Universidad Central “Martha Abreu” de Las Villas
Facultad de Matemática – Física – Computación
Licenciatura en Ciencia de la Computación

Trabajo de Diploma

Herramienta computacional para la detección de conglomerados usando los métodos Scan.

Autor:

Elaine Valdés Hernández

Tutores:

Dr. Gladis Casas Cardoso

M.Sc Laureano Rodríguez Corvea

Santa Clara, Julio 2008

Declaración de autoría

Hago constar que el presente Trabajo de Diploma ha sido realizado en la facultad de Matemática, Física y Computación de la Universidad Central “Marta Abreu” de Las Villas (UCLV) como parte de la culminación de los estudios de Licenciatura en Ciencia de la Computación, autorizando a que el mismo sea utilizado por la institución para los fines que estime conveniente, tanto de forma total como parcial y que además no podrá ser presentado en eventos ni publicado sin la previa autorización de la UCLV.

_____

Firma del Autor

Los abajo firmantes, certificamos que el presente trabajo ha sido realizado según acuerdo de la dirección de nuestro centro y que el mismo cumple con los requisitos que debe tener un trabajo de esta envergadura referido a la temática señalada.

_____ _____

Firma del Tutor

Firma del Jefe
Seminario de Bioinformática

A mi mamá

A Mimi y Pipo, por darme cada uno lo mejor de sí.

A tío Rafe, -es esto por lo que has tenido que manejar tanto-.

A mamita, por ser mi ángel de la guarda.

A Robe y Panchi, por quererme de la mejor manera.

A tía Viky y tía Matilde, no se que hubiera sido de mi sin su ayuda-.

A Yiset y Leduan, por regalarme su cariño.

A tía mima y abuela, por su preocupación.

A Enrique, -contigo todo se volvió más fácil-.

A Yolanda y Nieves, por ser unas tías magníficas.

A Rafe y Gastón, por estar cuando los necesito.

A Felix, por ser mi amigo más que mi médico.

A mi nueva familia, por aceptarme como un miembro más.

A Nirka, -tu cariño siempre fue excepcional-.

A Charly, La Minitra y Morelito, por su paciencia ante nuestras impertinencias (mías y de las abejitas).

A mis tutores Gladita y Laureano, por su comprensión, su ayuda, su sacrificio.

A Ricardo Grau –profe cuando se grande quiero ser como usted-.

A Liset, por ser la maestra que todo niño debería tener.

A las abejitas,... ¿¡qué puedo decir de ustedes!? ...

A todos; Gracias.

Resumen

Los métodos Scan son estadísticos confiables utilizados en la detección de conglomerados de enfermos. Con el objetivo de ampliar su aplicación a problemas bioinformáticos se les realiza una modificación que consiste en transformar la secuencia de datos en otra análoga, donde la categoría de interés sería representada por uno y las demás categorías por cero. Se incluye además el uso de técnicas de lógica borrosa para obtener mejores resultados. En este trabajo se propone una aplicación que incluye las variaciones del estadístico Scan. La aplicación fue implementada en el lenguaje de programación da alto nivel Java. Se hizo un trabajo de validación amplio que incluyó el contraste de los resultados con otros arrojados por funciones similares hechas sobre el paquete *Mathematica* además de un intenso estudio de simulación basado en algunas pruebas estadísticas no paramétricas y las curvas ROC. Para ilustrar la efectividad de los métodos en la bioinformática se presentó un ejemplo sencillo de este tipo, que ilustra la fortaleza e importancia de los métodos borrosos.

Índice

Introducción

La secuenciación de genomas ha generado una gran cantidad de datos de sucesiones de bases nucleotídicas o de aminoácidos y los científicos se han auxiliado de una amplia gama de instrumentos matemáticos e informáticos para intentar comprender esta enorme información y extraer conocimiento de ella. Varios problemas claves del análisis de secuencias bioinformáticas requieren el barrido y la detección de conglomerados de bases específicas, en estas secuencias. Aunque se trata de una necesidad esencialmente espacial, cuando se aplica sobre una secuencia unidimensional este problema hace recordar fácilmente el problema de la detección de conglomerados de "casos" en el tiempo, y en particular la detección de excesos de casos de una enfermedad que pueden evidenciar una epidemia.

La detección de conglomerados de enfermos, a partir del desarrollo y aplicación de métodos estadísticos específicos, constituye un problema epidemiológico en el que se ha venido trabajando también intensivamente desde hace relativamente poco tiempo. El objetivo fundamental de estas técnicas es detectar la presencia de un exceso de casos diagnosticados de una determinada enfermedad en espacio, tiempo o considerando ambos escenarios a la vez. Las técnicas clásicas de detección de conglomerados, (métodos jerárquicos, o de las *k*-medias), no resuelven el problema de manera correcta, por lo que fue necesario desarrollar e implementar métodos matemáticos más específicos.

En trabajos recientes se modifica una de las técnicas más usadas para la detección de conglomerados en el tiempo: el método de Scan, con el objetivo de ampliar su campo de aplicación a dominios bioinformáticos. El método Scan consiste esencialmente de un barrido de la secuencia a través de una ventana móvil y el conteo de "casos" hasta determinar si en alguna de las ventanas hay un "exceso" significativo de casos. Existen formulaciones clásicas de este método tanto para secuencias puramente unidimensionales, como para secuencias circulares, estas últimas aplicables por ejemplo, en la detección de epidemias que puedan tener un carácter periódico, digamos anual. Ambas versiones, lineal y circular tienen un campo de aplicación en bioinformática, para la detección de conglomerados de "bases" en genomas, expresados linealmente o circularmente, entre estos últimos, por ejemplo, los genomas de carácter mitocondrial. Se

han creado además variantes borrosas del clásico método Scan que proporciona mejores resultados en los estudios bioinformáticos, debido al posible ruido o variabilidad de secuencias biológicas. Dichas modificaciones constituyen instrumentos alternativos novedosos para la solución de diversos problemas de la ciencia. Pero hasta el momento actual estas variantes solo han sido implementadas parcialmente en el paquete *Mathematica* y no se cuenta con un producto de software específico, con posibilidades de acceso por múltiples usuarios y con la posibilidad de evaluar los métodos de detección frente a otras alternativas.

Formulación del problema.

En este trabajo se quiere desarrollar una aplicación que ejecute varias variantes de un método de detección de conglomerados: el método Scan (lineal y circular). La aplicación debe contener también las modificaciones borrosas de ambas técnicas. Se pretende además realizar experimentos de simulación que validen dichas modificaciones.

Preguntas de investigación

1. ¿Cómo crear de manera eficiente una aplicación que contenga todas las variantes de los métodos Scan?
2. ¿Cómo validar los métodos incorporados al software?
3. ¿Resolverán los nuevos métodos incorporados de manera eficiente, algunos problemas de bioinformática?

Se formula entonces el siguiente

Objetivo General:

Implementar en un lenguaje *open source*, una aplicación donde se incluyan las variantes de los métodos Scan Lineal y Circular, con criterios duros y borrosos, validarlos y mostrar su utilidad en la solución de problemas de bioinformática.

De manera muy resumida las tareas específicas pudieran expresarse como:

1. Diseñar el sistema.
2. Implementar el sistema.

3. Validar todos los métodos Scan utilizando datos simulados y la teoría de las curvas ROC
4. Mostrar un ejemplo de carácter bioinformático.

El contenido de la tesis se aborda en tres capítulos:

- En el Capítulo 1 se hace una descripción de los métodos Scan, de su surgimiento y desarrollo. Se describe en particular el método Scan clásico para la detección de conglomerados de enfermos y se explican: una primera modificación donde se generaliza el método para ser usado en la solución de problemas de bioinformática y una segunda modificación donde se emplea la lógica difusa para optimizar los resultados del método. En este capítulo se incluyen además dos epígrafes donde se tratan elementos de lógica difusa y curvas ROC.
- En el Capítulo 2 se desarrollan las implementaciones del método Scan. Se comentan otros sistemas que usan el estadístico Scan para la solución de diferentes tipos de problemas y se hace una explicación detallada del diseño del nuevo software.
- En el Capítulo 3 se muestra la validación del método y el software a través de resultados experimentales. Se hace en particular un análisis de los resultados del método sobre varias secuencias simuladas que difieran en composición y tamaño para de esta forma demostrar la superioridad de los métodos borrosos.

Capítulo 1. Los métodos Scan. Surgimiento y desarrollo

Los métodos Scan surgieron hace más de un cuarto de siglo para resolver problemas epidemiológicos. Ellos se utilizaron inicialmente para detectar aglomeraciones de casos diagnosticados con una determinada enfermedad en períodos de tiempo cortos (Casas, 2003a, Jacquez, 1996).

Las técnicas de detección de aglomeraciones temporales, en particular los métodos Scan, pretenden dar una respuesta acertada, en aquellos casos en los que los datos disponibles se refieren a series cortas de tiempo, que no pueden ser analizadas usando los métodos convencionales, (Jacquez, 1996).

1.1 Los métodos Scan para la detección de conglomerados de enfermos

Los métodos Scan asumen que los casos en cuestión se encuentran ordenados cronológicamente de acuerdo con la fecha de primeros síntomas o de diagnóstico de la enfermedad, de muerte o cualquier otro evento de salud que se considere.

1.1.1 El método Scan sobre una línea

Sean $X_1, X_2, \ldots, X_n$ variables aleatorias independientes e idénticamente distribuidas que denotan las fechas de ocurrencias de n eventos en el intervalo $(0, T]$. Se quiere probar la hipótesis nula de que los eventos están uniformemente distribuidos contra la alternativa de que existe un conglomerado dentro de algún subintervalo de $(0, T]$, (Nagarwilla, 1996).

Se define un intervalo o una ventana de tamaño fijo de acuerdo con la duración esperada de la epidemia. La ventana seleccionada se desplaza a lo largo de la línea del tiempo y se determinan en cada caso, la cantidad de enfermos asociados a ella, (Aldrich and Wanzer, 1993). Partiendo del supuesto de que si no hay conglomerados, el número de casos por unidad de tiempo se ajusta a una distribución de Poisson, puede calcularse un estadígrafo que detecte un exceso significativo de casos

Para la formulación, sean:

t : amplitud de la ventana,

T : período de tiempo total que se analiza,

$L = T/t$: fracción que representa el período de tiempo total que se analiza con relación al ancho de la ventana,

n : cantidad de enfermos diagnosticados en T,

λ : número esperado de casos por unidad de tiempo en un proceso de Poisson,

$w_{y,\,y+t}$: cantidad de enfermos en la ventana [y, y+t).

Hipotéticamente el estadístico: $w = w_t(T) = \max_{0 \le y \le T-t} \{w_{y,\,y+t}\}$ representa el número máximo de casos que aparecen en una ventana cuando se mueve continuamente a lo largo del tiempo. En la práctica, la ventana [y, y+t) se mueve discretamente a partir de una sucesión de puntos equidistantes y_1, y_2,...y_k que cubren todo el período de análisis de amplitud T. Se denomina paso del Scan o paso del desplazamiento a $\Delta y = y_k - y_{k-1}$(Casas, 2003a).

Realmente, el estadístico anterior se estima por su versión discreta:

$$\overline{w} = \overline{w}_{t,\Delta t}(T) = \max_{1 \le i \le k\,t} \left\{ w_{y_i,\, y_{i+t}} \right\}$$

La idea del método es que, si existe un conglomerado, el número máximo de casos hallados en una ventana debe ser grande. El test estadístico depende de varios de los parámetros explicados con anterioridad y en esencia calcula la probabilidad p de que aparezcan w o más casos en una ventana. La fórmula que se utilizó para p es la propuesta en (Nauss, 1982):

$$p = P^*(w,\, \lambda L,\, 1/L) = 1 - Q^*(w,\, \lambda L,\, 1/L) \qquad (1.1)$$

donde Q^* puede ser aproximado para cualquier $L>2$ a partir de sus valores con $L = 2$ y $L = 3$.

$$Q^*(w, \lambda L, 1/L) \approx Q^*(w, 2\lambda, 1/2)\left[Q^*(w, 3\lambda, 1/3)/Q^*(w, 2\lambda, 1/2)\right]^{L-2} \qquad (1.2)$$

La aproximación (1.2) es fácilmente calculable usando una microcomputadora personal. El cálculo exacto de $Q^*(w,2\lambda,1/2)$ y $Q^*(w,3\lambda,1/3)$ se basa en un teorema demostrado también en (Nauss, 1982) y cuya esencia se resume aquí:

Para $w>2$, $p_i = e^{-\lambda}\lambda^i/i!$, $F_w = \sum_{i=0}^{w} p_i$, $\lambda > 0$, se tiene que: (1.3)

$$Q^*(w,2\lambda,1/2) = F_{w-1}^2 - (w-1)p_w\, p_{w-2} - (w-1-\lambda)p_w\, F_{w-3}$$

$$Q^*(w,3\lambda,1/3) = F_{w-1}^3 - A_1 + A_2 + A_3 - A_4$$

donde:

$$A_1 = 2p_w\, F_{w-1}\left((w-1)F_{w-2} - \lambda F_{w-3}\right)$$

$$A_2 = 0.5\, p_w^2\left((w-1)(w-2)F_{w-3} - 2(w-2)\lambda F_{w-4} + \lambda^2 F_{w-5}\right)$$

$$A_3 = \sum_{r=1}^{w-1} p_{2w-r}\, F_{r-1}^2$$

$$A_4 = \sum_{r=2}^{w-1} p_{2w-r}\, p_r\left((r-1)F_{r-2} - \lambda F_{r-3}\right)$$

donde $F_i = 0$ para todo $i<0$.

La aproximación (1.2) puede calcularse para valores no enteros de *L*. Esto la diferencia de otras expresiones matemáticas que se usaban con estos fines anteriormente. Además de ser menos restrictiva, varios autores demuestran que (1.2) es mucho más precisa, (Glaz, 1993, Nauss, 1982, Sahu and col., 1993).

1.1.2 El método Scan sobre un círculo

Este método es una variación del anterior y se utiliza para enfermedades que tengan un comportamiento estacional. Los datos se encuentran ordenados cronológicamente a lo largo de la línea del tiempo y el círculo se forma uniendo la última fecha con la primera. La ventana se desplaza sobre el círculo y se determina en cada una, la cantidad de enfermos asociados a ella. Con este desplazamiento circular se pretende incorporar al análisis la cercanía de posibles casos a “finales del último período considerado” con los

del principio del “primer período considerado”, como si fueran “los del siguiente período”.

La probabilidad de observar *w* o más casos en un intervalo o ventana de tamaño fijo se estima por:

$$p = P_c^*(w, \lambda L, 1/L) = 1 - Q_c^*(w, \lambda L, 1/L) \tag{1.4}$$

donde ahora:

$$Q_c^*(w, \lambda L, 1/L) \approx Q^*(w, 4\lambda, 1/4)\left[Q^*(w, 3\lambda, 1/3)\right]^{L-2}\left[Q^*(w, 2\lambda, 1/2)\right]^{L-1} \tag{1.5}$$

Para hallar $Q^*(w, 4\lambda, 1/4)$ se utiliza *L=4* en (1.2). Después de simplificar se obtiene:

$$Q^*(w, 4\lambda, 1/4) \approx \left[Q^*(w, 3\lambda, 1/3)\right]^2 / Q^*(w, 2\lambda, 1/2) \tag{1.6}$$

Luego $Q^*(w, 4\lambda, 1/4)$ también queda en función de $Q^*(w, 2\lambda, 1/2)$ y de $Q^*(w, 3\lambda, 1/3)$. Estos últimos valores se calculan en forma exacta a partir de las fórmulas anteriores, (Nauss, 1982).

