FACULTAD DE INFORMÁTICA
UNIVERSIDAD POLITÉCNICA DE MADRID

TÉSIS DE MÁSTER
MÁSTER UNIVERSITARIO EN
INTELIGENCIA ARTIFICIAL

CARACTERIZACIÓN Y SIMULACIÓN DE ARBORIZACIONES DENDRÍTICAS CON REDES BAYESIANAS INCLUYENDO VARIABLES ANGULARES

AUTOR : Luis Rodríguez Luján
TUTORES : Pedro Larrañaga Múgica
Concha Bielza Lozoya

Jul, 2015

Reconocimientos

En primer lugar quiero expresar mi agradecimiento a Concha Bielza y Pedro Larrañaga por transmitirme una ínfima parte de su conocimiento y experiencia para la realización de este trabajo así como por su compromiso en mi desarrollo personal y académico durante este tiempo. Ciertamente, hubiera sido imposible desarrollar este trabajo sin su tutela.

Quiero agradecer a mis compañeros del Computational Intelligence Group (CIG) su apoyo incondicional, sin el cual las largas jornadas de trabajo hubiesen sido, sin lugar a dudas, insoportables.

Este trabajo no hubiese sido posible sin el apoyo económico del proyecto Human Brain Project.

Por último, pero no menos importante, quiero agradecer a mi familia y amigos por servirme de guía en todo momento, por aguantar mi completa dedicación que tan poco tiempo me ha permitido disfrutar de su compañía. En especial me gustaría agradecer a Amparo e Irene por sus valiosos consejos, aunque parezca que siempre hago lo contrario y a Gema, por aguantar mi ausencia durante este tiempo y ayudar en todo lo posible.

Resumen

El funcionamiento interno del cerebro es todavía hoy en día un misterio, siendo su comprensión uno de los principales desafíos a los que se enfrenta la ciencia moderna. El córtex cerebral es el área del cerebro donde tienen lugar los procesos cerebrales de más alto nivel, cómo la imaginación, el juicio o el pensamiento abstracto. Las neuronas piramidales, un tipo específico de neurona, suponen cerca del 80 % de los cerca de los 10.000 millones de que componen el córtex cerebral, haciendo de ellas un objetivo principal en el estudio del funcionamiento del cerebro.

La morfología neuronal, y más específicamente la morfología dendrítica, determina cómo estas procesan la información y los patrones de conexión entre neuronas, siendo los modelos computacionales herramientas imprescindibles para el estudio de su rol en el funcionamiento del cerebro. En este trabajo hemos creado un modelo computacional, con más de 50 variables relativas a la morfología dendrítica, capaz de simular el crecimiento de arborizaciones dendríticas basales completas a partir de reconstrucciones de neuronas piramidales reales, abarcando desde el número de dendritas hasta el crecimiento los los árboles dendríticos. A diferencia de los trabajos anteriores, nuestro modelo basado en redes Bayesianas contempla la arborización dendrítica en su conjunto, teniendo en cuenta las interacciones entre dendritas y detectando de forma automática las relaciones entre las variables morfológicas que caracterizan la arborización. Además, el análisis de las redes Bayesianas puede ayudar a identificar relaciones hasta ahora desconocidas entre variables morfológicas.

Motivado por el estudio de la orientación de las dendritas basales, en este trabajo se introduce una regularización L_1 generalizada, aplicada al aprendizaje de la distribución von Mises multivariante, una de las principales distribuciones de probabilidad direccional multivariante. También se propone una distancia circular multivariante que puede utilizarse para estimar la divergencia de Kullback-Leibler entre dos muestras de datos circulares. Comparamos los modelos con y sin regularización en el estudio de la orientación de la dendritas basales en neuronas humanas, comprobando que, en general, el modelo regularizado obtiene mejores resultados. El muestreo, ajuste y representación de la distribución von Mises multivariante se implementa en un nuevo paquete de R denominado *mvCircular*.

Abstract

The inner workings of the brain are, as of today, a mystery. To understand the brain is one of the main challenges faced by current science. The cerebral cortex is the region of the brain where all superior brain processes, like imagination, judge and abstract reasoning take place. Pyramidal neurons, a specific type of neurons, constitute approximately the 80 % of the more than 10.000 million neurons that compound the cerebral cortex. It makes the study of the pyramidal neurons crucial in order to understand how the brain works.

Neuron morphology, and specifically the dendritic morphology, determines how the information is processed in the neurons, as well as the connection patterns among neurons. Computational models are one of the main tools for studying dendritic morphology and its role in the brain function. We have built a computational model that contains more than 50 morphological variables of the dendritic arborizations. This model is able to simulate the growth of complete dendritic arborizations from real neuron reconstructions, starting with the number of basal dendrites, and ending modeling the growth of dendritic trees. One of the main differences between our approach, mainly based on the use of Bayesian networks, and other models in the state of the art is that we model the whole dendritic arborization instead of focusing on individual trees, which makes us able to take into account the interactions between dendrites and to automatically detect relationships between the morphologic variables that characterize the arborization. Moreover, the posterior analysis of the relationships in the model can help to identify new relations between morphological variables.

Motivated by the study of the basal dendrites orientation, a generalized L_1 regularization applied to the multivariate von Mises distribution, one of the most used distributions in multivariate directional statistics, is also introduced in this work. We also propose a circular multivariate distance that can be used to estimate the Kullback-Leibler divergence between two circular data samples. We compare the regularized and unregularized models on basal dendrites orientation of human neurons and prove that regularized model achieves better results than non regularized von Mises model. Sampling, fitting and plotting functions for the multivariate von Mises are implemented in a new R packaged called *mvCircular*.

Contenidos

Lista de imágenes IX

Lista de tablas XI

1. Introducción 1

1.1. Estadística direccional multivariable 3
1.2. Morfología de las neuronas piramidales 4
1.3. Objetivos 5
1.4. Planificación de la tésis 6

I Estadística circular multivariante 9

2. Muestreo de las distribuciones circulares 11

2.1. Muestreo de la distribución von Mises univariante 11
2.2. Muestreo por rechazo 12
2.3. Muestreador aleatorio de Gibbs 13
2.3.1. Convergencia: *burn-in* 14
2.3.2. Autocorrelación: *thinning* 15
2.4. Comparación y selección del método 16

3. Aprendizaje de parámetros 19

3.1. Verosimilitud 19
3.2. Pseudo-verosimilitud 19
3.2.1. Optimización 20
3.3. Aprendizaje regularizado 22
3.3.1. Minimización 22

4. Evaluación 25

4.1. Divergencia Kullback-Leibler circular 25

4.2. Experimentos 26
4.3. Resultados 29
4.4. Aplicación a datos morfológicos de neuronas humanas 31

II Arborizaciones dendríticas: Caracterización y simulación 35

5. Obtención, preprocesamiento y caracterización de la arborización 37
5.1. Introducción 37
5.1.1. Modelos en la literatura 38
5.2. Extracción y pre-procesamiento 40
5.2.1. Lectura y digitalización 41
5.2.2. Reparación y filtrado 43
5.2.3. Centrado y alineación 44
5.3. Descriptores 45
5.3.1. Elementos de la arborización 46
5.3.2. Variables de construcción 49
5.3.3. Variables por conjunto 49
5.3.4. Variables por segmento 50
5.3.5. Variables del nodo 51

6. Modelización 53
6.1. Radio del soma 53
6.2. Localización de la raíz de las dendritas basales 55
6.3. Crecimiento simulado 57
6.3.1. Función de selección 57
6.3.2. Obtención de descriptores 58
6.4. Redes Bayesianas 59
6.4.1. Aprendizaje estructural y paramétrico 61
6.5. Interpretación 62

7. Simulación 65
7.1. Descripción del proceso 65
7.2. Crecimiento de la arborización 66
7.2.1. Muestreo y obtención de variables de construcción 67
7.3. Resultados y evaluación 68
7.3.1. Visual 69

7.3.2. Variables emergentes 69

8. Conclusiones y trabajo futuro **71**

8.1. Estadística circular multivariante 71

8.2. Reconstrucción de la arborización dendrítica basal 72

Apéndices **75**

Orientación de las raíces dendríticas **77**

Redes Bayesianas del modelo **79**

Bibliografía **89**

Lista de imágenes

1.1. Córtex cerebral 2
1.2. Esquema de la morfología de una neurona piramidal 5

2.1. Comparación de muestreadores 17

3.1. Tiempo de ajuste por número de muestras 23

4.1. Resultados concentración baja y matriz lambda nula 30
4.2. Resultados concentración baja y matriz lambda nula para número de muestras muy bajo 30
4.3. Resultados concentración alta y matriz lambda densa 31
4.4. Resultados para concentración entre 0.1 y 5 con matriz $\mathbf{\Lambda}$ dispersa . . 32
4.6. Variación de la concentración con $p = 5$, $n = 1000$ y matriz $\mathbf{\Lambda}$ nula . 32
4.5. Divergencia KL para concentración variable y matriz Λ dispersa . . . 33
4.7. Ángulos entre dendritas 33
4.8. Ángulos entre dendritas de neuronas del lóbulo occipital 34

5.1. Neurona con dendrita ápical mal etiquetada 43
5.2. Neurona con dendrita apical reetiquetada 44
5.3. Proceso de alineación 46
5.4. Subconjuntos de la arborización 47
5.5. Segmentos de la arborización 48
5.6. Variables de construcción 50

6.1. Ajuste del radio del soma 54
6.2. Ajuste del número de dendritas 55
6.3. Ajuste de la orientación 56
6.4. Obtención de descriptores mediante crecimiento simulado 59
6.5. Distribución de la longitud del segmento actual 61
6.6. Red Bayesiana de los segmentos raíz 63

6.7. Sub sed con los diámetros de construcción de una bifurcación 63

7.1. Proceso de conversión de valores discretos en reales 67
7.2. Simulación de una arborización 68

1. Ajuste de la orientación para tres dendritas 77
2. Ajuste de la orientación para cuatro dendritas 78
3. Ajuste de la orientación para cinco dendritas 78

4. Red Bayesiana para los segmentos raíz 79
5. Red Bayesiana para los nodos de elongación de orden centrífugo 0 . . 80
6. Red Bayesiana para los nodos de elongación de orden centrífugo 1 . . 81
7. Red Bayesiana para los nodos de elongación de orden centrífugo 2 . . 82
8. Red Bayesiana para los nodos de elongación de orden centrífugo superior a 2 83
9. Red Bayesiana para los nodos de bifurcación de orden centrífugo 0 . . 84
10. Red Bayesiana para los nodos de bifurcación de orden centrífugo 1 . . 85
11. Red Bayesiana para los nodos de bifurcación de orden centrífugo 2 . . 86
12. Red Bayesiana para los nodos de bifurcación de orden centrífugo superior a 2 87

Lista de tablas

4.1. Configuraciones para número de muestras variables 28
4.2. Configuraciones para concentración variable 29
4.3. Divergencia KL para ángulos entre dendritas 34

LISTA DE TABLAS

1

Introducción

Comprender los mecanismos que gobiernan el funcionamiento del cerebro es, casi con total seguridad, el reto más complejo al que se ha enfrentado la ciencia moderna. Con su estudio, los neurocientíficos pretenden ser capaces de desvelar los mecanismos que regulan los procesos cerebrales que determinan nuestra identidad como seres inteligentes: juicio, razonamiento abstracto, imaginación, etc. Además, los descubrimientos realizados en este área pueden motivar el desarrollo de nuevos métodos de aprendizaje automático, como es el caso de las redes neuronales.

Es posible establecer el comienzo de la neurociencia moderna a principios del siglo XX con los estudios realizados por Santiago Ramón y Cajal acerca de la morfología y procesos conectivos de las células nerviosas, en los que sienta las bases lo que actualmente se conoce como doctrina de la neurona, la cual sostiene que las neuronas son la unidad estructural y funcional del sistema nervioso (Ramón y Cajal, 1913). A pesar de todos los medios tecnológicos disponibles para el estudio del cerebro, y en concreto de la morfología neuronal, todavía estamos muy lejos de desentrañar el funcionamiento del cerebro, como bien refleja la siguiente cita del propio *Ramón y Cajal*: Ël cerebro es un mundo que consiste en un número de continentes inexplorados y grandes extensiones de territorio desconocido.

De acuerdo a dicha doctrina, las neuronas son los elemento básicos del cerebro, aunque la estructura de la neurona dista mucho de poder ser considerada *simple*. Los árboles dendríticos y axonales que parten desde el soma forman patrones de ramificación-elongación extraordinariamente complejos en un espacio tridimensional. El descubrimiento experimental a finales del siglo XX de que la morfología neuronal tiene un impacto directo en su respuesta electrofisiológica, ha supuesto que investigadores y proyectos a gran escala como el *Human Brain Project* o el *BRAIN initiative* inviertan una gran cantidad de recursos económicos y humanos en la investigación

de la morfología neuronal.

De entre todas las regiones del cerebro humano, la corteza o córtex cerebral que recubre los hemisferios cerebrales es tal vez el área que despierta mayor interés (figura 1.1). Este interés nace del hecho de que en la corteza cerebral se localizan procesos como el procesamiento del lenguaje, el pensamiento abstracto, la memoria, etc. En la literatura habitualmente se divide la corteza cerebral en capas de la I a la VI, siendo la capa I la más externa. Cada capa cortical contiene una distribución característica de neuronas y patrones de conexión específicos con otras regiones del cerebro, existiendo también conexiones entre las diferentes capas corticales. Sobre esta estructura se define el concepto de columna cortical, un grupo de neuronas perteneciente a diferentes capas corticales, altamente interconectadas formando *microcircuitos*. En la actualidad muchos neurocientíficos sostienen que las columnas corticales son las unidades funcionales en el córtex cerebral.

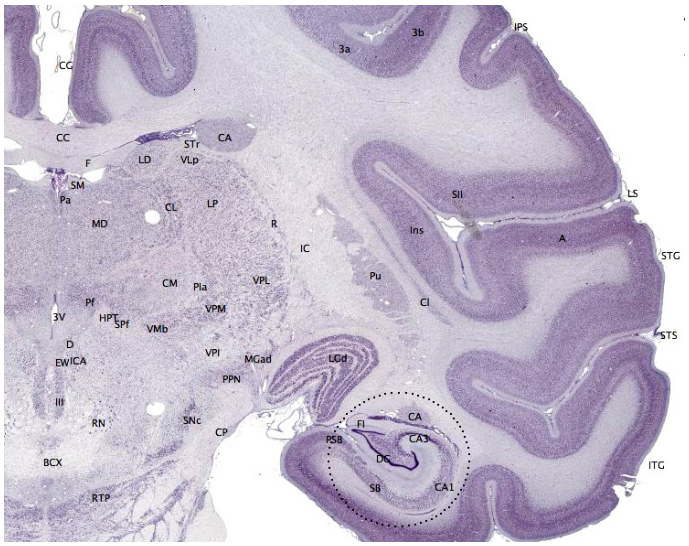


Figura 1.1: Imagen de una sección del cerebro. El córtex cerebral es la capa exterior en color violeta oscuro. Fuente: `http://brainmaps.org/ajax-viewer.jpg`

Dentro de la corteza cerebral destaca un tipo de neurona específico: la neurona piramidal. Llamada originalmente piramidal por la forma de su cuerpo celular (soma), las neuronas piramidales suponen entre el 75 % y el 85 % de las 10.000 millones de neuronas que componen el córtex cerebral. En este trabajo nos centraremos en crear un modelo capaz de explicar y simular la estructura de un componente específico de la neurona piramidal: la arborización basal. Éste y otros conceptos relativos a la morfología de las neuronas piramidales se explican en la sección 1.2 de esta introducción.

Motivado por la modelización de la orientación de las dendritas basales en neuronas piramidales, nos adentramos en el campo de la estadística direccional multivariante para el estudio de dicha orientación. La estadística direccional es un campo específico dentro de la estadística clásica que trabaja con direcciones, ángulos, rotaciones, etc. Si para la estadística clásica la distribución normal es la distribución más conocida, su análogo en el caso direccional es la distribución von Mises sobre la circunferencia. En la sección 1.1 realizaremos una breve introducción a los conceptos clave de la estadística direccional.

Finalmente, este trabajo se organiza como sigue:

- En la primera parte se detalla el estudio realizado sobre la distribución von Mises multivariante y la implementación del paquete *mvCircular* para trabajar con distribuciones circulares multivariantes

- En segundo lugar se expone el modelo creado para la caracterización y simulación de la arborización dendrítica basal de neuronas piramidales paso por paso, comenzando con la selección de los descriptores hasta finalizar con la simulación y evaluación de los resultados

- Finalmente, en una última sección expondremos las conclusiones obtenidas y las líneas de trabajo futuro

1.1. Estadística direccional multivariable

Los datos direccionales están presentes en múltiples campos científicos, desde la dirección del viento en meteorología hasta los ángulos de ramificación de los árboles, pasando por aplicaciones tan sorprendentes como la minería de textos. Por desgracia, los datos direccionales han sido tratados en la literatura como variables continuas lineales. La estadística direccional provee herramientas específicas para el tratamiento, estudio y modelado de datos circulares.

La distribución de von Mises es, con diferencia, la distribución circular más estudiada y extendida en la literatura; de hecho, se la conoce con frecuencia como la distribución normal en la circunferencia. Una de las principales ventajas de la distribución von Mises frente a la distribución normal envuelta en la circunferencia es que la von Mises pertenece a la familia exponencial, estando su función de densidad definida:

$$f_{VM}(\Theta;\mu,\kappa) = \frac{1}{2\pi I_0(\kappa)} \exp\{\kappa\cos(\theta-\mu))\}$$

donde μ es el ángulo medio, κ el parámetro de concentración, en otras palabras, la inversa de la varianza y I_0 la función de Bessel modificada de orden 0.

Tomando como referencia la definición exponencial de la distribución von Mises, podemos definir la distribución von Mises multivariante como la análoga de la distribución normal multivariante (Mardia et al., 2008). En este caso la función de densidad es:

$$f_{MVM}(\Theta;\boldsymbol{\mu},\boldsymbol{\kappa},\boldsymbol{\Lambda}) = \frac{1}{Z(\boldsymbol{\kappa},\boldsymbol{\Lambda})} \exp\{\boldsymbol{\kappa}^T c(\boldsymbol{\Theta}) + \frac{1}{2} s(\boldsymbol{\Theta})\boldsymbol{\Lambda} s(\boldsymbol{\Theta})\} \tag{1.1}$$

donde $c(\Theta)$ y $s(\Theta)$ son los vectores cuyas componente son $c_i(\boldsymbol{\Theta}) = \cos(\Theta_i - \mu_i)$ y $s_i(\boldsymbol{\Theta}) = \sin(\Theta_i - \mu_i)$ respectivamente.

Desafortunadamente, el principal problema de la distribución von Mises multivariante es que el término de normalización $Z(\boldsymbol{\kappa},\boldsymbol{\Lambda})$ no tiene una fórmula cerrada conocida cuando el número de dimensiones es mayor a 2, por lo que tiene que ser estimado de forma numérica.

1.2. Morfología de las neuronas piramidales

Las neuronas piramidales (Figura 1.2), descubiertas y estudiadas por Ramón y Cajal, deben su nombre a la forma piramidal del cuerpo celular o soma. Son el tipo de neurona más común en la corteza cerebral, aunque están también presentes en otras regiones como el hipocampo. Son consideradas una pieza clave en la conectividad neuronal y en los circuitos cerebrales encargados de los procesos cerebrales más avanzados.

Morfológicamente, se distinguen cuatro componentes principales en la neurona piramidal:

- El **soma** o cuerpo celular es la región de la neurona que contiene el núcleo y el resto de orgánulos comunes en las células eucariotas (ribosomas, mitocondrias, etc.). El soma es el encargado de realizar las tareas metabólicas de la neurona, generando los componentes necesarios para el mantenimiento estructural (microtúbulos) y la actividad químico-eléctrica (neurotransmisores)
- El **axón**, una prolongación que sale del cuerpo del cuerpo de la neurona hacia capas inferiores, con ramificaciones en la parte terminal. El axón suele ser le

prolongación neuronal (neurita) más larga y compleja. El axón es el canal de salida por el que la neurona se comunicará con otras neuronas, células musculares, glándulas, etc.

- La **dendrita apical**, otra extensión del soma hacia capas superiores, generalmente hasta la capa I. La dendrita apical suele presentar ramificaciones laterales a medida que asciende, con una ramificación final muy compleja denominada *apical tuft*. Tanto la dendrita apical como las basales se conectan con los axones de otras neuronas a través de las espinas dendríticas, formando lo que se conoce como sinapsis. Por lo tanto, las dendrita son los canales de entrada por los que se reciben estímulos del resto de neuronas

- Las **dendritas basales**, que parten de la base del soma y se propagan principalmente de forma horizontal en la misma capa. Al conjunto de dendritas basales se la denomina arborización basal, el cual suele formar complejos patrones de ramificación tridimensionales.

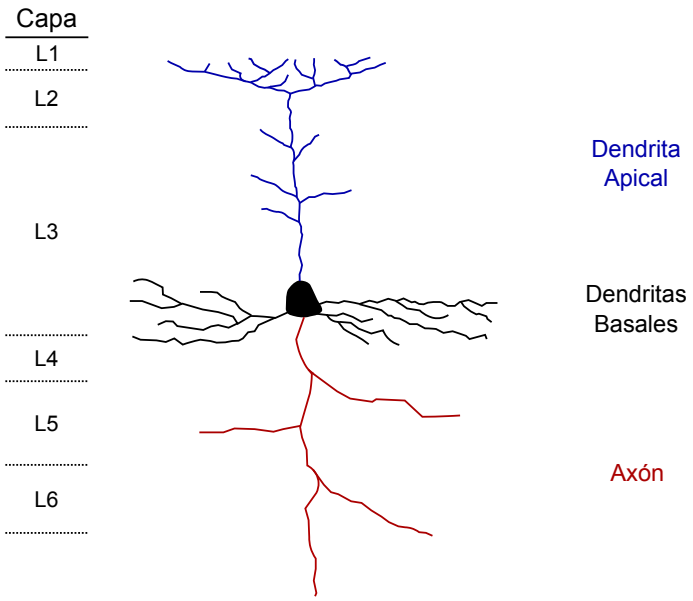


Figura 1.2: Representación esquemática de una neurona piramidal

1.3. Objetivos

Este proyecto tiene como objetivo el estudio morfológico de la arborización de las dendritas basales de neuronas piramidales. El trabajo está basado en el estu-

dio cuantitativo de la morfología dendrítica sobre el que se aplica un modelo de representación.

En la literatura encontramos una gran variedad de modelos y aplicaciones específicas que intentan modelizar, o bien un único árbol dendrítico, o bien una arborización completa. Encontraremos dos enfoques principales: modelos basados en el uso de reglas paramétricas para explicar y simular el crecimiento dendrítico, y por otra parte, modelos estadísticos basados en datos reales. No conocemos ningún modelo estadístico capaz de modelizar y simular la arborización basal en su conjunto teniendo en cuenta las interacciones entre los distintos árboles dendríticos, como sí hace nuestra aproximación.

Respecto a la estadística circular multivariante, nuestro objetivo es el estudio del impacto de la regularización generalizada en la distribución von Mises circular así como la implementación de un paquete informático que haga más accesible el uso de estas herramientas.

1.4. Planificación de la tésis

El desarrollo del proyecto ha seguido dos metodologías diferentes para cada una de las partes:

- En el estudio relativo a la estadística circular multivariante hemos seguido un proceso de desarrollo guiado por los experimentos a realizar. Partiendo de la idea inicial de que el uso de la regularización generalizada mejoraría el resultado del ajuste en cierto tipo de problemas con un fuerte conocimiento *a priori*, se determinaron los componentes funcionales necesarios en la implementación, la definición formal de los experimentos a realizar y, finalmente, se analizaron los resultados obtenidos en dichos experimentos.

- Para la creación del modelo de la arborización dendrítica basal se han seguido las fases típicas en el proceso de análisis de datos y extracción de conocimiento:

    1. Definir formalmente el problema y determinar el objetivo que se pretende conseguir con el proceso de análisis de datos
    2. Obtener un conjunto de datos sobre los que trabajar (*dataset*)
    3. Pre-procesar los datos. Lo que implica la interpretación del formato, definición de las estructuras de datos y labores de corrección y eliminación de ruido y errores

4. Transformar los datos, realizando un proceso de extracción de información
5. Elegir de un método para el realizar análisis datos adecuado para cumplir el objetivo marcado
6. Realizar el análisis los datos y elegir un algoritmo concreto del método escogido en el paso anterior
7. Detectar de patrones de interés en los datos
8. Interpretar de los patrones mediante representaciones gráficas y análisis estadístico
9. Incorporar y contrastar el conocimiento obtenido con el existente

Parte I

Estadística circular multivariante

2

Muestreo de las distribuciones circulares

Uno de los elementos imprescindibles para el tratamiento computacional de cualquier distribución de probabilidad, y por tanto de obligada implementación, es la capacidad de generar muestras aleatorias pertenecientes a dicha distribución. Si bien para las distribuciones univariantes más conocidas (e.g. la distribución normal) existen métodos específicos para el muestreo muy eficientes, en el caso de las distribuciones multivariantes existen muy pocos algoritmos específicos, teniendo que recurrir a métodos generales como el algoritmo de Metrópolis-Hastings (Hastings, 1970; Metropolis et al., 1953) o el muestrador de Gibbs (Gelfand y Smith, 1990), el cual puede ser entendido como un caso específico del método de Metropolis-Hastings.

En esta sección detallaremos los métodos de muestreo implementados en el paquete *mvCircular* para la distribución von Mises circular en varias variables, mencionando los pros y los contras de cada uno; en concreto, se han implementado dos algoritmos: el primero, un método basado en el algoritmo de muestreo por rechazo, y el segundo, un muestreador de Gibbs. Para el muestreo de la distribución *wrapped-normal* en varias dimensiones se ha adaptado la función de muestreo implementada en el paquete *tmvtnorm* para distribuciones normales truncadas en varias variables.

