

Trabajo de Fin de Máster
Máster Universitario en Ingeniería Industrial

**Diseño y entrenamiento de una red neuronal para
simular un modelo de remodelación ósea basado
en poblaciones celulares**

Autor: Ana Carrasco Mantis

Tutores: José Luis Calvo Gallego y Francisco Javier Martínez Reina

Dpto. Ingeniería Mecánica y Fabricación
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2020



Trabajo de Fin de Máster
Máster Universitario en Ingeniería Industrial

Diseño y entrenamiento de una red neuronal para simular un modelo de remodelación ósea basado en poblaciones celulares

Autor:

Ana Carrasco Mantis

Tutores:

José Luis Calvo Gallego

Profesor Ayudante Doctor

Y

Francisco Javier Martínez Reina

Profesor Titular de Universidad

Dpto. de Ingeniería Mecánica y Fabricación

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2020

Trabajo de Fin de Máster: Diseño y entrenamiento de una red neuronal para simular un modelo de remodelación ósea basado en poblaciones celulares

Autor: Ana Carrasco Mantis

Tutores: José Luis Calvo Gallego y Francisco Javier Martínez Reina

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2020

El Secretario del Tribunal

A mis padres, Antonio y Encarni

A mi familia y amigos

A mis profesores

AGRADECIMIENTOS

El Máster Universitario en Ingeniería Industrial ha sido una oportunidad de avance y mejora de lo estudiado en años anteriores, de nuevas metas y de superación de miedos. Una etapa que nos ha hecho experimentar emociones muy extremas y por la que hemos sido capaces de ofrecer lo mejor de nosotros mismos. Otro pasito más hacia un esperado futuro laboral.

En este trabajo de fin de máster he descubierto una parte de mí que ve la programación computacional como todo un reto. Sin duda, lo mejor de esta experiencia es haberla realizado con unos tutores a los que admiro, que me han hecho el camino mucho más fácil, siempre me han atendido en tutorías de una manera muy dispuesta y me han permitido seguir aprendiendo de la biomecánica. Muchas gracias, Javier y José Luis.

Quiero agradecer también a mi querida abuela por rezarle a sus santos cada vez que me presentaba a un examen y, por supuesto, a mis maravillosos padres por todo el ánimo transmitido, por cada abrazo, por acompañarme a la gran mayoría de los exámenes, por cada sonrisa que me hacían sacar, por ser los tres como uno... por infinitas cosas más para las que no hay suficiente espacio en estas páginas.

Gracias a todas esas personas, he descubierto que todo el mundo tiene un gran potencial y que el primer paso para exteriorizarlo es confiar en nosotros mismos. Durante el camino hacia nuestras metas, es esencial contar con personas que nos enseñen el valor del esfuerzo y nos ayuden a avanzar siempre hacia mejor, con la actitud más positiva posible.

ÍNDICE DE TEMAS

1. Introducción	17
1.1 Justificación del trabajo	17
1.2 Aplicaciones de las Redes Neuronales Artificiales	18
2. Biología ósea. Patología: Osteoporosis	19
2.1 Hueso: funciones, tipos y composición	19
2.2 Remodelación ósea	21
2.2.1 Poblaciones celulares	22
2.2.2 Unidad multicelular básica, <i>BMU</i>	24
2.2.3 Factores bioquímicos	25
2.3 Osteoporosis	26
2.3.1 Definición, tipos y factores de riesgo	26
2.3.2 Impacto y fracturas más frecuentes	27
2.3.3 Osteoporosis postmenopáusica	27
3. Modelo	29
3.1 Descripción del modelo	29
3.1.1 Porosidad vascular y superficie específica	35
3.1.2 Funciones de Hill	35
3.1.3 Regulación geométrica	37
3.1.4 Regulación mecánica	38
3.1.5 Concentración de los factores bioquímicos	38
3.1.6 Mineralización	39
3.1.7 Inclusión de la osteoporosis y del denosumab	41