1.2 Los métodos Scan generalizados

El método Scan clásico se ha utilizado ampliamente para la detección de conglomerados temporales de enfermos, pero sin dudas puede generalizarse. Varios son los intentos que se han realizado en este sentido, entre ellas se tienen las realizadas en nuestra Universidad, para aplicaciones a la bioinformática. Para lograr estas aplicaciones, se propone realizar una cierta transformación al método. de manera que ellos puedan utilizarse para detectar conglomerados en un sentido más universal, (Casas and Rodríguez, 2008).

Para ello se propone ordenar los datos por algún criterio determinado que depende del campo de aplicación. Si se trabaja con fechas, los datos se ordenan cronológicamente, si se trabaja con secuencias de bases que representan algún gen completo, o una porción de este, sería correcto asumir que tal juego de datos ya está ordenado.

El segundo paso consiste en transformar dicha secuencia en una secuencia análoga, pero dicotómica. El valor uno se colocará cada vez que aparezca la categoría de interés: una

base, un aminoácido o una subsecuencia determinada dentro de una secuencia del ADN o de proteínas, una fecha u otro evento que se considere. El valor cero se asociará a todas las demás categorías, (Casas and Rodríguez, 2008). Los datos transformados se representan en una línea. El nuevo problema que surge es el de determinar si en la secuencia formada por ceros y unos existen conglomerados de unos.

Por ejemplo, supóngase que se tiene una determinada secuencia de un gen y que dentro de ella resulta de interés determinar si existen conglomerados de la subsecuencia GCG. La transformación de la secuencia original en una dicotómica se realiza como se muestra en la Figura 1.1.

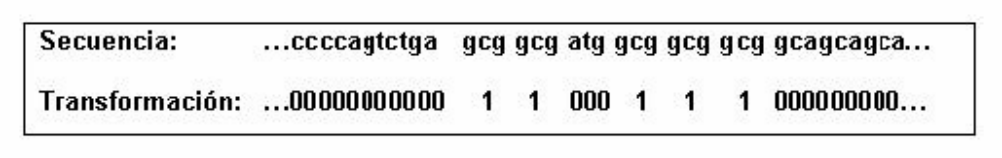

Figura 1.1 Ejemplo de conversión a cadena binaria de una porción de la secuencia de un gen

Obsérvese que la categoría de interés: subsecuencia GCG, se sustituyó por 1, mientras que el resto de los casos considerados se sustituyó por el valor cero.

El método clásico define un intervalo o ventana de tamaño fijo que se mueve, con un determinado paso, por la línea del tiempo que se analiza. En el caso generalizado, la ventana se movería por la secuencia de unos y ceros en que se transformó el problema original. La idea del método radica en que si existe un conglomerado, el número máximo de la categoría de interés (1s) hallado en una ventana, debe ser muy grande al compararla con las cantidades que aparecen en la mayoría de las ventanas restantes.

De manera general se pueden definir las hipótesis de la forma siguiente:

H_0: La categoría representada por 1s se distribuye uniformemente dentro de la secuencia considerada.

H_1: Existe al menos un conglomerado dentro de alguna de las ventanas analizadas.

1.3 Elementos de lógica borrosa

La lógica borrosa (o difusa) no es un concepto moderno. Hace 2500 años ya Aristóteles consideraba que existían ciertos grados de veracidad y falsedad y Platón había trabajado con grados de pertenencia. (Buckley, 2006).

Un **conjunto borroso** es aquel que no está formado por números sino por etiquetas lingüísticas. Una **etiqueta lingüística** es una palabra o conjunto de palabras, que representan los nombres de los conjuntos borrosos.

En los conjuntos clásicos se sabe si un elemento de un universo de discurso pertenece o no a él acudiendo a la lógica booleana. Es decir, estos conjuntos se pueden definir con un predicado que asigne a cada elemento del conjunto el valor 1 ó 0, en función de su pertenencia o no al conjunto. En los conjuntos borrosos esto no es posible. Así, cada elemento tendrá un valor asociado dentro del conjunto que indicará en qué "cantidad" pertenece a dicho conjunto. Esto es lo que se define como grado de pertenencia. Por ello, un conjunto borroso es la unión de los grados de pertenencia de todos aquellos elementos que forman parte de su universo de discurso.

El **universo de discurso** de un conjunto borroso es el intervalo en el que se incluyen los posibles valores que pueden tomar los elementos del conjunto. Con independencia de los valores que formen este universo, debe indicarse que siempre estará normalizado al intervalo [0,1]. Luego, un **conjunto borroso** es un conjunto de pares $(x, \mu(x))$, de forma que el primer elemento del par es el elemento x y el segundo, un número real $\mu(x)$ o en el intervalo [0,1] que indica el grado de pertenencia de x al conjunto. Realmente, si A es el conjunto borroso a describir, la función de pertenencia de cada x del universo, es típica de A, esto es $\mu_A(x)$

La existencia del grado de pertenencia para saber si un elemento pertenece a un conjunto o no, puede utilizarse para tratar problemas de imprecisión o incertidumbre en bases de datos, reconocimiento de patrones, clasificación, entre otras (Buckley, 2006).

1.3.1 Operaciones básicas con conjuntos borrosos

Las operaciones básicas que se pueden realizar en un conjunto clásico también pueden realizarse en un conjunto borroso. De forma general quedan definidas las operaciones:

1. **Intersección**: el resultado de esta operación entre dos conjuntos borrosos, será un conjunto borroso en el que se encuentren aquellos elementos que están en ambos conjuntos, esto es que tengan un grado de pertenencia diferente de cero en ambos conjuntos. Si lo pensamos gráficamente, según las funciones de pertenencia, se puede afirmar que: si se superponen ambas gráficas, la intersección es la zona en la que coinciden ambas funciones en el sentido de la intersección clásica, ver Figura 1.2.
2. **Unión**: el resultado es un conjunto en el que se encuentren todos aquellos elementos de alguno de los conjuntos, esto es, que tengan un grado de pertenencia diferente de cero en alguno de los conjuntos. En la representación gráfica, sería aquella zonas que resulta de la unión clásica de ambas funciones de pertenencia, ver Figura 1.3.
3. **Negación o complemento**: esta es una operación válida sobre un único conjunto. La negación se define como todos aquellos elementos que forman parte de su universo de discurso, pero no forman parte del conjunto borroso con pertenencia 1. A cada elemento del complemento del conjunto A, se le asigna un grado de pertenencia 1- $\mu_A(x)$ ver Figura 1.4.

Las figuras siguientes aclaran gráficamente las ideas expresadas con anterioridad (Dadone, 2001).

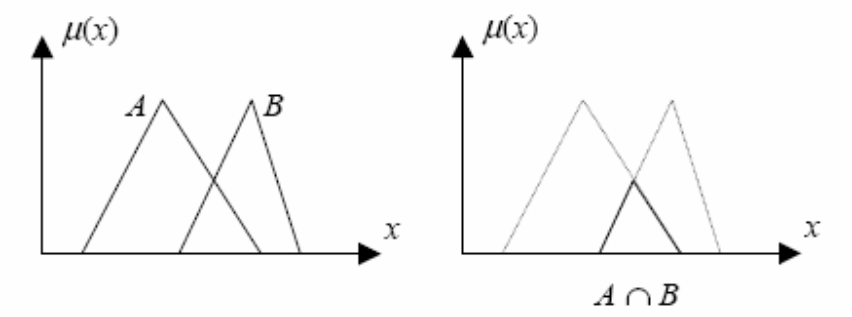

Figura 1.2 Conjuntos borrosos A y B. Operación intersección

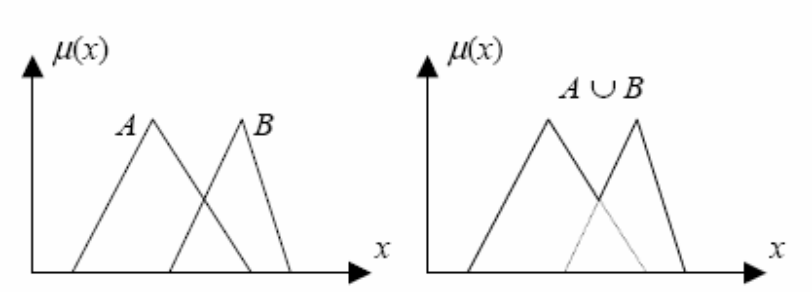

Figura 1.3 Conjuntos borrosos A y B. Operación unión

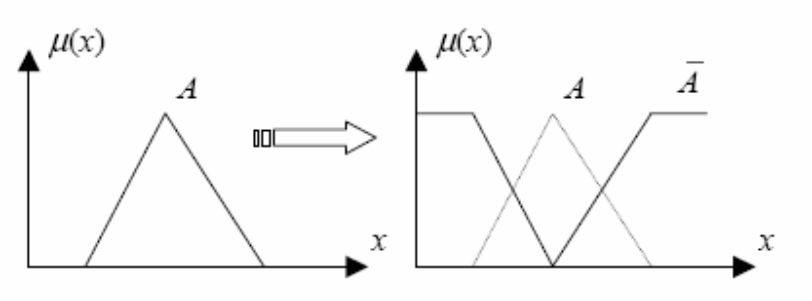

Figura 1.4 Conjunto borroso A. Operación negación

1.3.2 Tipos fundamentales de funciones de pertenencia

Dado un universo de discurso *X*, los subconjuntos borrosos de *X* son rigurosamente pares (A, μ_A) donde $\mu_A ; X \rightarrow [0,1]$ es la function de pertenencia que caracteriza a los elementos de *A*. Como esta función de pertenencia está definida sobre todo el universo, identificamos el conjunto borroso con la función de pertenencia y lo escribimos como: $A : X \rightarrow [0,1]$

En principio puede afirmarse que cualquier función A es válida. Su definición exacta depende de la aplicación. Pero en general, es preferible usar funciones típicas y sencillas, debido a que simplifican muchos cálculos y no se pierde exactitud (Dadone, 2001).

A continuación se mostrarán algunas funciones de pertenencia típicas:

1.3.2.1 Función de pertenencia triangular

Se define por sus límites inferior a y superior b, y el valor modal m, tal que a < m < b.

$$A(x)=\begin{cases}0 & \text{si } x \le a\\ (x-a)/(m-a) & \text{si } x \in (a,m]\\ (b-x)/(b-m) & \text{si } x \in (m,b)\\ 0 & \text{si } x \ge b\end{cases}$$

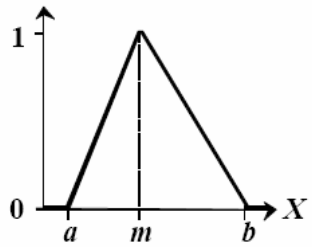

Figura 1.5 Función de pertenencia triangular

También puede representarse así: A(x;a,m,b) = máx { mín{ (x-a)/(m-a), (b-x)/(b-m) }, 0 }

1.3.2.2 Función de pertenencia trapezoidal

Definida por sus límites inferior a y superior d, y los límites de su soporte, b y c, inferior y superior respectivamente.

$$A(x)=\begin{cases}0 & \text{si } (x \le a) \text{ o } (x \ge d)\\ (x-a)/(b-a) & \text{si } x \in (a,b]\\ 1 & \text{si } x \in (b,c)\\ (d-x)/(d-c) & \text{si } x \in (b,d)\end{cases}$$

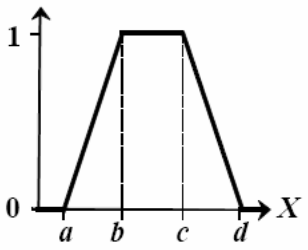

Figura 1.6 Función de pertenencia trapezoidal

1.3.2.3 Función de pertenencia Gausiana

Definida por su valor medio m y el valor k>0.

Es la típica campana de Gauss. Cuanto mayor es el valor de k, más estrecha es la campana.

$$A(x)=e^{-k(x-m)^2}$$

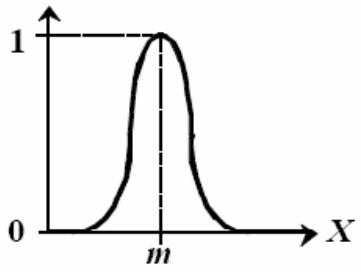

Figura 1.7 Función de pertenencia Gaussiana

1.3.2.4 Función de pertenencia S

La Función S está definida por sus límites inferior a y superior b, y el valor m, o punto de inflexión tal que a < m < b.

Un valor típico es: m=(a+b) / 2. El crecimiento es más lento cuanto mayor sea la distancia a-b.

$$A(x)=\begin{cases}0 & \text{si} \quad x \le a \\ 2\,\{(x-a)/(b-a)\}^2 & \text{si} \quad x \in (a,m] \\ 1-2\,\{(x-b)/(b-a)\}^2 & \text{si} \quad x \in (m,b) \\ 1 & \text{si} \quad x \ge b \end{cases}$$

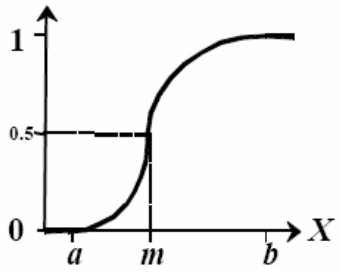

Figura 1.8 Función de pertenencia S

1.3.2.5 Función de pertenencia Gamma

Está definida por su límite inferior a y el valor k>0. Existen dos variantes:

$$A(x)=\begin{cases}0 & \text{si} \quad x \le a \\ 1-e^{-k(x-a)^2} & \text{si} \quad x > a \end{cases}$$

$$A(x)=\begin{cases}0 & \text{si} \quad x \le a \\ \dfrac{k(x-a)^2}{1+k(x-a)^2} & \text{si} \quad x > a \end{cases}$$

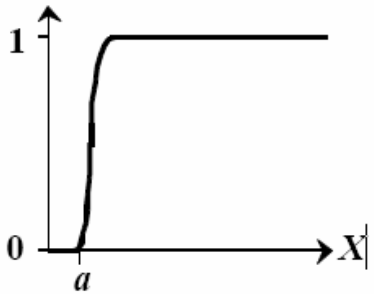

Figura 1.9 Función de pertenencia Gamma

Esta función se caracteriza por un rápido crecimiento a partir de a.

Cuanto mayor es el valor de k, el crecimiento es más rápido aún. La primera variante tiene un crecimiento más rápido. En ambas nunca se alcanza el valor 1, aunque tienen una asíntota horizontal en él.

1.3.3 Desfuzzyficación

Fuzzyficar es un anglicismo que se podría traducir como un verbo que explica la acción de transformar un valor numérico duro (crisp) en un valor difuso, esto es, una categoría

con un grando de pertenencia a ella. Por ejemplo transformar el dato de temperatura corporal de 38 grados a la categoría “fiebre” con grado de pertenencia 0,7. Desfuzzyficar es esencialmente el proceso inverso. En muchos problemas, aunque se estimen las variables que lo describen mediante operaciones con números borrosos, será necesario acabar cuantificando las magnitudes que se pretenden estimar finalmente mediante un valor cierto, es decir, debemos asignarlas un valor “crisp” o duro. Esto es exactamente lo que en la literatura borrosa se conoce como “desfuzzyficar”.

Otra definición análoga de “desfuzzyficar” es el proceso de encontrar el valor del universo que mejor “represente” al conjunto borroso. Para ello se traza alguna recta perpendicular al eje de las abscisas (donde se representa el universo del conjunto borroso), el punto donde se interceptan esta recta y ese eje es el valor que se asume. Los diferentes métodos de desfuzzyficacion determinan el lugar por donde se traza esa recta. Se pueden emplear los siguientes:

- Centroide del área: esta técnica del centroide o centro de gravedad encuentra el punto “balance” de la solución borrosa calculando la media ponderada de esa región borrosa. Dada una región borrosa A, la expresión de calculo es:

$$z = \frac{\sum_{i=0}^{n} d_i * \varphi_A(d_i)}{\sum_{i=0}^{n} \varphi_A(d_i)}$$

donde d es el i-ésimo valor del universo y φ(d) es el valor de pertenencia para ese punto. (Una formulación más general para el caso del universo continuo sustituiría las sumatorias por integrales, pero sigue la misma idea)

Este método es uno de los mas usados por varias razones: los valores desfuzzyficados tienden a moverse suavemente alrededor de la superficie borrosa, es decir, cambios en la topología del conjunto borroso provocan cambios suaves del valor desfuzzyficado. Además es relativamente fácil de calcular.