2.1. Muestreo de la distribución von Mises univariante

Como paso previo a la exposición de los métodos de muestreo de la distribución von Mises multivariante, es necesario determinar cómo obtener muestras de una

distribución von Mises univariante, ya que ésta será utilizada en ambos métodos multivariantes. En ésta sección describiremos el algoritmo de rechazo implementado en el paquete *mvCircular* (Best y Fisher, 1979) para el muestreo de la distribución von Mises univariante.

Con el fin de proporcionar una intuición rápida del funcionamiento del muestreo por rechazo, que se utilizará tanto para el muestreo de la distribución univariante como para la distribución multivariante, consideremos el siguiente ejemplo: supongamos que queremos obtener muestras de una distribución con función de densidad $p(\mathbf{x})$, para la que conocemos otra función de densidad $q(\mathbf{x})$ de la cual podemos obtener muestras de manera sencilla y además $p(\mathbf{x}) \leq Cq(\mathbf{x})$ con $C \geq 1$. El algoritmo de rechazo genera muestras $\mathbf{x}^{(i)} = q(\mathbf{x})$ (muestrea q) y las acepta con probabilidad $\frac{p(\mathbf{x}^{(i)})}{Cq(\mathbf{x}^{(i)})}$.

Para el muestreo de una distribución von Mises con parámetros μ y κ, la cual tiene una como función de densidad $f(\theta) = \frac{1}{2\pi I_0(\kappa)} \exp(\kappa \cos(\theta - \mu))$ utilizaremos una distribución *Wrapped-Cauchy* como cota superior para el método de rechazo, cuya función de densidad se expresa:

$$q(\theta) = \frac{(2\rho/\kappa)\exp\{\kappa(1+\rho^2)/2\rho - 1\}}{(1+\rho^2 - 2\rho\cos(\theta))(2\pi I_0(\kappa))}$$

donde:

$$\rho = \frac{\tau - \sqrt{2\tau}}{2\kappa}, \tau = \sqrt{1 + (1 + 4\kappa^2)}$$

La implementación del muestro por rechazo de la distribución von Mises univariante en el paquete *mvCircular* se ha realizado en ANSI C sin incluir una llamada a la rutina en R, ya que se trata una función auxiliar para los métodos multivariantes. El paquete *circular* de R proporciona otra implementación de este mismo algoritmo.

2.2. Muestreo por rechazo

Como se explicó en la sección anterior, para utilizar este algoritmo basta con definir una función de densidad q de la que podamos obtener muestras de manera sencilla, y calcular C de forma que se cumpla la desigualdad $p(\mathbf{x}) \leq Cq(\mathbf{x})$ en todo punto. Cuánto más se acerque la cota C a 1, mayor será la eficiencia del muestreador.

El algoritmo implementado en *mvCircular* (Mardia y Voss, 2014) permite generar muestras de una distribución von Mises multivariante con parámetros $\boldsymbol{\mu}$, $\boldsymbol{\kappa}$, $\boldsymbol{\Lambda}$ sujetos las siguientes restricciones:

- El número de variables p debe de ser pequeño, ya que la eficiencia del método decae exponencialmente a medida que aumenta p

- La matriz $\mathbf{P} = diag(\kappa_1, ..., \kappa_p) - \mathbf{\Lambda}$ debe ser definida positiva. Denotaremos $\lambda_{min} > 0$ al menor autovalor de $\mathbf{P}$

Además, supondremos sin pérdida de generalidad al ser una distribución rotacionalmente equivariante, que el vector media $\boldsymbol{\mu}$ es el vector nulo . Bajo estas premisas, la función q propuesta utiliza p ángulos independientes, obtenidos de distribuciones von Mises unidimensionales, para establecer la cota superior:

$$q(\mathbf{\Theta}) = \prod_{i=1}^{p} \frac{\exp\{\frac{\lambda_{min}}{4}\cos(2\theta_i)\}}{2\pi I_0(\frac{\lambda_{min}}{4})}$$

Con estas funciones, una muestra $\mathbf{\Theta}$ se acepta con probabilidad $\frac{f(\mathbf{\Theta})}{Cg(\mathbf{\Theta})}$, la cual se puede calcular de forma exacta:

$$p(\mathbf{\Theta}) = \exp\left\{\sum_{i=1}^{p} \kappa_i(c_i - 1) + \frac{1}{2}s^T(\mathbf{\Lambda} + \lambda_{min}I)s\right\}$$

donde:

$$s_i = \sin(\theta_i), c_i = \cos(\theta_i)$$

e I_0 corresponde a la función de Bessel modificada de primera especie de orden 0.

El método de muestreo ha sido implementado íntegramente en ANSI C con el objetivo de minimizar en la medida de lo posible el tiempo de ejecución del mismo junto con un procedimiento en R que hace uso de esta rutina.

En el mismo artículo (Mardia y Voss, 2014), se ofrece una cota superior para la probabilidad de aceptación a medida que aumenta el valor del parámetro de concentración $\boldsymbol{\kappa}$ dada por la expresión:

$$\frac{1}{2^p}\sqrt{\frac{\lambda_{min}^p}{|\mathbf{P}|}}$$

Tras analizar la expresión es posible ver que la eficiencia del algoritmo decrece exponencialmente con la dimensionaliad, algo que ya se preveía en las restricciones al comienzo del apartado, así como que el valor de los parámetros $\boldsymbol{\kappa}$ y $\mathbf{\Lambda}$ tiene impacto directo en la eficiencia, ya que cuanto mayor sea la diferencia entre los autovalores de la matriz $\mathbf{P}$, peor será la eficiencia del algoritmo.

2.3. Muestreador aleatorio de Gibbs

El muestreador de Gibbs (Gelfand y Smith, 1990) es uno de los métodos más populares dentro de los métodos Monte Carlo de cadenas de Markov (MCMC). El algoritmo permite obtener muestras de un vector aleatorio $\mathbf{X}$ de una distribución

aleatoria $\pi(\mathbf{X})$ a partir de las distribuciones condicionales univariantes $\pi(X_i|\mathbf{X}_{-i})$, siempre y cuándo sea posible obtener muestras aleatorias de forma sencilla a partir de éstas últimas.

En nuestro caso, las condicionales univariantes de la von Mises multivariante son distribuciones von Mises univariantes (Mardia et al., 2008; Razavian et al., 2011), para las cuales disponemos de un método rápido de muestreo. En concreto se tiene que para $\pi = MV(\boldsymbol{\mu}, \boldsymbol{\kappa}, \boldsymbol{\Lambda})$:

$$\pi(\theta_i|\boldsymbol{\theta}_{-i}) = VM(\mu_{(p)}, \kappa_{(p)})$$

donde:

$$\mu_{(p)} = \mu_p + \arctan(\kappa_p^{-1} \textstyle\sum_{j=1, j\neq p}^{p} [\lambda_{jp} \sin(\theta_j - \mu_j)]),$$
$$\kappa_{(p)} = \sqrt{\kappa_p^2 + \left(\textstyle\sum_{j=1, j\neq p}^{p} [\lambda_{jp} \sin(\theta_j - \mu_j)]\right)^2}$$

De entre todos los posibles esquemas de actualización para el muestreador de Gibbs, *mvCircular* implementa el escaneo aleatorio, el cual selecciona la siguiente componente a actualizar utilizando un vector de probabilidades $\boldsymbol{\alpha} = (\alpha_1, ..., \alpha_p)$ hasta que se alcanza la convergencia. Al igual que en el caso anterior este procedimiento ha sido implementado en ANSI C con su correspondiente función en R.

2.3.1. Convergencia: *burn-in*

Cuándo se utiliza el muestreador de Gibbs una de las cuestiones más importantes en su implementación es cómo determinar el número de iteraciones necesarias para converger (Raftery y Lewis, 1992), ya que, aunque teóricamente el método converge siempre para un número infinito de iteraciones, en la práctica se descarta un número de iteraciones predeterminado o se utiliza un criterio de convergencia, lo que se conoce como periodo de *burn-in*.

Uno de los criterios de convergencia más utilizados en la literatura es el *potential scale reduction factor (PSRF)* (Cowles y Carlin, 1996; Gelman y Rubin, 1992). Para calcular el PSRF es necesario generar m cadenas en paralelo con puntos de origen diferentes y dispersos en el espacio de variables, sobre las que computa la varianza entre cadenas (B/n) y la varianza intrínseca media de cada cadena (W). El ratio entre la varianza entre cadenas y la varianza intrínseca media se aproxima a 1 a medida que avanza la simulación y usualmente se determina convergencia cuando su valor es inferior a $1,1$ (Gelman y Shirley, 2011).

En *mvCircular* utilizaremos la versión multivariante del PSRF (MPSRF) como estimador de la convergencia del muestreador de Gibbs para la distribución von-Mises multivariante (Brooks y Gelman, 1998). El paquete *coda* en R, enfocado al análisis de procesos MCMC, implementa este y otros diagnósticos de convergencia para el análisis de simulaciones MCMC. El método genera muestras utilizando un número de cadenas distintas (superior o igual a tres), obtiene el valor del estimador MPSRF y continúa la simulación hasta que el valor del estimador se encuentra por debajo del umbral preestablecido, cuyo valor por defecto es $1,1$.

En las pruebas realizadas para verificar el funcionamiento del método, se detectó que, pese a que su implementación era correcta, el criterio basado en el MPSRF dictaminaba convergencia prematura en la mayoría de los casos, alcanzando un valor muy próximo a la unidad aún y cuando claramente las cadenas se encontraban lejos de converger. El uso de la varianza lineal con variables circulares y el espacio de variables acotado, puede explicar este comportamiento. Por ello, en la evaluación realizada en los siguientes apartados, y como comportamiento por defecto del muestreador, se descarta un elevado número de iteraciones (≈ 1000000) al inicio de la simulación como periodo de *burn-in*.

2.3.2. Autocorrelación: *thinning*

Otro problema asociado al uso del muestreador de Gibbs (y en general para cualquier muestreador MCMC), es la correlación entre los valores de la cadena a lo largo del tiempo o autocorrelación. En concreto, para una cadena $\mathbf{X^{(i)}}_{i=0}^{n}$ obtenida a partir de un muestreador aleatorio de Gibbs, la autocorrelación entre dos elementos depende exclusivamente de la distancia τ entre los elementos de la cadena (Levine y Casella, 2006), y viene dada por:

$$R(\tau) = \frac{E[(\mathbf{X}_t - \boldsymbol{\mu})(\mathbf{X}_{t+\tau} - \boldsymbol{\mu})]}{\sigma^2}$$

El mecanismo más común para obtener muestras lo más independientes posibles a partir de una cadena $\mathbf{X^{(i)}}_{i=0}^{n}$ con una autocorrelación significativa entre términos es utilizar una técnica de sub-muestreo denominada *thinning* que consiste en seleccionar uno de cada m ($m \geq 0$) elementos en la cadena, descartando el resto.

Es importante destacar que el propósito del *thinning* no es mejorar la precisión de las estimaciones del método (propósito estadístico), sino computacional. El uso del *thinning* no mejora la precisión del método pero reduce la cantidad de memoria necesaria para su almacenamiento y el tiempo necesario para su procesamiento (Link y Eaton, 2012).

En nuestro caso, al utilizar un muestrador aleatorio de Gibbs, dos elementos consecutivos de la cadena difieren únicamente en un elemento. Por ello, estableceremos un valor por defecto para el *thinning* el cual será proporcional al número de iteraciones necesarias para que dos muestras consecutivas (tras el *thinning*) no tengan ningún componente idéntico, en otras palabras, que todos los componentes del vector aleatorio hayan sido actualizados al menos una vez. Dado el vector de probabilidades $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, ..., \alpha_p)$ donde α_i es la probabilidad de actualizar el i-ésima componente del vector en cada iteración, la probabilidad de actualizar todos los elementos en n iteraciones viene dada por:

$$p(todos|n) = 1 - \sum_{i=1}^{n} (1 - \alpha_i)^n$$

Dado un valor umbral para dicha probabilidad, es sencillo establecer una cota inferior para el número de iteraciones n que es óptima si $\alpha_i = \frac{1}{p}$, $i = 1...p$. Si definimos una tolerancia β de forma que la probabilidad de actualizar todos los componentes sea superior a $1 - \beta$, una cota inferior para el número de iteraciones es:

$$n \geq \left\lceil \frac{\log(\beta/p)}{\log(1 - \text{mín}(\alpha)} \right\rceil$$

En *mvCircular* el usuario puede establecer tanto el parámetro β como el número de muestras a descartar entre iteraciones de forma directa, incluso desactivando la funcionalidad de *thinning* al establecer un valor igual a cero.

2.4. Comparación y selección del método

En el momento de generar muestras a partir de una distribución von-Mises multivariante dada, el procedimiento deberá seleccionar el muestreador más rápido en cada caso, siempre y cuando el usuario no especifique de forma explícita el método a utilizar. Con el fin de establecer unas reglas rápidas que permitan seleccionar con el mínimo coste posible un método u otro, se ha realizado un breve análisis preliminar de la eficiencia de cada uno de los métodos. En todos los casos utilizaremos los parámetros por defecto de *thinning* y *burn-in* para el muestreador aleatorio de Gibbs.

En primer lugar, estableceremos un umbral muy sencillo según número de variables de la distribución objetivo. Es de esperar que el método de rechazo sea más eficiente para un número reducido de variables, pero el decrecimiento exponencial de la tasa de aceptación debería hacer preferible el método de Gibbs a partir de cierto

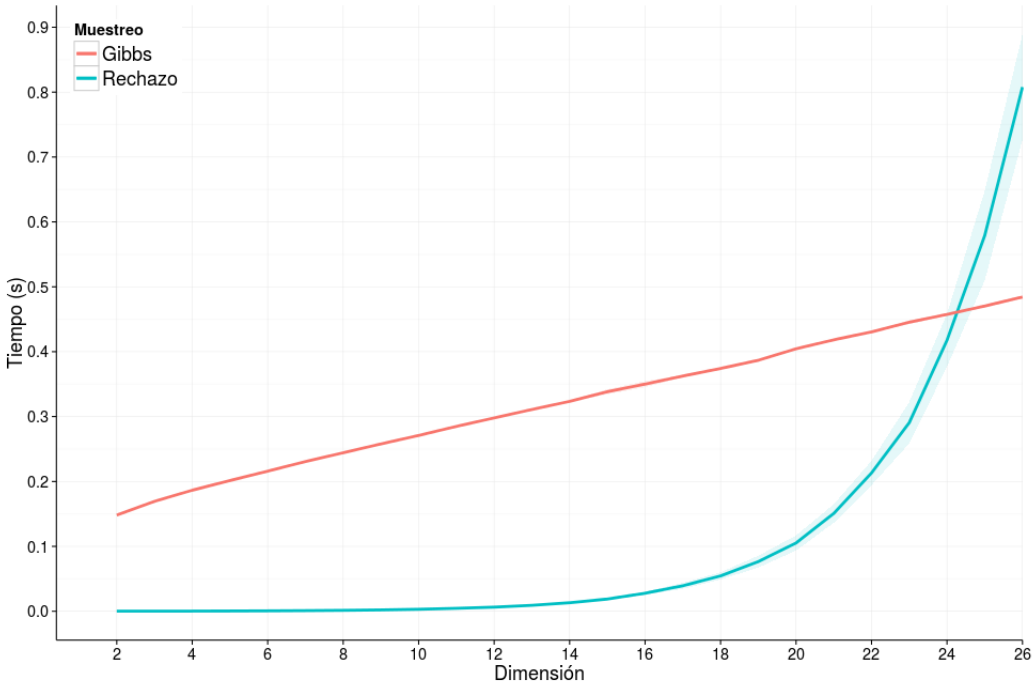


Figura 2.1: Comparación de métodos Tiempo de generación de 1000 muestras con 100 repeticiones

número de variables. Utilizando el paquete *microBenchmark* medimos el tiempo de generación de 1000 muestras de cada generador, repitiendo cada medición 100 veces, para un número de variables entre 2 y 26 , en condiciones óptimas para el método de rechazo, es decir, con todos los autovalores idénticos. Como se observa en la figura 2.1, aún en condiciones óptimas, a partir de 25 variables el muestreador de Gibbs es superior al método de rechazo.

En segundo lugar, una vez superado el primer umbral, calcularemos el valor de la cota inferior dada por Mardia para concentraciones elevadas (Mardia y Voss, 2014), esto es:

$$\frac{1}{2^p}\sqrt{\frac{\lambda_{min}^p}{|\boldsymbol{P}|}}$$

Seleccionaremos el método de rechazo si el valor de esta cota se encuentra por debajo de 10^{-5}, mientras que se utilizará el método de Gibbs en caso contrario.

En resumen, el selector del muestreador es:

1. Si el número de variables es superior a 25, se utiliza el método de Gibbs
2. En caso contrario, se calculan los autovalores de la matriz $\boldsymbol{P}$

3. Si algún autovalor no es positivo o el valor de la cota es inferior a 10^{-5} se utiliza el método de Gibbs

4. En caso contrario, se utiliza el método de rechazo

3

Aprendizaje de parámetros

3.1. Verosimilitud

El aprendizaje de parámetros basado en la maximización de la verosimilitud de la muestra, como su propio nombre indica, busca el conjunto de parámetros tales que la probabilidad de observar dicha muestra de la distribución paramétrica objetivo sea lo mayor posible. Como recordatorio, la función de densidad de la distribución von Mises multivariante viene dada por la siguiente formulación (Mardia et al., 2008) :

$$f_{MVM}(\boldsymbol{\Theta}; \boldsymbol{\mu}, \boldsymbol{\kappa}, \boldsymbol{\Lambda}) = \frac{1}{Z(\boldsymbol{\kappa}, \boldsymbol{\Lambda})} \exp\left\{\kappa^T c(\boldsymbol{\Theta}) + \frac{1}{2} s(\boldsymbol{\Theta}) \boldsymbol{\Lambda} s(\boldsymbol{\Theta})\right\}$$

donde $c_i(\boldsymbol{\Theta}) = \cos(\Theta_i - \mu_i)$ and $s_i(\boldsymbol{\Theta}) = \sin(\Theta_i - \mu_i)$ y $Z(\boldsymbol{\kappa}, \boldsymbol{\Lambda})$ es una constante de normalización sin forma cerrada conocida para más de dos variables. A partir de la definición, es sencillo ver que la verosimilitud de una muestra $\{\boldsymbol{\Theta}^{(i)}\}_{i=1}^N$ se calcula como:

$$\mathcal{L}(\boldsymbol{\Theta}|\boldsymbol{\mu}, \boldsymbol{\kappa}, \boldsymbol{\Lambda}) = \prod_{i=1}^N f_{MVM}(\boldsymbol{\Theta^{(i)}}; \boldsymbol{\mu}, \boldsymbol{\kappa}, \boldsymbol{\Lambda})$$

Por desgracia, el valor del término de normalización Z debe ser calculado mediante integración numérica en p dimensiones, lo que hace extremadamente costoso, desde un punto de vista computacional, el uso de la función de verosimilitud para el aprendizaje de los parámetros de la distribución.

3.2. Pseudo-verosimilitud

Como se menciona en el apartado anterior, debido a la complejidad del cálculo de término de normalización en la función de densidad de la distribución von Mises

multivariante, no es aconsejable utilizar la verosimilitud como función objetivo para ajustar los parámetros de la distribución. En su lugar, se propone utilizar una aproximación consistente de la verosimilitud denominada pseudo-verosimilitud (Mardia et al., 2007, 2008) definida:

$$P\mathcal{L}(\boldsymbol{\Theta}|\boldsymbol{\mu},\boldsymbol{\kappa},\boldsymbol{\Lambda}) = (2\pi)^{-}pN \prod_{i=1}^{N}\prod_{j=1}^{p} P_{MVM}(\boldsymbol{\Theta}_{\boldsymbol{j}}^{(\boldsymbol{i})}|\boldsymbol{\Theta}_{-\boldsymbol{j}}^{(\boldsymbol{i})})$$

Ya que las distribuciones marginales condicionadas de la distribución von Mises multivariante son distribuciones von Mises univariantes, este término es conocido y puede ser calculado sin necesidad de recurrir a métodos numéricos:

$$P\mathcal{L}(\boldsymbol{\Theta}|\boldsymbol{\mu},\boldsymbol{\kappa},\boldsymbol{\Lambda}) = (2\pi)^{-pN} \prod_{i=1}^{N}\prod_{j=1}^{p} \frac{1}{I_0(\kappa_{(j)}^{(i)})} \exp\left\{\kappa_{(j)}^{(i)} \cos(\theta_{i,j} - \mu_{(p)}^{(i)})\right\}$$

donde

$$\mu_{(p)}^{(i)} = \mu_p + \arctan\left(\kappa_p^{-1} \sum_{j=1, j\neq p}^{p} \lambda_{jp} \sin\left(\theta_j^{(i)} - \mu_j\right)\right),$$

$$\kappa_{(j)}^{(i)} = \sqrt{\kappa_p^2 + \left(\sum_{j=1, j\neq p}^{p} \lambda_{jp} \sin\left(\theta_j^{(i)} - \mu_j\right)\right)^2}$$

3.2.1. Optimización

Para calcular los parámetros de una distribución von Mises multivariante que maximice la pseudo-verosimilitud descrita anteriormente, redefiniremos el problema como un problema de minimización, donde la función de pérdida (L) es menos el logaritmo natural de la función de pseudo-verosimilitud:

$$L(\boldsymbol{\Theta}|\boldsymbol{\mu},\boldsymbol{\kappa},\boldsymbol{\Lambda}) = (np)\log(2\pi) + \textstyle\sum_{i=1}^{N}\sum_{j=1}^{p}\left\{\log(I_0(\kappa_{(j)}^{(i)})) - \kappa_{(j)}^{(i)}\cos(\theta_{i,j} - \mu_{(p)}^{(i)})\right\}$$

Si bien la función de pérdida puede calcularse de forma exacta en un tiempo relativamente pequeño, ésta todavía puede ser optimizada para reducir su complejidad computacional, limitando el número de operaciones trigonométricas y expresando las sumas como productos de matrices. Es importante destacar que, aunque el tiempo cálculo del valor de la función pueda ser pequeño, los algoritmos de optimización requieren evaluar la función de pérdida un gran número de veces, por lo que cualquier reducción en el tiempo de cálculo repercutirá notablemente en el tiempo final.

En primer lugar, ya que la distribución es rotacionalmente equivariante y el parámetro $\boldsymbol{\mu}$ óptimo se calcula como la media circular componente a componente, podemos asumir, sin pérdida de generalidad, que $\mu_i = 0$, $i = 1...p$. Para reducir el número de operaciones trigonométricas, y en concreto para eliminar el cómputo de la inversa de la tangente, haremos uso de las siguientes igualdades trigonométricas:

$$\sin(\arctan(x)) = \frac{x}{\sqrt{1+x^2}}$$

$$\cos(\arctan(x)) = \frac{1}{\sqrt{1+x^2}}$$

para reducir el término $\kappa_{(j)}^{(i)} \cos(\theta_{i,j} - \mu_{(p)}^{(i)})$:

$$\kappa_{(j)}^{(i)} \cos(\theta_{i,j} - \mu_{(p)}^{(i)}) =$$

$$\kappa_{(j)}^{(i)} (\cos(\theta_{i,j}) \cos(\mu_{(p)}^{(i)}) + \sin(\theta_{i,j}) \sin(\mu_{(p)}^{(i)})) =$$

$$\kappa_{(j)}^{(i)} (\cos(\theta_{i,j}) \frac{1}{\sqrt{1 + \kappa_p^{-1} \sum_{j=1, j\neq p}^{p} \lambda_{jp} \sin(\theta_j^{(i)})}} + \sin(\theta_{i,j}) \frac{\kappa_p^{-1} \sum_{j=1, j\neq p}^{p} \lambda_{jp} \sin(\theta_j^{(i)})}{\sqrt{1 + \kappa_p^{-1} \sum_{j=1, j\neq p}^{p} \lambda_{jp} \sin(\theta_j^{(i)})}} =$$

$$\cos(\theta_{i,j}) \kappa_j + \sin(\theta_{i,j}) \sum_{j=1, j\neq p}^{p} \lambda_{jp} \sin(\theta_j^{(i)})$$

A su vez, es posible expresar los sumatorios de la expresión como elementos de una matriz $\boldsymbol{A}$, cuyo cálculo puede expresarse como el producto de dos matrices. Si denotamos $\boldsymbol{S}$ a la matriz $N \times p$ tal que $\boldsymbol{S} = (s_{i,j}) = \sin(\theta_{i,j})$, con los vectores en filas, entonces la matriz $\boldsymbol{A}$ donde $a_{i,j} = \sum_{j=1, j\neq p}^{p} \left[\lambda_{jp} \sin \theta_j^{(i)}\right]$ puede calcularse como:

$$\boldsymbol{A} = \boldsymbol{S}\boldsymbol{\Lambda}$$

Además, si eliminamos el término constante $(np)\log(2\pi)$, la expresión de la función de pérdida queda reducida a:

$$L_c(\boldsymbol{\Theta}|\boldsymbol{\mu}, \boldsymbol{\kappa}, \boldsymbol{\Lambda}) = \sum_{i=1}^{N} \sum_{j=1}^{p} \{\log(I_0(\kappa_j^{(i)}) - \cos(\theta_{i,j})\kappa_j - sin(\theta_{i,j})a_{i,j}\}$$

3.3. Aprendizaje regularizado

El aprendizaje regularizado es una técnica que habitualmente se utiliza en problemas donde la obtención de datos es costosa y/o compleja y por lo tanto es necesario lidiar con un conjunto muy reducido de muestras. El aprendizaje regularizado de los parámetros de la distribución von Mises multivariante ya ha sido llevado a cabo en la literatura (Razavian et al., 2011) mediante la aplicación de una penalización L_1 uniforme a todos los elementos de la matriz $\mathbf{\Lambda}$. No obstante, aplicar una penalización uniforme a todos los componentes de la matriz, tal y como se hace en la regularización L_1 habitual, supone añadir, desde una perspectiva Bayesiana, la creencia de que todos los elementos $\lambda_{i,j}$ son de magnitud similar, lo que no puede corresponderse con el conocimiento *a priori* del problema (Tan et al., 2014).

