

for $j = 1\ N$ **do** $A(i,j) \leftarrow \frac{A_{\text{num}}(i,j)}{A_{\text{den}}(i)}$

for $k = 1\ M$ **do** $B(i,k) \leftarrow \frac{B_{\text{num}}(i,k)}{B_{\text{den}}(i)}$ **return** π, A, B

Funciones Auxiliares

Algorithm 2 Forward Algorithm with Scaling

Secuencia $O = (o_0, \ldots, o_{T-1})$, matrices A, B, y vector π α y escala c Initialize: $\alpha_0(i) \leftarrow \pi(i) \cdot B(i, o_0)$, for $i = 1, \ldots, N$ $c_0 \leftarrow \sum_{i=1}^{N} \alpha_0(i)$
for $i = 1\ N$ **do** $\alpha_0(i) \leftarrow \alpha_0(i)/c_0$

for $t = 1\ T-1$ **do**

for $j = 1\ N$ **do** $\alpha_t(j) \leftarrow \left(\sum_{i=1}^{N} \alpha_{t-1}(i) \cdot A(i,j)\right) \cdot B(j, o_t)$ $c_t \leftarrow \sum_{j=1}^{N} \alpha_t(j)$

for $j = 1\ N$ **do** $\alpha_t(j) \leftarrow \alpha_t(j)/c_t$ **return** α, c

Algorithm 3 Backward Algorithm with Scaling

Secuencia O, matrices A, B, y escala c obtenida en Forward β Initialize: $\beta_{T-1}(i) \leftarrow 1/c_{T-1}$ for all $i = 1, \ldots, N$
for $t = T - 2$ 0 **do**
for $i = 1$ N **do** $\beta_t(i) \leftarrow \frac{\sum_{j=1}^{N} A(i,j) \cdot B(j, o_{t+1}) \cdot \beta_{t+1}(j)}{c_t}$ **return** β

Algorithm 4 Entrenamiento, Evaluación y Predicción con SVM

Matriz de características X y etiquetas y obtenidas del DataFrame de entrenamiento (con variables: `full_length`, `viterbi_score`, `pred_length`, `ratio`);
Conjunto de características para el transcriptoma: X_{trans} Clasificador SVM entrenado, métricas de evaluación y predicciones sobre el transcriptoma
Dividir X y y en conjuntos de entrenamiento $(X_{\text{train}}, y_{\text{train}})$ y prueba $(X_{\text{test}}, y_{\text{test}})$ utilizando una proporción 80/20
Inicializar SVM con kernel RBF, parámetros $C \leftarrow 1.0$ y γ configurado (e.g., `"scale"`)
Entrenar el clasificador con X_{train} y y_{train}
Predecir etiquetas $\hat{y}$ usando X_{test} con el clasificador entrenado Calcular las métricas:

- Accuracy $\leftarrow$ accuracy$(y_{\text{test}}, \hat{y})$
- Precisión $\leftarrow$ precision$(y_{\text{test}}, \hat{y})$
- Recall $\leftarrow$ recall$(y_{\text{test}}, \hat{y})$
- F1 Score $\leftarrow$ f1$(y_{\text{test}}, \hat{y})$

Calcular y visualizar la matriz de confusión con los valores de y_{test} y $\hat{y}$
Si se desea obtener la curva ROC, reentrenar SVM con opción `probability=True`
Predecir probabilidades $p \leftarrow$ predict_proba(X_{test}) para la clase madura Calcular los valores de FPR, TPR y AUC con las funciones adecuadas Graficar la curva ROC y mostrar el AUC
Para cada vector de características $x \in X_{trans}$ Predecir clase $\hat{y}_{trans} \leftarrow$ SVM.predict(x) Almacenar el ID y la predicción en un registro de resultados Fin Para
return Clasificador SVM, métricas de evaluación y resultados de predicción en el transcriptoma

3 Resultados y discusión

De acuerdo al modelo oculto de Markov desarrollado a partir de solo los peptidos maduros y la maquina de soporte vectorial se obtuvieron los siguientes resultados en el entrenamiento con las secuencias completas del conoserver.