- Medio de máximos: es el promedio de los valores del universo de discurso donde se alcanza el máximo grado de pertenencia.

- Menor de máximos: es el valor menor del universo de discurso con el cual se alcanza el máximo grado de pertenencia.
- Mayor de máximos: es el valor mayor del universo de discurso con el cual se alcanza el máximo grado de pertenencia.

A diferencia de la técnica del centroide, la de los máximos tiene algunos atributos que la hacen generalmente aplicable a determinadas clases de problemas. Entre ellas: el valor desfuzzyficado es sensitivo a que una sola regla domine el conjunto de reglas; además el valor tiende a variar significativamente cuando cambia la región borrosa.

1.3.4 Borrosidad y probabilidad

Se debe evitar confundir la función de pertenencia de un conjunto borroso con una función de densidad probabilística. Debe tenerse presente que la función de pertenencia de un conjunto borroso indica hasta que punto, cierto valor de una magnitud puede ser incluido en un conjunto borroso, mientras que la probabilidad, por su parte, indica la frecuencia con la que aparecen los diversos valores de una magnitud dada.

Ello puede explicarse con un ejemplo concreto. Supóngase que se tiene una botella con cierta cantidad de líquido. Una función de pertenencia indicará el grado en que se puede incluir esa botella dentro del conjunto de las botellas “vacías” y dentro del conjunto de las botellas “llenas” (vacías y llenas son aquí los términos linguísticos que caracterizan estos conjuntos borrosos). La probabilidad por su parte, brindará información acerca de cuantas botellas de las encontradas, se pueden incluir en cada uno de dichos conjuntos. Una probabilidad 0.33 de botellas vacías indica que de cada 100 botellas que se tomen 33 estarán vacías, mientras que una pertenencia de 0.33 al conjunto botellas vacías puede indicar que la botella en cuestión incluye un tercio de la capacidad del líquido de que se trate.

Aunque muchas de las expresiones matemáticas de la lógica borrosa son similares a otras del campo de la probabilidad, su sentido es diferente. Las funciones de pertenencia a un conjunto se fijan “arbitrariamente” por el observador, indicando así cierto significado que este le asigna a cada una de las variables lingüísticas que definen 1os conjuntos. Por el

contrario, la probabilidad se determina por la observación de la ocurrencia de 1os valores de una magnitud.

1.4 Los métodos Scan Borrosos

En este epígrafe se propone otra modificación a los métodos Scan clásicos que se han estado tratando en este primer capítulo. De esta forma surgen los métodos Scan Borrosos. La principal idea para la creación de las nuevas técnicas es cambiar la ventana móvil de tamaño fijo por una ventana móvil con los extremos borrosos. En los extremos del intervalo estará presente la función de pertenencia. (Rodríguez, 2007). Como ya se explicó en epígrafes anteriores, visualizamos la función de pertenencia con una representación gráfica de la magnitud de participación de cada valor.

Mediante este procedimiento se le asigna un peso a los casos que se encuentran en los extremos de la ventana móvil. Los métodos Scan borrosos (lineal y circular), utilizan esos valores de pertenencia para determinar su influencia en la salida borrosa que representa el número de casos detectados en la ventana.

La ventana ahora se redefine (se transforma en borrosa) y tiene la forma:

$$\text{Ventana Borrosa}_{k} = \begin{cases} i * \dfrac{s_{i-k}}{(g_i + 1)} & i = k - g_i, \ldots, g_i \\ s_i & i = k, \ldots, k + t \\ (1 - i) * \dfrac{s_{i-k}}{(g_i + 1)} & i = k + t + 1, \ldots, k + t + g_i \end{cases}$$

donde: $s_1, s_2, \ldots, s_n$ es la secuencia binaria,

t : amplitud de la parte fija de la ventana,

g_i : longitud de la parte borrosa de la ventana. A esta parte se le denominará en lo adelante tamaño del suavizamiento.

La formulación matemática del test es esencialmente la misma: el método “escanea” la secuencia binaria utilizando una ventana móvil borrosa. Debido a que la ventana es borrosa, el número de casos (de “unos”) reportados en cada intervalo, o sea, el estadístico η^*_{max} del método Scan es ahora un valor real, no un entero como en el método clásico.

Este detalle trae consigo que aparezcan modificaciones en las fórmulas originales para el cálculo de la significación. Las fórmulas referidas en (1.3) son las funciones de probabilidad y de distribución de Poisson, que es una función discreta, o sea, está definida sólo para valores enteros de la variable. Con la inclusión de una función de pertenencia, la variable que representa el número de casos hallados en una ventana se transforma en una magnitud real, y para resolver el problema existen diferentes variantes:

1. Aproximar el valor real al entero más próximo. Las funciones de probabilidad y de distribución de Poisson se utilizan repetidamente en el cálculo del valor de la significación, por lo que es posible que exista una propagación del error. En lo adelante a este método se le llamará aproximación 1, ver Figura 1.10.

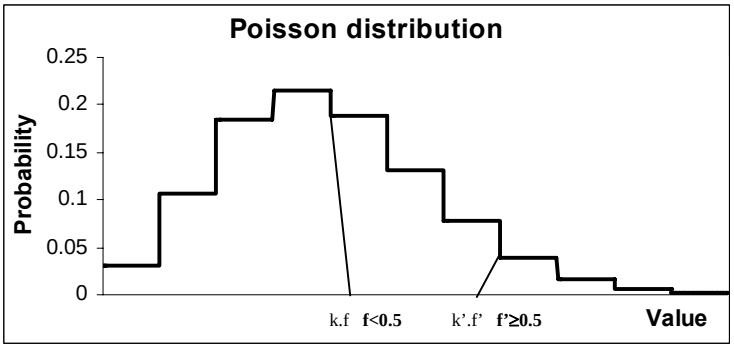

Figura 1.10 Aproximación 1 para el cálculo de la significación en el método Scan borroso

2. Aproximar el valor real usando la combinación de dos distribuciones: la Poisson hasta el valor entero anterior y luego uniforme para estimar la parte real. En lo adelante a este método se le llamará aproximación 2, ver Figura 1.11.

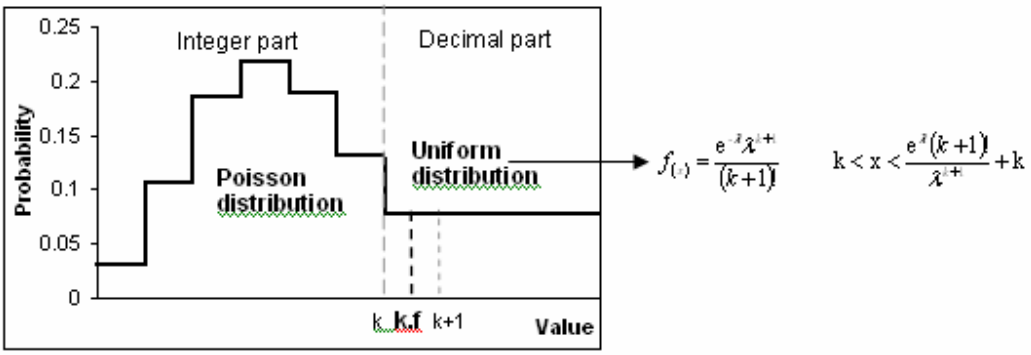

Figura 1.11 Aproximación 2 para el cálculo de la significación en el método Scan borroso

Las formulas originales presentadas por Nauss a partir de (1.2) necesitan modificarse de la manera siguiente:

- $P[\, x <= k.f \,] = \sum_{n=0}^{k} \frac{\lambda^n}{n!} e^{-\lambda} + f * \frac{e^{-\lambda} \lambda^{k+1}}{(k+1)!}$

- $P[\, x = k.f \,] = \frac{e^{-\lambda} \lambda^k}{k!} + f * \left(\frac{e^{-\lambda} \lambda^k}{k!} - \frac{e^{-\lambda} \lambda^{k+1}}{(k+1)!} \right)$

- $A3 = \sum_{r=1+f}^{k.f} P[\, x = 2 * k.f - r \,] * P[\, x <= r - 1 \,]^2$

- $A4 = \sum_{r=2+f}^{k.f} P[\, x = 2 * k.f - r \,] * P[\, x = r \,] \left((r-1) P[\, x <= r - 2 \,] - \lambda P[\, x <= r - 3 \,] \right)$

3. Aproximar el valor real por medio de dos funciones de interpolación. De forma general, la interpolación es un método que permite obtener nuevos puntos de datos a partir de un conjunto discreto de estos. En nuestro caso, los conjuntos discretos serían las funciones de probabilidad y de distribución de Poisson. Con ellos pueden construirse dos funciones de interpolación como se muestra en la Figura 1.12. En ambos casos se utilizaron polinomios de grado cuatro. En lo adelante a este método se le llamará aproximación 3.

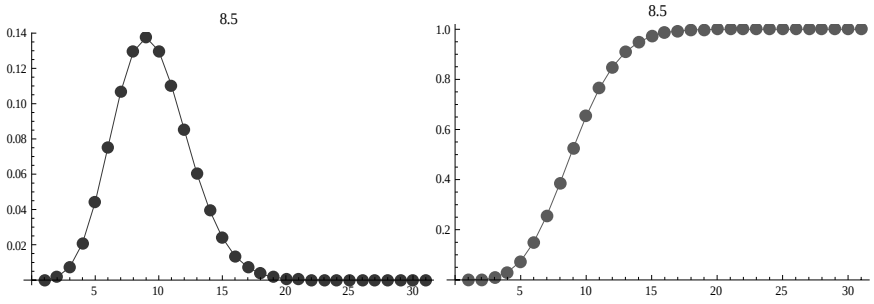

Figura 1.12 Aproximación 3 para el cálculo de la significación en el método Scan borroso

Finalmente, se crean dos conjuntos borrosos: significativo y no significativo. Cada uno de ellos tiene una función de pertenencia en forma de S:

No significativo:

$$S_{(u,0.05,0.0625,0.075)} = \begin{cases} 0 & u \le 0.05 \\ 2 * \dfrac{u - 0.05}{0.025^2} & 0.05 < u < 0.0625 \\ 1 - 2 * \dfrac{u - 0.075}{0.025^2} & 0.0625 \le u < 0.075 \\ 1 & u \ge 0.075 \end{cases}$$

Significativo:

$$S_{(u,0.075,0.0875,0.1)} = \begin{cases} 1 & u \le 0.05 \\ 1 - 2 * \dfrac{u - 0.05}{0.025^2} & 0.05 < u < 0.0625 \\ 2 * \dfrac{u - 0.075}{0.025^2} & 0.0625 \le u < 0.075 \\ 0 & u \ge 0.075 \end{cases}$$

La Figura 1.13 muestra gráficamente las ideas anteriores:

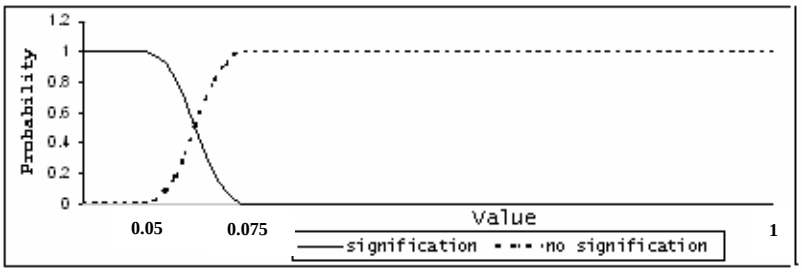

Figura 1.13 Conjuntos significativo y no significativo

Para realizar el proceso de "desfuzzyficación" se utilizó el método del mayor de los máximos.

Las ideas desarrolladas en este epígrafe son generales. Pueden aplicarse tanto al método Scan lineal (con el que se ha ejemplificado), como a su variante circular.

1.5 Curvas ROC

El problema de la detección de si hay conglomerados o no es en esencia un problema de clasificación. En este trabajo se pretende implementar y comparar varios métodos para resolver este problema y por tanto necesitamos de técnicas de comparación de desempeño de clasificadores

Las curvas ROC, acrónimo de *Receiver Operating Characteristics*, ofrecen una representación gráfica, fácil de comprender, sobre el desempeño de un clasificador que dependa de cierto parámetro, por ejemplo de un punto de corte para clasificar los positivos o los negativos (en nuestro ejemplo: hay o no conglomerados en cada secuencia analizada). La selección del punto de corte durante el aprendizaje a partir de varias secuencias de datos donde se conoce a priori si hay o no conglomerados, determinará una matriz de confusión en la cual hay cierta cantidad (y razón o proporción) de Verdaderos Positivos (*TP*), Falsos Negativos (*FN*), Verdaderos Negativos (*TN*) y Falsos Positivos (FP). En el plano de coordenadas *X, Y* la curva ROC queda dentro del cuadrado [0, 1] x [1, 0] y plotea, para cada valor del parámetro del clasificador la Razón de Falsos Positivos (*FP*) en el eje *X* y la Razón de Verdaderos Positivos (*TP*) en el eje *Y*. El punto (0,1) representa al clasificador perfecto porque clasificaría todos los casos positivos y todos los casos negativos correctamente; por otro lado el punto (1, 0) representa todo lo contrario pues no hace ninguna clasificación correcta. Los puntos (0,0) y (1, 1) representan clasificadores que predicen todos los casos como, negativos y positivos, respectivamente (Bradley, 1997).

En las curvas ROC cada valor del parámetro del clasificador aporta un par (*FP, FT*) de modo que con un conjunto de estos puntos es posible plotear la curva. Los parámetros de un clasificador pueden ser ajustados para encontrar el punto de la curva más cercano a (0,1) o a la razón de FP y TP deseada, acorde al problema. Un clasificador que no dependa de parámetros se representa apenas mediante un punto (punto ROC en lugar de curva ROC), correspondiente a su par (*FP, FT*).

La curva ROC no depende de la cantidad casos negativos o positivos que existen en la base de aprendizaje. Resume toda la información contenida en la matriz de confusión ya que *FN* es el complemento de *TP* y *TN* es el complemento de *FP*. Permiten examinar el equilibrio entre la habilidad de un clasificador para identificar correctamente los casos positivos y el número de casos negativos que están incorrectamente clasificados.

El área bajo la curva ROC puede tomarse como una medida de la exactitud en muchas aplicaciones. Si se comparan dos clasificadores a través de sus curvas ROC podemos decir que el de mayor área bajo su correspondiente curva ROC es mejor clasificador. Para

un clasificador no paramétrico, la eficiencia puede medirse usando la distancia entre su correspondiente punto (*FP, TP*) y el punto (0, 1); mientras menor sea esta distancia más eficiente será el clasificador. En ambos criterios, pueden introducirse pesos en términos de la importancia relativa de los *FP* o los *TP*.

1.6 Consideraciones finales del capítulo

Hasta aquí se ha mostrado de manera resumida una revisión de la literatura acerca de la evolución de los métodos Scan. Ellos fueron concebidos en sus inicios para resolver problemas de detección de conglomerados anormales de enfermos en ambientes epidemiológicos en períodos cortos de tiempo.

Con el paso de los años han ido apareciendo generalizaciones en la literatura, lo que ha incrementado de manera notable su campo de aplicación. La bioinformática no constituye una excepción. Con el objetivo de aplicar tales técnicas a secuencias de ADN, surgen los métodos Scan generalizados que transforman una secuencia cualquiera (de ADN por ejemplo) a una secuencia binaria, que es mucho más fácil de trabajar.