Para incluir éste conocimiento previo del problema en la regularización, en caso de que lo hubiera, hemos propuesto e implementado una versión generalizada de la penalización L_1 para la distribución von Mises multivariante donde cada $\lambda_{i,j}$ se penaliza de forma independiente. Con este fin, definimos una matriz de penalización simétrica $\boldsymbol{P}$, del mismo tamaño que $\mathbf{\Lambda}$, con todos los elementos $p_{i,j}$ positivos o iguales a cero.

Incluyendo éste término de penalización en la función de pérdida definida en el apartado anterior, el problema del aprendizaje de parámetros consiste ahora en minimizar una nueva función de pérdida:

$$f(\mathbf{\Theta}|\boldsymbol{\mu}, \boldsymbol{\kappa}, \mathbf{\Lambda}, \mathbf{P}) = L_c(\mathbf{\Theta}|\boldsymbol{\mu}, \boldsymbol{\kappa}, \mathbf{\Lambda}) + \sum_{i=1}^{p-1} \sum_{j=i+1}^{p} |\lambda_{i,j}| p_{i,j}$$

3.3.1. Minimización

Para encontrar el conjunto de parámetros que minimizan la función de pérdida utilizaremos el método de optimización quasi-Newton L-BFGS-B (Zhu et al., 1997) en su última versión (Morales y Nocedal, 2011), una extensión del método L-BFGS que permite la inclusión de restricciones simples en el espacio de parámetros, tales como $\kappa_i > 0$ en nuestro caso. El método está especialmente diseñado para funciones con un gran número de parámetros.

En cada iteración el algoritmo trata de encontrar una aproximación de la inversa matriz Hessiana, para lo cual es necesario proveerle tanto de la función a minimizar, como de su gradiente. Para ello, calcularemos las derivadas de la función de pérdida con respecto a $\lambda_{i,j}$ y κ_j:

$$\frac{\partial L_c}{\partial \kappa_j} = \sum_{i=1}^{N} \left[A_0(\kappa_j^i) \frac{\kappa_j}{\kappa_j^{(i)}} - \cos(\theta_{i,j}) \right]$$

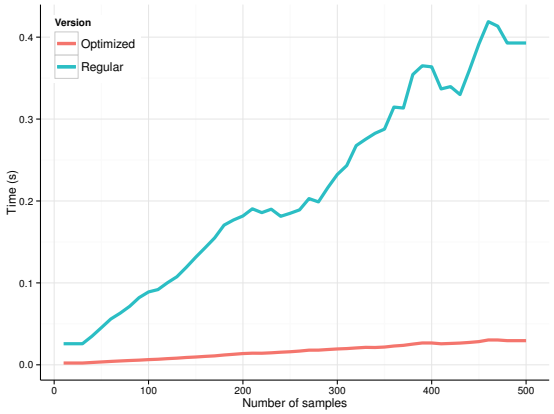


Figura 3.1: Tiempo de ajuste. Fuente: Rodriguez-Lujan et al. (2015)

$$\frac{\partial L_c}{\partial \lambda_{j,k}} = \sum_{i=1}^{N} \left[\sin(\theta_{i,k}) \left(\frac{A_0(\kappa_j^{(i)})}{\kappa_j^{(i)}} \rho_{j,k} - \sin \theta_{i,j} \right) \right] + sgn(\lambda_{j,k}) p_{j,k}$$

donde $A_0 = \frac{I_1}{I_0}$ y sgn es la función signo que evalúa $sgn(0) = 0$

Es importante destacar que, pese a que el valor absoluto no es una función diferenciable en 0, es posible definir una función diferenciable que aproxime al valor absoluto con precisión arbitraria. Considerando que el número real de valores representables en un ordenador es finito y que la distancia mínima entre dos números consecutivos viene dada por la capacidad de representación de la máquina, podemos tratar la función valor absoluto como si fuese diferenciable en todo punto.

Tanto la implementación de la función de pérdida, como el proceso de minimización han sido implementados en ANSI C con sus correspondientes llamadas en R. Para ello, se ha utilizado una versión en C del código de L-BFGS-B denominada L-BFGS-B-C (Becker, 2014) incluida junto con el paquete. De igual forma, se implementa la versión no optimizada de la función de pérdida (Razavian et al., 2011) con fines comparativos.

Se realizó un análisis comparativo de los tiempos de ejecución en el ajuste de los parámetros entre la versión optimizada de la función de penalización y la versión propuesta en la literatura, utilizando en ambos casos el algoritmo L-BFGS-B con los mismos parámetros. Para ello se midió el tiempo de ejecución mediante la librería *microBenchmark* en el ajuste de una distribución 5-dimensional con un número de muestras entre 10 y 500. Los resultados en la figura 3.1 muestran que la versión optimizada es notablemente más rápida en todos los casos, incrementando la diferencia a medida que aumenta el número de muestras.

4

Evaluación

Es un hecho conocido que para valores altos del parámetro de concentración $\boldsymbol{\kappa}$, la distribución von Mises se aproxima a una distribución Normal en la circunferencia. Este mismo fenómeno se extiende al caso multivariable, aunque no está claro cómo afectan a este comportamiento el número de variables o el valor de la matriz $\boldsymbol{\Lambda}$ (Razavian et al., 2011).

En esta sección realizaremos una comparación exhaustiva del comportamiento del ajuste a partir de datos generados de una distribución von Mises conocida entre la propia distribución von Mises y la normal multivariante. Estudiaremos el comportamiento de las distribuciones a medida que varían los parámetros de la distribución original y la dimensionalidad, tratando de ver como las diferentes configuraciones paramétricas afectan al número de muestras necesarias para la convergencia y a la *normalidad* de la distribución von Mises.

Además, incluiremos en las comparaciones la versión regularizada de la distribución von Mises, con el fin de comprobar su comportamiento cuando el número de muestras es reducido y su evolución a medida que la cantidad de muestras disponibles aumenta.

4.1. Divergencia Kullback-Leibler circular

La presencia del término de normalización en la expresión de la función de densidad de la distribución von Mises multivariante restringe el uso de las medidas más habituales de divergencia entre distribuciones de probabilidad como la divergencia de Kullback-Leibler (KL), impidiendo además el uso de test de bondad de ajuste sofisticados. En la literatura se ha utilizado la comparación del valor de verosimilitud como medida de la bondad de ajuste y comparación entre distribuciones (Mardia

et al., 2008); o el valor del coseno del ángulo entre los parámetros (Razavian et al., 2011), pero éstas medidas no son adecuadas cuando se comparan distribuciones con parametrizaciones distintas y el valor de la verosimilitud, además de ser costoso de calcular, no es aplicable en todos los casos.

Para superar estos inconvenientes, utilizaremos una aproximación de la divergencia KL para datos multivariable basada en la distancia al k-ésimo vecino más cercano entre dos muestras dadas (Pérez-Cruz, 2008). En esencia, la aproximación de la divergencia KL entre dos distribuciones d-dimensionales P y Q dada una muestra de cada distribución, $\{\mathbf{X}\}_1^n$ y $\{\mathbf{Y}\}_1^m$ respectivamente, se define:

$$\hat{D}_k(P||Q) = \frac{d}{n}\sum_{i=1}^{n}\{\log\left(\frac{r_k(\mathbf{x_i})}{s_k(\mathbf{x_i})}\right)\} + \log\frac{m}{n-1}$$

donde $r_k(\mathbf{x_i})$ y $s_k(\mathbf{x_i})$ son, respectivamente la distancia al k-ésimo vecino más cercano de $\mathbf{x_i}$ en $\mathbf{X}$ e $\mathbf{Y}$.

En lugar de la distancia euclídea para calcular la distancia al k-ésimo vecino más cercano en cada uno de los conjuntos necesitamos una distancia que tenga en cuenta la naturaleza circular de los datos. Haremos uso de la métrica dada por la función determinada por el algoritmo 1 para medir la distancia circular entre dos puntos a, b en el dominio $[0, 2\pi)^p$

Algorithm 1 MV Circular distance

```
procedure MvCircularDist
    for i = 1..p do
        if |a_i − b_i| <= π then
            c_i ← b_i
        else
            if a_i > π then
                c_i ← b_i + 2π
            else
                c_i ← b_i − 2π
    return ||a − c||
```

Es importante destacar que la función $f : [0, 2\pi)^p \times [0, 2\pi)^p \to \mathbb{R}$ definida por el algoritmo 1 es una distancia en el dominio $[0, 2\pi)^p$ para cualquier valor de p.

4.2. Experimentos

En los experimentos de esta sección generaremos muestras a partir de una distribución von Mises multivariante con diferentes configuraciones de parámetros; en

concreto se variará: (a) El número de variables p entre 5 y 100; (b) el número de muestras n de 5 a 1.000 dependiendo de p; (c) El valor del vector de concentraciones $\boldsymbol{\kappa}$; y (d) la matriz $\boldsymbol{\Lambda}$ para diferentes configuraciones. En todos los casos el vector de medias $\boldsymbol{\mu}$ quedará fijado a π en todas las componentes.

El siguiente procedimiento se repetirá para cada combinación de parámetros r veces y se calculara la media y desviación típica de la divergencia KL calculada en las r repeticiones:

1. Se generarán n muestras p-dimensionales de una distribución von Mises multivariante con parámetros $\boldsymbol{\mu}_0$, $\boldsymbol{\kappa}_0$ and $\boldsymbol{\Lambda}_0$

2. Se ajustarán los parámetros de las distribuciones von Mises y Normal, utilizando para ello las n muestras generadas en el paso anterior

3. Se generarán 1.000 muestras p-dimensionales tanto de la distribución original como de las distribuciones ajustadas y se calculará la estimación de la divergencia KL entre la distribución original y las ajustadas utilizando estas muestras

Además, para cada configuración experimental, se calculará el umbral de convergencia (es decir, el valor de la divergencia a partir del cual podemos decir que las distribuciones son semejantes) de la distribución objetivo mediante un método de *bootstrapping*:

1. Se generan m muestras de la distribución objetivo como conjunto de test con la configuración paramétrica dada

2. Se generan otras m muestras de la misma distribución objetivo y se calcula la divergencia con el conjunto de test. Este proceso de repite 100 veces

3. Se calcula el percentil 95 de las divergencias calculadas. Este valor se toma como umbral de convergencia.

El objetivo de dichos experimentos es el estudio del número de muestras necesarias para la convergencia de cada una de las distribuciones así como el impacto de los distintos parámetros al comportamiento de cada una de ellas. En concreto, se recogen dos conjuntos de configuraciones experimentales:

1. **Número de muestras variable**: Para diferentes configuraciones de los parámetros $\boldsymbol{\Lambda}$ y $\boldsymbol{\kappa}$, y dimensión variable, se comprueba la evolución del valor del estimador de la divergencia KL a medida que varía el número de muestras disponibles para el ajuste

2. **Concentración variable**: Para un número fijo de muestras y una matriz $\mathbf{\Lambda}$ predeterminada, se varía el valor del vector de concentraciones $\boldsymbol{\kappa}$

En todos los casos, la matriz simétrica $\mathbf{\Lambda}$ se generará de forma aleatoria, donde cada elemento $\lambda_{i,j}$ tomará un valor distinto de cero con probabilidad ρ entre 1 y -1. Igualmente, la matriz de penalización $\boldsymbol{P}$ vendrá dada por $\boldsymbol{P} = \mathbf{\Lambda} + \boldsymbol{R}$, donde $\boldsymbol{R}$ es una matriz simétrica de ruido Gaussiano de parámetro $\sigma = 0,25$ con $r_{i,j} \sim N(0, \sigma)$. Generaremos matrices $\mathbf{\Lambda}$ aleatorias, con tres configuraciones:

- **Matriz vacía**: La probabilidad ρ de que un elemento sea no nulo es 0
- **Matriz dispersa**: La probabilidad ρ de que un elemento sea no nulo es 0.3
- **Matriz densa**: La probabilidad ρ de que un elemento sea no nulo es 0.7

De igual manera, contemplaremos tres posibles valores para el vector de concentraciones κ:

- **Concentración baja**: $\kappa_i = 0,1\ i = 1...p$
- **Concentración media**: $\kappa_i = 2\ i = 1...p$
- **Concentración alta**: $\kappa_i = 7\ i = 1...p$

En los experimentos con un número variable de muestras diferenciaremos entre los experimentos para dimensionalidad baja, media y alta. Para valores de p bajos (menor a 10) se evaluarán todas las configuraciones posibles de los parámetros $\boldsymbol{\kappa}$ y $\mathbf{\Lambda}$ (9 en total), con el número de muestras variando entre 5 y 500 y repitiendo 10 veces cada ejecución, mientras que para valores de p medios y altos (entre 20 y 100) se probarán igualmente todas las configuraciones posibles, con el número de muestras variando de 10 a 10000 y 5 repeticiones de cada ejecución. Las configuraciones experimentales se resumen en la tabla 4.1.

Dimensión	Muestras	Repeticiones	Concentración	$\mathbf{\Lambda}$
5,10	5 - 500	10	Baja, media y alta	Vacía, dispersa y densa
20,50,100	10 - 10000	5	Baja, media y alta	Vacía, dispersa y densa

Tabla 4.1: Configuraciones para número de muestras variables

El mismo esquema se repite en la pruebas con concentración variable. Para este grupo de experimentos el valor de la concentración de todas las componentes variará entre 0.01 y 5; se repetirá cada experimento para matrices $\mathbf{\Lambda}$ vacías, dispersas y

densas; y el número de muestras será de 1000 para los casos con 5,10 y 20 variables, mientras que para 50 y 100 variables se tomarán 10000 muestras. De igual manera en la tabla 4.2 se recogen las configuraciones de los experimentos.

Dimensión	Muestras	Repeticiones	Concentración	$\mathbf{\Lambda}$
5,10,20	1000	10	0,01 - 5	Vacía, dispersa y densa
50,100	10000	5	0,01 - 5	Vacía, dispersa y densa

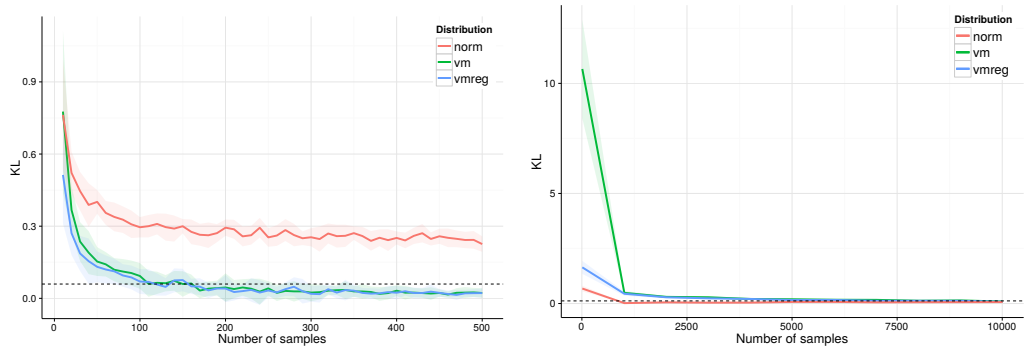
Tabla 4.2: Configuraciones para concentración variable

4.3. Resultados

En está sección expondremos los resultados obtenidos de la ejecución de los experimentos planteados en la sección anterior junto con las conclusiones derivadas de dichos resultados. Debido al elevado número de ejecuciones y gráficas producto de los experimentos, nos centraremos únicamente en aquellas que ayuden a la justificación de las conclusiones obtenidas (Rodriguez-Lujan et al., 2015).

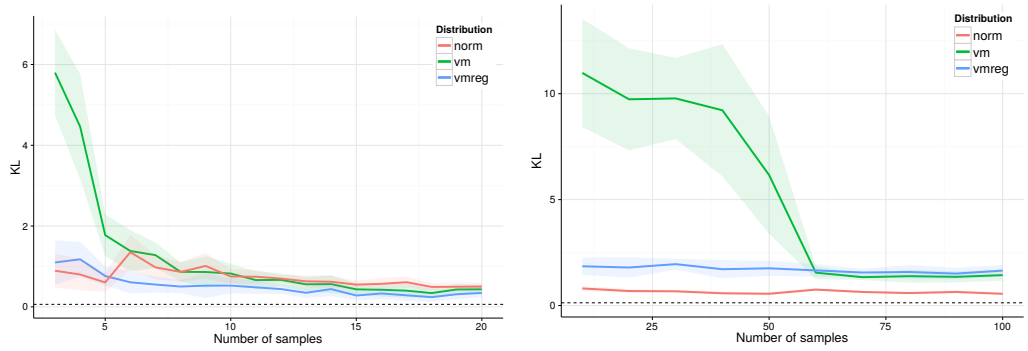
En primer lugar, en la figura 4.1 observamos como para valores de concentración muy bajos y matriz $\mathbf{\Lambda}$ vacía, ambas distribuciones von Mises obtienen mejores resultados que la normal para un número de variables bajo ($p = 5$), mientras que para un número de variables medio-alto ($p = 100$) el rendimiento de las von Mises cae drásticamente, probablemente como resultado del uso de la pseudo-verosimilitud en el ajuste. Como se esperaba, la versión regularizada obtiene mejores resultados cuando el número de muestras es reducido, sin empeorar su rendimiento respecto a la versión no regularizada a medida que el número de muestras aumenta. En la figura 4.2, puede observarse como este comportamiento se hace notable cuanto menor es el número de muestras respecto al número de variables en ambos casos.

El caso opuesto al anterior, con valores de concentración altos y matriz $\mathbf{\Lambda}$ densa se muestra en la figura 4.3 para el mismo número de variables. Los resultados obtenidos concuerdan con los resultados teóricos esperados, la distribución von Mises con alta concentración se asemeja a la distribución normal, teniendo para un número de variables bajo ($p = 5$) un valor de divergencia similar. Al igual que en el caso anterior observamos como para un número reducido de muestras la versión regularizada obtiene resultados significativamente mejores, por lo que vemos que este comportamiento es consistente, independientemente de la concentración y la matriz $\mathbf{\Lambda}$. De igual manera, parece que el rendimiento del ajuste de la distribución von Mises decrece a medida que aumenta el número de variables de la distribución.



(a) $p = 5$ **(b)** $p = 100$

Figura 4.1: Divergencia KL para concentración baja y matriz lambda nula

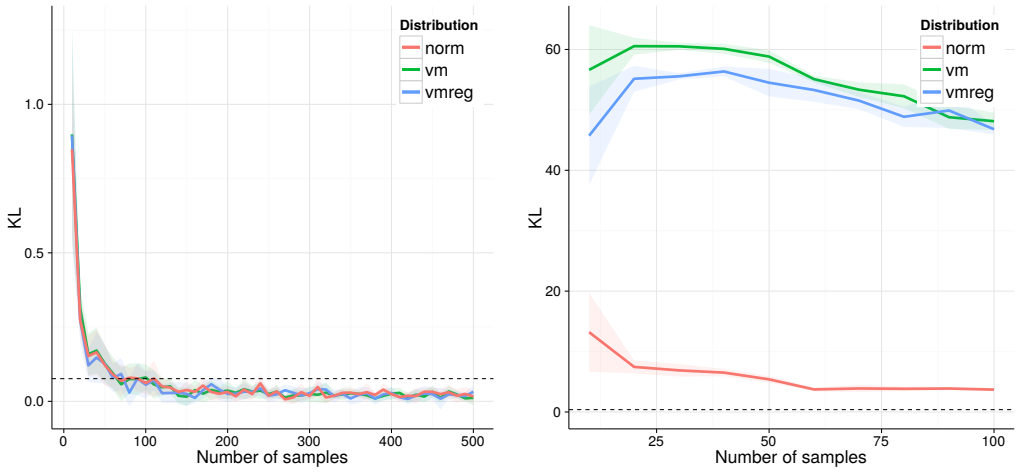


(a) $p = 5$ **(b)** $p = 100$

Figura 4.2: Divergencia KL para concentración baja y matriz $\mathbf{\Lambda}$ nula para número de muestras muy reducido

Los resultados de los experimentos de variación del valor de la concentración se muestran, para $\mathbf{\Lambda}$ dispersa en la figura 4.4. A partir de las gráficas podemos sacar dos conclusiones: a medida que aumenta el número de variables, el valor del parámetro de concentración para el que la distribución von Mises se asemeja a una distribución normal aumenta, siendo este valor cercano a 1,5 para $p = 5$, mientras que para $p = 100$ se encuentra cerca de 4; y que el número de muestras necesario para la convergencia decrece a medida que aumenta la concentración, como se observa claramente para $p = 100$ donde la divergencia decrece a medida que aumenta la concentración pero el umbral de rechazo se mantiene constante.

En la figura 4.5 podemos ver como, para distintos valores de la matriz $\mathbf{\Lambda}$ el comportamiento difiere notablemente. En concreto para mismo número de variables $p = 10$ vemos como el aumento del número de elementos distintos de cero en la matriz parece tener el efecto contrario al aumento de la concentración. Este efecto



(a) $p = 5$ **(b)** $p = 100$

Figura 4.3: Divergencia KL para concentración alta y matriz $\mathbf{\Lambda}$ densa

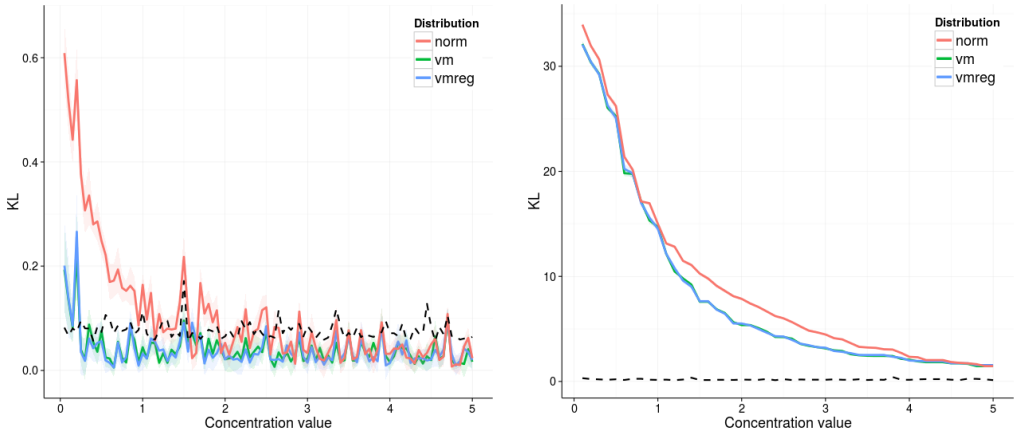
se hace especialmente notable en la figura 4.6, donde observamos como para una matriz $\mathbf{\Lambda}$ nula y $p = 5$ la distribución normal iguala o supera a la von Mises a partir de una concentración extraordinariamente baja ($\approx 0,2$)

4.4. Aplicación a datos morfológicos de neuronas humanas

Con el fin de evaluar la mejora que supone la aplicación de la regularización al aprendizaje de la distribución von Mises multivariante sobre un conjunto de datos reales, compararemos los ajustes obtenidos por la versión no regularizada y la regularizada al modelar un conjunto de variables angulares obtenidas a partir de datos morfológicos de neuronas humanas.

Como se expone en la introducción, las dendritas basales de neuronas pirámidales son un tipo concreto de neuritas que crece desde la base del cuerpo neuronal de forma más o menos plana. En este caso, mediremos los ángulos entre cada par de dendritas basales a partir de los datos morfológicos de la neuronas e intentaremos ajustar nuestras distribuciones circulares a dichos datos.

Para ello, hemos descargado un conjunto de reconstrucciones tridimensionales de neuronas humanas (Jacobs et al., 2001) de Neuromorpho.org (Ascoli et al., 2007), un repositorio público de reconstrucciones neuronales que contiene datos de más de 30000 neuronas de distintas especies animales. Junto con las reconstrucciones, hemos obtenido metadatos asociados a cada reconstrucción que contienen, entre otros



(a) $p = 5, n = 1000$ **(b)** $p = 100, n = 10000$

Figura 4.4: Resultados para concentración entre 0.1 y 5 con matriz $\mathbf{\Lambda}$ dispersa

valores: la técnica de reconstrucción utilizada, la edad del individuo, su género y la región del cerebro a la que pertenece la muestra. De entre todas las reconstrucciones pertenecientes al mismo estudio (Jacobs et al., 2001) hemos seleccionado únicamente aquellas que pertenecen a los lóbulos occipital y frontal.

El procedimiento para la extracción y el preprocesado de las muestras se explicará con más detalles en la segunda parte de este trabajo. En resumen, las muestras originales, en texto plano, se procesan y se miden los ángulos entre las dendritas basales. Ante la ausencia de una referencia externa que nos pueda indicar la orientación global de la neurona, se tomará la dendrita de mayor tamaño como la principal y se medirán los ángulos en el orden contrario a las agujas del reloj como se expone en la figura 4.7. El último ángulo, a estar completamente determinado por el resto, se omite en las mediciones.