3.1.8 Sistema de ecuaciones diferenciales	42
3.2 Análisis de sensibilidad	43
3.3 Proceso de automatización	55
4. Redes Neuronales	57
4.1. Neurobiología	57
4.2 Red Neuronal Artificial	58
4.3 Tipos de entrenamiento	59
4.4 Validación cruzada	61
4.4.1 Datos necesarios	61
4.4.2 Configuración de la red	62
4.4.3 Entrenamiento y test	63
5. Resultados	65
5.1 Procedimiento de obtención de los resultados	65
5.2 Dos variables	66
5.2.1 Entrenamiento desacoplado	66
5.2.2 Entrenamiento exhaustivo	67
5.2.3 Entrenamiento pseudoaleatorio	69
5.2.4 Entrenamiento aleatorio	71
5.2.5 Elección del tipo de entrenamiento	73
5.3 Tres variables	75
5.4 Cuatro variables	78
5.5 Cinco variables	83
5.6 Análisis del coeficiente de variación para dos, tres, cuatro y cinco variables	90
5.6.1 Parámetros de la red	91
5.6.1.1 Influencia del número de capas	91
5.6.1.2 Influencia del número de neuronas	92
5.6.1.3 Influencia de la función de entrenamiento	93

5.6.1.4 Selección de la mejor opción	94
5.7 Importancia de la distribución de los puntos de entrenamiento	96
6. Conclusiones y trabajos futuros	99
7. Bibliografía	101
8. Anexos	105
8.1 Creación de la matriz de entrada al modelo matemático	105
8.2 Programa principal	110
8.3 Llamada al programa principal	134
8.4 Creación de las matrices de entrada y objetivo de la red neuronal	136
8.5 Creación de submatrices	139
8.6 Búsqueda de puntos de entrenamiento	140
8.7 Validación cruzada	142
8.8 Coeficiente de variación	146

ÍNDICE DE TABLAS

Tabla 2.2.1. Factores que influyen en la remodelación ósea	25
Tabla 3.1.1. Descripción de las cantidades dinámicas del modelo	29
Tabla 3.1.2. Descripción de las constantes del modelo	30
Tabla 3.1.3. Constantes del modelo, valores numéricos y unidades	32
Tabla 5.6.1. Coeficiente de variación	95

1. INTRODUCCIÓN

“Hay una fuerza motriz más poderosa que el vapor, la electricidad y la energía atómica: la voluntad”.

-Albert Einstein-

En el siguiente trabajo, se presenta un modelo matemático que trata de simular una de las enfermedades óseas de mayor impacto a nivel mundial, la osteoporosis. Dicho modelo cuenta con gran variedad de parámetros de entrada y a través de las simulaciones realizadas, se trata de estudiar el efecto que producen en los parámetros de salida. Para ello, se necesitan realizar gran número de simulaciones, con el consecuente coste computacional y de tiempo.

Por ello, surge la idea de utilizar la herramienta de las redes neuronales artificiales. Disponemos de datos de entrada y de salida del modelo de simulaciones previamente realizadas. Con esta información, se pretende construir una red neuronal que mediante un aprendizaje sea capaz de aproximar la salida del modelo, con el objetivo de probar en un futuro múltiples combinaciones de parámetros de entrada sin necesidad de realizar los cálculos computacionales.

Finalmente, en el diseño de la red hay que tener en cuenta aspectos como el tipo de entrenamiento, el número de capas, el número de neuronas en cada capa y la función de entrenamiento con el fin de obtener una red cuyos datos de salida sean lo más parecido posible a los de salida del modelo matemático.

1.1 Justificación del trabajo

A continuación, se muestra la justificación de la elección de las redes neuronales artificiales como una herramienta para la resolución de problemas (Izaurieta & Saavedra, 2000).

El cerebro humano es un extraordinario sistema de cálculo capaz de resolver problemas no lineales y muchas operaciones a la vez. Su eficacia radica en la plasticidad que posee, es decir, es capaz de procesar patrones tridimensionales en un breve periodo de tiempo, de responder correctamente a estímulos no percibidos, etc.

Las neuronas son aproximadamente 5 órdenes de magnitud más lentas que una computadora. Sin embargo, dicha lentitud se compensa con el elevado número de interconexiones. El cerebro humano cuenta con alrededor de 10^{10} neuronas y 10^{13} conexiones neuronales. Por este motivo, la eficiencia energética del cerebro es 10^{10} veces mayor que la de las computadoras.