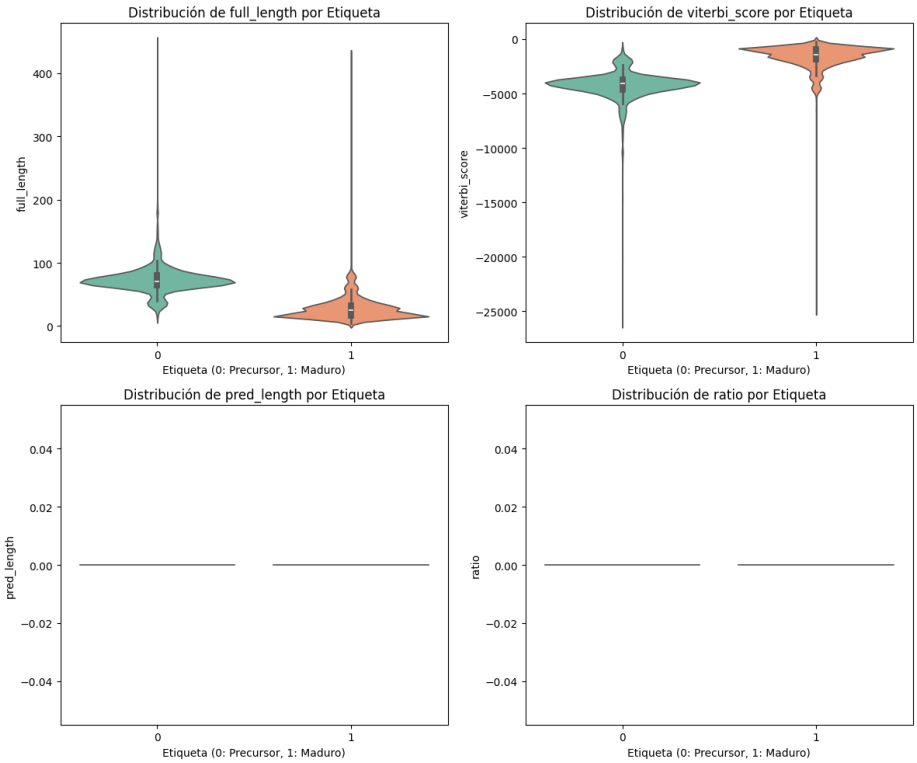

Fig. 3. Se presenta la distribución de la longitud de la secuencia completa por etiqueta (maduro y no maduro. Asi como la puntuación de vitterbi para el peptido y maduro y no maduro.)

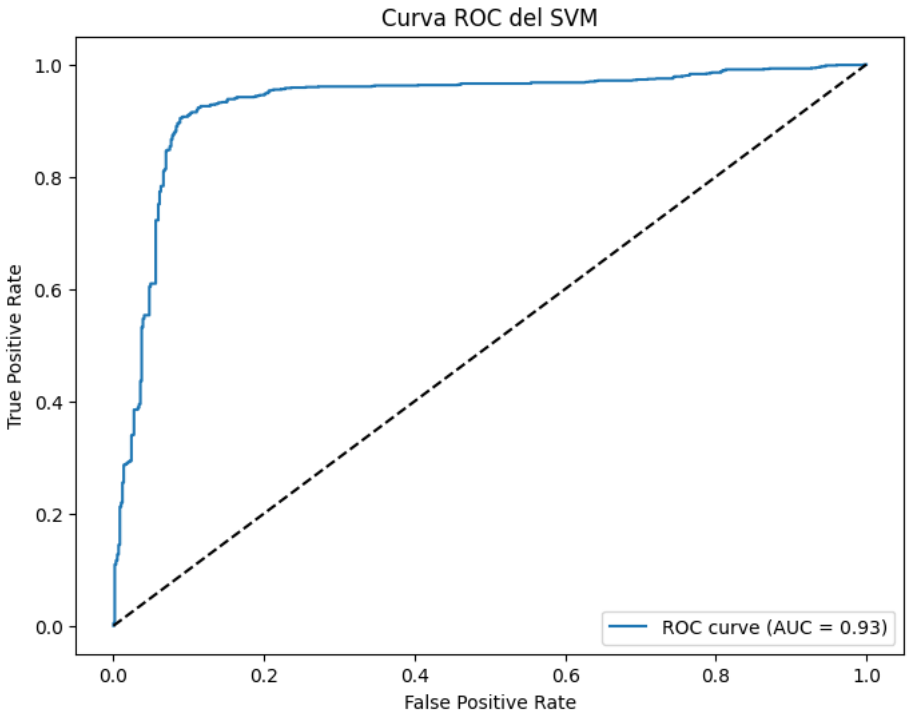

Fig. 4. Los resultados del algoritmo de maquina de soporte vectorial con kernel radial y caracteristicas de entrada obtenidas del HMM a traves del algoritmo de vitterbi alcanzo un valor AUC de 93% en la identificación de las secuencias clasificadas como peptidos maduros.

Por ultimo al utilizar el algoritmo de clasificación entrenado con las caracteristicas de la Fig. 2 se obtuvieron las siguientes predicciones para la traducción de transcriptoma a proteoma.