Para construir la matriz de penalización utilizaremos la distancia entre las dendritas como conocimiento *a priori* del problema. Si el término $\lambda_{i,j}$ de la matriz expresa la relación entre en ángulo formado por la i-ésma dendrita y su siguiente y el ángulo entre la dendrita j-ésima y su siguiente es de esperar que esta correlación disminuya a medida que aumenta la distancia entre dichos ángulos, dicho de otra forma, esperamos que la correlación sea

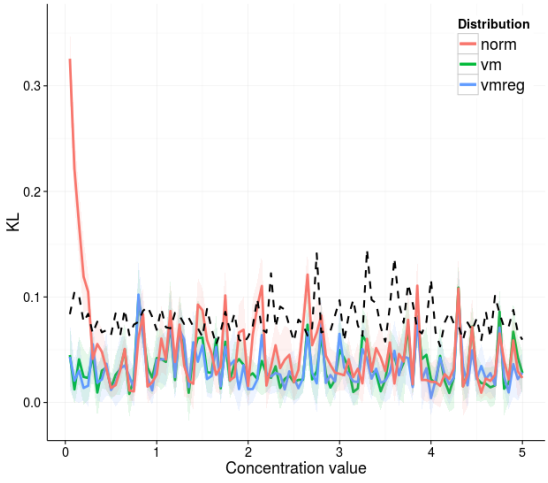
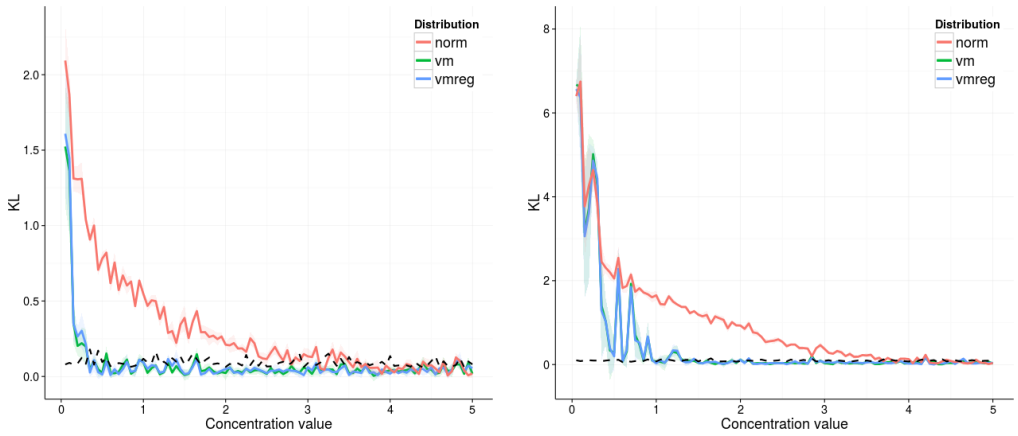


Figura 4.6: $p = 5, n = 1000$, $\mathbf{\Lambda}$ nula



(a) $p = 10, n = 1000$, Λ dispersa **(b)** $p = 100, n = 1000$, Λ densa

Figura 4.5: Resultados para concentración entre 0.1 y 5 con matriz $\mathbf{\Lambda}$ dispersa

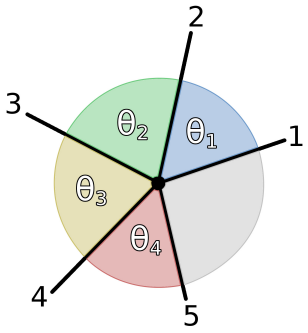


Figura 4.7: Ángulos entre dendritas. Fuente: Rodriguez-Lujan et al. (2015)

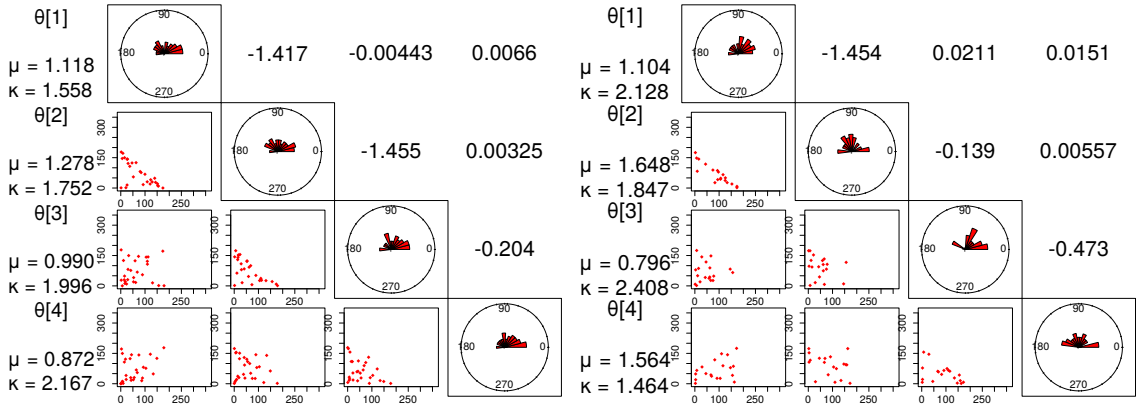
mayor cuanto más cercanos sean los ángulos. Por ello, utilizaremos la distancia entre los ángulos como valor para la matriz de penalización $\boldsymbol{P} = (p_{i,j})$. Es interesante ver como, tomando como referencia la figura 4.7, el valor de $p_{1,4}$ es 1 en lugar de 2.

Para la obtención de los resultados hemos dividido los datos en conjuntos de entrenamiento y test, ajustando las distribuciones con la primera y calculando la divergencia KL con respecto a la segunda, repitiendo todo el proceso en 100 ocasiones (*repeated train and test*). Como se puede observar en la tabla 4.3 la distribución regularizada obtiene mejores resultados en todos los casos, especialmente en aquellos donde el número de muestras es menor.

De forma adicional se ha implementado en el paquete *mvCircular* una función que crea una gráfica resumen de la función von Mises ajustada, incluyendo los datos a partir de los que se realizó dicho ajuste. En la diagonal incluye diagramas de rosas que muestran las distribuciones marginales en los datos originales; la parte triangular superior muestra el valor de los parámetros $\lambda_{i,j}$ para cada par de variables mientras

Tabla 4.3: Divergencia KL para ángulos entre dendritas

Dendritas	Edad	Género	Región	Muestras	vM Regularizada	vM
5	Adulto	Hombre	Occipital	19	**0,63**	0,68
5	Adulto	Mujer	Occipital	28	**0,49**	0,57
5	Adulto	Ambos	Occipital	47	**0,35**	0,37
5	Adulto	Hombre	Frontal	21	**0,55**	0,71
5	Adulto	Mujer	Frontal	21	**0,53**	0,65
5	Adulto	Ambos	Frontal	42	**0,35**	0,40
6	Adulto	Hombre	Frontal	16	**0,87**	1,19
6	Adulto	Mujer	Frontal	12	**1,04**	1,43
6	Adulto	Ambos	Frontal	28	**0,50**	0,64



(a) Mujer **(b)** Hombre

Figura 4.8: Ángulos entre dendritas de neuronas del lóbulo occipital de adultos. Rodriguez-Lujan et al. (2015)

que la parte inferior expone las muestras originales dos a dos; por último, en la primera columna se muestran los parámetros correspondientes a la media $\boldsymbol{\mu}$ y la concentración. $\boldsymbol{\kappa}$. Un ejemplo de dicha representación puede verse en la figura 4.8.

Parte II

Arborizaciones dendríticas: Caracterización y simulación

5

Obtención, preprocesamiento y caracterización de la arborización

En esta segunda parte del trabajo nos centraremos en la modelización y simulación del conjunto de dendritas basales de neuronas piramidales, también llamado arborización dendrítica basal. Analizaremos los modelos existentes en la literatura y discutiremos en que se diferencia nuestra propuesta de las anteriores. Abordaremos en este primer capítulo la caracterización de la arborización dendrítica basal de una neurona piramidal.

En líneas generales, seguiremos el esquema típico de desarrollo, donde tras este análisis inicial, pasaremos a la obtención de datos a partir de reconstrucciones reales, elaboraremos un modelo que explique el crecimiento para finalmente intentar ser capaces de crear neuronas sintéticas similares a las reales.

5.1. Introducción

El desarrollo del sistema nervioso es un proceso dinámico de gran complejidad, las neuronas se diferencian unas de otras, establecen y destruyen conexiones sinápticas unas con otras a lo largo del tiempo, etc. El interés en el patrón de arborización dendrítico radica en el hecho de que dicho patrón determina en gran medida la forma en la que se combinan las señales procedentes de cada una de las entradas (sinapsis), localizadas en estas mismas dendritas, hasta generar una salida a través del axón. Por tanto, para comprender la forma en la que las neuronas procesan y combinan las señales recibidas para producir diferentes salidas es imprescindible conocer los mecanismos que rigen la creación del patrón de arborización dendrítica (Major et al., 2013).

La construcción de modelos computacionales que permitan el estudio en profundidad de los procesos moleculares que gobiernan el comportamiento electro-fisiológico de la neurona y su crecimiento puede ayudar a elaborar de teorías acerca del desarrollo neuronal y, en última instancia, desentrañar el funcionamiento el cerebro (Van Ooyen, 2011), aunque, en la actualidad todavía estamos lejos de dicho objetivo. Además, los modelos computacionales reducen el elevado coste de producir reconstrucciones dendríticas a partir de datos reales, ya sea complementando dicho proceso, o asistiendo la creación de neuronas sintéticas en la creación de simulaciones cerebrales a gran escala.

Por si todo lo anterior no fuera suficiente, estudios recientes asocian malformaciones en las ramificaciones dendríticas con enfermedades como la esquizofrenia, el autismo o el alzheimer, o como resultado de malformaciones genéticas como el síndrome de Down (Kulkarni y Firestein, 2012). La creación de modelos y simulaciones bio-químicas realistas puede servir, en un futuro cercano, para encontrar dianas terapéuticas para estas y otras afecciones, así como para comprender su origen.

5.1.1. Modelos en la literatura

El desarrollo de modelos capaces de modelar y simular la estructura dendrítica y axonal de una neurona es un área de investigación en pleno auge con motivo de la aparición de proyectos de gran tamaño relacionados con el cerebro, como el *Blue Brain* y su sucesor, el *Human Brain Project* (HBP), en Europa y el *BRAIN initiative* en Estados Unidos. El objetivo de estos proyectos es el estudio del cerebro desde un punto de vista funcional, la creación de modelos y posterior simulación a gran escala mediante el uso de grandes supercomputadores.

Grosso modo podemos distinguir dos aproximaciones en la literatura: en primer lugar los modelos basados en el uso de reglas paramétricas para explicar y simular el crecimiento dendrítico, y por otra parte los modelos estadísticos basados en datos (*data-driven*).

En el primer grupo encontramos, entre otros, el software L-Neuron (Ascoli y Krichmar, 2000) basado en conjuntos de reglas recursivas que describen la geometría dendrítica utilizando sistemas Lindenmeyer. Los parámetros de las reglas se derivan de las medidas tomadas en dendritas reales, determinando que, por ejemplo, el orden centrífugo de un segmento determina la probabilidad de ramificación en la mayoría de casos.

También en este grupo encontramos la herramienta NeuGen (Eberhard et al., 2006), otro modelo basado en reglas cuyo objetivo no es crear células individuales,

si no redes completas 3D utilizando distribuciones estadísticas, ajustadas según los datos reales y restricciones geométricas. Este método no contempla la competición por los recursos entre dendritas en el proceso de generación y se centra en neuronas de capas IV y V.

El último representante de interés es NeuroConstruct (Gleeson et al., 2007), un modelo que baja al nivel de procesos biomoleculares para explicar la creación de las redes neuronales, utilizando fórmulas para calcular, por ejemplo, la resistencia estructural o la conductancia eléctrica.

Respecto al segundo grupo, tenemos también varias opciones. NETMORPH (Koene et al., 2009) es una herramienta cuyo objetivo es creare redes tridimensionales con morfologías neuronales realistas; introduce el concepto de conos de crecimiento, el cual utilizaremos en nuestro modelo, con tres acciones posibles: elongación, ramificación y giro. Simula cada cono de forma independiente y tiene en cuenta la competencia por los recursos entre los conos.

El trabajo que nos sirve de punto de partida (López-Cruz et al., 2011) encaja también dentro del segundo grupo. Crea un modelo para árboles dendríticos individuales teniendo en cuenta un gran número de variables, aunque al contemplar exclusivamente bifurcaciones, crea árboles dendríticos simplificados. Una de las características más destacadas es el uso de redes distintas para las bifurcaciones según el orden centrífugo de estas. Existen trabajos posteriores (Lin y Li, 2013) con variables similares que utilizan redes de regulación génicas para crear árboles independientes.

Fuera de estos dos grupos, tenemos aproximaciones más ingeniosas o creativas, como puede ser la creación de árboles dendríticos mediante crecimiento por difusión (Luczak, 2006), en el que las neuronas crecen mediante la incorporación de pequeñas partículas que se mueven de forma aleatoria por el espacio hasta que chocan con una estructura fija, una dendrita. Aunque el resultado puede llegar a ser estéticamente similar en algunos casos, el modelo no refleja en absoluto el mecanismo real por el que se produce el crecimiento dendrítico.

Por último, queda destacar dos algoritmos TREES (Cuntz et al., 2011) y cx3D (Zubler y Douglas, 2009) que tratan el problema del crecimiento dendrítico como un problema de minimización, en el que tenemos que reducir la cantidad de cableado a utilizar sujetos a una serie de restricciones geométricas.

5.2. Extracción y pre-procesamiento

La obtención de datos experimentales de la morfología de una única neurona específica es un proceso complejo, costoso y generalmente propenso a errores. Pese a la existencia de multitud de métodos para el marcado y tinción de neuronas individuales, el proceso dista mucho de ser perfecto, ya que, o bien la neurona es modificada durante el mismo (por ejemplo, se encoje tras el marcado), o el marcado puede ser incompleto.

La herramienta más habitual para la extracción de las variables morfológicas de una neurona sigue siendo hoy en día el microscopio óptico, que en el mejor de los casos ofrece una resolución cercana a $100nm$ (Gustafsson et al., 1999). Las limitaciones propias de la herramienta quedan acentuadas debido al procedimiento utilizado para la preparación de las muestras: una única muestra de tejido se divide en láminas muy finas que se fijan y observan individualmente para, finalmente, obtener una reconstrucción a partir de la pila de imágenes. Más recientemente, el uso del microscópio electrónico ha reducido alguna de estas limitaciones, especialmente aquellas asociadas a la resolución del microscopio óptico.

Una vez obtenidas las imágenes, el proceso de extracción y digitalización de los datos morfológicos se realiza a través de un proceso semi-automático denominado *tracing* (Türetken et al., 2011), en el cual se marca el recorrido de cada neurita en la imagen, asistido por herramientas especializadas como NEUROLUCIDA (Glaser y Glaser, 1990). Aún y con el uso las técnicas más recientes, el proceso de *tracing* requiere una gran cantidad de supervisión manual y hace que el resultado del proceso quede sujeto al criterio subjetivo del neurocientífico encargado de realizar la reconstrucción. Aunque, por lo general, el resultado de éste proceso suele tener una tasa de errores aceptable, todavía quedan por resolver algunos problemas como las *intersecciones* de neuritas en la imagen, los cortes en la reconstrucción (especialmente en el eje z), la imprecisión en la medida del diámetro de las neuritas y las reconstrucciones incompletas como resultado de los cortes realizados en la preparación (Ascoli et al., 2001).

En los siguientes apartados detallaremos el proceso realizado para la lectura de las reconstrucciones dendríticas digitalizadas de un conjunto de neuronas determinado, la selección de dichas neuronas, el proceso de filtrado así como todas las tareas de pre-procesamiento realizadas durante la adquisición de los datos.

5.2.1. Lectura y digitalización

La representación digital de la estructura dendrítica tridimensional 3D más utilizada en la literatura es la representación de las ramificaciones dendríticas (y en general de las neuritas) como una secuencia de cilindros con diferente diámetro, orientación y longitud. Los formatos digitales más extendidos para la descripción de la morfología dendrítica son el formato *SWC* (una tabla con coordenadas cartesianas, diámetros, etc.) y el formato Neurolucida ASCII, (*ASC*), de interpretación más compleja.

En este trabajo hemos implementado un intérprete de ficheros *ASC* en R. El formato Neurolucida se basa en la descripción, línea a línea, de cada cilindro en el árbol dendrítico. En concreto, para cada cilindro tenemos sus coordenadas cartesianas y diámetro. La conectividad de cada elemento del fichero Neurolucida (el cilindro que le precede y los que le siguen) viene dada por la propia estructura del árbol y no se especifica en cada línea; esta estructura queda determinada a través del número de indentaciones en cada línea, donde en cada bifurcación los cilindros hijo son indentados con respecto al padre, reforzado mediante el uso de anidaciones, similares a la existentes en cualquier lenguaje de programación. A continuación se incluye un fragmento de un fichero *ASC* real con fines ilustrativos.

```
( (Color Blue)
  (Dendrite)
; (    x       y        z         w   )  ;
  (   -1.96    10.19    -1.60     1.45)  ; Root
  (    0.82    14.08    -1.60     1.34)  ; R, 1
  (    1.83    16.08    -2.80     1.22)  ; R, 2
  (
    (    5.49    15.58   -50.79     0.78)  ; R-1, 1
    (    7.05    16.58   -49.99     0.78)  ; R-1, 2
    ...
```

Las reconstrucciones 3D de arborizaciones dendríticas reales, necesarias para el aprendizaje del modelo, se han obtenido de *Neuromorpho.org* (Ascoli et al., 2007), un repositorio web donde es posible encontrar reconstrucciones catalogadas por especie, género, etc. De todos los conjuntos de reconstrucciones disponibles, hemos seleccionado un conjunto de 41 reconstrucciones de neuronas piramidales de capa II y III pertenecientes a ratas con 36 días de vida en formato *ASC* (reconstrucciones pertenecientes al archivo Svoboda (Shepherd y Svoboda, 2005) en Neuromorp-

ho.org). Las reconstrucciones incluyen además de la arborización dendrítica basal, reconstrucciones del axón y la dendrita ápical en gran detalle.

El intérprete de ficheros en formato Neurolucida implementado en R, procesa los ficheros originales y los transforma a un formato interno de tabla, más sencillo de procesar y manipular que el árbol original. En la tabla, cada fila se corresponderá con una entrada en el fichero original con coordenadas cartesianas y diámetro, a la que se le añade información adicional. Para cada entrada en el fichero original, tendremos una fila en la tabla con los siguientes campos:

- Un identificador numérico único para cada punto
- Un identificador numérico único de la célula a la que pertenece cada punto
- Un identificador numérico del árbol dendrítico al que pertenece el punto. Dicho identificador será único para cada dendrita dentro de cada célula
- Un campo que identifica el tipo de estructura a la que pertenece el punto: dendrita basal, apical, axón o soma
- Las coordenadas cartesianas del punto respecto al sistema de referencia de cada célula
- El diámetro de la neurita en dicho punto
- El identificador único del punto anterior al actual (padre). En caso de no existir antecesor tomará el valor NA
- Los identificadores únicos de sus descendientes, hasta un máximo de dos. En caso de no tener descendientes tomará el valor NA
- El identificador de la rama a la que pertenece el punto dentro del árbol dendrítico. Por ejemplo: R-1-2-1
- La longitud del cilindro, medida desde el antecesor
- La longitud del camino hasta la raíz del árbol al que pertenece el punto (camino hasta el soma)
- El volumen total del camino hasta la raíz del árbol al que pertenece el punto (volumen acumulado hasta el soma)

A lo largo de todo el proceso trabajaremos directamente sobre esta tabla, añadiendo nuevas columnas en caso que sea necesario, pero manteniendo como mínimo estos campos.

5.2.2. Reparación y filtrado

Aunque el conjunto de neuronas seleccionado se hizo, entre otros motivos, siguiendo un criterio basado en la calidad de las reconstrucciones, entre las 41 reconstrucciones seleccionadas existen diferentes errores que necesitan ser subsanados, o incluso pueden ser motivo de descarte de la reconstrucción en cuestión.

En primer lugar, tras la carga inicial de los datos, realizamos una inspección visual de los mismos mediante la representación 3D de la arborización dendrítica basal utilizando la librería *RGL*; representaremos el soma como una esfera, asignaremos a cada árbol dendrítico un color característico, y para cada fila de la tabla de datos pintaremos un punto en el espacio y uniremos cada punto con su padre (si lo hubiese) mediante un segmento, ambos del color asociado al árbol al que pertenecen. A partir de dicha inspección, se detecta que en una gran cantidad de casos la dendrita apical está etiquetada como dendrita basal, por ejemplo en la figura 5.1 vemos que la dendrita amarilla que está marcada como dendrita basal se diferencia claramente del resto al tratarse de una dendrita apical.

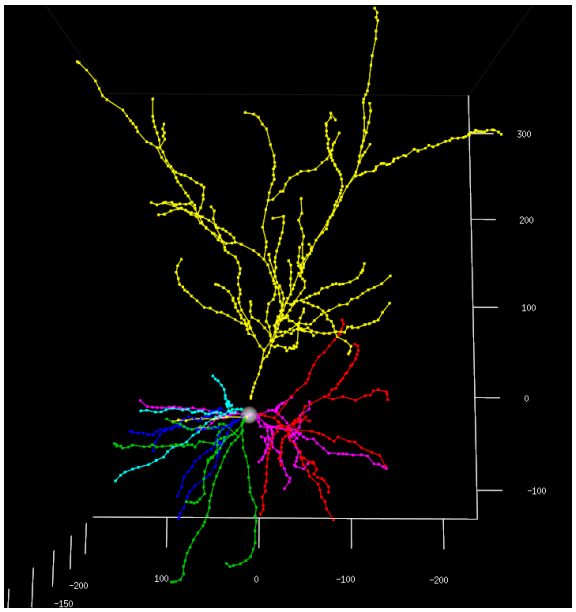


Figura 5.1: Ejemplo de neurona con dendrita ápical mal etiquetada

Para solucionar este problema se realiza una inspección visual de la arborización dendrítica de cada una de las 41 reconstrucciones en el conjunto de datos, encontrando que en la mayoría de ellas (más de 30) existe dicho problema. En una segunda vuelta se modifican de forma manual las dendritas apicales mal etiquetadas. Continuando con el ejemplo anterior, en la figura 5.2 se muestra el resultado tras el

reetiquetado de la dendrita apical.

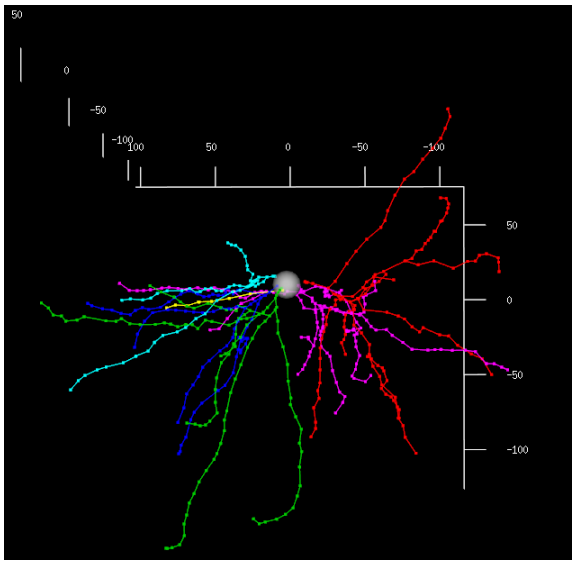


Figura 5.2: Ejemplo de neurona con dendrita apical reetiquetada

De forma paralela, el análisis de la estructura de las arborizaciones dendríticas en los datos reveló la existencia de 3 neuronas (reconstrucciones) con trifurcaciones en el árbol. De acuerdo a nuestro modelo, y siguiendo lo expuesto en la literatura, consideramos dichas trifurcaciones como un error en la reconstrucción, ya que se asume que un árbol dendrítico únicamente puede tener bifurcaciones en un mismo nodo. Ante la imposibilidad de establecer un criterio sólido y fundamentado para reparar este problema se opta por excluir estas tres reconstrucciones del conjunto de datos final, quedando un total de 38 reconstrucciones válidas. La detección y descarte de trifurcaciones se implementa como una comprobación automática durante el proceso de carga e interpretación de los ficheros *ASC*.

5.2.3. Centrado y alineación

Con el objetivo de simplificar los cálculos posteriores y construir un modelo acorde a lo establecido en la literatura, durante el proceso de carga y extracción se realizarán una serie de transformaciones sobre los datos originales para que cumplan las siguientes condiciones:

1. El centro del cuerpo neuronal (soma) se encuentra en el origen de coordenadas (0,0,0)

2. El origen de las dendritas basales se encuentra en el plano $z = 0$

Para cumplir con la primera condición es necesario calcular la posición actual del centro del soma en cada célula y, posteriormente, desplazar las coordenadas cartesianas de todos los puntos de cada célula para desplazar la posición del centro al origen de coordenadas. Para calcular la posición del soma, y en general a lo largo del modelo, asumiremos que tiene forma esférica (Quan et al., 2013; Stelescu et al., 2012), aunque estudios más avanzados sugieren que se asemeja más a un elipsoide (Luengo Sánchez, 2014). Estableceremos, por lo tanto, el centro del soma como el baricentro de las raíces de los árboles dendríticos en la neurona y tomaremos como radio del soma la distancia media desde este baricentro a cada una de las raíces.

Una vez centrada la neurona en el origen de coordenadas, el siguiente paso es rotar la neurona de forma que la raíces de las dendritas basales se encuentren tan cerca como sea posible del plano $z = 0$. Para llevar a cabo esta tarea haremos uso de la interpretación geométrica del análisis de componentes principales (PCA) (Jolliffe, 2002), una técnica de extracción de variables. El análisis de componentes principales devuelve una nueva base ortogonal del espacio de variables, de forma que el primer vector recoge la varianza de mayor tamaño del conjunto de datos, el segundo vector la segunda mayor y así sucesivamente. En el caso tridimensional, centrado en el origen de coordenadas, el PCA nos ofrece la recta y/o el plano que mejor explican los datos; en este último caso, el tercer vector de la base ortonormal resultado de la aplicación del PCA resulta ser el vector normal del plano que recoge la mayor varianza de los datos.