Las redes neuronales artificiales son capaces de adquirir conocimiento (aprender) a través de la experiencia, poseen la plasticidad antes mencionada, se adaptan bien a los cambios dado que son capaces de ofrecer buenas respuestas a entradas con pequeñas variaciones, buena tolerancia a fallos, comportamiento no-lineal, capacidad de considerar por separado las características de un objeto, etc.

1.2 Aplicaciones de las Redes Neuronales Artificiales

Las aplicaciones de las redes neuronales artificiales son múltiples, entre las que podemos mencionar (Basogain, 1998, Basheer & Hajmeer, 2000 y Azam, 2000):

- Análisis financiero: evaluación de préstamos hipotecarios.
- Procesamiento de imágenes: clasificación de imágenes, pruebas médicas automatizadas, reconocimientos automatizados, terminación de patrón e imagen, inspección industrial, etc.
- Diagnóstico: evaluación de fallo, diagnóstico dermatológico, fallo de reactor, de sistemas de control automático, etc.
- Control y robótica: balanceo de objetos, brazos robóticos, etc.
- Procesado de voz: reconocimiento de fenómenos, de velocidad, de texto, etc.
- Microbiología: modelaje del crecimiento microbiano de curvas, análisis de espectros de masas de pirólisis para la identificación de microorganismos, duración de los alimentos, etc.
- Otros: reconocimiento de voz automático, reconocimiento de caracteres escritos a mano, optimización, modelos estadísticos, problemas de inteligencia artificial, etc.

2. BIOLOGÍA ÓSEA. PATOLOGÍA: OSTEOPOROSIS

“Mi cerebro es solo un receptor, en el Universo hay un núcleo del cual obtenemos conocimiento, fuerza e inspiración”.

-Nikola Tesla-

El hueso es un biomaterial que se encuentra en continua renovación, bien porque cada parte del esqueleto se remodela periódicamente o como consecuencia de haber sufrido algún daño. Para ello, dispone de un mecanismo de autoreparación conocido como remodelación ósea, llevada a cabo por las poblaciones celulares y los factores bioquímicos. Además, en este capítulo, estudiaremos una de las enfermedades óseas que tiene lugar cuando se produce un desequilibrio en dicha remodelación, la osteoporosis.

2.1 Hueso: funciones, tipos y composición

Entre las múltiples funciones óseas, podemos mencionar: dar forma a la estructura resistente del cuerpo, mantener los grados de libertad necesarios para el movimiento (por este motivo, en las extremidades hay más huesos en la zona distal que en la proximal), ofrecer apoyo a los músculos, proteger de los impactos, proteger a la médula ósea, etc. Además, participa en importantes funciones fisiológicas como son la homeostasis del calcio y del fósforo (Riancho & Delgado-Calle, 2011). Para cumplir con la primera, el hueso almacena reservas de calcio, mineral muy importante en funciones como la mineralización, la actividad muscular, la transmisión de impulsos nerviosos, la activación de reacciones enzimáticas o la coagulación sanguínea, entre otras (Martínez-Reina, García-Aznar, Domínguez & Doblaré, 2008). Por otro lado, mantener los niveles de fósforo dentro del rango fisiológico es importante para el metabolismo y para la mineralización de la matriz ósea, donde la hormona paratiroidea juega un papel importante (Quarles, 2008).

Por un lado, podemos clasificar a los huesos en función de su morfología en largos, planos y cortos (Figuras 2.1.1-2.1.3), mientras que, por otro lado, atendiendo a sus características internas, en cortical y trabecular (Figura 2.1.4) (Pérez, Merí & Ruano, 2004):

Los huesos largos son aquellos en los que una dimensión es mayor que las otras dos y en ellos se distinguen varias partes: la diáfisis o parte central, la epífisis localizada en las extremidades y entre ambas, la metáfisis, donde se da el proceso de osificación. Cuando este finaliza tras completarse el crecimiento, queda la línea epifisaria. Por ejemplo, el húmero y el fémur. En los huesos planos, dos dimensiones predominan sobre la tercera. En ellos se suelen diferenciar una cara cóncava y otra convexa y presentan mayor densidad ósea por los bordes que por el interior. Entre ellos, se encuentran el cráneo y el esternón. En los huesos cortos, no hay ninguna dimensión característica y en ellos, su superficie es más densa que su interior. Algunos ejemplos son los del carpo y el tarso.