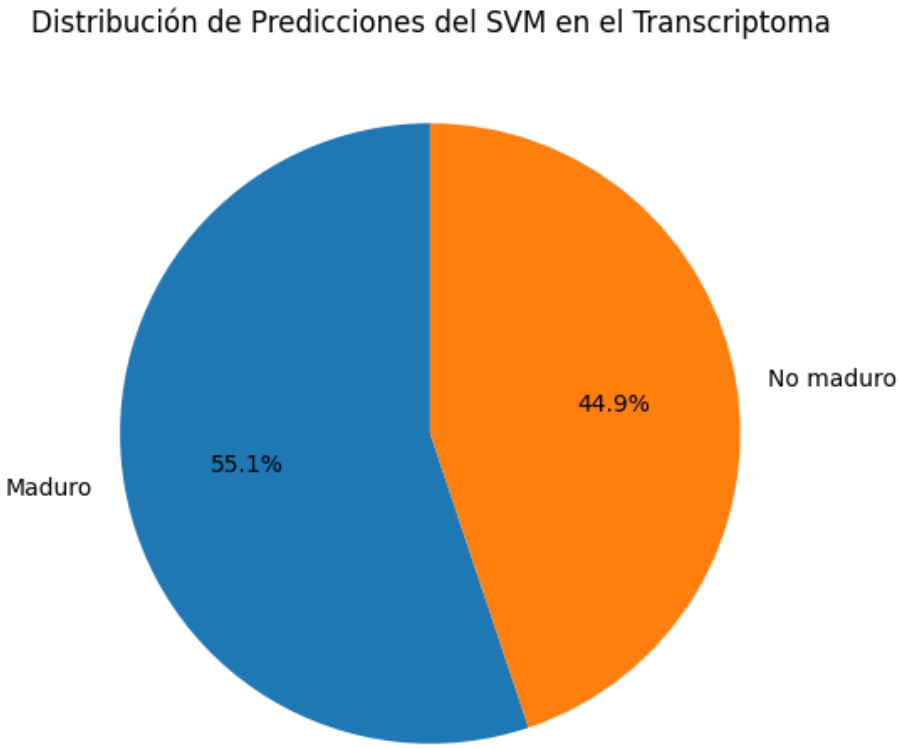

Fig. 5.

4 Discusión y conclusiones

De acuerdo a los resultados anteriores se identifica que los modelos ocultos de markov a pesar de ser una metodología clásica a diferencia de metodologías contemporaneas como los modelos de lenguaje siguen siendo herramientas suficientemente poderosas para la identificación de conotoxinas (Carrillo et al., 2025). Adicionalmente que es un método que es comprensible el proceso de obteneción de resultados a diferencia de los modelos con paradigma conexionista. (Xiong et al., 2025; Xao et al., 2025)

A su vez se identifica que los modelos ocultos de markov pueden ser un método eficiente en la extracción de caracteristicas mejorando el desempeño de métodos de aprendizaje supervisado como las maquinas de soporte vectorial.

References

Fusetani, N., Kem, W. (Eds.). (2009). Marine toxins as research tools (Vol. 46). Springer Science Business Media.

Kaas Q, Yu R, Jin AH, Dutertre S and Craik DJ. ConoServer: updated content, knowledge, and discovery tools in the conopeptide database. Nucleic Acids Research (2012) 40(Database issue):D325-30

Kaas Q, Westermann JC, Halai R, Wang CK and Craik DJ. ConoServer, a database for conopeptide sequences and structures. Bioinformatics (2008) 24(3):445-6

Li, R., Yu, J., Ye, D., Liu, S., Zhang, H., Lin, H., ... Deng, K. (2025). Conotoxins: Classification, Prediction, and Future Directions in Bioinformatics. Toxins, 17(2), 78.

Laht, S., Koua, D., Kaplinski, L., Lisacek, F., Stöcklin, R., Remm, M. (2012). Identification and classification of conopeptides using profile Hidden Markov Models. Biochimica et Biophysica Acta (BBA)-Proteins and Proteomics, 1824(3), 488-492.

López-Carrillo J, Bernáldez-Sarabia J, Pawar TJ, Jiménez S, Dueñas S, Figueroa-Montiel A, Olivares-Romero JL, Granados-Soto V, Licea-Navarro AF, Caram-Salas NL. Systemic antihyperalgesic effect of a novel conotoxin from Californiconus californicus in an inflammatory pain model. Front Pain Res (Lausanne). 2025 Jan 24;5:1500789. doi: 10.3389/fpain.2024.1500789. PMID: 39925365; PMCID: PMC11802583.

Xiong, D., Ming, Y., Li, Y., Li, S., Chen, K., Liu, J., ... He, X. (2025). EvoNB: A Protein Language Model-Based Workflow for Nanobody Mutation Prediction and Optimization. Journal of Pharmaceutical Analysis, 101260.

Zhao, G., Ge, C., Han, W., Yu, R., Liu, H. (2025). ConoGPT: Fine-Tuning a Protein Language Model by Incorporating Disulfide Bond Information for Conotoxin Sequence Generation. Toxins, 17(2), 93.