Teniendo en cuenta la suposición de que las raíces de las dendritas basales se distribuyen a lo largo de un plano, aplicamos PCA a las coordenadas cartesianas de las raíces junto con el origen de coordenadas y extraemos el tercer vector de la nueva base, el vector normal $\mathbf{v}$ del plano que forman en los datos originales las raíces de las dendritas basales. Una vez obtenido el vector normal, calculamos la matriz R de la rotación que transforma $\mathbf{v}$ en $\mathbf{z} = (0, 0, 1)$ con respecto al eje dado por su producto vectorial y el origen de coordenadas. Finalmente, multiplicamos todas las coordenadas de los puntos en la neurona por la matriz de rotación obtenida para realizar dicha alineación (figura 5.3).

5.3. Descriptores

En esta sección se detallan las variables seleccionadas para describir la arborización dendrítica basal. En general, el criterio seguido para la elección de las variables descriptoras ha sido su sentido biológico, su disponibilidad (debemos poder calcu-

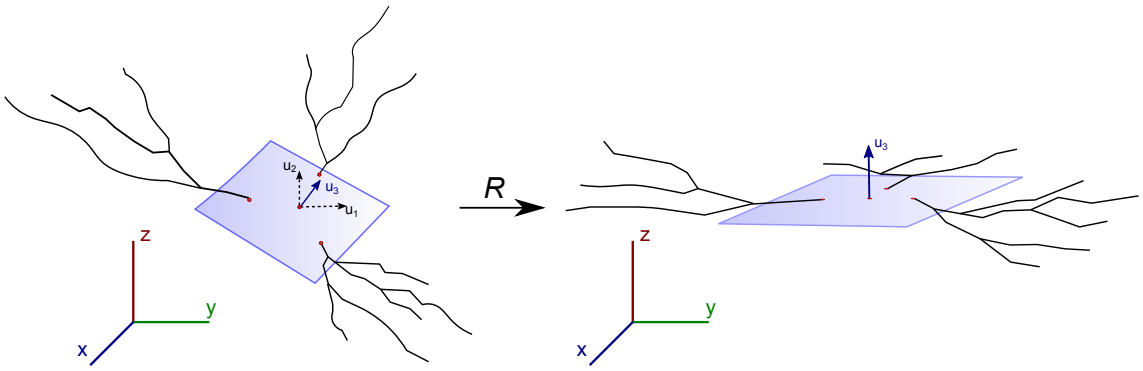


Figura 5.3: Alineación de las raíces de las dendritas basales con el plano $z = 0$

larlo), y su presencia en otros modelos del estado del arte o estudios experimentales. Partiremos del conjunto base de variables del trabajo de referencia (López-Cruz et al., 2011).

Una excepción a la regla anterior es el diámetro: El diámetro de la dendrita es un elemento clave, ya que determina su capacidad conductiva (Chklovskii y Stepanyants, 2003). El problema con el diámetro no está ni en su sentido biológico, ni en su presencia en otros modelos si no en su disponibilidad; con frecuencia en las reconstrucciones encontramos diámetros erróneos (por ejemplo constantes), principalmente debido a limitaciones en la resolución del microscopio y en el algoritmo de *tracing*. Pese a estos problemas, se ha decidido incluir el diámetro en la variables descriptoras por ser un elemento de importancia principal y con la esperanza de disponer en el futuro de datos con mediciones de diámetro correctas.

5.3.1. Elementos de la arborización

Con el fin de facilitar la comprensión de las variables en el modelo seleccionadas para caracterizar el patrón de arborización dendrítica, previamente necesitamos definir ciertos segmentos y subconjuntos dentro de una reconstrucción, generalmente contemplados en la literatura, sobre los cuales efectuaremos una serie de medidas que se utilizarán como descriptores de la arborización.

Distinguiremos entre conjuntos, grupos de nodos con alguna característica en común, y segmentos,series de nodos entre bifurcaciones o desde un nodo terminal hasta una bifurcación.

Dado un nodo concreto n, perteneciente a un árbol dendrítico T, definimos los siguientes conjuntos (figura 5.4):

- La **sub-dendrita**: Es el conjunto de antecesores del nodo n en el árbol T. Es

decir, el conjunto de nodos que construyen el camino desde n hasta la raíz del árbol T

- El **sub-árbol**: Es el conjunto de nodos pertenecientes al árbol T anteriores (generados o seleccionados antes) que el nodo n

- La **sub-arborización**: El conjunto de nodos de la misma célula que n, generados o seleccionados con anterioridad al nodo n y que no pertenecen al árbol T

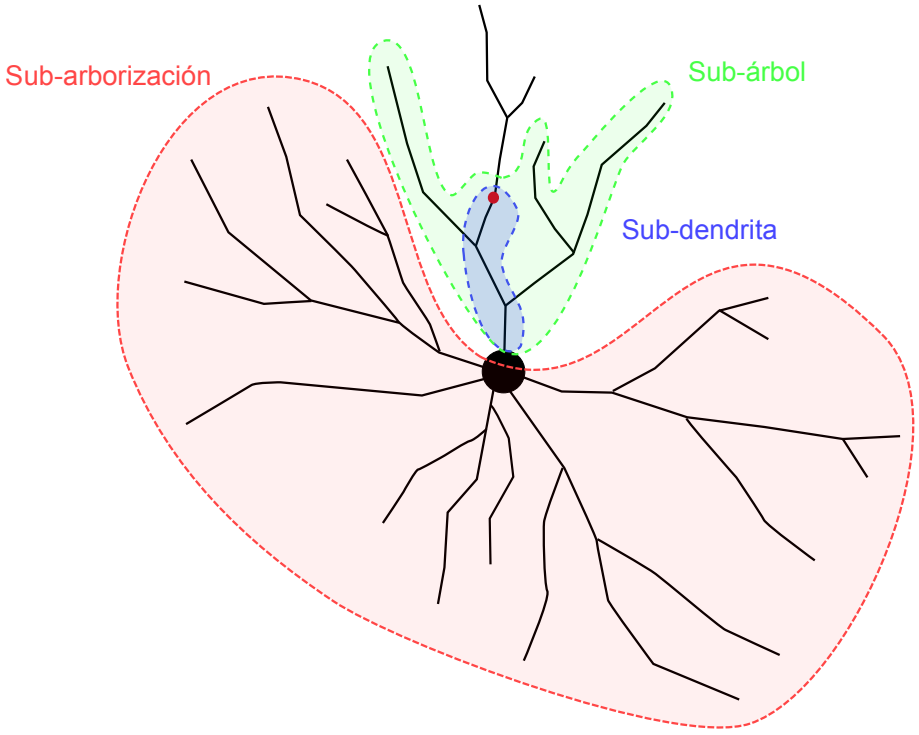


Figura 5.4: Dado un nodo concreto (rojo intenso) se definen tres conjuntos representativos en la arborización: La subdendrita (azúl), el subárbol (verde) y la sub-arborización(rojo)

De igual manera, para el mismo nodo n definimos cinco segmentos representativos (figura 5.5), entendiendo segmento como el conjunto de nodos entre dos bifurcaciones, entre una bifurcación y un nodo terminal, una bifurcación y la raíz desde la raíz hasta el nodo terminal si no existe ninguna bifurcación:

- El **segmento actual**: Compuesto por los nodos desde n hasta la primera bifurcación o la raíz

- El **segmento hermano**: Compuesto por los nodos pertenecientes al segmento que nace de la misma bifurcación que el segmento actual de n
- El **segmento padre**: Compuesto por los nodos entre las dos últimas bifurcaciones desde el nodo n o desde la última bifurcación hasta la raíz
- El **segmento más cercano en el sub-árbol**: Compuesto por los nodos que pertenecen al segmento del nodo más cercano a n dentro del sub-arbol que no pertenece a la sub-dendrita de n
- El **segmento más cercano en la sub-arborización**: Compuesto por los nodos que pertenecen al segmento del nodo más cercano a n dentro de la sub-arborización

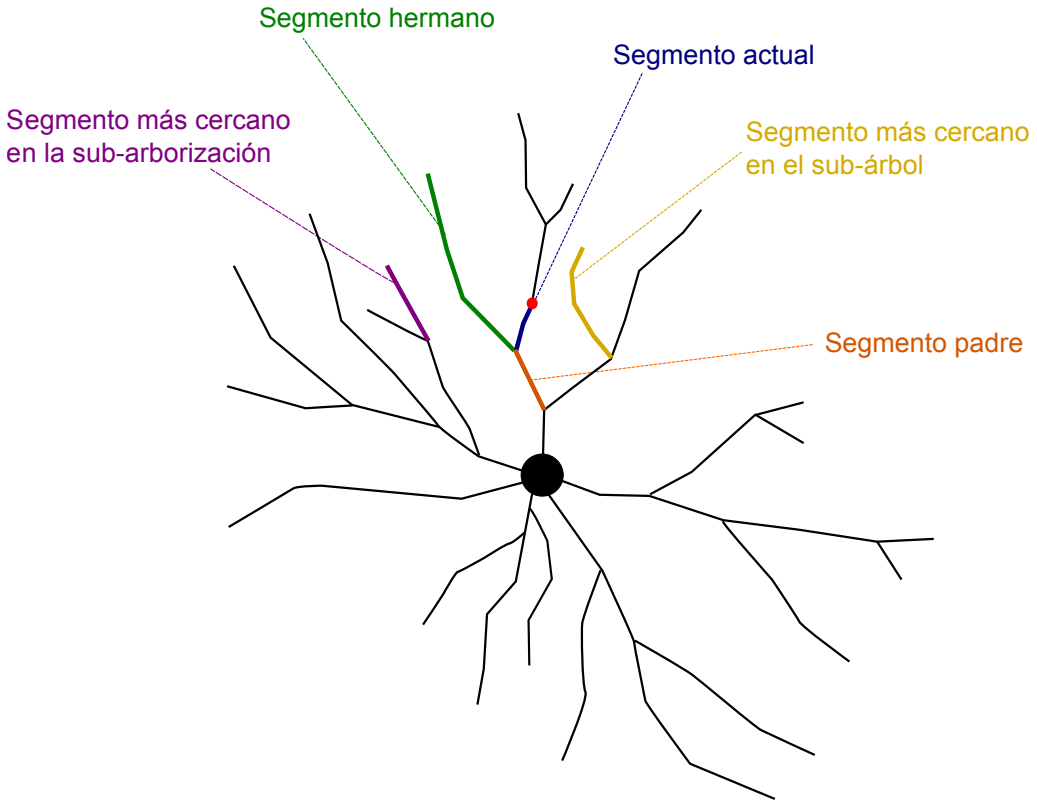


Figura 5.5: Dado un nodo concreto (rojo intenso) se definen cinco segmentos representativos en la arborización: El segmento actual (azúl), el segmento hermano (verde), el segmento padre (naranja), el segmento más cercano dentro del sub-árbol (amarillo) y el segmento más cercano dentro de la sub-arborización(púrpura)

No todos los conjuntos y segmentos tienen sentido para todos los nodos, en concreto para el primer segmento de un árbol dendrítico ninguna de las definiciones

anteriores tiene sentido, mientras que para nodos con orden menor a 1 no tenemos sub-árbol, segmento hermano, segmento padre ni segmento más cercano en el sub-árbol.

Todas las medidas de orientación para cada uno de los segmentos y conjuntos se tomarán con respecto al sistema de referencia local al nodo actual. Dicho sistema de referencia, compuesto por los vectores $\mathbf{u}_x, \mathbf{u}_y, \mathbf{u}_z$, se calculará como el resultado de aplicar una rotación positiva al sistema de referencia canónico que transforme el vector $\mathbf{x} = (1, 0, 0)$ en el vector unitario dado por el segmento anterior. El azimut se calculará como el ángulo formado entre el vector $\mathbf{u}_x$ y la proyección del segmento actual sobre el plano formado por $\mathbf{u}_x, \mathbf{u}_y$; de manera recíproca, la inclinación se medirá como el ángulo formado entre el vector $\mathbf{u}_x$ y la proyección del segmento actual sobre el plano formado por $\mathbf{u}_x, \mathbf{u}_y$.

5.3.2. Variables de construcción

Las variables de construcción son las encargadas de describir el segmento cuyo origen está en un nodo dado. La idea detrás de la definición de estas variables es la posibilidad de, una vez aprendido el modelo, ser capaces de crear arborizaciones dendríticas en función de los valores que tomen en el modelo dichas variables de construcción. Para ser capaces de obtener las coordenadas cartesianas del siguiente nodo dado un punto de origen utilizaremos el azimut y la inclinación para determinar la orientación del segmento respecto al segmento anterior, la longitud y el diámetro para determinar su tamaño y una variable adicional que denominaremos *tipo* que determinará el comportamiento del nodo destino y cuyo valor podrá ser: Terminal, elongación o bifurcación. En la figura quedan representadas las variables de orientación de un segmento con respecto al anterior.

5.3.3. Variables por conjunto

Basándonos en otros modelos existentes en la literatura y en los hallazgos experimentales relativos al crecimiento dendrítico, seleccionamos seis variables para describir cada uno de los tres subconjuntos de nodos de la arborización:

- Número de nodos terminales en el conjunto
- Número de bifurcaciones en el conjunto
- Número de conos de crecimiento en el conjunto, con el fin de recoger la competitividad existente entre dendritas

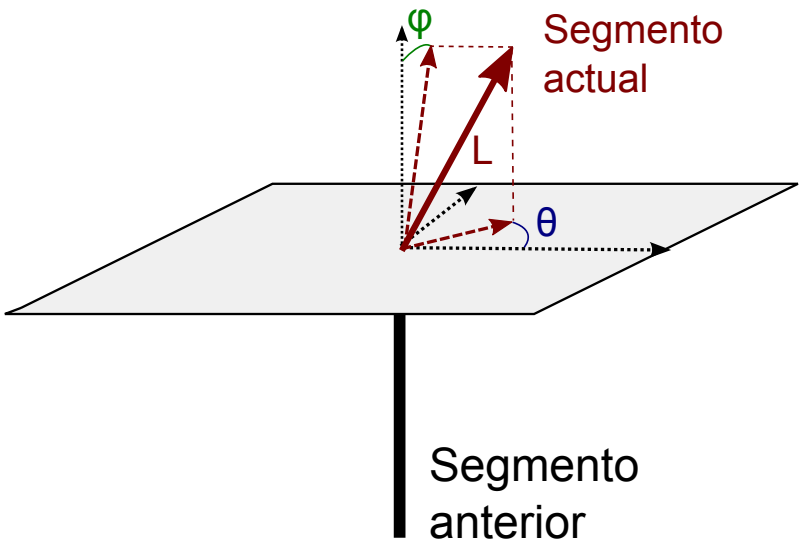


Figura 5.6: Variables de construcción de un segmento: orientación y longitud. Para la orientación se utiliza el azimut (θ) y la inclinación (ϕ) locales al nodo actual

- Longitud total del conjunto, medida como la suma de la longitud de los cilindros
- Volumen total del conjunto, medida como la suma del volumen de los cilindros
- Orden centrífugo máximo en los nodos del conjunto, entendiendo el orden centrifugo de un nodo como el número de bifurcaciones en el camino desde el nodo hasta la raíz de la dendrita. La probabilidad de bifurcación de una dendrita parece decrecer exponencialmente a medida que aumenta el orden centrífugo

De forma adicional, para el sub-árbol y la sub-dendrita se calculará la orientación promedio desde la raíz (azimut e inclinación) en el sistema de referencia local del nodo, cuyo cálculo queda descrito en el subapartado anterior. La orientación promedio del conjunto se calcula como la media ponderada por la longitud del segmento de los vectores unitarios desde la raíz hasta el punto intermedio de cada segmento.

Para la sub-árborización el calculo de la orientación promedio carece de sentido ya que estamos tratando con varios árboles dendríticos de forma simultánea, siendo el resultado final de escalo valor informativo.

5.3.4. Variables por segmento

Al igual que en el caso anterior, para cada uno de los segmentos relevantes para cada nodo utilizaremos una serie de variables para su descripción, basándonos una

vez más en la literatura para su elección. Las variables utilizadas para todos los segmentos son:

- Longitud total del segmento, medida como la suma de las longitudes de los cilindros del segmento
- Volumen total del segmento, medida como la suma del volumen de los cilindros del segmento
- La tasa de encogimiento, medida como el ratio entre el diámetro inicial del segmento y el diámetro final
- Orientación (azimut e inclinación) del segmento medido en el sistema de referencia local del nodo actual. Para el cálculo de esta orientación e utiliza la orientación promedio del segmento calculada de forma similar a la expuesta para los subconjuntos

Además, para los segmentos más cercanos en el sub-árbol y la sub-arborización se incluirán las siguientes variables con el fin de reflejar en qué dirección y a que distancia del nodo actual se encuentran cada uno de estos segmentos:

- Distancia entre el nodo actual y el segmento
- Orientación (azimut e inclinación) del vector entre el nodo actual y el nodo más cercano, medida en el sistema de referencia local del nodo actual

5.3.5. Variables del nodo

En último lugar incluiremos tres variables que representan características propias del nodo actual en la reconstrucción, en concreto mediremos:

- El orden centrífugo del nodo, es decir el número de bifurcaciones entre el nodo y la raíz de la dendrita
- El diámetro del nodo
- La distancia en línea recta hasta el soma desde el nodo
- La longitud y el volumen total del camino hasta el soma. Estas variables coinciden por completo con las medidas para la sub-dendrita

- El volumen del soma esférico, constante para todos los nodos pertenecientes a una misma neurona. Según el modelo de crecimiento por difusión de tubulina desde el soma, la cantidad de neuritas (en volumen) que puede mantener una neurona viene dada por su volumen

6

Modelización

Una vez seleccionados los descriptores que caracterizan la arborización dendrítica basal de una neurona piramidal, nuestro objetivo es la construcción de un método capaz de explicar desde cero el proceso completo.

El modelado de la arborización presenta complejidades adicionales, ya que es necesario establecer un modelo jerárquico, donde cada paso depende del anterior, siendo la arborización dendrítica el último eslabón:

1. El radio del soma, supuesta esfericidad
2. El número de dendritas basales a construir
3. Dado el número de dendritas basales, establecer su orientación en el plano $z = 0$
4. Finalmente, la arborización dendrítica basal

6.1. Radio del soma

La forma y tamaño del soma en las herramientas y modelos existentes para reconstrucción de árboles dendríticos existentes en la literatura ha sido un factor sorprendentemente obviado. Pese a que, si asumimos un modelo de crecimiento limitado por la difusión de una sustancia, tubulina, desde el soma hacia los conos de crecimiento, el tamaño del soma tiene un impacto directo en la longitud y complejidad de las arborizaciones que una neurona puede mantener, con frecuencia la información relativa al contorno o el tamaño del soma es imprecisa o incluso inexistente.

Debido a esta última limitación, y teniendo en cuenta que la modelización del soma no es el objetivo principal de este trabajo, asumiremos la esfericidad del soma al igual que se hizo en el capítulo anterior (Quan et al., 2013; Stelescu et al., 2012), midiendo el radio del soma de las reconstrucciones reales como la distancia media entre el origen de coordenadas (el centro del soma) a las raíces de las dendritas basales.

A partir de dichas medidas, y ante la ausencia de estudios similares en la literatura que pudieran servir como referencia, se decide adoptar un modelo en el que el radio del soma sigue una distribución lognormal con dos parámetros el logaritmo de la media (μ) y el logaritmo de la desviación típica (σ). Utilizamos el paquete *MASS* de R para realizar el ajuste de los parámetros de la distribución, cuyo resultado puede verse en la figura 6.1. Tras el ajuste, aplicamos el test no paramétrico de Kolmogorov–Smirnov para medir la bondad del ajuste, obteniendo un p-valor de $p = 0,5090$, aunque este resultado no puede considerarse significativo debido al reducido número de muestras (38).

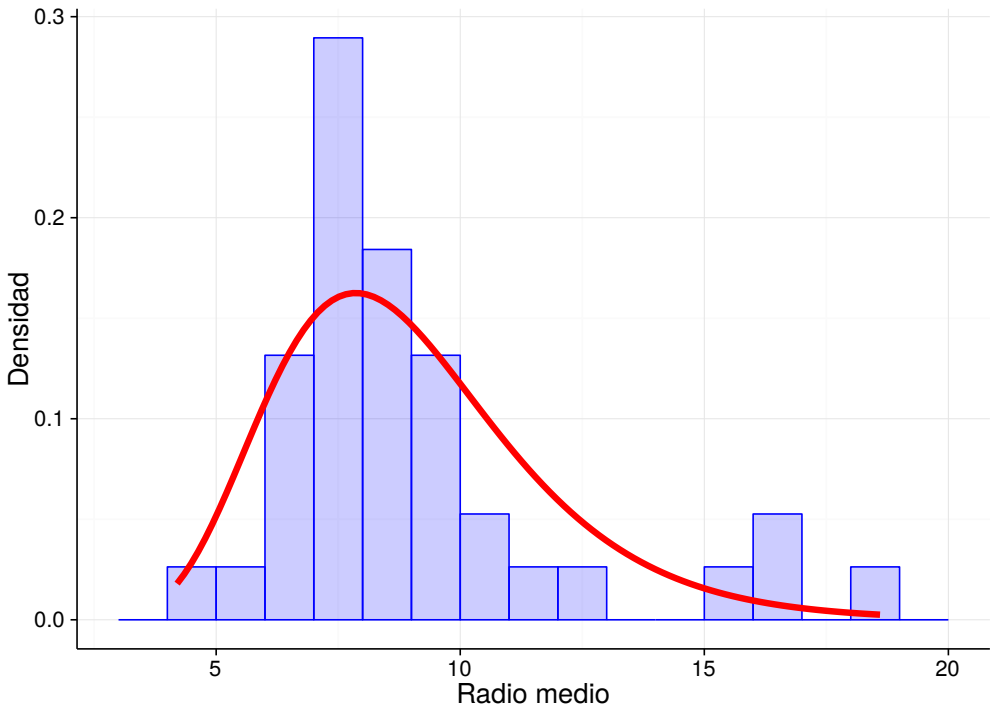


Figura 6.1: Datos reales (azul) y su ajuste a una distribución log-normal con parámetros $\log \mu = 2,15$ y $\log \sigma = 0,299$ (línea roja)

6.2. Localización de la raíz de las dendritas basales

Para determinar la localización en el plano $z = 0$ de la raíz de las dendritas basales de una neurona es necesario establecer un modelo en dos pasos en el que en primer lugar determinamos el número de dendritas basales de la neurona, para, después modelizar los ángulos entre dichas dendritas.

Para modelizar el número de dendritas basales hemos seleccionado una distribución discreta con dominio en los enteros positivos Conway-Maxwell-Poisson (Shmueli et al., 2005), una generalización de la distribución de Poisson en la que se incluye un parámetro adicional para controlar la dispersión.

Utilizaremos el paquete *compoisson* de R que implementa todas las funciones necesarias para trabajar la distribución, incluyendo el ajuste de parámetros a partir de muestras reales. Al igual que en el caso anterior, el resultado del ajuste y la distribución real de los datos puede verse en la figura 6.2. Para comprobar la bondad de ajuste de la distribución utilizaremos una revisión del test de Kolgomorov-Smirnov implementada en el paquete *dgof* para la evaluación de distribuciones discretas, obteniendo un p-valor superior a $0,99$.

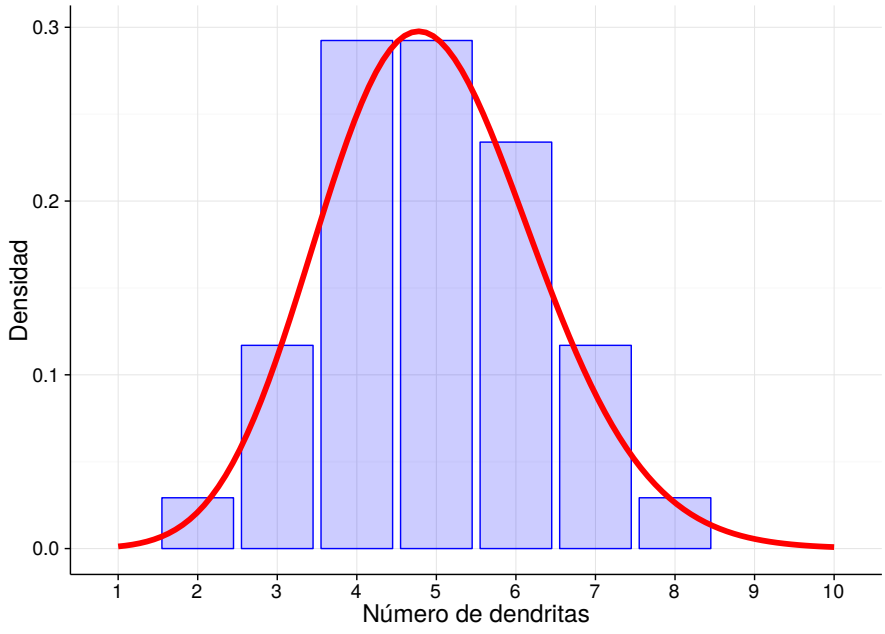


Figura 6.2: Datos reales (azul) y su ajuste a una distribución Conway-Maxwell-Poisson con parámetros $\lambda = 131,51$ y $\nu = 2,93$ (línea roja)

En el siguiente paso, estudiaremos la orientación plana de las raíces de las den-

dritas basales mediante el uso de la ditribución von-Mises circular multivariante y el paquete *mvCircular* expuesto en la primera parte de este trabajo. Realizaremos el ajuste paramétrico en aquellos casos en los que el número de muestras para el número de dendritas basales sea superior a 3 (3, 4, 5, 6, y 7 dendritas), considerando que con menos muestras no es posible obtener resultados consistentes en ningún caso.