Figuras 2.1.1-2.1.3. Pérez, Meri y Ruano (2004). Manual y atlas fotográfico de anatomía del aparato locomotor. De izquierda a derecha, extremo distal del fémur, hueso parietal del cráneo y huesos del carpo.

El hueso cortical es más denso, presenta una porosidad aproximada del 30% y se compone de láminas de matriz extracelular concéntricas alrededor de los canales de Havers. Se localiza en la diáfisis de los huesos largos y en la capa más externa de los huesos. El hueso trabecular es menos denso, con una porosidad que oscila entre el 50 y 90% y está formado por trabéculas entre cuyos espacios se encuentra la médula. Se localiza en la epífisis de los huesos largos y en la mayoría de los huesos planos y cortos (Barba, 2011). Esta diferencia de densidades se justifica por las cargas a las que el hueso está sometido. Por el desuso se reabsorbe hueso mientras que por el sobreuso se deposita, resultando en diferencia de densidad y anisotropía, todo ello para lograr un diseño óptimo (Martínez-Reina, Ojeda & Mayo, 2016).

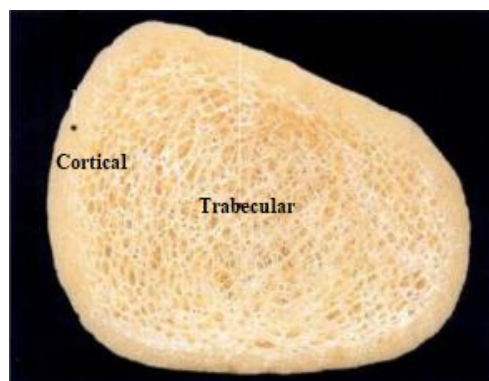


Figura 2.1.4. Pérez, Meri y Ruano (2004). Manual y atlas fotográfico de anatomía del aparato locomotor. Sección transversal de un hueso largo en una zona próxima a la epífisis.

En cuanto a su composición, si tomamos como referencia un determinado volumen de tejido óseo, podemos comprobar que está compuesto de matriz ósea y de poros. Dicha matriz, a su vez se compone de fase mineral, donde el componente principal es la hidroxiapatita, orgánica, fundamentalmente colágeno de tipo I y agua. El porcentaje de las fases depende de varios factores, como por ejemplo, la especie (Figura 2.1.5), la edad, la zona corporal, el género, la presencia de enfermedad o tratamiento, pero suele estar en 70%, 20% y 10%, respectivamente (Boskey, 2013).

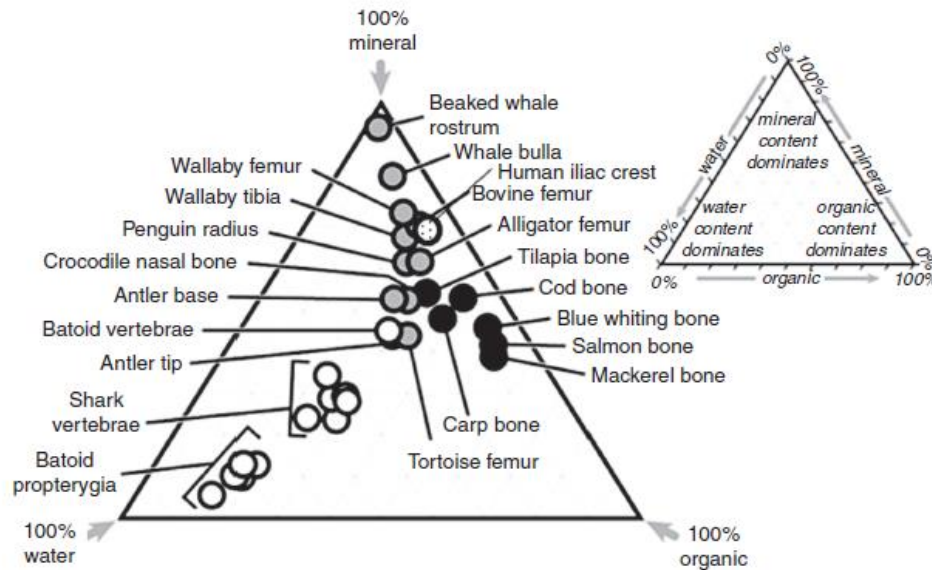


Figura 2.1.5. Boskey (2013). Bone composition: relationship to bone fragility and antiosteoporotic drug effects. Composición de los huesos en diferentes especies.