En este capítulo se muestran además los fundamentos matemáticos de los cambios realizados a los métodos Scan originales, empleando elementos de conjuntos borrosos y de la lógica borrosa, para crear los métodos Scan borrosos.

Por último, se describen las curvas ROC que serán utilizadas para comparar el desempeño de las diferentes variantes del método Scan, vistas como clasificadores, en el proceso de detección de conglomerados

Capítulo 2. Implementaciones del método Scan

En este capítulo se comentarán algunos sistemas orientados al estudio de eventos epimediológicos o geográficos y problemas de carácter bioinformático que implementan el método Scan.

Se explicará el diseño de la aplicación que realiza las nuevas implementaciones del método sobre software libre. Para la implementación del sistema en Java fue necesario un estudio previo del lenguaje de programación, para conocer de las facilidades que brinda, cuáles podrían resultar útiles para lograr la mayor eficiencia posible durante la ejecución del software.

2.1 Antecedentes de implementaciones del método Scan

2.1.1 EpiDet

El EpiDet es un sistema para la detección temprana de focos epidémicos, fundamentalmente enfermedades poco comunes o poco conocidas, no necesariamente infecciosas, como algunos tipos de cáncer o neuritis (Casas, 2003b).

EpiDet usa las técnicas de "clustering" que serán enumeradas a continuación:

- Pearson: Detecta conglomerados espaciales, no trabaja con el eje de coordenadas *X, Y, Z* sino con áreas geográficas o celdas.
- Cuzick and Edwards: Trabaja con el eje de coordenadas *X, Y, Z*. precisa de dos ficheros, uno de casos (individuos enfermos) y otro de controles (individuos sanos); ambos con la misma estructura, un encabezado de la forma *X, Y, Z* y los datos acordes al encabezado.
- Knox: Usa ficheros con encabezado de la forma *Día, Mes, Año*, *X, Y, Z*.
- Grimson temporal: Puede analizar dos tipos distintos de datos: agrupados y separados. El método Grimson-Temporal para datos separados utiliza ficheros con encabezamiento y estructura de la forma *Día, Mes, Año*; el método Grimson-Temporal para datos agrupados parte de ficheros que tienen como encabezado la

palabra reservada *Casos* y los datos se agrupan de acuerdo a alguna unidad temporal que sea de interés del usuario.

- Grimson espacial: Usa los mismos ficheros de datos que el método Cuzick and Edwards, pero este método requiere además un fichero de adyacencias que indica la disposición geográfica de las celdas en las que se ha dividido el área de estudio.
- Grimson espacio-temporal: Usa los mismos ficheros de datos que el método Knox y el fichero de adyacencias.
- Scan: Detecta conglomerados temporales. Usa ficheros con el mismo encabezamiento que los del método Grimson-Temporal para datos separados.

2.1.2 Paquetes implementados sobre el software *Mathematica*

En su versión más simple el paquete *Mathematica* se define como un software para hacer "Matemáticas" por Computadoras. Según su uso es:

- Una calculadora de tipo numérico con muchas funciones y con "cualquier precisión".
- Un paquete avanzado de subrutinas de cálculo numérico.
- Una calculadora que trabaja con expresiones simbólicas, en realidad es una potente herramienta de cálculo simbólico.
- Un paquete potente de gráficos en dos y tres dimensiones.
- Un lenguaje de programación de alto nivel.
- Un sistema para crear documentos interactivos que incluyan textos, gráficos, animaciones y sonidos entre otros.
- Un sistema de apoyo a otros programas.

Siguiendo este último uso se han creado varios paquetes en el *Mathematica* que implementan muchos de los métodos de "clustering" mencionados anteriormente (Casas, 2003a). Esto se realizó con el propósito de validar los resultados de las implementaciones realizadas y de realizar estudios complejos de Simulación.

En el Anexo 1 se puede apreciar la implementación del método de Scan clásico en el *Mathematica*.

2.1.3 SatScan

El SatScan es un software que analiza datos de espacio, tiempo y espacio-tiempo usando el estadístico scan. Está diseñado para la vigilancia de afectaciones geográficas, la detección de conglomerados de estas afectaciones en el tiempo o el espacio-tiempo y para determinar si son estadísticamente significativas. El sistema es capaz de estimar cuando una afectación geográfica está distribuida de forma aleatoria, evaluar si un conglomerado alarmante de afectaciones es estadísticamente significativo y realizar una vigilancia temporal periódica para la detección temprana de afectaciones, en su etapa inicial (Kulldorff, 2007).

SatScan usa varios modelos, uno basado en la distribución de Poisson para eventos distribuidos en un área geográfica de acuerdo a una población en riesgo ya conocida; un modelo de Bernoulli sobre eventos dicotómicos (0 ó 1) tales como casos y control; un modelo de permutación espacio-tiempo usando solo casos; o un modelo normal para otros tipos de datos continuos. Los datos pueden tener un formato consistente con el eje de coordenadas para cada evento o pueden ser agregados en alguno de los niveles geográficos que ofrece el software. SatScan puede ser ajustado para cualquier número de parámetros proporcionados por el usuario, tales como conglomerados de espacio-tiempo conocidos y datos perdidos. Es posible escanear simultáneamente varios conjuntos de datos para buscar conglomerados que existen en uno o varios conjuntos.

2.2 Diseño del nuevo sistema

Los Diagramas UML están formados por un conjunto de patrones de diseño que permiten conocer el funcionamiento de un software desde diferentes perspectivas. Los diagramas de clases, por ejemplo constituyen un modelo estructural estático que muestra las clases e interfaces, sus estructuras y relaciones con otras clases e interfaces.

Los diagramas de casos de uso describen la relación entre el usuario y el sistema, desde la perspectiva del usuario, no aportan información sobre el funcionamiento interno del

software. Los diagramas de actividades, por otra parte, son una variación de las máquinas de estado donde cada estado representa la ejecución de una acción y las transiciones dependen de las ejecuciones de las acciones.

Partiendo de las facilidades que proveen los diagramas UML se hará uso de ellos para explicar de forma sencilla el diseño del nuevo sistema.

2.2.1 Especificación de requisitos.

En este epígrafe se realiza una breve descripción de los principales requisitos del sistema que se implementa.

El sistema debe ser capaz de:

- Capturar los datos de entrada para las ejecuciones del método Scan desde un archivo texto, con valores 0 y 1. Los valores 1 significan categoría de interés y 0 el resto de las categorías.
- Se implementarán los métodos Scan Lineal Generalizado, Scan Lineal Fuzzy, Scan Circular Generalizado y Scan Circular Fuzzy. Los métodos se describen posteriormente mediante algunos diagramas de actividades.
- Todos los métodos están sujetos a dos parámetros generales: ancho de la ventana móvil y paso del scan (paso del barrido).
- En el caso del Scan Lineal Fuzzy y el Scan Circular Fuzzy aparecen dos parámetros adicionales: factor de incertidumbre y técnica de aproximación.
- Los parámetros mencionados anteriormente tienen las siguientes características:
    - Ancho de la ventana móvil: valor entero entre uno y el largo de la secuencia.
    - Paso de scan: valor entero entre uno y el valor del ancho de la ventana móvil.
    - Factor de incertidumbre: valor entero entre 0 y 8.
    - Técnica de aproximación: elegible entre redondeo, distribución de Poisson y Uniforme y polinómica.
- Pueden aparecer inicialmente los valores por defecto que se le asignan a cada uno de los tres primeros parámetros mencionados con valores numéricos, en este caso 20, 1 y 0 respectivamente.
- Los resultados deben ser mostrados en un área de texto y pueden ser guardados en caso de que el usuario así lo decida.

2.2.2 Diagrama general de casos de uso

En el diagrama de casos de usos de la Figura 2.1 se hace un bosquejo de todos los casos de uso del sistema, que serán explicados detalladamente, más adelante.

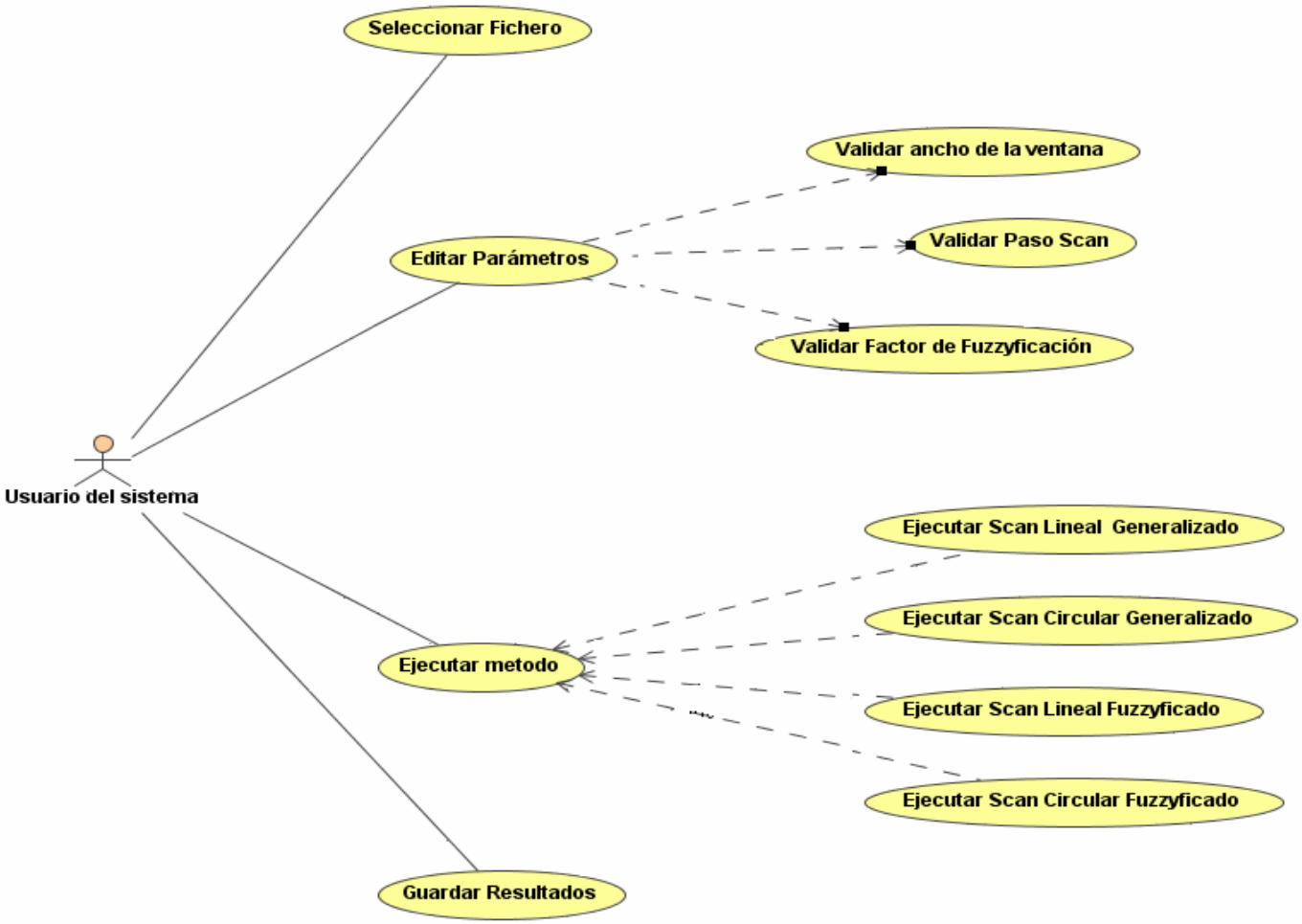

Figura 2.1 Casos de uso del Usuario del sistema.

2.2.3 Especificación de los casos de uso

Caso de uso: Seleccionar Fichero

Actor: Usuario del sistema.

Propósito: Proporcionar el nombre del archivo que contiene los datos.

Curso normal de los eventos

Acción del actor	**Respuesta del sistema**
1. Este caso de uso comienza cuando el usuario hace clic en el botón Buscar.	2. El sistema muestra un cuadro de dialogo que permite moverse por los distintos directorios y escoger un archivo específico.
3. El usuario escoge el archivo deseado	4. El sistema muestra en un campo de texto el nombre completo del archivo.

Curso alternativo

- Línea 3: el usuario decide no escoger archivo alguno, cancelando la acción.

Caso de uso: **Editar parámetros**

Actor: Usuario del sistema.

Propósito: Editar los valores, de los parámetros necesarios, para ejecutar el método Scan.

Sección Principal

Curso normal de los eventos

Acción del actor	**Respuesta del sistema**
1. Este caso de uso comienza cuando el usuario hace clic en alguno de los siguientes cuadros de texto. a) Ancho de ventana móvil. b) Paso Scan. c) Factor de incertidumbre.	
2. El usuario escribe un valor en el cuadro de texto.	3. El sistema valida el valor escrito.
4. El usuario puede reeditar el valor o editar otro campo.	

Sección: Ancho de ventana móvil.

Curso normal de los eventos

Acción del actor	**Respuesta del sistema**
1. El usuario escribe un valor en el cuadro de texto Ancho de ventana móvil.	2. El sistema reconoce el valor como numérico.

Cursos alternativos

- Línea 2: el sistema detecta que el carácter escrito por el usuario, no es numérico. Se muestra un mensaje de error y se restituye el valor por defecto.
- Línea 2: el usuario abandona el cuadro de texto dejándolo vacío o con valor cero. Se restituye el valor por defecto.

Sección: Paso Scan.

Curso normal de los eventos

Acción del actor	**Respuesta del sistema**
1. El usuario escribe un valor en el cuadro de texto Paso Scan.	2. El sistema reconoce el valor como numérico.

Cursos alternativos

- Línea 2: el sistema detecta que el carácter escrito por el usuario, no es numérico. Se muestra un mensaje de error y se restituye el valor por defecto.
- Línea 2: el sistema detecta que el valor de Paso Scan es mayor que el valor de Ancho de ventana móvil. Se muestra un mensaje de error y se restituye el valor por defecto.
- Línea 2: el usuario abandona el cuadro de texto dejándolo vacío o con valor cero. Se restituye el valor por defecto.

Sección: Factor de incertidumbre.

Curso normal de los eventos

Acción del actor	Respuesta del sistema
1. El usuario escribe un valor en el cuadro de texto Factor de incertidumbre.	2. El sistema reconoce el valor como numérico; si este valor es mayor que cero habilita el combobox Técnica de aproximación; si es igual a cero lo deshabilita.

Cursos alternativos

- Línea 2: el sistema detecta que el carácter escrito por el usuario, no es numérico. Se muestra un mensaje de error y se restituye el valor por defecto.
- Línea 2: el usuario abandona el cuadro de texto dejándolo vacío. Se restituye el valor por defecto.

Caso de uso: Ejecutar método

Actor: Usuario del sistema.

Propósito: Seleccionar la variante del método Scan que desea ejecutar Visualizar los resultados de la ejecución.

Prerrequisitos: el sistema debe contener el camino completo del fichero donde se encuentran los datos y todos los parámetros que necesita el método.

Curso normal de los eventos

Acción del actor	Respuesta del sistema
1. Este caso de uso comienza cuando el usuario escoge un tipo de método en el combobox Tipo de Scan.	
2. El usuario hace clic en el botón Aceptar.	2. El sistema carga los datos del fichero, cuyo camino esta especificado en el cuadro

	de texto Archivo origen, ejecuta el método Scan, y escribe el resultado en el área de texto.

Cursos alternativos

- Línea 1: si el combobox Técnica de desfuzzyficación está habilitado, el actor puede escoger una de ellas
- Línea 2: existe un error durante la ejecución del método. El sistema devuelve un textdialog mostrando el tipo de error.

Caso de uso: Guardar resultados

Actor: Usuario del sistema

Propósito: Almacenar los resultados en el archivo que indique el usuario.