Utilizaremos una matriz de penalización con estructura similar a la expuesta en la aplicación de la distribución von Mises multivariante en el estudio anterior (Rodriguez-Lujan et al., 2015) donde la penalización crece a medida que aumenta la distancia entre lo ángulos. En los casos de 6 y 7 dendritas, el reducido número de muestras frente a la elevada complejidad paramétrica (22 y 28 parámetros respectivamente) hace que el resultado obtenido sea extremadamente pobre, con una estimación de la divergencia KL excesivamente elevada, por lo que se rechazan dichos resultados; en el resto de casos se acepta el ajuste y sus resultados se adjuntan en el apéndice 8.2, la gráfica del resultado para cinco dendritas se muestra en la figura 6.3.

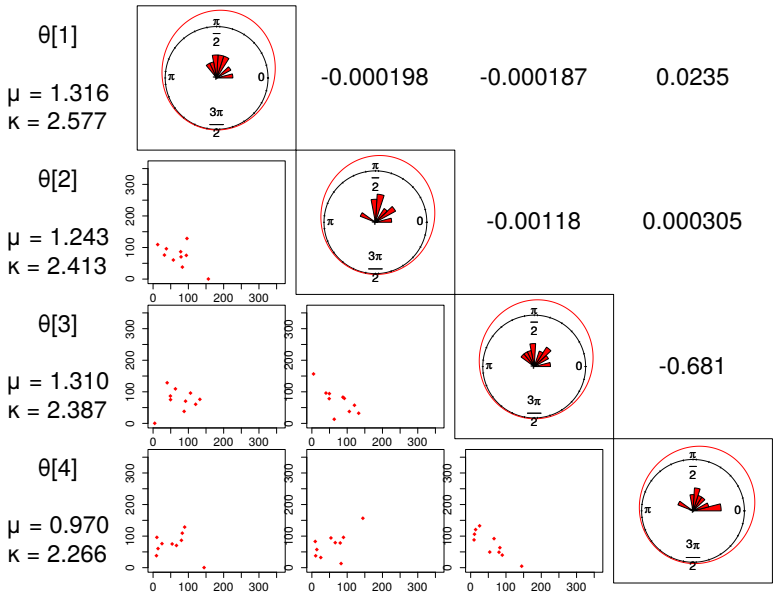


Figura 6.3: Ajuste de la orientación de las dendritas basales a una distribución von Mises multivariante para cinco dendritas

6.3. Crecimiento simulado

La construcción de la arborización dendrítica es un proceso de crecimiento, bifurcación y retracción que depende tanto de factores intrínsecos a la neurona, como de factores externos. Su forma es, por lo tanto, el resultado de un proceso dinámico con estructuras que se ramifican, crecen o decrecen en función de la actividad sináptica, la competición con otras dendritas por los recursos y otros factores intrínsecos a la neurona, como es la difusión de proteínas asociadas a los microtúbulos desde el soma.

Pese a existir visualizaciones en tiempo real del crecimiento neuronal, la inmensa mayoría de reconstrucciones 3D proceden de estudios que utilizan técnicas que requieren el fijado de los tejidos, por lo que únicamente disponemos de una captura del resultado del proceso dinámico en un momento determinado, sin ningún tipo de información sobre cómo se ha llegado al estado actual ni de los estados anteriores.

Nuestra capacidad de construir un modelo basado en los datos queda limitada en este aspecto por la evidente ausencia de reconstrucciones secuenciales para una misma neurona. Por este motivo, para alcanzar el objetivo de modelizar y simular el crecimiento de la arborización es necesario adoptar un modelo teórico en este aspecto. En este sentido, existen en la literatura modelos estocásticos que describen el crecimiento de una neurona a lo largo del tiempo mediante el concepto de conos de crecimiento, principalmente representados en la herramienta NETMORPH (Koene et al., 2009).

Para nuestro modelo, acorde con lo implementado en NETMORPH, asumiremos que el crecimiento de una u otra dendrita dentro de una misma célula se rige mediante la competición por un recurso limitado (tubulina), producido en el soma y que llega hasta los conos de crecimiento mediante transporte activo o difusión desde el soma (Graham y van Ooyen, 2004; Van Ooyen et al., 2001).

6.3.1. Función de selección

Con el fin de ser capaces de modelar y simular el proceso de crecimiento es necesario que, dado un conjunto de nodos en crecimiento o activos, seamos capaces de establecer la probabilidad de cada nodo para ser el siguiente en crecer. Denominaremos función de selección a la función que dado el conjunto de nodos, junto con una serie de características, devuelve el siguiente nodo a ser procesado.

En este trabajo, de acuerdo al modelo de competición por recursos (Van Ooyen et al., 2001), queremos una función de selección que:

- Otorgue más probabilidad a las dendritas con mayor tamaño en volumen, ya que suponemos que dichas dendritas recibirán un aporte mayor tanto por difusión como, especialmente, por transporte activo desde el soma
- Limite la probabilidad de crecimiento a medida que la distancia, entendida como camino, desde el nodo activo hasta el soma aumenta, siguiendo una argumentación similar, pero opuesta en resultado, al caso anterior.
- Tenga en cuenta para el cálculo de la probabilidad la presencia de otros nodos activos, su distancia y tamaño

A partir de las condiciones anteriores, implementamos una función de selección que otorga a cada nodo activo una probabilidad proporcional a la siguiente cantidad:

$$p_{crecimiento}(nodo) \approx \frac{Volumen\,hasta\,el\,soma}{(Camino\,hasta\,el\,soma)^3}$$

En el caso de que entre los nodos activos existiese un nodo raíz, cuyo volumen y camino hasta el soma es nulo, este será seleccionado siempre. Si hubiese más de un nodo raíz, la función seleccionara uno de los nodos raíz al azar de forma uniforme.

6.3.2. Obtención de descriptores

Una vez se ha definido cómo simular el crecimiento dendrítico, aplicaremos este procedimiento sobre reconstrucciones morfológicas reales, obteniendo las variables descriptoras de cada nodo seleccionado en cada iteración. El algoritmo para la obtención de instancias a partir de una reconstrucción real se basa en los siguientes pasos

1. En primer lugar marcamos todos los nodos como *"no vistos"* salvo los nodos raíz de cada dendrita, los cuales marcamos como ïvistosÿ activos (en crecimiento)
2. Mientras quede algún nodo activo en la reconstrucción:
   a) Seleccionamos el siguiente nodo aplicando la función de selección sobre los nodos activos
   b) Obtenemos los descriptores de dicho nodo en el estado actual de la reconstrucción, es decir, teniendo en cuenta únicamente los nodos *"vistos"*. Hay que tener en cuenta que el conjunto de descriptores varía en función del tipo de nodo (bifurcación y elongación) y su orden centrífugo.

c) Marcamos los descendientes del nodo, si los hubiera, en la reconstrucción como v̈istos:̈ También marcamos los descendientes que no sean un nodo terminal como äctivos:̈

d) Eliminamos el nodo seleccionado del conjunto de nodos activos

En la figura 6.4 se muestra de forma esquemática sobre una reconstrucción simplificada el método propuesto en el párrafo anterior.

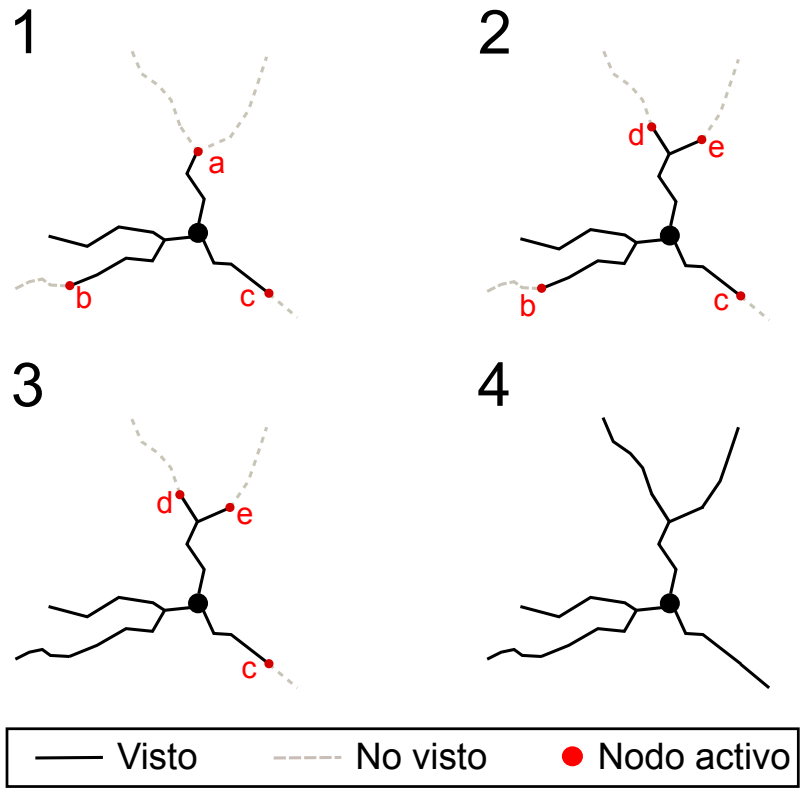


Figura 6.4: Ejemplo de obtención de descriptores mediante crecimiento simulado: (1) Punto intermedio del proceso con tres nodos activos (a,b,c). (2) Estado tras la selección del nodo bifurcación (a), se incluyen sus descendientes (d,e) en la lista de nodos activos. (3) Estado tras la selección del nodo de elongación (b), su nodo descendiente, al ser terminal no se incluye en la lista de nodos activos. (4) Estado final del proceso tras varias iteraciones al quedar vacía la lista de nodos activos

6.4. Redes Bayesianas

Las redes Bayesianas son un modelo gráfico probabilístico (Koller y Friedman, 2009) compuesto, principalmente, por dos elementos:

- El grafo dirigido acíclico $\mathcal{G}$ que captura las independencias condicionales entre las variables del problema
- El componente probabilístico $\mathcal{P}$ que incluye las distribuciones de probabilidad condicionales para cada nodo dado sus padres en el grafo

La elección de un modelo Bayesiano se realiza por su capacidad de representar el problema de una forma compacta, gracias a la factorización; y por ser un modelo comprensible e interpretable, a diferencia de otros modelos de *caja negra* como las máquinas de vectores soporte o las redes neuronales. Además las redes Bayesianas son capaces de efectuar cualquier tipo de razonamiento sobre las variables del problema, esta flexibilidad nos permite ser capaces de simular el crecimiento dendrítico mediante inferencia.

En un principio se contempló la posibilidad de utilizar redes Bayesianas híbridas, donde las variables de problema pueden ser Gaussianas o discretas, con la única restricción de que una variable discreta no podía tener un padre continuo. Esta opción finalmente fue descartada por dos motivos principalmente:

- La distribución de algunas variables era claramente no Gaussiana, y por tanto el modelo obtendría una representación muy pobre e imprecisa de dichas variables, siendo algunas tan relevantes como la longitud del segmento (figura 6.5)
- El proceso de aprendizaje con el paquete de R *deal* era intratable desde un punto de vista computacional debido al elevado número de variables (hasta 63). Una estimación optimista del tiempo necesario para el aprendizaje estructural con dicho algoritmo determinaba el tiempo de ejecución necesario en varias semanas.

En su lugar, aplicaremos discretización no supervisada para cada variable no discreta, dividiendo el rango de valores en cuatro intervalos de igual longitud. Una vez realizada la discretización, utilizaremos una red Bayesiana discreta como modelo.

De forma similar a la realizada en el trabajo que nos sirve como punto de partida (López-Cruz et al., 2011), aprenderemos diferentes redes Bayseianas según el orden centrífugo y el tipo de nodo. Específicamente, crearemos 9 redes:

- Una red Bayesiana para los segmentos raíz de cada dendrita, i.e. el primer segmento de cada árbol, que constará únicamente de las 5 variables de construcción y el volumen del soma

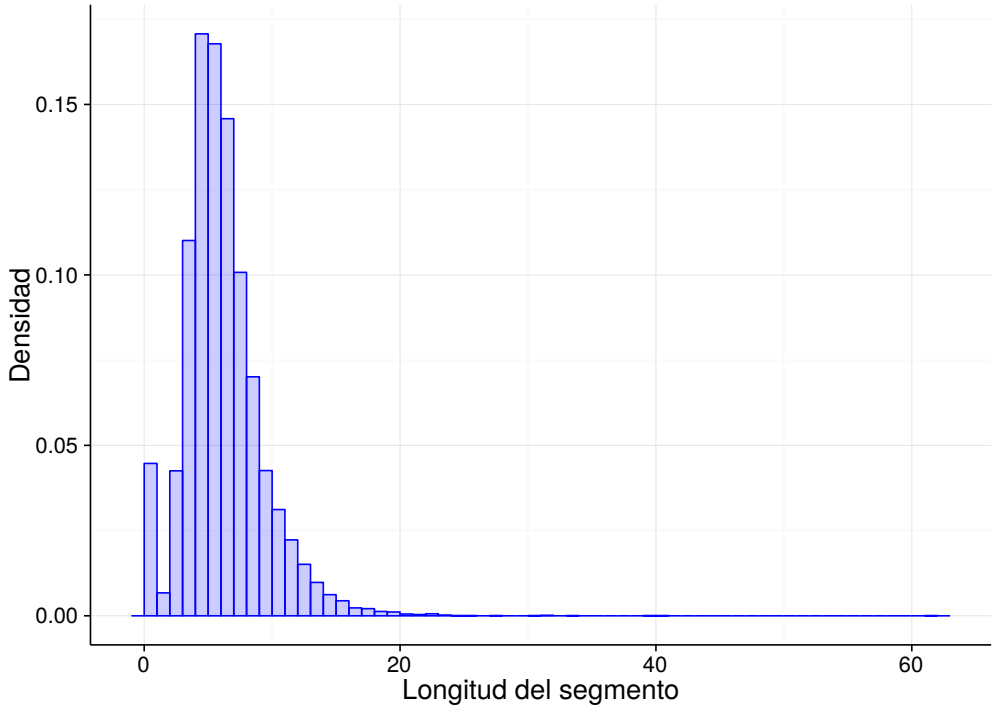


Figura 6.5: Histograma de la distribución de valores para la longitud del segmento actual en elongaciones de orden 2 o más

- Cuatro redes Bayesianas para las elongaciones de orden 0,1,2 y superior, compuestas por hasta 58 variables (incluidas las cinco variables de construcción del siguiente segmento)

- Cuatro redes Bayesianas para las bifurcaciones de orden 0,1,2 y superior, compuestas por hasta 63 variables, con dos conjuntos de variables de construcción, unas para el segmento derecho (el segmento con mayor azimuth) y otras para el izquierdo

6.4.1. Aprendizaje estructural y paramétrico

Tanto el aprendizaje del componente gráfica de las redes Bayesianas, como del componente paramétrico se han realizado a través del paquete de R *bnlearn*, el cual incluye una amplia variedad de algoritmos implementados de forma eficiente y capaz de aprender redes con un gran número de variables en tiempo razonable.

En el aprendizaje de la estructura de la red Bayesiana tenemos dos vías para atajar el problema: Detectar independencias condicionales entre variables mediante tests estadísticos para generar el grafo; o definir una métrica que estime lo bien que

se ajusta una estructura a los datos observados y utilizar un algoritmo de búsqueda que minimice (maximice) dicha cantidad.

En nuestro caso hemos aplicado la segunda aproximación, utilizando con métrica el Criterio de Información Bayesiano (Schwarz et al., 1978), el cual evalúa la verosimilitud de los datos dada la estructura incluyendo una penalización sobre el número de parámetros necesarios para definir una red Bayesiana con dicha estructura. Para la búsqueda se ha utilizado el algoritmo de *Hill Climbing* implementado en el mismo paquete, con un elevado número de reinicios (50) y perturbaciones (20) para conseguir escapar de mínimos locales.

Una vez determinada la estructura, el aprendizaje de las tablas de probabilidad condicionada para cada variable se ha realizando mediante el método de máxima verosimilitud, que en este caso coincide con el recuento de la frecuencia relativa de cada valor de la variable dados los valores de sus padres en el grafo.

6.5. Interpretación

Tras el aprendizaje estructural y paramétrico de las nueve redes Bayesianas del modelo, en esta sección vamos a realizar un breve análisis de las redes obtenidas, pudiendo consultar las nueves redes completas en el apéndice B del documento. Hay que tener en cuenta que, el número de muestras varía según la red, siendo las redes correspondientes a las bifurcaciones de orden 0 y los segmentos raíz las que disponen de menos muestras y la red de elongaciones de orden superior a dos la que más. En las redes, las variables de construcción de la elongación y del segmento derecho de una bifurcación están marcadas en rojo, mientras que las correspondientes al segmento izquierdo están marcadas en azul.

La red de los segmentos raíz es con diferencia la más sencilla (6 variables), esto, unido a que el número de muestras es muy bajo hace que la estructura obtenida contenga únicamente un arco (6.6) entre el azimut y la inclinación. Esta relación es bastante lógica e intuitiva ya que ambos valores determinan la orientación de forma conjunta y de hecho aparecerá en todas o casi todas las redes.

Cómo esperábamos, la complejidad de la red (número de arcos) crece a medida que aumenta el orden centrifugo en las redes aprendidas para bifurcaciones y elongaciones. En general, las redes Bayesianas de nodos de elongación son mucho más complejas que sus contrapartes para bifurcaciones, principalmente debido a que se disponen de menos muestras para el aprendizaje de estas últimas.

Pese a la falta de precisión en la medición de los diámetros, observamos tanto en

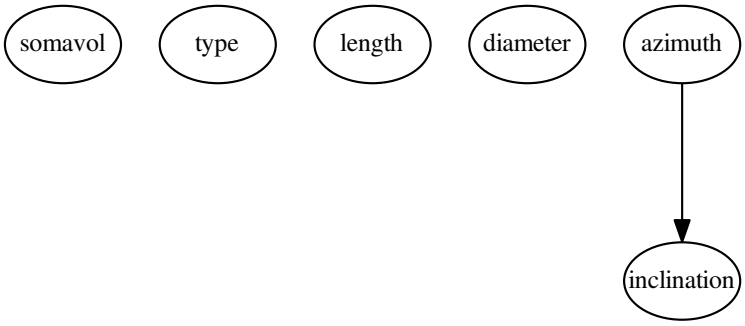


Figura 6.6: Estructura de la red Bayesiana aprendida para los segmentos raíz

las redes de elongación como en las redes de bifurcación como en las de elongación que el diámetro de construcción depende del diámetro del nodo actual para todos los órdenes centrífugos, como se muestra en la figura 6.7.

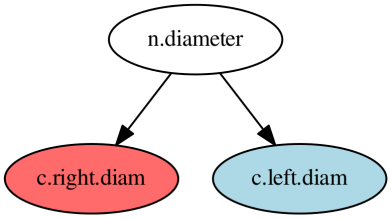
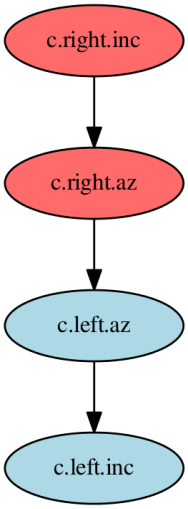


Figura 6.7: Sub red con los diámetros de construcción de una bifurcación (Orden 2)

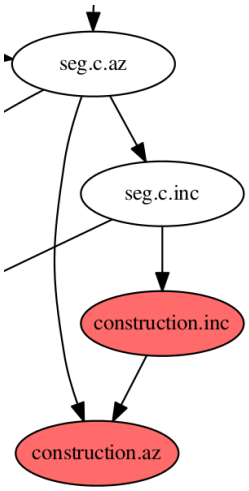
Otro aspecto interesante, y acorde a lo esperado, es la relación entre la orientación de las dos ramas de una bifurcación (figura 6.8a. En este sentido también esperábamos ver cierta dependencia del resto de orientaciones, especialmente las del segmento actual y los segmentos cercanos, como veremos ocurre para las elongaciones. Una vez más, el reducido número de muestras puede explicar la ausencia de esta relación.

En la red correspondiente a elongaciones de orden centrífugo dos y superior, vemos que la orientación del segmento a construir sí que depende de la orientación del segmento actual, la cual es una relación lógica ya que esperamos que el siguiente segmento siga, más o menos, la misma orientación que sus antecesores, sin realizar giros bruscos en la dirección (figura 6.8b).

Por último, es interesante ver que en (casi) todas las redes aprendidas, la longitud del segmento en las variables de construcción es independiente del resto de nodos en



(a) Sub-red con la orientación de las ramas de una bifurcación de orden 2



(b) Sub-red con la orientación del segmento de una elongación de orden 2

la red. Una posible causa es que la longitud del segmento no es una característica propia de la dendrita, que en cierto modo crece de un modo continuo, si no que depende del algoritmo de *tracing* utilizado en la reconstrucción y del criterio del neurocientifico.

7

Simulación

El último componente del trabajo realizado, y objetivo final del mismo, es la creación de arborizaciones dendríticas completas similares a las reales. Para llevar a cabo la simulación obtendremos los valores de las variables de construcción de las redes Bayesianas dada las evidencias mediante inferencia para, progresivamente, hacer crecer cada árbol dendrítico.

En las siguientes secciones describiremos el proceso de forma detallada, mostraremos los resultados (las dendritas virtuales) y realizaremos una breve evaluación de los mismos.

7.1. Descripción del proceso

El proceso de simulación extremo a extremo se compone de 4 pasos secuenciales, muy similares a los expuestos para el aprendizaje del modelo:

- En primer lugar obtenemos el radio del soma mediante muestreo de la distribución log-normal con los parámetros aprendidos durante la modelización. Construimos un soma esférico centrado en el origen de coordenadas con dicho radio
- El siguiente paso es obtener el número de dendritas a partir de la distribución Conway-Maxwell-Poisson ajustada en el capítulo anterior. Si no disponemos del modelo von Mises multivariante para tal número de dendritas, repetimos el muestreo para obtener otro número de dendritas
- Seleccionamos la distribución von Mises multivariante para el número de dendritas seleccionado y obtenemos una muestra de los ángulos entre dendritas.

- Según los ángulos entre dendritas y el radio del soma, colocamos las raíces en el plano $z = 0$ comenzando por el eje x, donde colocaremos la primera dendrita en el punto $(r, 0, 0)$, con r el radio del soma, y las marcamos como activas
- Simularemos el crecimiento de los árboles dendríticos siguiendo un algoritmo similar al descrito en la modelización y ejemplificado en la figura 6.4. En la siguiente sección describiremos en detalle este proceso

7.2. Crecimiento de la arborización

El crecimiento de los árboles dendríticos sigue un proceso muy similar al detallado en la sección , realizando el siguiente procedimiento para cada nodo seleccionado hasta que no quede ningún nodo activo:

- Obtenemos los descriptores (sección 5.3), salvo las variables de construcción, las cuales, obviamente, son desconocidas ya que el descendiente todavía no es conocido
- Discretizamos las evidencias obtenidas utilizando los mismo puntos de corte determinados durante discretización llevada a cabo durante la modelización
- Introducimos las evidencias en la red Bayesiana correspondiente según el tipo de nodo y su orden centrífugo. Mediante inferencia obtenemos una muestra de los valores discretos de las variables de construcción
- Transformamos las variables de construcción discretas en continuas mediante un proceso que se describe en detalle en la subsección 7.2.1
- Construimos los siguientes segmentos a partir de las variables de construcción y, si el tipo del nodo en las variables de construcción no es *terminal*, lo marcamos como activo
- Actualizamos los descendientes del nodo seleccionado en la tabla y lo marcamos como no-activo

Para la el muestreo de las redes Bayesianas utilizaremos el paquete de R *gRain*, que nos permite establecer la evidencia en la red, es decir, asignar valores fijos a ciertos nodos y obtener una muestra completa de la red en función de dicha evidencia, la estructura de la red y las tablas de probabilidad.

7.2.1. Muestreo y obtención de variables de construcción

Para la transformación de los valores discretos obtenidos mediante muestreo de las redes Bayesianas, en valores continuos que nos permitan reconstruir el segmento en cuestión utilizaremos una técnica, basada en los datos reales, que permite realizar este proceso sin realizar ninguna asunción sobre la forma de la distribución de la variable (López-Cruz et al., 2011).

Dada una variable $\mathbf{x}$ y $\mathbf{\Pi}(\mathbf{x})$ el conjunto de padres de x en la red Bayesiana y sus valores discretos obtenidos por inferencia, el método selecciona M muestras en los datos reales discretizados donde el valor de $\mathbf{\Pi}(\mathbf{x})$ y de la propia variable sean los mismos que los obtenidos por inferencia.

En segundo lugar, se seleccionan los valores reales de $\mathbf{x}$ correspondientes a las muestras seleccionadas y se dividen en $\sqrt{M}$ secciones de igual longitud.

Finalmente, se selecciona una de las secciones con probabilidad proporcional al número de muestras reales en dicha sección y se toma la mediana de la sección como valor continuo final. En este trabajo, debido al reducido número de muestras disponibles, trabajaremos con $M = 16$.

La figura 7.1 muestra de forma esquemática el proceso descrito en el párrafo anterior.