2.2 Remodelación ósea

La remodelación ósea es un proceso de renovación natural del esqueleto de los seres vivos, como consecuencia de algún daño estructural o de una remodelación periódica. Las actividades de la rutina diaria o esfuerzos físicos pueden generar microgrietas cuya acumulación puede desembocar en fractura. Para evitarla, disponemos de dicho mecanismo celular de autoreparación. Además, la remodelación ósea contribuye a la homeostasis mineral, ya que proporciona acceso a almacenes de calcio y fosfato. Alteraciones en dicha renovación ósea puede tener como consecuencia enfermedades como la osteoporosis (Komarova, Smith, Dixon, Sims & Wahl, 2003).

Se produce gracias a la acción coordinada de las células óseas y los factores bioquímicos en lo que se conoce como una unidad multicelular básica (en inglés, *Basic Multicellular Unit, BMU*), descritos a continuación (ver Figura 2.2.1). Su duración depende del tipo de hueso: menor en el cortical y mayor en el esponjoso, metabólicamente más activo.

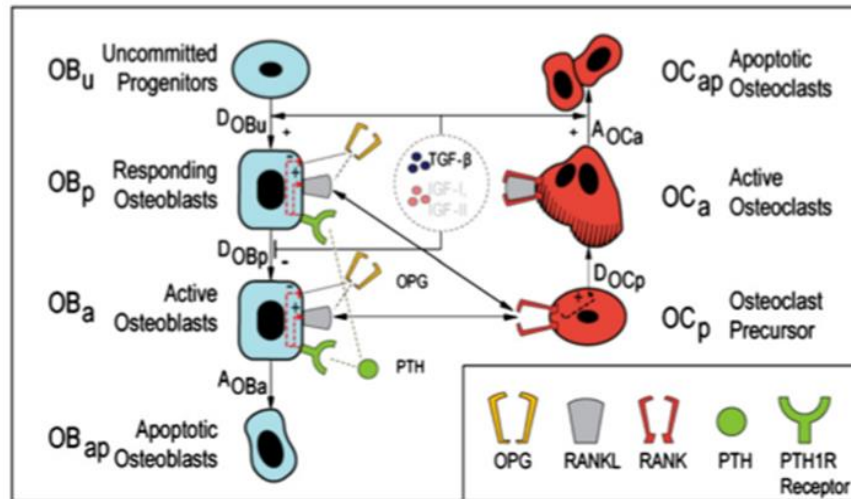


Figura 2.2.1. Pivonka et al. (2008). Model structure and control of bone remodeling: A theoretical study. Esquema de remodelación ósea donde se puede distinguir el linaje osteoblástico, el osteoclástico, las diferentes etapas de las células óseas y los factores bioquímicos implicados.

2.2.1 Poblaciones celulares

A continuación, se describen los cuatro tipos de poblaciones celulares estudiados: osteoblastos (OB), osteoclastos (OC), osteocitos y células de borde. En la Figura 2.2.1, podemos diferenciar entre células del linaje osteoblástico y las del linaje osteoclástico. En ambos, distinguimos varias etapas a lo largo de su vida celular: no comprometidos (OB_u), precursores (OB_p), activos (OB_a) y los que sufren apoptosis (OB_{ap}).

Los osteoblastos son células óseas mononucleadas que se encargan del proceso de formación tras la actividad de los osteoclastos. Depositan la fase orgánica y posteriormente el hueso se mineraliza. Cuando cumplen su función, pueden sufrir apoptosis o se diferencian a osteocito o célula de borde, como se detalla en la Figura 2.2.2. Las células madre mesenquimales se pueden diferenciar a varios tipos celulares, entre ellos, los osteoblastos no comprometidos, OB_u . Posteriormente, gracias a la acción de determinados factores, se produce la diferenciación de los no comprometidos a los precursores, OB_p y más tarde, de los precursores a los activos, OB_a , que son los que se encargan de la formación ósea (Pivonka et al., 2008).