Curso normal de los eventos

Acción del actor	**Respuesta del sistema**
1. Este caso de uso comienza cuando el usuario hace clic en el botón Guardar.	2. El sistema muestra un cuadro de dialogo que permite moverse por los distintos directorios y escoger un archivo específico o crearlo.
3. El usuario escoge el archivo deseado	4. El sistema escribe los datos en el archivo y muestra un mensaje indicando que la acción ha finalizado.

Curso alternativo

- Línea 2: el usuario decide no escoger archivo alguno.

2.2.4 Diagramas de actividades de los métodos Scan

Los diagramas de actividades de la Figura 2.2 y la Figura 2.3 permitirán conocer como se ejecutan los métodos Scan Lineal Generalizado y Scan Lineal Fuzzyficado respectivamente. Dado que los métodos circulares constituyen una variación, (en la que se unen en el inicio y el fin de la cadena) de los métodos lineales, no se mostrarán sus diagramas de actividades pues sus comportamientos pueden conocerse por analogía.

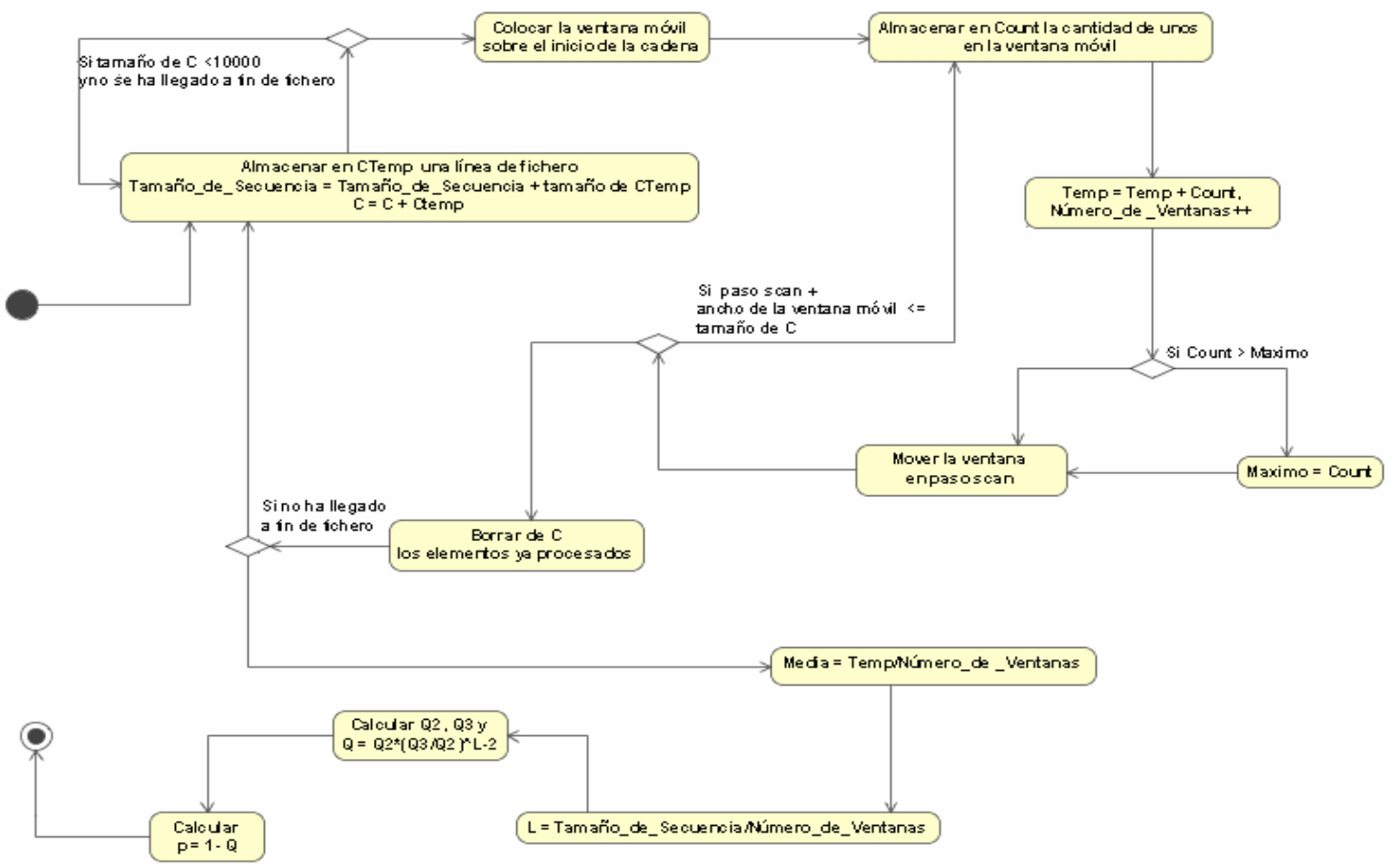

Figura 2.2 Diagrama de Actividades del método Scan Lineal Generalizado.

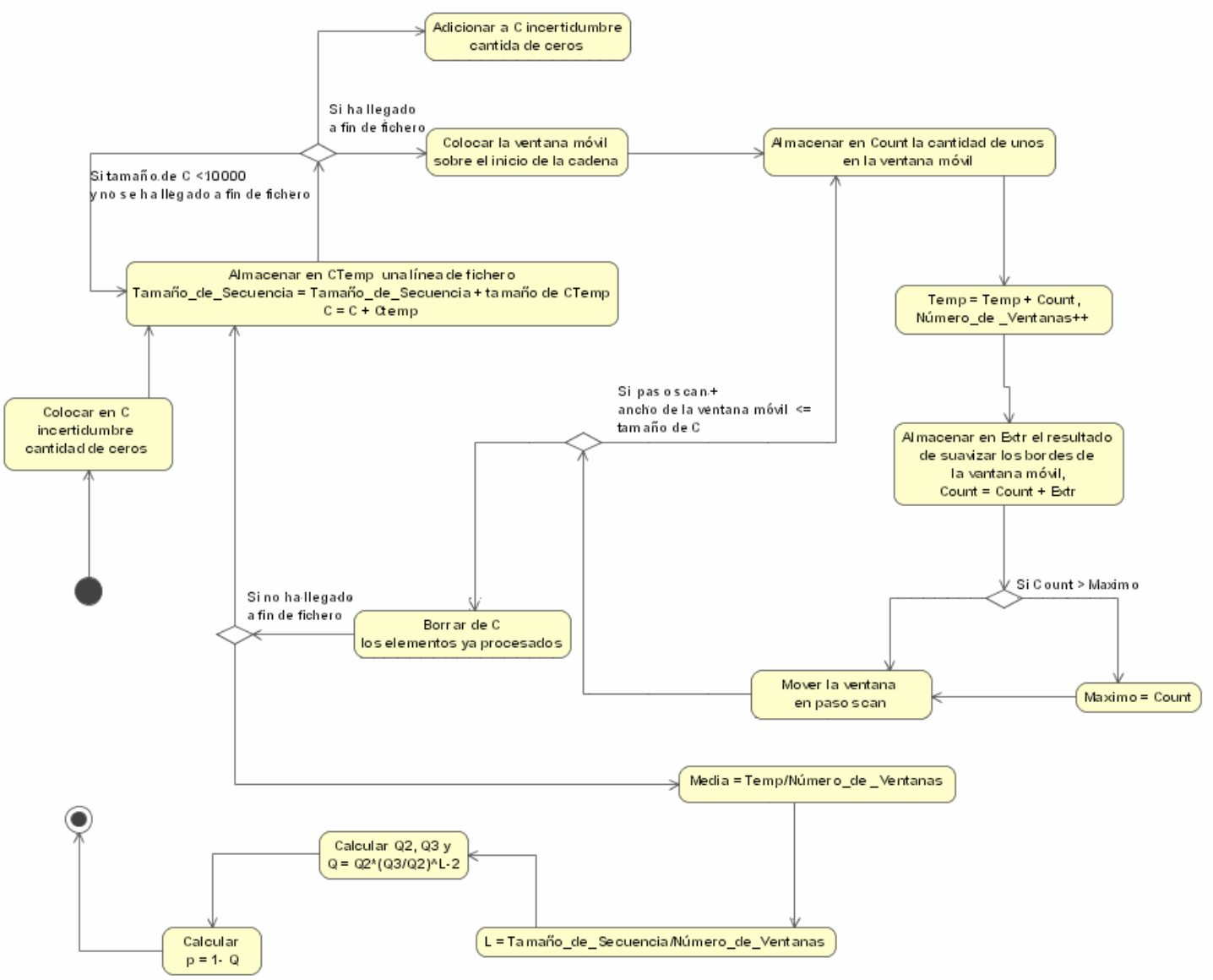

Figura 2.3 Diagrama de Actividades del método Scan Lineal Fuzzyficado.

Es necesario aclarar que en el método Scan Circular Fuzzyficado no se incluyen ceros en los extremos de la secuencia simplemente se unen igual que en el Scan Circular Generalizado.

2.2.5 Diagrama de clases

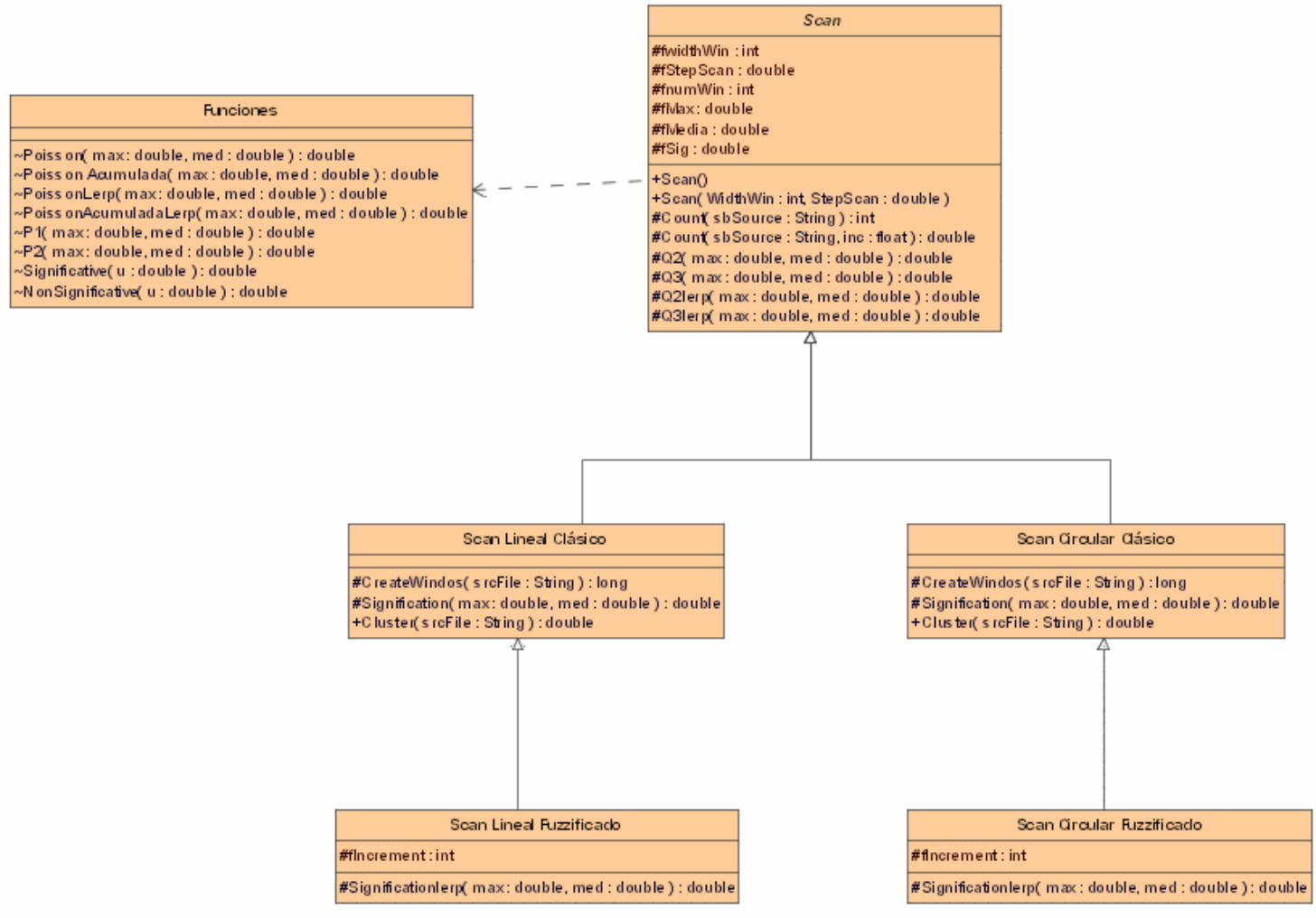

Figura 2.4 Diagrama de clases.

2.2.6 Interfaz del sistema

En la interfaz del sistema se observan los campos para la edición de los parámetros y el área de texto, donde se mostrarán datos como, el tamaño de la secuencia que contenía el fichero, el número de veces que se movió la ventana sobre la cadena, el estadístico $\overline{w}$ y la probabilidad de que exista al menos un conglomerado en la secuencia.

Los botones Aceptar y Guardar no están habilitados inicialmente, al primero solo se habilitará después que todos los parámetros estén listos y el segundo después que se haya ejecutado el método.

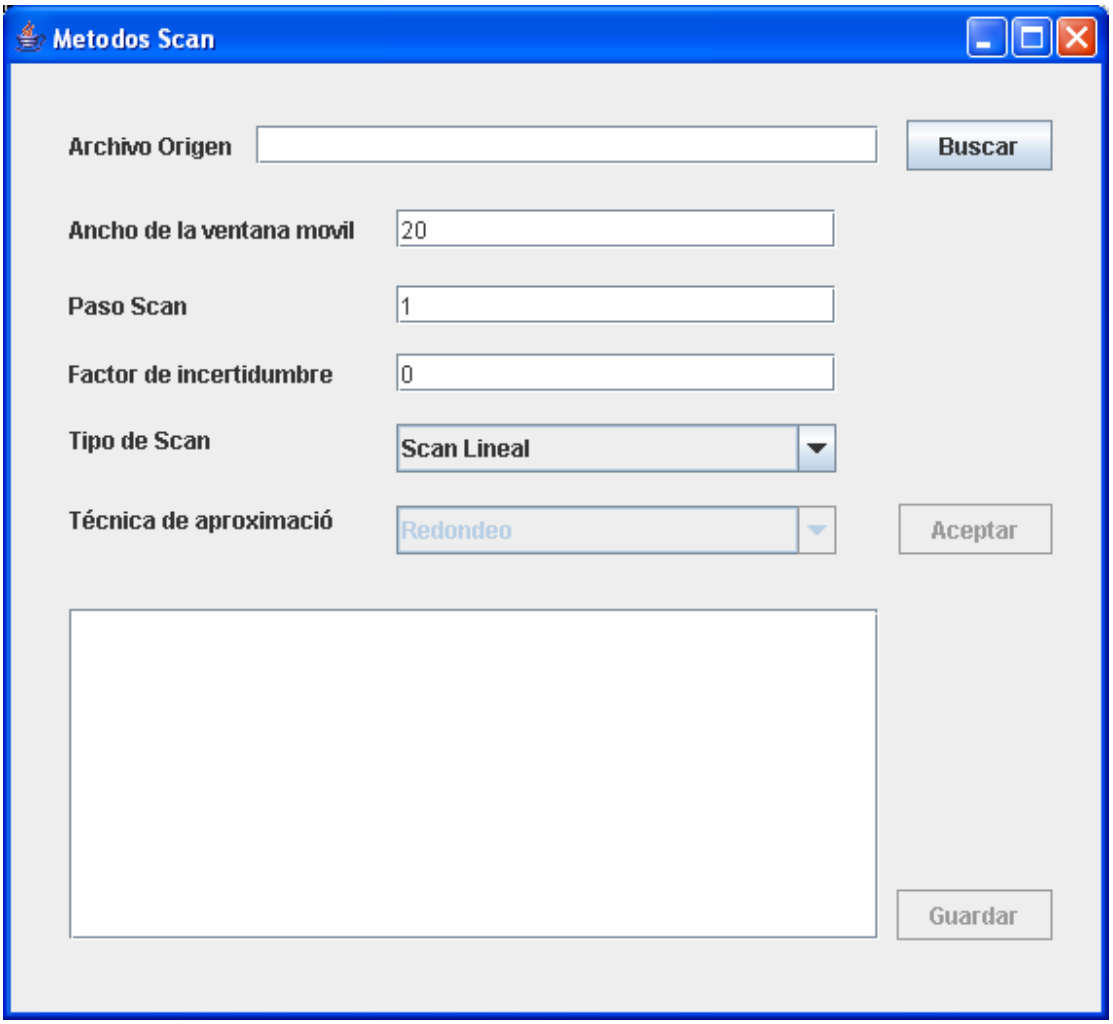

Figura 2.5 Interfaz del sistema

2.3 Consideraciones finales del capítulo

En este capítulo se describen algunos detalles relacionados con softwares dedicados a la detección de conglomerados. En este sentido se mencionan el EpiDet, algunos paquetes específicos implementados sobre el *Mathematica* y el SatScan.