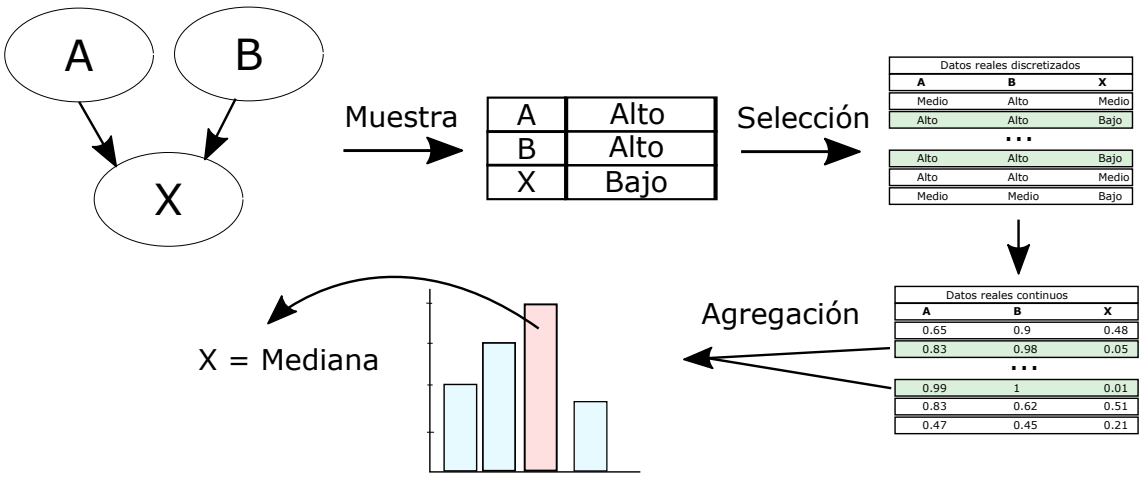
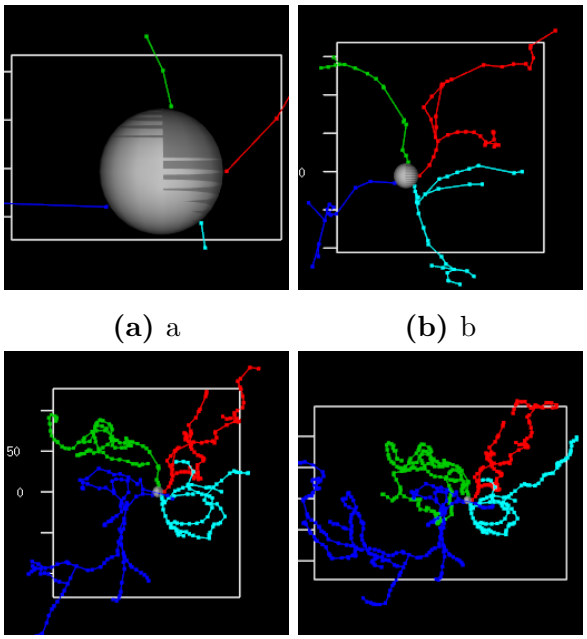


Figura 7.1: Proceso de conversión de valores discretos en reales basado en las muestras originales



(a) a **(b)** b

(c) c **(d)** d

Figura 7.2: Simulación de una arborización dendritica completa con cuatro dendritas

7.3. Resultados y evaluación

Siguiendo el método propuesto en la sección anterior utilizando el modelo aprendido a partir de los datos del archivo Svoboda, se generan 50 arborizaciones basales completas de forma independiente. El proceso de creación de cada neurona es relativamente rápido, tomando cerca de 5 minutos la creación de una arborización completa, de los cuales, la mayoría del tiempo de ejecución se emplea en la inferencia.

Utilizando la librería *RGL*, representamos en un entorno 3D interactivo las neuronas creadas. Durante la simulación se toman capturas del estado de la arborización en intervalos regulares, permitiendo observar el crecimiento de las dendritas. La figura 7.2 contiene una secuencia temporal obtenida mediante simulación de una neurona con 4 dendritas.

Para evaluar el modelo se comparan las 50 arborizaciones simuladas con las 38 reconstrucciones reales a partir de las cuales se construyó dicho modelo. En primer lugar se realiza una inspección visual de las simulaciones a través de las representaciones 3D, detectando errores en la simulación en comparación con las reconstrucciones reales. Finalmente, se reliza una evaluación estadística comparando la distribución de una serie de variables denominadas emergentes: variables no

incluidas en el modelo que son resultado directo de la morfogénesis (Ascoli et al., 2001).

7.3.1. Visual

A partir del análisis visual de las simulaciones frente a las reconstrucciones originales podemos extraer las siguientes conclusiones:

- En general las dendritas generadas en las simulaciones presentan un patrón de elongación mucho más propenso al giro que las reconstrucciones reales. Las dendritas generadas comienzan a curvarse formando estructuras muy complejas y poco realistas, incluso llegando a crecer de vuelta hacia el soma.
- Por otra parte, obviando la elongación, los patrones de ramificación y el número de ramificaciones parece similar en ambos casos.
- Las dendritas simuladas parecen no tener en cuenta la posición del resto de dendritas en la neurona. Esto concuerda con la ausencia de dependencias entre la orientación de los segmentos cercanos y la orientación de construcción
- El crecimiento de la arborización en cada iteración parece ser acorde con lo esperado, en general la parte próxima al soma se desarrolla al inicio de la simulación de forma homogénea, mientras que el crecimiento en profundidad de cada árbol parece seguir un patrón menos regular

7.3.2. Variables emergentes

El análisis estadístico de la distribución de las variables emergentes en las arborizaciones simuladas frente a la distribución en las reconstrucciones reales es una técnica ya contemplada en la literatura, tanto en el caso univariante con la aplicación de test Kolgomorov-Smirnov y Wilcoxon-Rank o el cálculo la divergencia KL, como en el caso multivariante con el uso de la estimación de la divergencia KL multivariable (Lin y Li (2013), López-Cruz et al. (2011) y Ascoli y Krichmar (2000)).

En este trabajo nos centraremos en la evaluación de variables emergentes relativas a la célula al completo, dejando la evaluación de árboles individuales para futuros refinamientos del modelo. Se reduce la selección a cuatro variables representativas: Número de bifurcaciones y terminaciones, longitud dendrítica total de la neurona y volumen cúbico (*box volume*).

Aplicamos los test de Kolgomorov-Smirnov y Wilcoxon-Rank para cada variable emergente, comparando los valores de las reconstrucciones reales contra los de las simulaciones. Para todas las variables, ningún test rechaza la suposición de que ambos conjuntos provienen de la misma distribución subyacente, aunque con un p-valor muy cercano al rechazo ($\approx 0,15$). Aunque, una vez más, el reducido número de muestras hace que la confianza en los resultados estadísticos de esta sección sea baja o nula. En el futuro sería aconsejable contar con un número más elevado de muestras y simulaciones para realizar la evaluación.

8

Conclusiones y trabajo futuro

8.1. Estadística circular multivariante

El trabajo realizado a introducido una reformulación de la función de pseudo-verosimilitud optimizada desde un punto de vista computacional para la distribución von Mises multivariante, reducidendo el tiempo de ajuste de forma significativa y permitiendo el uso de la distribución en escenarios de alta dimensionalidad.

Además, se ha propuesto una distancia circular multivariante, que junto con la aproximación por vecinos más próximos de la distancia de Kullback-Leibler permite la introducción de una medida de comparación estadística más robusta para datos circulares.

Por último, se ha propuesto una penalización L_1 generalizada para esta misma distribución, comprobando como su aplicación, especialmente en casos dónde el número de muestras es muy limitado, mejora los resultados del ajuste.

Como líneas de estudio futuro, quedan abiertos varios frentes de trabajo:

- Es necesario estudiar en profundidad cómo afecta la escala los valores de la matriz de penalización al resultado final
- Los resultados preliminares parecen indicar que el parámetro de concentración $\boldsymbol{\kappa}$ y la matriz $\boldsymbol{\Lambda}$ actúan de manera conjunta sobre la distribución, siendo necesario ahondar en este comportamiento desde un enfoque más teórico
- El muestreador de Gibbs implementado en el paquete *mvCircular* necesita ser estudiado en profundidad. La idea de utilizar un método adaptativo para reducir el tiempo de convergencia, así como la autocorrelación, parece prometedora

Por último destacar que como resultado de este trabajo se presentará un artículo

en la conferencia CAEPIA 2015 y se publicará en el volumen especial de la conferencia en *Lecture Notes on Artificial Inteligence*. Se espera que el paquete *mvCircular* esté disponible públicamente antes de final de año.

8.2. Reconstrucción de la arborización dendrítica basal

En este trabajo hemos propuesto la primera modelización basada en datos que aborda el problema de la reconstrucción de la arborización dendrítica basal teniendo en cuenta las interacciones entre los distintos árboles.

A diferencia de modelos anteriores, nuestras simulaciones reproducen la tortuosidad de las ramas originales, alejándonos de las reconstrucciones estructurales formadas por segmentos completamente rectos que si bien son estadísticamente similares, visualmente ofrecen una sensación completamente artificial

Aunque el resultado obtenido presenta fallos evidentes, creemos que éstos son subsanables con las siguientes modificaciones a realizar en el futuro:

- La discretización de las variables continuas, en especial de las relativas a la orientación, introduce un error que si bien puede ser pequeño, al reproducirse un elevado número de veces durante la elongación, produce resultados indeseados, como por ejemplo una tortuosidad demasiado elevada. Una línea de trabajo futura es el uso de redes Bayesianas capaces de trabajar con variables continuas no Gaussianas, como por ejemplo las redes con mixturas de polinomios; otra opción es el uso de técnicas más sofisticadas de discretización no supervisada como la propuesta en Friedman et al. (1996), que la realiza durante el aprendizaje de la red.

- La ausencia de ciertas relaciones entre variables de la red, lógicas a priori, puede ser un indicador de que el número de muestras con las que se realiza el aprendizaje es insuficiente, por lo que en el futuro será necesario evaluar la inclusión de reconstrucciones de distintas fuentes así como la reducción del número de redes a entrenar según el orden centrífugo, por ejemplo fusionando las redes para orden 0,1 y 2

- La ausencia de una referencia externa a la hora de medir los ángulos entre dendritas puede ser una fuente de imprecisión en la modelización. El método actual únicamente es consistente si asumimos simetría radial de la neurona,

propiedad supuesta en algunos casos, pero sin fundamento teórico. Es necesario establecer un criterio de ordenación consistente y global

Finalmente, el objetivo a largo plazo de este trabajo es su integración como parte del proyecto *Human Brain Project*. Con el fin de facilitar su uso a usuarios no expertos en el ámbito de la informática, se elaborará una interfaz gráfica web que permitirá realizar todos los pasos de la construcción del modelo y posterior simulación de forma sencilla y clara para un neurocientífico.

Apéndices

Orientación de las raíces dendríticas

A continuación se muestran las gráficas con el ajuste de la distribución von Mises multivariante para los casos con 3, 4 y 5 dendritas.

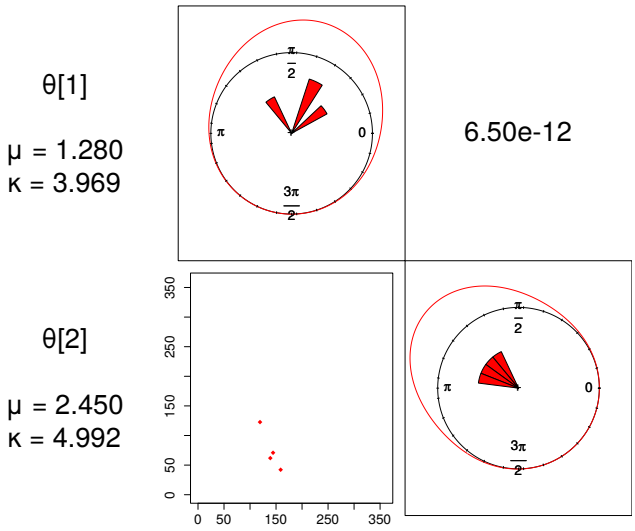


Figura 1: Ajuste de la orientación de las dendritas basales a una distribución von Mises multivariante para tres dendritas (KL = 0.39)

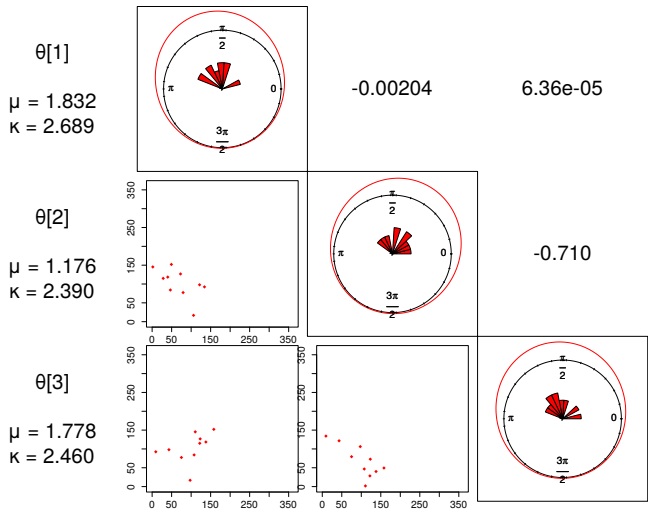


Figura 2: Ajuste de la orientación de las dendritas basales a una distribución von Mises multivariante para cuatro dendritas (KL = 0.81)

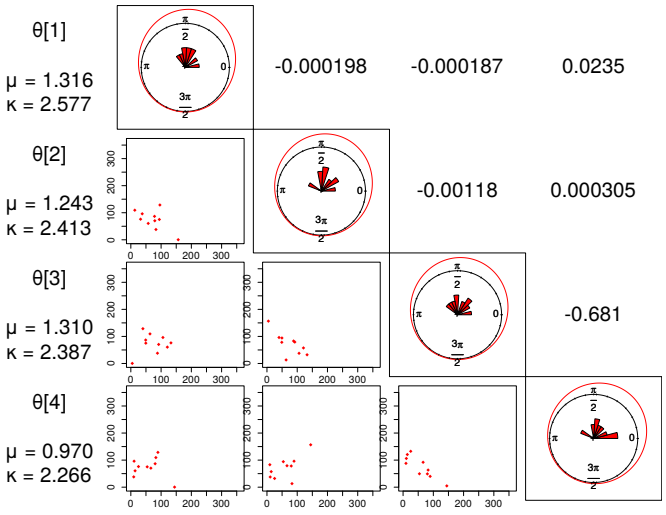


Figura 3: Ajuste de la orientación de las dendritas basales a una distribución von Mises multivariante para cinco dendritas (KL = 0.60)

Redes Bayesianas del modelo

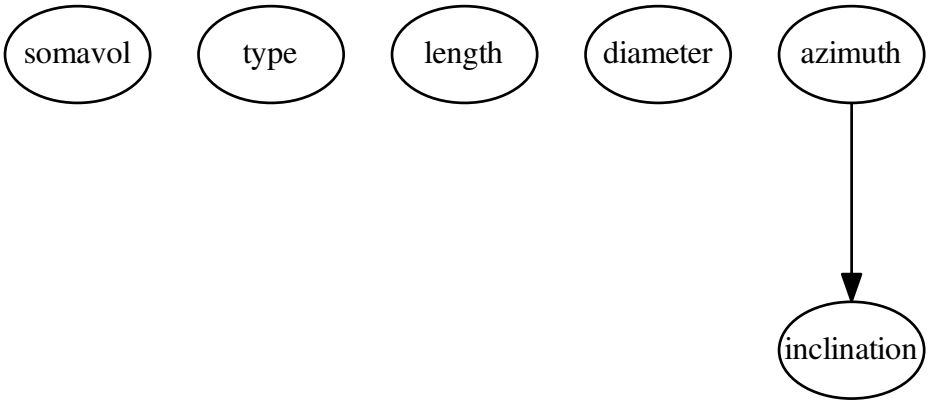


Figura 4: Estructura de la red Bayesiana aprendida para los segmentos raíz de la arborización dendrítica

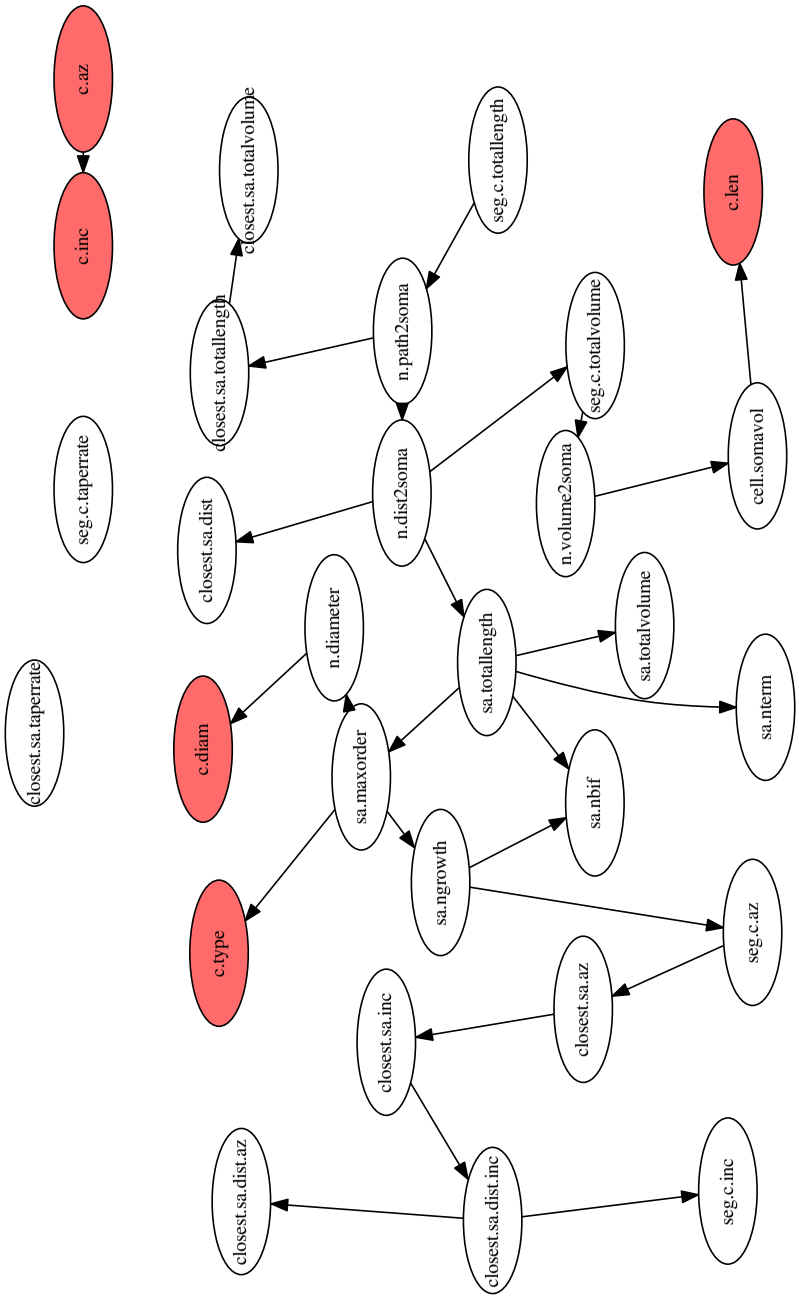


Figura 5: Estructura de la red Bayesiana aprendida para los nodos de elongación con orden centrífugo 0 en la arborización dendrítica

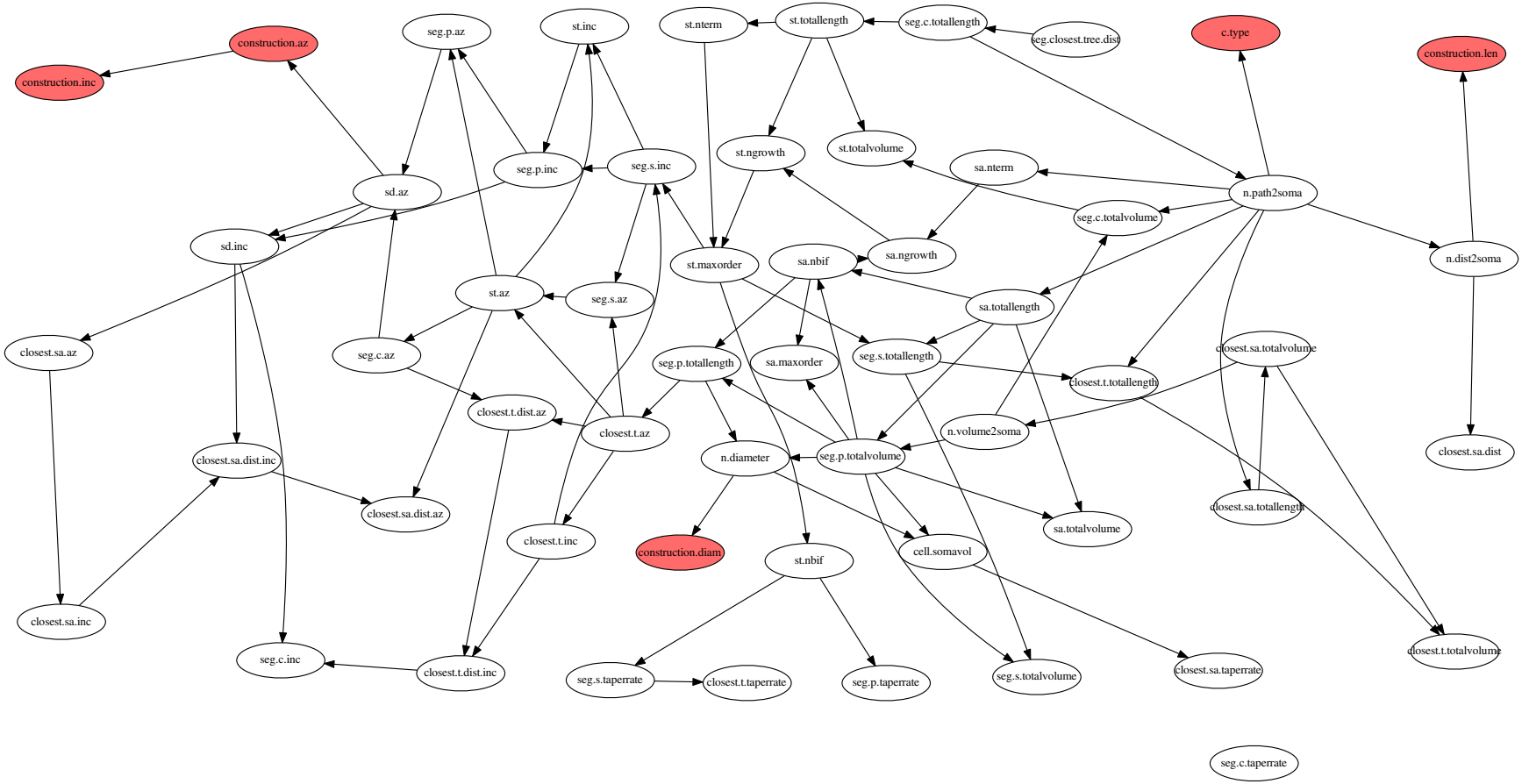


Figura 6: Estructura de la red Bayesiana aprendida para los nodos de elongación con orden centrífugo 1 en la arborización dendrítica

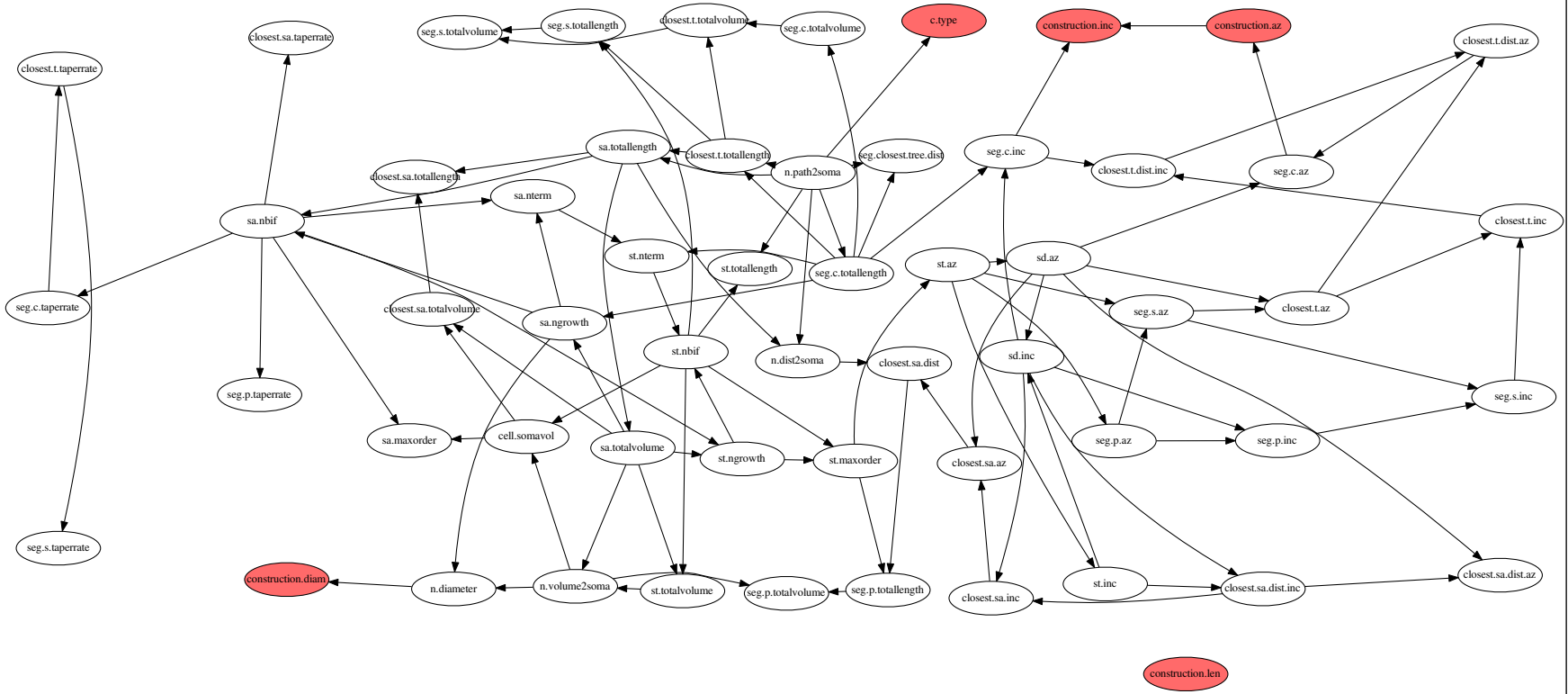


Figura 7: Estructura de la red Bayesiana aprendida para los nodos de elongación con orden centrífugo 2 en la arborización dendrítica