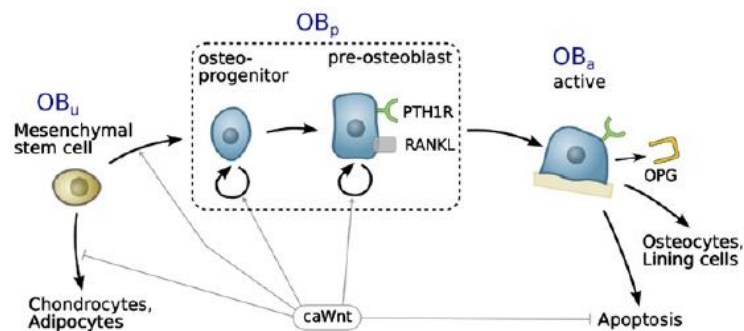


Figura 2.2.2. Buenzli, Pivonka, Gardiner y Smith (2012). Modelling the anabolic response of bone using a cell population model. Diferentes etapas del linaje osteoblástico y factores bioquímicos implicados.

Los osteoclastos son células óseas multinucleadas que llevan a cabo la reabsorción ósea, a una velocidad de decenas de micra por día. En la superficie pegada al hueso segregan ácidos que disuelven la fase mineral y con enzimas disuelven la fase orgánica (ver Figura 2.2.3). Al cumplir su objetivo, sufren apoptosis y no se diferencian a ningún otro tipo de célula ósea, a diferencia de los osteoblastos. Las células hematopoyéticas dan lugar a los osteoclastos no comprometidos, OC_u . Más tarde, gracias a la acción de determinados factores, se produce la diferenciación de los no comprometidos a los precursores, OC_p y finalmente, de los precursores a los activos, OC_a , que son los que se encargan de la reabsorción ósea (Pivonka et al., 2008).

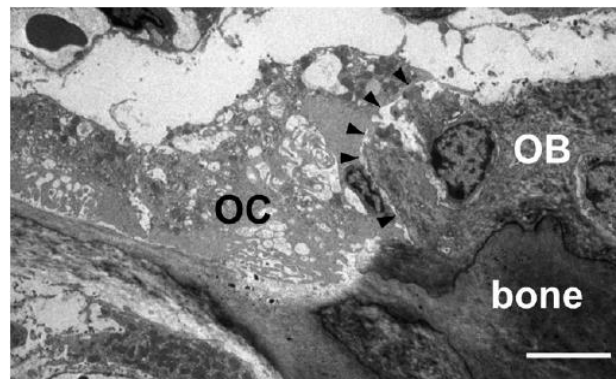


Figura 2.2.3. Matsuo e Irie (2008). Osteoclast-osteoblast communication. Contacto entre un osteoclasto y un osteoblasto. Se aprecia cómo los osteoclastos son células óseas multinucleadas mientras que los osteoblastos son mononucleadas.

Los osteocitos son células óseas que no proliferan y que derivan de los OB_a . Al diferenciarse estos últimos, se estiran y las ramificaciones ocupan túneles dentro de la matriz, denominados canalículos, que les permite conectarse entre sí y con las células de borde (Figura 2.2.4). Se encuentran tanto en la matriz ósea mineralizada como en los osteoides recién formados (Noble, 2008). Son los encargados del mantenimiento del hueso. Si algo falla pueden enviar una señal a las células de borde que activan la remodelación. Actúan como mecanosensores que traducen las señales mecánicas a respuestas bioquímicas (Bilbao, Sánchez & Gil-Antuñano, 2011). El osteocito, es una célula capaz de iniciar el proceso de remodelación en respuesta a estímulos mecánicos. Según Parfitt (2009) dicho estímulo se produce por el movimiento del fluido entre las ramificaciones de los osteocitos y las paredes óseas.

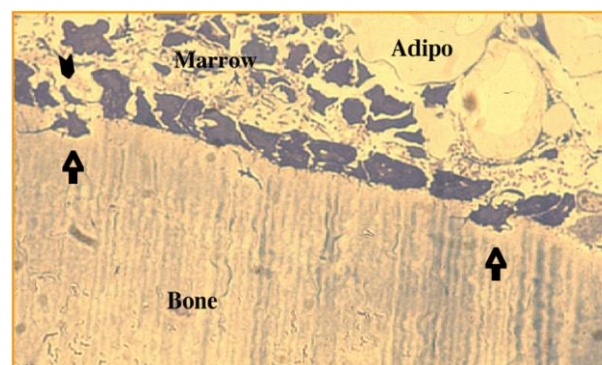


Figura 2.2.4. Noble (2008). The osteocyte lineage. La imagen muestra cómo parte de los osteoblastos activos tras depositar el osteoide se diferencian a osteocitos.