Se presentan además aspectos relacionados con la modelación del nuevo software, que ilustran como este garantiza la edición de los parámetros de forma segura para lograr una

ejecución más eficiente del método, podemos asegurar que el software cumple con los requerimientos que se plantearon inicialmente.

Capítulo 3. Análisis de los resultados experimentales

En este capítulo se realiza un intensivo estudio experimental con el objetivo de mostrar la superioridad de los métodos borrosos con respecto a los clásicos. Se muestra también una aplicación en el campo de la bioinformática. A continuación se comentarán brevemente las bases de la simulación realizada.

3.1 Bases de la simulación realizada

Para realizar los estudios de simulación, se generaron verdaderos y falsos conglomerados utilizando secuencia aleatoria de ceros y unos, generados con la distribución de Bernoulli. Además se definen diferentes tamaños de secuencia (n), pues los métodos de detección de conglomerados no responden de la misma forma ante secuencias de diferentes longitudes (Rodríguez, 2007).

3.1.1 Generación de conglomerados verdaderos

Para generar verdaderos conglomerados se siguen los siguientes pasos:

- Un quinto de la secuencia se genera con una probabilidad grande, para garantizar una elevada presencia de unos (categoría de interés). Esta probabilidad puede ser modificada.
- El resto de la secuencia, (4/5) se genera con una probabilidad pequeña, para garantizar una pobre presencia de unos (categoría de interés).
- En el conjunto mayor, se selecciona aleatoriamente una posición aleatoria y se inserta en ella el conjunto menor (que es el de mayor cantidad de unos).

De esta forma se obtiene una secuencia de ceros y unos que tiene al menos un conglomerado.

Ejemplo 1: Secuencia de tamaño = 40

1er paso. Con una probabilidad 0.95 se generan ocho valores:

1 1 1 1 1 1 1 1

2do paso. Se general el resto de la secuencia con probabilidad menor a 0.2 de presencia de unos (32 valores):

0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0

3er paso. Se genera aleatoriamente un valor entre 1 y 32: 17

Conjunto con al menos un conglomerado:

0 0 0 0 01 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0

Como puede apreciarse a simple vista en la secuencia generada existe un conglomerado de unos.

3.1.2 Generación de falsos conglomerados

Para generar falsos conglomerados se genera la secuencia con una probabilidad inferior a 0.5.

Ejemplo 2: pob = 40

0 0 1 0 1 1 0 1 0 0 1 0 1 0 1 1 0 0 1 0 1 0 0 1 0 1 0 0 1 1 0 0 0 1 0 0 1 0 0 1

Como puede apreciarse a simple vista en la secuencia generada no existe ningún conglomerado de unos.

3.1.3 Consideraciones generales

Para realizar los estudios de simulación, se generan 1000 conjuntos de verdaderos y falsos conglomerados independientemente. Estos ficheros de datos se les presentan a los métodos Scan y se contabilizan las repuestas de los mismos. En el caso de los métodos borrosos se decidió utilizar parámetros de suavizado de tres y de cinco valores a ambos lados de la ventana.

El resultado de está evaluación para cada incertidumbre aporta dos curvas correspondientes a los conjuntos borrosos significativo y no significativo, las cuales evaluadas en una misma ventana móvil, suman 1000, que es el total de casos considerados en cada juego de datos. La incertidumbre (o el suavizado cero) corresponde con el método Scan Clásico Generalizado.

Se consideraron todos los valores posibles para el parámetro más importante de esos métodos: el tamaño de la ventana móvil. En este sentido, los valores se toman desde el más pequeño posible: uno, hasta el mayor de todos: el largo de la secuencia generada. En la práctica la elección de valores extremos resulta un tanto inadecuada y tiene muy pocas probabilidades de ser elegida por un usuario consciente. Con el objetivo de poder hacer recomendaciones certeras acerca de los valores de los parámetros que resultarían adecuados, se decidió considerarlas todas.

3.2 Análisis de los resultados del los métodos lineales en secuencias de tamaño 100

Se hicieron corridas con secuencias pequeñas (de tamaño 100), con otras intermedias (tamaño 250) y con otras mayores (tamaño 500). A continuación se muestran detalles de los experimentos realizados.

3.2.1 Verdaderos conglomerados en secuencia de tamaño 100

Como ya se sabe, para obtener las variantes borrosas existen tres aproximaciones diferentes. Todas se calcularon y graficaron, ver Figura 3.1.

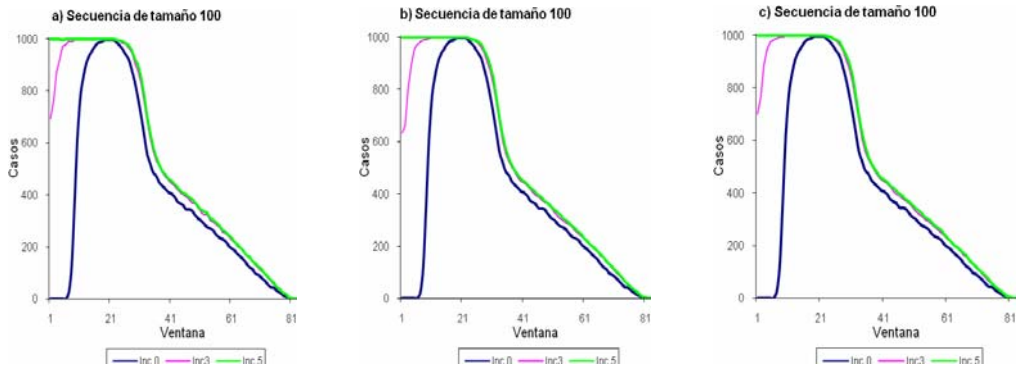

Figura 3.1 Métodos Scan Lineal en secuencias de tamaño 100. Los métodos borrosos se calcularon según a) variante aproximada, b) variante Poisson-Uniforme, c)variante Polinomial.

El gráfico ilustra que los métodos borrosos ayudan a detectar mejor los conglomerados cuando el tamaño de la ventana es pequeño. Cuando los tamaños de la ventana aumentan, su comportamiento es aparentemente el mismo que el del método clásico.

Visualmente no parece haber diferencias significativas entre las tres variantes diferentes para el cálculo de la significación borrosa (ver Figura 3.1). Para corroborar esta hipótesis se aplicó un test de Friedman. Las Tablas 3.1 muestran los resultados para los datos que se obtuvieron con parámetro de suavizamiento tres.

Ranks

	Mean Rank
Aprox3	2.46
Unif3	1.47
Pol3	2.08

Test Statistics[a]

N	100
Chi-Square	82.872
df	2
Asymp. Sig.	.000

a. Friedman Test

Tablas 3.1 Test de Friedman para secuencias de tamaño 100 y suavizamiento tres

Como puede apreciarse, a pesar de que visualmente no se pueden apreciar las diferencias, estas sí existen. Se repitió el análisis para los datos con suavizamisnto cinco y los resultados coincidieron, ver Tablas 3.2.

Ranks

	Mean Rank
Aprox5	2.37
Unif5	1.56
Pol5	2.08

Test Statistics[a]

N	100
Chi-Square	61.113
df	2
Asymp. Sig.	.000

a. Friedman Test

Tablas 3.2 Test de Friedman para secuencias de tamaño 100 y suavizamiento cinco.

No es objetivo del presente trabajo determinar cuál de las aproximaciones borrosas es la mejor, luego no se realizan análisis complementarios para determinarla.

Con el objetivo de demostrar la superioridad del método borroso, se decidió aplicar un test de Wilcoxon para detectar las diferencias entre el método clásico y las tres variantes del borroso. Los resultados se muestran en las Tablas 3.3 considerando suavizamiento tres y en la 3.4 considerando suavizamiento cinco.

Test Statistics[b]

	Aprox3 - ClasicoL	Unif3 - ClasicoL	Pol3 - ClasicoL
Z	-7.866[a]	-7.867[a]	-7.867[a]
Asymp. Sig. (2-tailed)	.000	.000	.000

a. Based on negative ranks.

b. Wilcoxon Signed Ranks Test

Tabla 3.3 Test de Wilcoxon para secuencias de tamaño 100 y suavizamiento tres.

Test Statistics[b]

	Aprox5 - ClasicoL	Unif5 - ClasicoL	Pol5 - ClasicoL
Z	-7.867[a]	-7.866[a]	-7.867[a]
Asymp. Sig. (2-tailed)	.000	.000	.000

a. Based on negative ranks.

b. Wilcoxon Signed Ranks Test

Tabla 3.4 Test de Wilcoxon para secuencias de tamaño 100 y suavizamiento cinco.

En ambos casos los resultados fueron siempre altamente significativos. En el primer análisis se demuestra en particular la superioridad de los métodos borrosos, en casi la totalidad de los casos analizados (82/100).

Como puede apreciarse en la Figura 3.1, los métodos borrosos son mucho mejores para tamaños de ventana pequeños. Se decidió entonces, repetir los análisis anteriores, eliminando de la muestra los 25 primeros casos, es decir los tamaños de ventana pequeños, desde 1 hasta 25. Los resultados de la aplicación del test de Wilcoxon aparecen en las Tablas 3.5 y 3.6 respectivamente.

Test Statistics[b]

	Aprox3 - ClasicoL	Unif3 - ClasicoL	Pol3 - ClasicoL
Z	-6.568[a]	-6.568[a]	-6.568[a]
Asymp. Sig. (2-tailed)	.000	.000	.000

a. Based on negative ranks.

b. Wilcoxon Signed Ranks Test

Tablas 3.5 Test de Wilcoxon para secuencias de tamaño 100 y suavizamiento tres, sin los tamaños de ventana pequeños

A pesar de haber eliminado los tamaños de ventana pequeños, los métodos borrosos siguen mostrado mejores resultados que el Scan Clásico Generalizado. Los resultados altamente significativos del test de Wilcoxon así lo confirman.

Test Statistics[b]

	Aprox5 - ClasicoL	Unif5 - ClasicoL	Pol5 - ClasicoL
Z	-6.568[a]	-6.567[a]	-6.567[a]
Asymp. Sig. (2-tailed)	.000	.000	.000

a. Based on negative ranks.

b. Wilcoxon Signed Ranks Test

Tablas 3.6 Test de Wilcoxon para secuencias de tamaño 100 y suavizamiento cinco, sin los tamaños de ventana pequeños.

Al realizar el análisis de los métodos borrosos con incertidumbre cinco, se mantienen los resultados de forma altamente significativos.

Para terminar, se decidió comparar los datos correspondientes a los diferentes suavizados, para determinar cual de ellos es el mejor. Se utilizó un test de Wilcoxon. Los resultados se muestran en la Tabla 3.7.

Test Statistics[b]

	Aprox5 - Aprox3	Unif5 - Unif3	Pol5 - Pol3
Z	-4.978[a]	-4.823[a]	-4.960[a]
Asymp. Sig. (2-tailed)	.000	.000	.000

a. Based on negative ranks.

b. Wilcoxon Signed Ranks Test

Tablas 3.7 Test de Wilcoxon en secuencias de tamaño 100 para comparar los datos resultantes de los suavizados tres y cinco en las tres varientes borrosas.

Como puede apreciarse existen diferencias significativas. Puede concluirse que aumentar el suavizado produce resultados significativamente mejores a la hora de detectar conglomerados verdaderos.

3.2.2 Falsos conglomerados en secuencia de tamaño 100

La Figura 3.2 muestra el comportamiento del método Scan Clásico Generalizado ante falsos conglomerados.

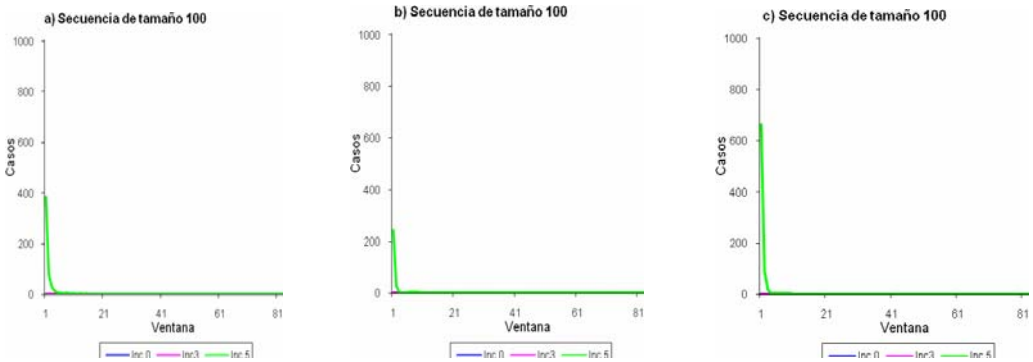

Figura 3.2 Métodos Scan Lineal en secuencias de tamaño 100 ante falsos conglomerados. Los métodos borrosos se calcularon según a) variante aproximada, b) variante Poisson-Uniforme, c)variante Polinomial.

Ante falsos conglomerados, el método Clásico responde acertadamente en el 100% de los casos. Este comportamiento es imposible de mejorar. Parece ser que, en la medida en la que se incrementa el suavizado, comienza el método, erróneamente, a detectar como verdaderos, falsos conglomerados. Probemos esto con ayuda de un test de Friedman.

Ranks

	Mean Rank
ClasicoL	2.36
Aprox3	2.62
Unif3	2.50
Pol3	2.52

Test Statistics[a]

N	100
Chi-Square	20.640
df	3
Asymp. Sig.	.000

a. Friedman Test

Tablas 3.8 Test de Friedman para secuencias de tamaño 100 en falsos conglomerados y suavizamiento tres.

Ranks

	Mean Rank
ClasicoL	1.97
Aprox5	2.84
Unif5	2.52
Pol5	2.67

Test Statistics[a]

N	100
Chi-Square	91.243
df	3
Asymp. Sig.	.000

a. Friedman Test

Tablas 3.9 Test de Friedman para secuencias de tamaño 100 en falsos conglomerados y suavizamiento cinco.

Como puede apreciarse en ambos casos existen diferencias significativas. Si fijamos la atención en la tabla de los rangos medios, puede apreciarse que el valor menor

corresponde al método clásico. Puede concluirse entonces que aumentar el suavizado produce resultados significativamente peores a la hora de detectar falsos conglomerados.

3.2.3 Curvas ROC para secuencias de tamaño 100

¿Qué decisión se debe tomar entonces? Los métodos borrosos lineales producen mejores resultados que su alternativa clásica, pero ¿se debe aumentar o no el suavizamiento? Si se quiere detectar conglomerados verdaderos con mayor seguridad debemos aumentarlo, pero se correrá el riesgo de detectar como verdaderos falsos conglomerados. Si por el contrario lo más importante es no detectar falsos, entonces se debe seleccionar un tamaño de suavizado pequeño.