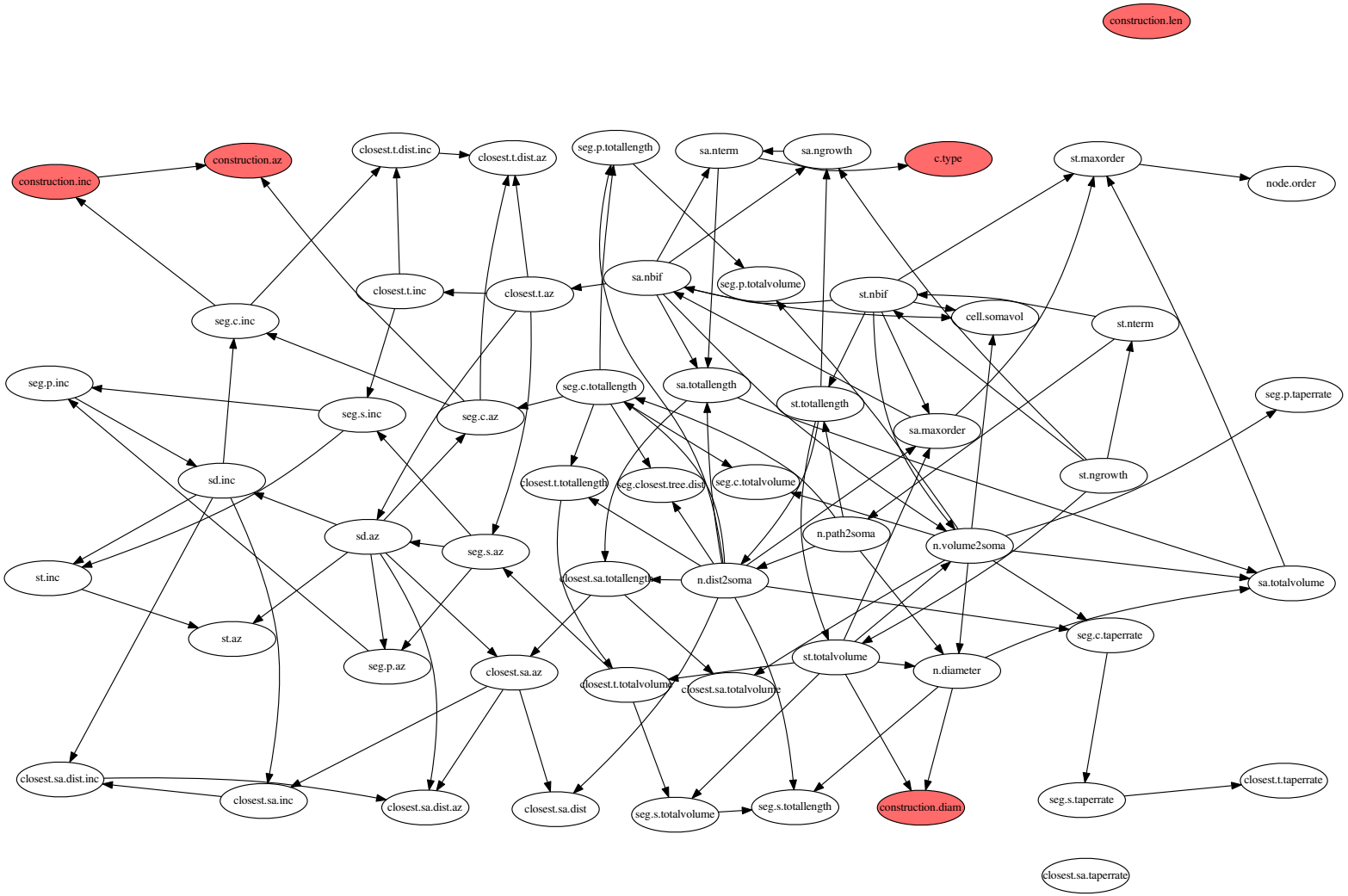


Figura 8: Estructura de la red Bayesiana aprendida para los nodos de elongación con orden centrífugo superior a 2 en la arborización dendrítica

Figura 9: Estructura de la red Bayesiana aprendida para los nodos de elongación con orden centrífugo 0 en la arborización dendrítica

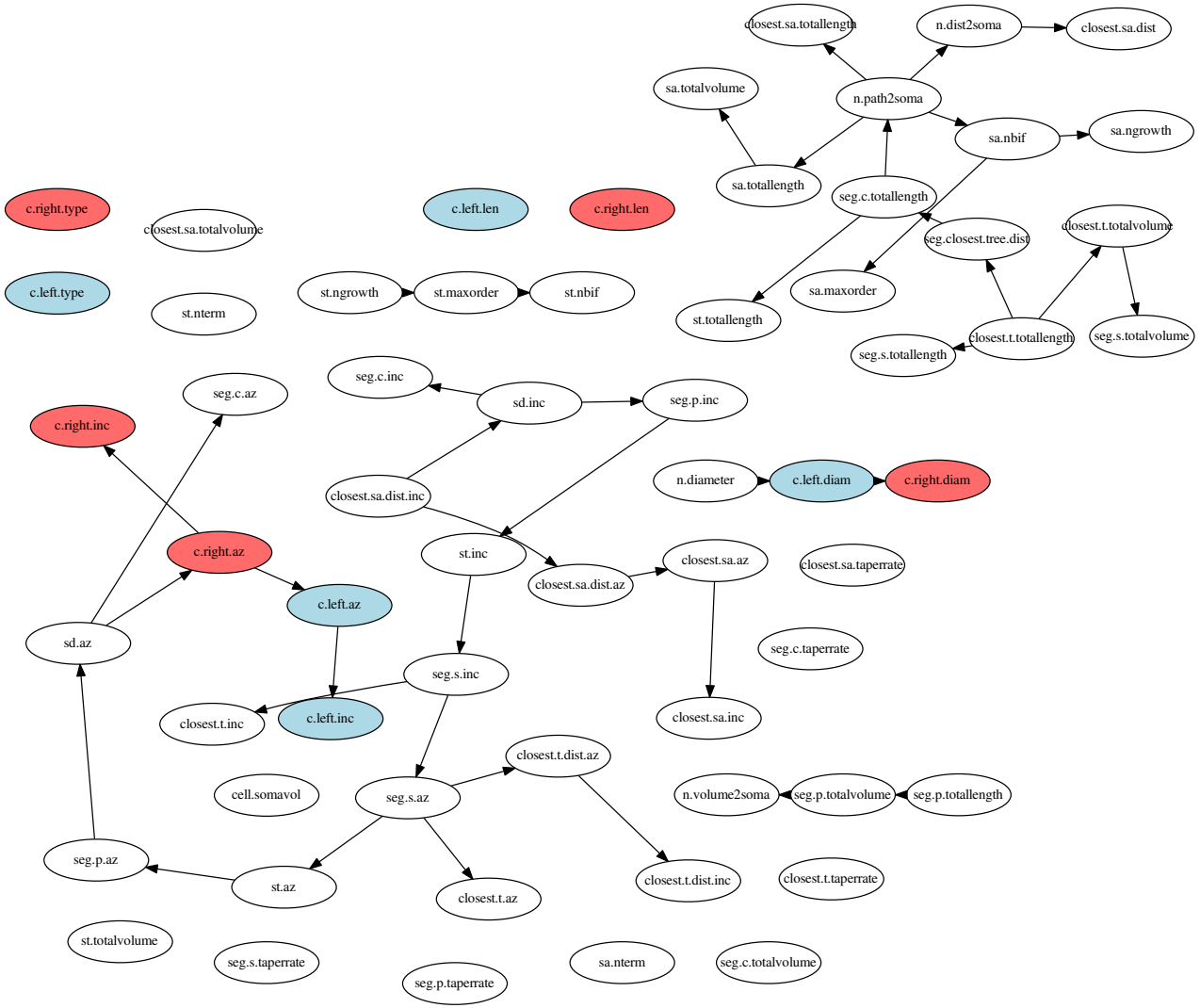


Figura 10: Estructura de la red Bayesiana aprendida para los nodos de elongación con orden centrífugo 1 en la arborización dendrítica

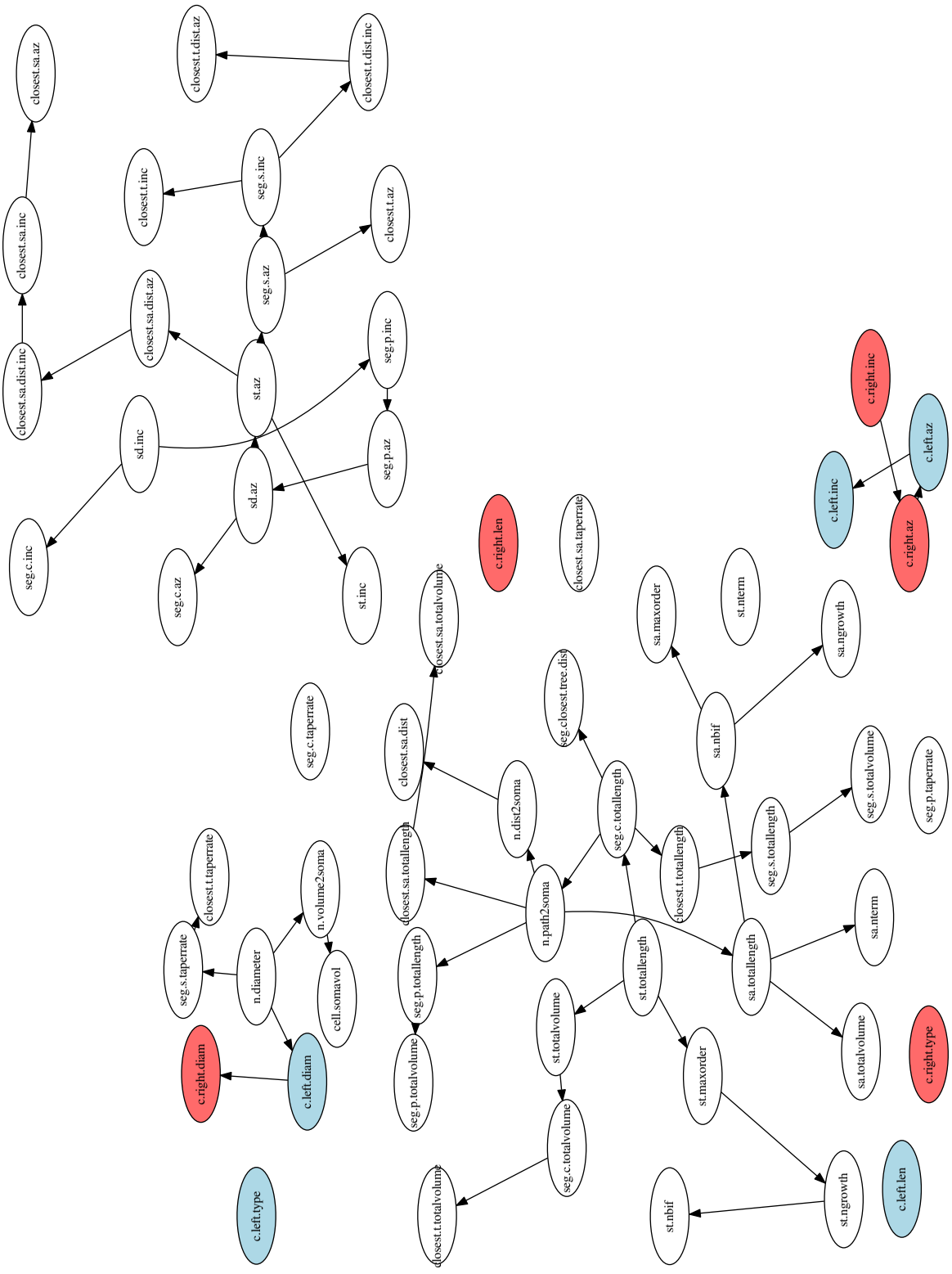


Figura 11: Estructura de la red Bayesiana aprendida para los nodos de elongación con orden centrífugo 2 en la arborización dendrítica

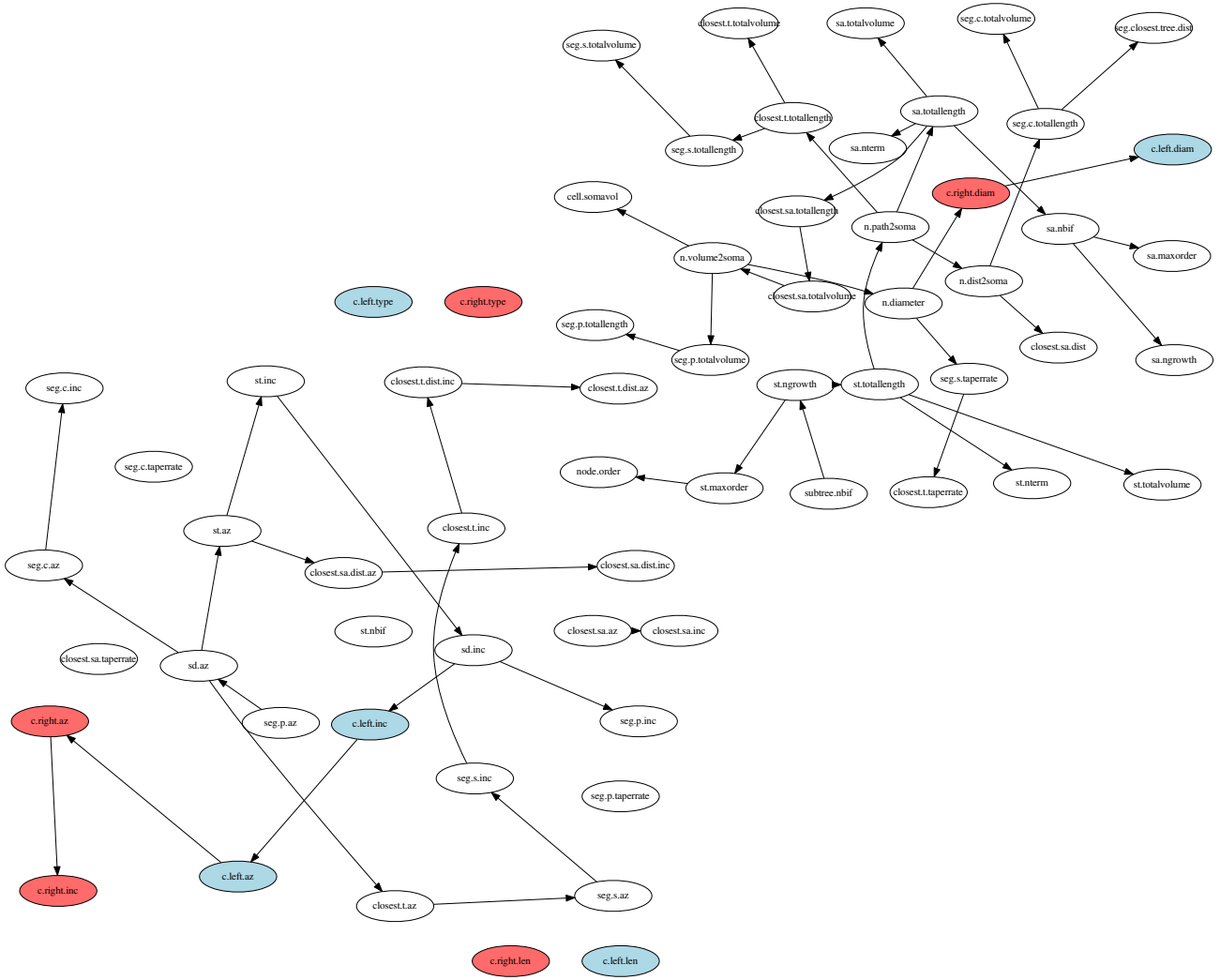


Figura 12: Estructura de la red Bayesiana aprendida para los nodos de elongación con orden centrífugo superior a 2 en la arborización dendrítica

Bibliografía

Giorgio A Ascoli y Jeffrey L Krichmar. L-neuron: a modeling tool for the efficient generation and parsimonious description of dendritic morphology. *Neurocomputing*, 32: 1003–1011, 2000. 38, 69

Giorgio A Ascoli, Jeffrey L Krichmar, Slawomir J Nasuto, y Stephen L Senft. Generation, description and storage of dendritic morphology data. *Philosophical Transactions of the Royal Society of Biological Sciences*, 356(1412):1131–1145, 2001. 40, 69

Giorgio A Ascoli, Duncan E Donohue, y Maryam Halavi. Neuromorpho. org: a central resource for neuronal morphologies. *The Journal of Neuroscience*, 27(35):9247–9251, 2007. 31, 41

Stephen Becker. L-BFGS-B-C l-bfgs-b, converted from fortran to c. `https://github.com/stephenbeckr/L-BFGS-B-C`, 2014. 23

DJ Best y Nicholas I Fisher. Efficient simulation of the von Mises distribution. *Applied Statistics*, pages 152–157, 1979. 12

Stephen P Brooks y Andrew Gelman. General methods for monitoring convergence of iterative simulations. *Journal of Computational and Graphical Statistics*, 7(4):434–455, 1998. 15

Dmitri B Chklovskii y Armen Stepanyants. Power-law for axon diameters at branch point. *BMC Neuroscience*, 4(1):18, 2003. 46

Mary Kathryn Cowles y Bradley P Carlin. Markov chain monte carlo convergence diagnostics: a comparative review. *Journal of the American Statistical Association*, 91(434): 883–904, 1996. 14

Hermann Cuntz, Friedrich Forstner, Alexander Borst, y Michael Häusser. The trees toolbox—probing the basis of axonal and dendritic branching. *Neuroinformatics*, 9(1): 91–96, 2011. 39

BIBLIOGRAFÍA

Jens P Eberhard, Alexander Wanner, y Gabriel Wittum. Neugen: a tool for the generation of realistic morphology of cortical neurons and neural networks in 3d. *Neurocomputing*, 70(1):327–342, 2006. 38

Nir Friedman, Moises Goldszmidt, et al. Discretizing continuous attributes while learning bayesian networks. In *ICML*, pages 157–165, 1996. 72

Alan E Gelfand y Adrian FM Smith. Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association*, 85(410):398–409, 1990. 11, 13

Andrew Gelman y Donald B Rubin. Inference from iterative simulation using multiple sequences. *Statistical Science*, pages 457–472, 1992. 14

Andrew Gelman y Kenneth Shirley. Inference from simulations and monitoring convergence. *Handbook of Markov Chain Monte Carlo*, pages 163–174, 2011. 14

Jacob R Glaser y Edmund M Glaser. Neuron imaging with neurolucida—a pc-based system for image combining microscopy. *Computerized Medical Imaging and Graphics*, 14(5):307–317, 1990. 40

Padraig Gleeson, Volker Steuber, y R Angus Silver. neuroconstruct: a tool for modeling networks of neurons in 3d space. *Neuron*, 54(2):219–235, 2007. 39

Bruce P Graham y Arjen van Ooyen. Transport limited effects in a model of dendritic branching. *Journal of Theoretical Biology*, 230(3):421–432, 2004. 57

MGL Gustafsson, DA Agard, JW Sedat, et al. I5m: 3d widefield light microscopy with better than 100nm axial resolution. *Journal of Microscopy*, 195(1):10–16, 1999. 40

W Keith Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109, 1970. 11

Bob Jacobs, Matthew Schall, Melissa Prather, Elisa Kapler, Lori Driscoll, Serapio Baca, Jesse Jacobs, Kevin Ford, Marcy Wainwright, y Melinda Treml. Regional dendritic and spine variation in human cerebral cortex: a quantitative golgi study. *Cerebral Cortex*, 11(6):558–571, 2001. 31, 32

Ian Jolliffe. *Principal component analysis*. Wiley Online Library, 2002. 45

Randal A Koene, Betty Tijms, Peter van Hees, Frank Postma, Alexander de Ridder, Ger JA Ramakers, Jaap van Pelt, y Arjen van Ooyen. Netmorph: a framework for the stochastic generation of large scale neuronal networks with realistic neuron morphologies. *Neuroinformatics*, 7(3):195–210, 2009. 39, 57

Daphne Koller y Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009. 59

Vaishali A Kulkarni y Bonnie L Firestein. The dendritic tree and brain disorders. *Molecular and Cellular Neuroscience*, 50(1):10–20, 2012. 38

Richard A Levine y George Casella. Optimizing random scan gibbs samplers. *Journal of Multivariate Analysis*, 97(10):2071–2100, 2006. 15

Xianghong Lin y Zhiqiang Li. Generation and analysis of 3d virtual neurons using genetic regulatory network model. In *Advances in Neural Networks–ISNN 2013*, pages 9–18. Springer, 2013. 39, 69

William A Link y Mitchell J Eaton. On thinning of chains in mcmc. *Methods in Ecology and Evolution*, 3(1):112–115, 2012. 15

Pedro L López-Cruz, Concha Bielza, Pedro Larrañaga, Ruth Benavides-Piccione, y Javier DeFelipe. Models and simulation of 3d neuronal dendritic trees using bayesian networks. *Neuroinformatics*, 9(4):347–369, 2011. 39, 46, 60, 67, 69

Artur Luczak. Spatial embedding of neuronal trees modeled by diffusive growth. *Journal of Neuroscience Methods*, 157(1):132–141, 2006. 39

Sergio Luengo Sánchez. Clustering basado en redes bayesianas con predictoras continuas: aplicaciones en neurociencia. Master's thesis, ETSI Informatica, 2014. 45

Guy Major, Matthew E Larkum, y Jackie Schiller. Active properties of neocortical pyramidal neuron dendrites. *Annual Review of Neuroscience*, 36:1–24, 2013. 37

Kanti V Mardia y Jochen Voss. Some fundamental properties of a multivariate von Mises distribution. *Communications in Statistics-Theory and Methods*, 43(6):1132–1144, 2014. 12, 13, 17

Kanti V Mardia, Gareth Hughes, y Charles C Taylor. Efficiency of the pseudolikelihood for multivariate normal and von Mises distributions. *Preprint*, 2007. 20

Kanti V Mardia, Gareth Hughes, Charles C Taylor, y Harshinder Singh. A multivariate von Mises distribution with applications to bioinformatics. *Canadian Journal of Statistics*, 36(1):99–109, 2008. 4, 14, 19, 20, 25

Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, y Edward Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 1953. 11

BIBLIOGRAFÍA

José Luis Morales y Jorge Nocedal. Remark on .ªlgorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound constrained optimization". *ACM Transactions on Mathematical Software (TOMS)*, 38(1):7, 2011. 22

Fernando Pérez-Cruz. Kullback-leibler divergence estimation of continuous distributions. In *IEEE International Symposium on Information Theory, 2008*, pages 1666–1670. IEEE, 2008. 26

Tingwei Quan, Ting Zheng, Zhongqing Yang, Wenxiang Ding, Shiwei Li, Jing Li, Hang Zhou, Qingming Luo, Hui Gong, y Shaoqun Zeng. Neurogps: Automated localization of neurons for brain circuits using l1 minimization model. *Scientific Reports*, 3, 2013. 45, 54

Adrian E Raftery y Steven Lewis. How many iterations in the gibbs sampler. *Bayesian Statistics*, 4(2):763–773, 1992. 14

Santiago Ramón y Cajal. *Contribucion al conocimiento de la neuroglia del cerebro humano*. 1913. 1

Narges Razavian, Hetunandan Kamisetty, y Christopher James Langmead. The von Mises graphical model: Regularized structure and parameter learning. *Tech Rep CMU-CS-11-108, Carnegie Mellon University, Department of Computer Science*, 2011. 14, 22, 23, 25, 26

Luis Rodriguez-Lujan, Pedro Larrañaga, y Concha Bielza. Regularized multivariate von Mises distribution. *Lecture Notes in Artificial Intelligence*, 2015. 23, 29, 33, 34, 56

Gideon Schwarz et al. Estimating the dimension of a model. *The Annals of Statistics*, 6 (2):461–464, 1978. 62

Gordon MG Shepherd y Karel Svoboda. Laminar and columnar organization of ascending excitatory projections to layer 2/3 pyramidal neurons in rat barrel cortex. *The Journal of Neuroscience*, 25(24):5670–5679, 2005. 41

Galit Shmueli, Thomas P Minka, Joseph B Kadane, Sharad Borle, y Peter Boatwright. A useful distribution for fitting discrete data: revival of the conway–maxwell–poisson distribution. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 54 (1):127–142, 2005. 55

András Stelescu, János Sümegi, Ildikó Wéber, András Birinyi, y Ervin Wolf. Somatodendritic morphology and dendritic signal transfer properties differentiate between fore- y hindlimb innervating motoneurons in the frog rana esculenta. *BMC Neuroscience*, 13 (1):68, 2012. 45, 54

Kean Ming Tan, Palma London, Karthik Mohan, Su-In Lee, Maryam Fazel, y Daniela Witten. Learning graphical models with hubs. *The Journal of Machine Learning Research*, 15(1):3297–3331, 2014. 22

Engin Türetken, Germán González, Christian Blum, y Pascal Fua. Automated reconstruction of dendritic and axonal trees by global optimization with geometric priors. *Neuroinformatics*, 9(2-3):279–302, 2011. 40

Arjen Van Ooyen. Using theoretical models to analyse neural development. *Nature Reviews Neuroscience*, 12(6):311–326, 2011. 38

Arjen Van Ooyen, Bruce P Graham, y Ger JA Ramakers. Competition for tubulin between growing neurites during development. *Neurocomputing*, 38:73–78, 2001. 57

Ciyou Zhu, Richard H Byrd, Peihuang Lu, y Jorge Nocedal. Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization. *ACM Transactions on Mathematical Software (TOMS)*, 23(4):550–560, 1997. 22

Frederic Zubler y Rodney Douglas. A framework for modeling the growth and development of neurons and networks. *Frontiers in Computational Neuroscience*, 3, 2009. 39