Al igual que los osteocitos, las células de borde también derivan de los OB_a , que tras cumplir su función cubren la superficie ósea cambiando su forma a otra más plana y alargada. Su misión es la de proteger a la matriz ósea

de determinados agentes químicos presentes en la sangre que pueden disolver el mineral. Se comunican con los osteocitos y reclutan a los OC_p iniciándose así el proceso de remodelación ósea (Matsuo & Irie, 2008).

2.2.2 Unidad multicelular básica, *BMU*

Tras haber definido las células óseas, nos encontramos en disposición de definir su grupo de trabajo, la unidad multicelular básica, BMU. Su activación se produce por daño y/o desuso y su actividad consiste en la remodelación ósea: resorber tejido viejo y sustituirlo por osteoide. Para ello, tras la activación (A), se produce la resorción (R) por osteoclastos, después tiene lugar una inversión y finalmente la formación (F) por osteoblastos. A esa secuencia, se la denomina A-R-F (Martínez-Reina, García-Aznar, Domínguez & Doblaré, 2009). Sin embargo, otros autores dividen el proceso en tres fases: fase de iniciación, transición y final (Figura 2.2.5). En la primera se reclutan a los osteoclastos precursores, se diferencian a activos y se inicia la reabsorción, teniendo una duración de aproximadamente tres semanas. Posteriormente, en la segunda fase los osteoclastos cumplen su función, sufren apoptosis y se inicia la actividad de los osteoblastos. Finalmente, los osteoblastos depositan el osteoide y se produce la mineralización del hueso, durando alrededor de tres meses (Matsuo & Irie, 2008). La población y la actividad celular en dicha estructura está controlada por la homeostasis, es decir, balance entre resorción y formación ósea (Pivonka, Buenzli, Scheiner, Hellmich & Dunstan, 2013).

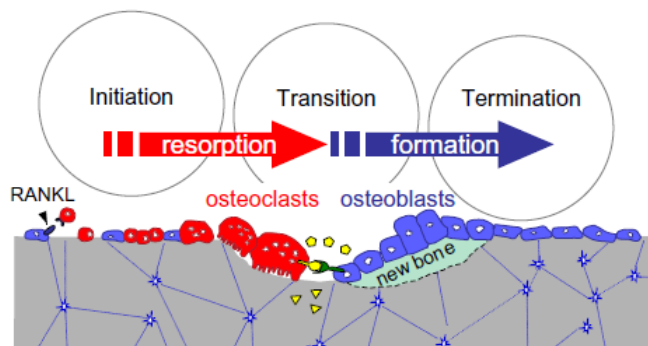


Figura 2.2.5. Matsuo et al. (2008). Osteoclast-osteoblast communication. Etapas del proceso de remodelación ósea teniendo en cuenta la comunicación entre osteoclastos y osteoblastos.

Algunos datos interesantes relacionados con la BMU son los siguientes (Parfitt, 1994):

- Dimensiones de una BMU: 1-2 mm de largo y 0,2-0,4 mm de ancho.
- Esperanza de vida de la BMU: 6-9 meses.
- Velocidad: 25 $\mu\text{m}/\text{día}$.
- Volumen de hueso reemplazado por una BMU: 0,025 mm^3 .
- Esperanza de vida de los osteoclastos: 2 semanas.
- Esperanza de vida de los osteoblastos activos: 3 meses.
- Intervalo de tiempo entre sucesivas remodelaciones óseas en una misma zona: 2-5 años.
- Tasa de reemplazo del esqueleto: 10% por año.

2.2.3 Factores que influyen en la remodelación ósea

En el proceso de remodelación ósea, influyen factores tanto mecánicos como no mecánicos (Tabla 2.2.1). Un ejemplo de un factor mecánico es por ejemplo la fuerza de la gravedad: cuando los astronautas viajan al espacio y están sometidos durante un tiempo a un estado de ingravidez, pierden densidad ósea. Por otro lado, un ejemplo de factor no mecánico es por ejemplo la edad: cuando las mujeres con cierta edad presentan la menopausia tienden a sufrir osteoporosis, lo que se traduce en pérdida de densidad ósea.