¿Cuál es entonces el mejor clasificador? ¿Cuál de ellos toma la mejor decisión y se equivoca menos (considerando igualmente malos ambos tipos de errores)? En este epígrafe se realizará un análisis de las curvas ROC.

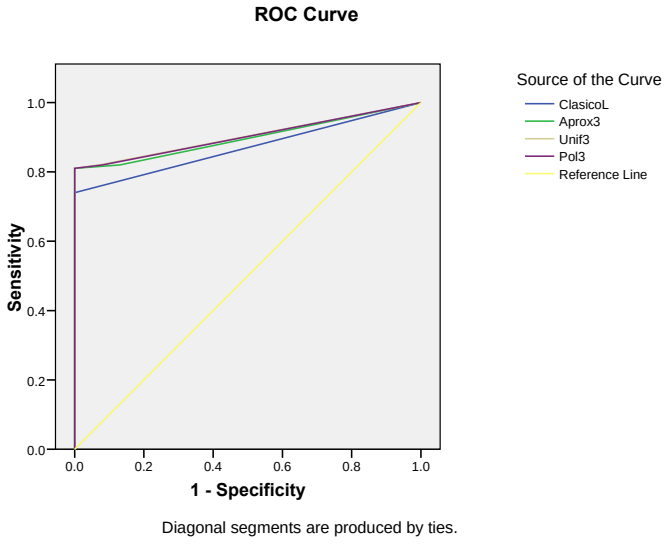

Figura 3.3 Curvas ROC en secuencias de tamaño 100. Los métodos borrosos tienen incertidumbre tres.

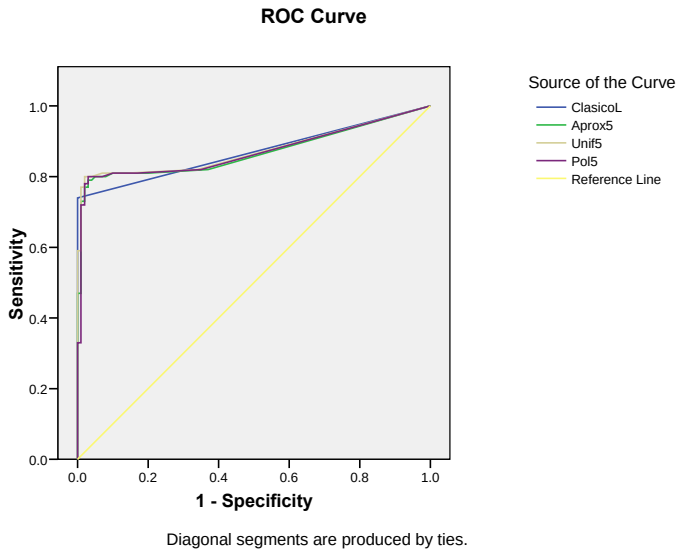

Figura 3.4 Curvas ROC en secuencias de tamaño 100. Los métodos borrosos tienen incertidumbre cinco.

A continuación se muestra una tabla resumen con los valores de las áreas bajo la curva.

Secuencia	**Método Scan Lineal**				
	Clásico	**Borroso**			
		Incertidumbre	**Aproximado**	**Poiss-Unif**	**Polinomial**
100	**0.870**	**3**	**0.898**	**0.903**	**0.902**
		5	**0.869**	**0.874**	**0.869**

Tablas 3.10 Área bajo la curva ROC en secuencias de tamaño 100.

Nótese que los mejores valores del área bajo la curva se observan en la variante dos (aproximación según las funciones Poisson Uniforme), utilizando un suavizamiento tres. Con esto quedan respondidas las interrogantes anteriores. La pregunta interesante ahora es: ¿se mantendrán estos resultados si se incrementa el tamaño de la secuencia?

3.3 Análisis de los resultados del los métodos lineales en secuencias de tamaño superiores

3.3.1 Verdaderos conglomerados en secuencia de tamaños superiores

A continuación se muestran los resultados de aplicar los métodos de Scan Lineal a secuencias mayores (250 y 500) que representan verdaderos conglomerados.

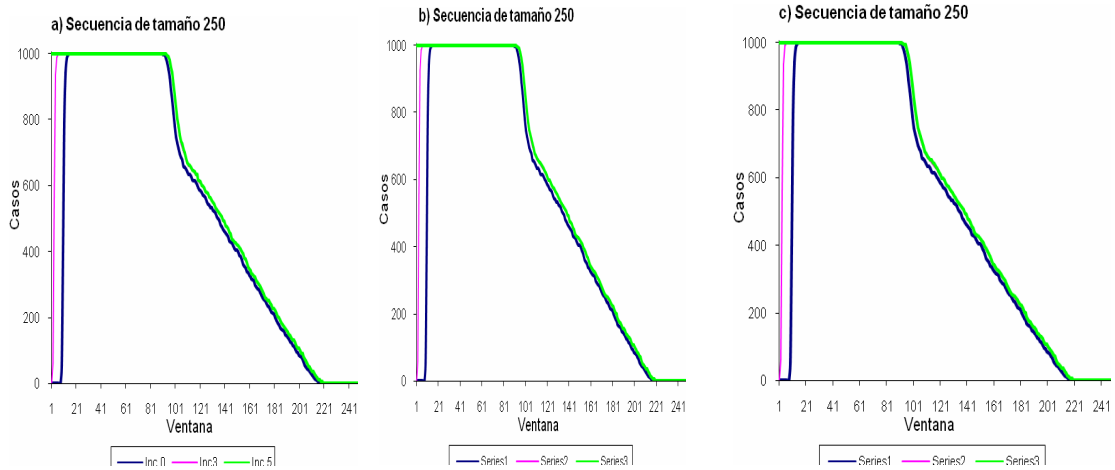

Figura 3.5 Métodos Scan Lineal en secuencias de tamaño 250. Los métodos borrosos se calcularon según a) variante aproximada, b) variante Poisson-Uniforme, c) variante Polinomial.

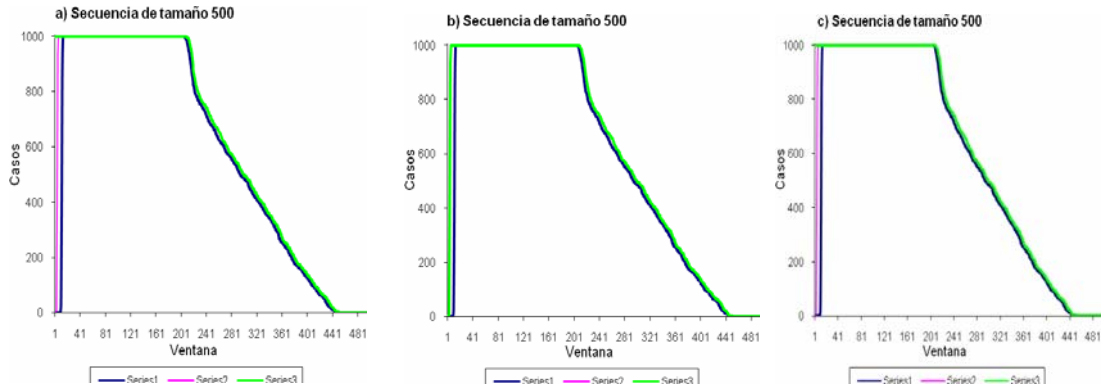

Figura 3.6 Métodos Scan Lineal en secuencias de tamaño 500. Los métodos borrosos se calcularon según a) variante aproximada, b) variante Poisson-Uniforme, c) variante Polinomial.

Como puede apreciarse, el comportamiento es muy similar en ambos gráficos. En todos ellos existe un máximo de tipo meseta que se propaga hasta un poco antes de la mitad de la secuencia que se analiza.

3.3.2 Falsos conglomerados en secuencia de tamaños superiores

A continuación se muestran los resultados de aplicar los métodos de Scan Lineal a secuencias mayores (250 y 500) que no representan conglomerados.

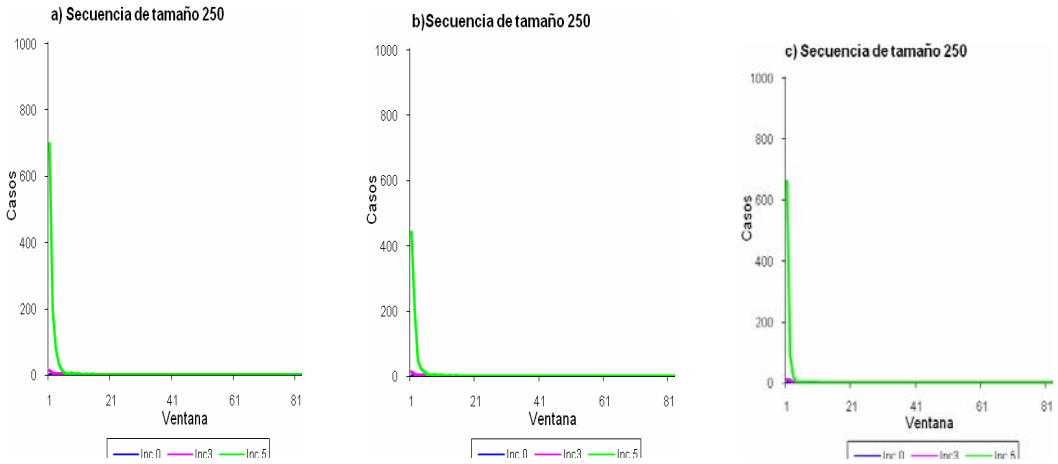

Figura 3.7 Métodos Scan Lineal en secuencias de tamaño 250 ante falsos conglomerados. Los métodos borrosos se calcularon según a) variante aproximada, b) variante Poisson-Uniforme, c) variante Polinomial.

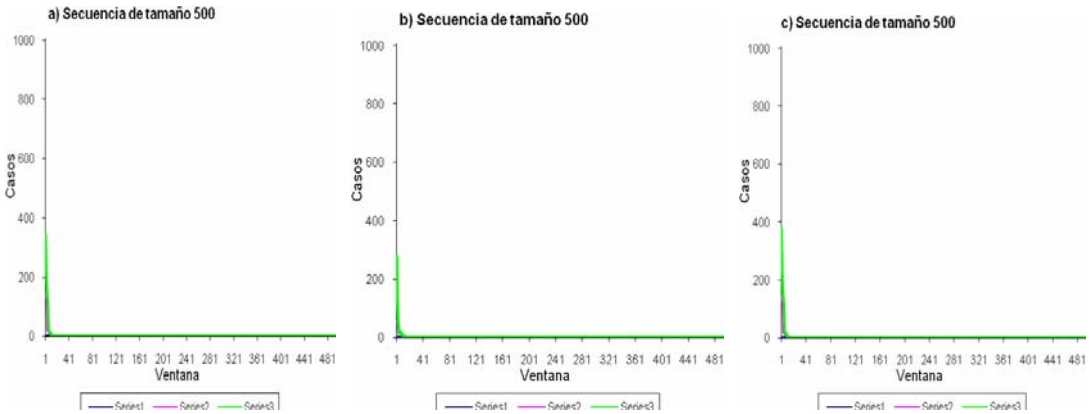

Figura 3.8 Métodos Scan Lineal en secuencias de tamaño 500 ante falsos conglomerados. Los métodos borrosos se calcularon según a) variante aproximada, b) variante Poisson-Uniforme, c) variante Polinomial.

Puede apreciarse que el comportamiento del método ante estas secuencias es muy similar al analizado para secuencias menores (de tamaño 100).

3.3.3 Análisis de las curvas ROC para los métodos lineales

La detección de conglomerados usando las técnicas del Scan Lineal puede considerarse un problema de clasificación. Dada una secuencia de longitud n habrá que determinar si existe o no cluster de conglomerados utilizando el Scan Lineal Clásico (o Scan Lineal Borroso con incertidumbre 0) y tres variantes de Scan Lineal Borroso: Aproximado y distribución de Poisson y Uniforme e Interpolación de polinómio. El área por debajo de la curva ROC se calculó para cada variante y suavizado usados en cada uno de los clasificadores. Los resultados son mostrados en la Tabla 3.11 (Nótese que se adicionaron los resultados ya analizados para secuencias de tamaño 100).

Población	**Método Scan Lineal**				
	Clásico	**Borroso**			
		Suavizamiento	**Aproximado**	**U**	**P**
100	**0.870**	**3**	**0.898**	**0.903**	**0.902**
		5	**0.869**	**0.874**	**0.869**
250	**0.912**	**3**	**0.925**	**0.935**	**0.937**
		5	**0.920**	**0.930**	**0.931**
500	**0.939**	**3**	**0.949**	**0.949**	**0.949**
		5	**0.949**	**0.947**	**0.949**

Tablas 3.11 Area bajo la curva ROC en los métodos lineales.

Es de destacar que en todos los casos, cualquiera de las variantes borrosas analizadas, muestra resultados más favorables que la versión clásica correspondiente a ella. Este es un hecho importante porque muestra la superioridad de los métodos borrosos con respecto al lineal clásico.

3.4 Análisis de los resultados de los métodos circulares.

A continuación se repite el mismo análisis, ahora considerando los métodos circulares.

3.4.1 Verdaderos conglomerados en secuencia de tamaños superiores

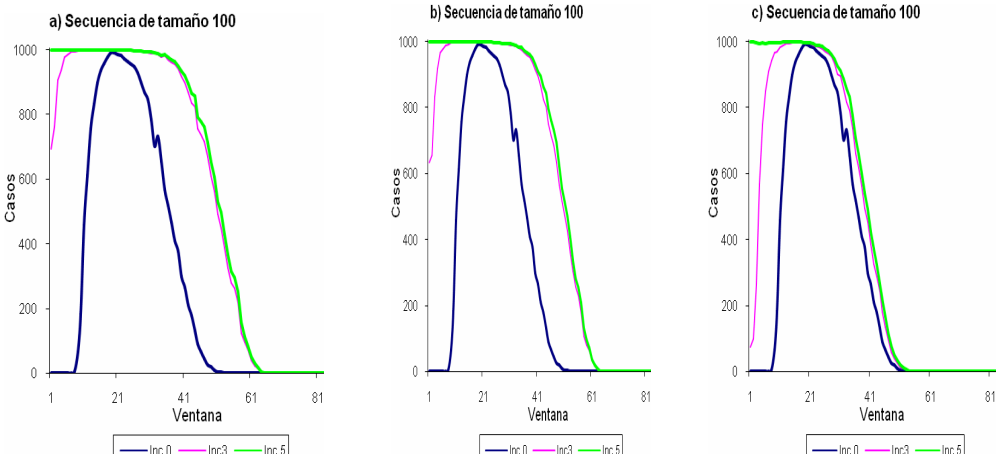

Figura 3.9 Métodos Scan Circular en secuencias de tamaño 100 ante verdaderos conglomerados. Los métodos borrosos se calcularon según a) variante aproximada, b) variante Poisson-Uniforme, c) variante Polinomial.

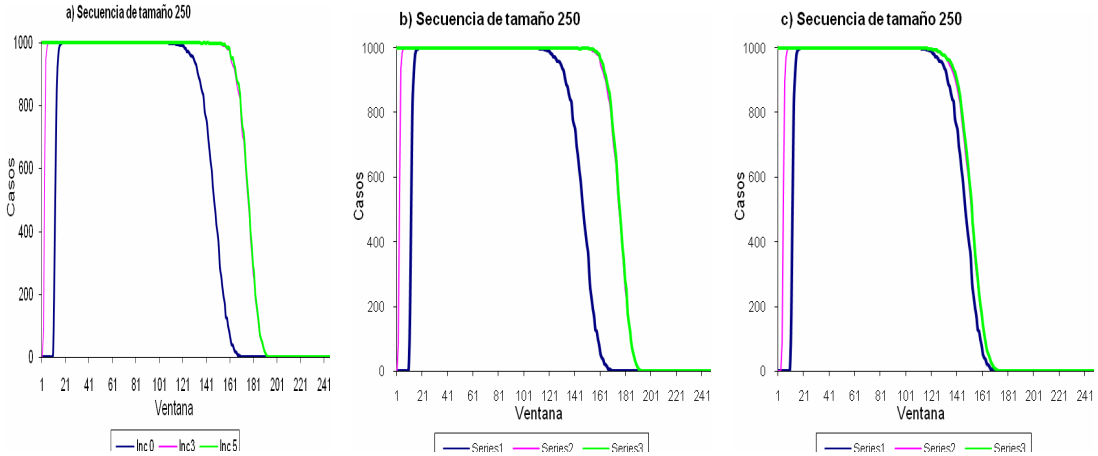

Figura 3.10 Métodos Scan Circular en secuencias de tamaño 250 ante verdaderos conglomerados. Los métodos borrosos se calcularon según a) variante aproximada, b) variante Poisson-Uniforme, c) variante Polinomial.

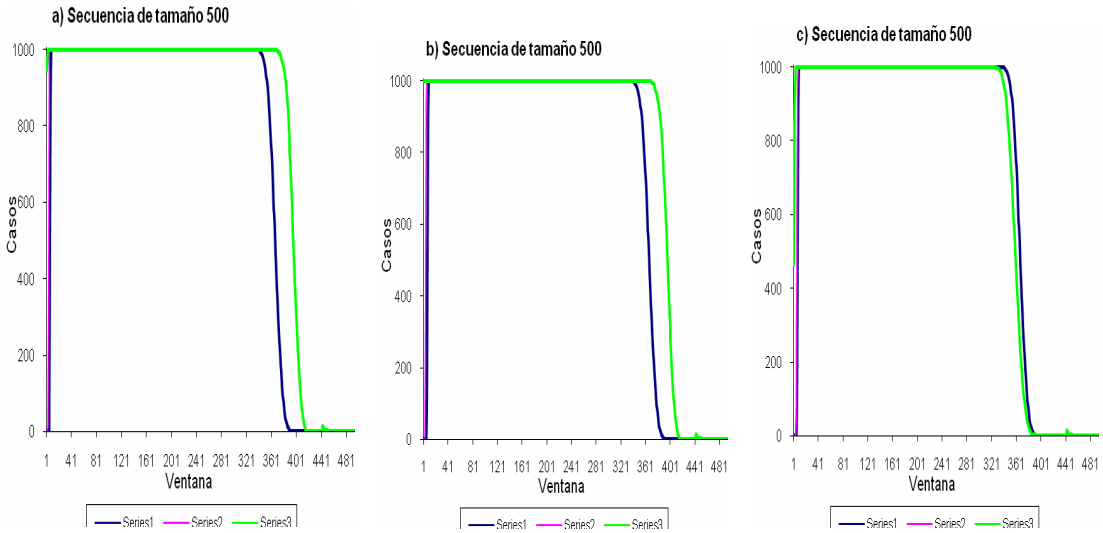

Figura 3.11 Métodos Scan Circular en secuencias de tamaño 500 ante verdaderos conglomerados. Los métodos borrosos se calcularon según a) variante aproximada, b) variante Poisson-Uniforme, c) variante Polinomial.

Si la secuencia es corta (n=100), resulta evidente la superioridad de los métodos borrosos. En la medida en la que aumenta el tamaño de la secuencia, todos los métodos circulares tienden a coincidir. En estos gráficoa también aparecen máximos de tipo meseta, que se incrementan con el aumento de la secuencia a considerar.

3.4.2 Falsos conglomerados en secuencia de tamaños superiores

Gráficamente el análisis de los métodos circulares ante secuencias con falsos conglomerados se comportó de la misma forma que frente a los métodos lineales. En la medida en la que se incrementa el suavizado, los métodos comienzan a detectar como verdaderos a los falsos conglomerados, ver Figuras 3.7 y 3.8.

3.4.3 Análisis de las curvas ROC para los métodos circulares

En este epígrafe se realiza el análisis de las curvas ROC para los métodos circulares. Los resultados se muestran en la Tabla 3.11.

Población	Método Scan Circular				
	Clásico	Borroso			
		Suavizamiento	Aproximado	U	P
100	0.730	3	0.798	0.813	0.770
		5	0.753	0.749	0.764
250	0.824	3	0.887	0.888	0.854
		5	0.886	0.887	0.852
500	0.888	3	0.928	0.928	0.898
		5	0.929	0.929	0.901

Tabla 3.11 Area bajo la curva ROC en los métodos circulares

De forma general se muestra la superioridad de los métodos borrosos, siendo esta más notable cuando el tamaño de la secuencia es mayor.

3.5 Una aplicación bioinformática

La Epilepsia Progresiva Miclónica de tipo Unverrricht-Lundborg (EPM1) es una enfermedad congénita autosómica recesiva. La causa de esta enfermedad es una mutación en el gen (CSTB) que codifica un inhibidor de la cistein proteasa, la cual consiste en la repetición anormal (35-70 veces) del dodecámero (CCCCGCCCCGCG) que se encuentra repetido de dos a tres veces en la región cinco del gen en los individuos sanos (Rodríguez, 2007). A continuación se muestran los resultados de la detección de conglomerados de este dodecámero en varias secuencias del gen de diferentes personas sanas o enfermas y con distintos métodos se scan

Observe que en la secuencias de personas sanas ninguno de los métodos del Scan detecta conglomerados, mientras que para personas enfermas el Scan Borroso detecta siempre conglomerados para los tamaños de ventanas móvil indicados, pero el Scan Clásico Generalizado solo detecta conglomerados para ventanas relativamente mayores o iguales a 15.

Pacientes	Secuencia	Ventana móvil	Scan	
			Clásico	Borroso (Suavizado 3)
Sanos	$(CCCCGCCCCGCG)_2$	1	0.8647	0.3890
		5	0.3565	0.3565
		15	0.7448	0.7448
	$(CCCCGCCCCGCG)_3$	1	0.9502	0.1380
		5	0.1523	0.1523
		15	0.6813	0.6813
Enfermos	$(CCCCGCCCCGCG)_{35}$	1	1.0000	0.0004
		5	0.1202	0.0000
		15	0.0000	0.0000
	$(CCCCGCCCCGCG)_{70}$	1	1.0000	0.0063
		5	0.7435	0.0017
		15	0.0006	0.0000

Table 3.12. Resultados del Scan Clásico Generalizado y Borroso aplicado a la secuencia del gen CSTB

3.6 Consideraciones finales del capítulo

En este capítulo se desarrolló un estudio de simulación sobre los métodos Scan estudiados con anterioridad. Se generaron secuencias con verdaderos y falsos conglomerados y se determinó el comportamiento de los métodos ante ellas.

Se utilizaron algunos tests no paramétricos (Friedman y Wilcoxon) sobre secuencias de tamaño 100 a manera de ejemplo, para mostrar el comportamiento de las tres variantes del método lineal borroso presentadas. Finalmente, las curvas ROC demostraron estadísticamente la superioridad de los métodos borrosos, sobre sus alternativas clásicas.

El capítulo culmina con la presentación de un problema sencillo de bioinformática.

Conclusiones

En este trabajo se arriban a las siguientes conclusiones:

1. Se diseñó un sistema para la detección de conglomerados utilizando los métodos Scan Lineal y Circular en sus variantes generalizadas y borrosas.
2. Se creó una aplicación de tal sistema utilizando el Java como lenguaje de programación de alto nivel. Todos los resultados se validaron con funciones similares implementadas en el paquete *Mathematica*.
3. Se validaron todos los métodos implementados realizando un estudio de simulación. Para lograrlo se utilizaron algunas pruebas estadísticas no paramétricas y las curvas ROC. Se demostró la superioridad de los métodos borrosos.
4. Se presentó un ejemplo bioinformático sencillo que ilustra la fortaleza e importancia de los métodos borrosos.

Recomendaciones

1. Ampliar el estudio de simulación a secuencias mucho mayores, de 1000 en lo adelante.
2. Paralelizar los métodos Scan, de manera que pueda implementarse una aplicación que corra sobre un cluster de computadoras.
3. Buscar otros ejemplos bioinformáticos para la validación de los métodos.

Bibliografía

ALDRICH, T. & WANZER, D. (1993) “Cluster” The agency for Toxic Substances and Disease Registry Division of Health Studies.

BRADLEY, A. P. (1997) The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*.

BUCKLEY, J. J. (2006) Fuzzy Probability and Statistics.

CASAS, C. G. (2003a) Técnicas de detección de conglomerados incluyendo factores adicionales. *Departamento de Computación.* Santa Clara, Universidad Central "Marta Abreu" de Las Villas.

CASAS, C. G. (2003b) Manual de usuario del EpiDet.

CASAS, C. G. & RODRÍGUEZ, C. L. (2008) Generalización de dos métodos de detección de conglomerados. Aplicaciones en bioinformática. *Revista de Matemática: Teoría y Aplicaciones*

DADONE, P. (2001) Design Optimization of Fuzzy Logic Systems Faculty of the Virginia Polytechnic Institute and State University.

GLAZ, J. (1993) Approximations for the tail probabilities and moments of the Scan statistics. *Statistics in Medicine,* 12: 1845-52.

JACQUEZ, G. (1996) *Infection Control and Hospital Epimediology*.

KULLDORFF, M. (2007) SaTScan.

NAGARWILLA, N. (1996) *Scan statistic with a variable window*.

NAUSS, J. (1982) Approximations for distributions of Scan statistics. *Journal of the Am. Stat. Assoc.*

RODRÍGUEZ, C. L. (2007) Validacion del metodo scan con verdaderos y falsos conglomerados *Compumat.*

SAHU, S. & COL. (1993) Effect of relative risk and cluster configuration on the power of the one-dimensional Scan statistics. *Statistics in Medicine,* 12: 1853-65.

Anexo

*** * * Método Scan * * ***

```
Needs["Statistics`DiscreteDistributions`"]

Needs["Statistics`DescriptiveStatistics`"]

Needs["Statistics`Common`DistributionsCommon`"]

Fnn[m_,n_]:=Module[{},If[n<0,p:=0,p:=CDF[PoissonDistribution[m],n]];Return[N[p,10]];]

Psi[m_,i_]:=Module[{},p:=PDF[PoissonDistribution[m],i];Return[N[p,10]]]

Clear[A1];

A1[m_,n_]:=2 Psi[m,n] Fnn[m,n-1] ((n-1) Fnn[m,n-2]-m Fnn[m,n-3])

Clear[A2];

A2[m_,n_]:=0.5 Psi[m,n]^2 ((n-1) (n-2) Fnn[m,n-3]-2 (n-2) m Fnn[m,n-4]+m^2 Fnn[m,n-5])

Clear[A3];

A3[m_,n_]:=Sum[Psi[m,2 n-r] Fnn[m,r-1]^2,{r,1,n-1}]

Clear[A4];

A4[m_,n_]:=Sum[Psi[m,2 n-r] Psi[m,r] (r-1) Fnn[m,r-2]-m Fnn[m,r-3],{r,2,n-1}]

Clear[Q2];

Q2[m_,n_]:=Fnn[m,n-1]^2-(n-1) Psi[m,n] Psi[m,n-2]-(n-1-m) Psi[m,n] Fnn[m,n-3]

Clear[Q3];

Q3[m_,n_]:=Fnn[m,n-1]^3-A1[m,n]+A2[m,n]+A3[m,n]-A4[m,n]

Clear[Q];

Q[m_,n_,L_]:=Q2[m,n] (Q3[m,n]/Q2[m,n])^(L-2)

Clear[Pfinal];
```

```
Pfinal[m_,n_,L_]:=1-Q[m,n,L]

Clear[Scanfinal];

Scanfinal[n_,lambdaL_,r_]:=Module[{},L:=1/r;m:=lambdaL/L;Print[N[Pfinal[m,n,L],10
]]]

Clear[Scanfinal1];

Scanfinal1[n_,lambda_,L_]:=Module[{},m:=lambda;Print[N[Pfinal[m,n,L],10]]]

ScanTemp[n_,p_,r_,Ancho_,Paso_,ProbL_,k_]:=

    CompoundExpression[ Print["******** n=",n," p=",p," r=",r," ********"];

     si=0; no=0; med=0;

     For[i=1,i<=k,i++,

           ndiv2=Quotient[n,2];

        u=Table[Random[],{ ndiv2}];

        SumUnif=Table[∑_{d=1}^{j} u_d ,{j,1,ndiv2}] ;

         (* Para generar la exponencial *)

         unif=Table[Random[],{n}];

          exponenc[j_]:=-Log[1-unif[[j]]]/((p+j-1) (n-j+1) r);

        SumExp=Table[ SumUnif[[ndiv2]]+ ∑_{d=1}^{j} exponenc_d ,{j,1,n}];

         Resul=Join[SumUnif,SumExp];

         conv=Resul;

  ExpNegList=conv;

  (* Incluyendo factores de riesgo segun principio fundamental *)

        CantFactores=Length[ProbL];

        For[fr=1,fr≤CantFactores,fr++,

   For [y=Length[conv],y≥1,y--,bernoulli=Random[];
```

```
   If[bernoulli>ProbL[[fr]],conv[[y]]=0]];
 ]; (* For fr *)
 For[y=n+ndiv2,y≥1,y--, If[conv[[y]]==0,conv=Delete[conv,y]]];
       ExpNegList=conv;
 inic=ExpNegList[[1]];
       fin=Last[ExpNegList]; (* fin es el ultimo dato *)
       MinWin=inic;(* es el inicio de la ventana movil *)
       MaxWin=inic+Ancho;(* es el fin de la ventana movil *)
       Window={};
       While [MaxWin <= fin,
          TempList=Select[ExpNegList,#1>=MinWin&];
          contador=Length[Select[TempList,#1<=MaxWin&]];
          Window=Insert[Window,contador,Length[Window]+1];
          MinWin=MinWin+Paso;
          MaxWin=MaxWin+Paso;
        ]; (* While *)
     maximo=Max[Window];
     suma=0;
     media=Mean[Window];
     L=(fin-inic)/Ancho;(* Print["L=",L]; *)
     signif=N[Pfinal[media,maximo,L],10];
     If[(signif<0.05),si++,If[(signif<=0.1),med++,no++]]
 ];
Print["Si=",si," No=",no," Med=",med];
```

```
 ] (*ScanTemp*)
ScanTempPot[n_,p_,r_,Ancho_,Paso_,k_]:=
   CompoundExpression[ Print["******** n=",n," p=",p," r=",r," ********"];
    si=0; no=0; med=0;
    For[i=1,i<=k,i++,
          u=Table[Random[],{n}];
        SumUnif=Table[ u[[j]] ,{j,1,n}] ;
         Resul=SumUnif;
  IntResul=Round[Resul];
          conv=86400 IntResul+3124051200;
        ExpNegList=conv;
 inic=ExpNegList[[1]];
       fin=Last[ExpNegList]; (* fin es el ultimo dato *)
       MinWin=inic;(* es el inicio de la ventana movil *)
       MaxWin=inic+Ancho;(* es el fin de la ventana movil *)
       Window={};
       While [MaxWin <= fin,
          TempList=Select[ExpNegList,#1>=MinWin&];
          contador=Length[Select[TempList,#1<=MaxWin&]];
          Window=Insert[Window,contador,Length[Window]+1];
          MinWin=MinWin+Paso;
          MaxWin=MaxWin+Paso;
       ]; (* While *)
 maximo=Max[Window];
```

```
      suma=0;
      media=Mean[Window];
      L=(fin-inic)/Ancho;
      signif=N[Pfinal[media,maximo,L],10];
      If[(signif<0.05),si++,If[(signif<=0.1),med++,no++]]
 ];
Print["Si=",si," No=",no," Med=",med];  ] (*ScanTempPot*)
```