Tabla 2.2.1. Bilbao et al. (2011). Regulación del metabolismo óseo a través del sistema RANK-RANKL-OPG. Factores que influyen en la remodelación ósea.

Mecánicos	No mecánicos
• Fuerza de la gravedad	• Hormonas
• Peso del individuo	• Factores locales (autocrinos/paracrinos)
• Actividad física-sedentarismo	• Edad/sexo/genética
• Contractilidad muscular	• Dieta (calcio, vitaminas, minerales,...)
• Esfuerzo	• Ocupación laboral/ergonomía
• Ingravidez-reposo prolongado	• Algunas enfermedades (p. ej. artritis reumatoide)

En lo que sigue, analizaremos más en detalle el proceso de remodelación ósea desde el punto de vista bioquímico. Para ello, partiremos de la Figura 2.2.1. Los factores bioquímicos que encontramos son los siguientes: RANK, RANKL, OPG, PTH Y TGF- β . Cuando se produce algún desajuste entre ellos, se da el desequilibrio entre formación y resorción y tienen lugar las enfermedades óseas como la osteoporosis, que analizaremos más en detalle a continuación (Bilbao et al., 2011 y Boyce & Xing, 2008).

RANK es el receptor activador del factor nuclear $\kappa\beta$ (NF- $\kappa\beta$). Se encuentra en la superficie de OC_p . Como consecuencia de la interacción de RANKL con RANK, se favorece la diferenciación de OC_p a OC_a , por lo tanto, mayor concentración de osteoclastos activos y mayor resorción ósea (Fuller, Wong, Fox, Choi & Chambers, 1998). Su ligando, RANKL, es un polipéptido que se encuentra en las células del estroma y en el linaje osteoblástico, concretamente en OB_p . La osteoprotegerina, OPG, es una proteína sintetizada por los osteoblastos y las células estromales, inhibidora de la resorción. Se une a RANKL para prevenir la interacción de RANKL con RANK, regulando así la osteoclastogénesis (Pivonka et al., 2010).

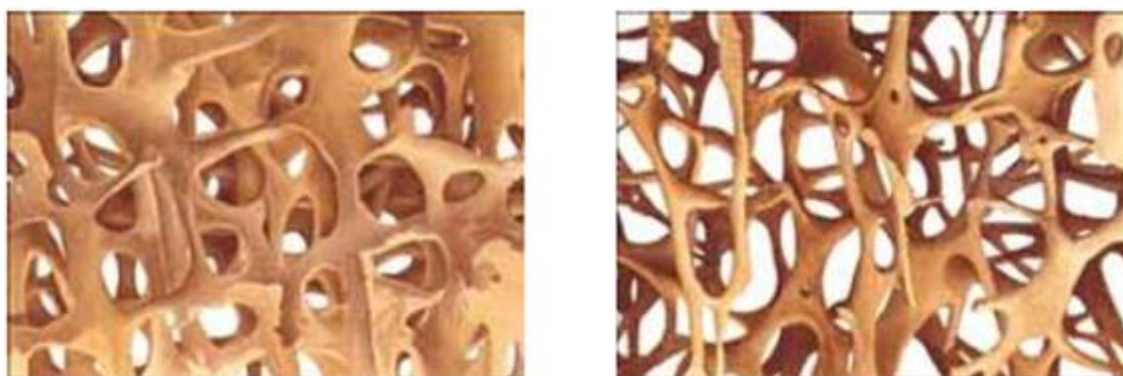
La hormona paratiroidea, PTH, se produce en las glándulas paratiroides. Se trata de un polipéptido de 84 aminoácidos y destaca por su papel en la homeostasis del calcio. Además, favorece la expresión de RANKL y disminuye la de OPG, estimulando la formación de osteoclastos a través de los osteoblastos. Por último, el factor de crecimiento transformador, TGF- β , se encuentra en la matriz ósea. Durante la resorción de dicha matriz por los OC_a , se libera al microambiente de las BMUs. Actúa como agente autorregulador sobre las células óseas. Cuenta con receptores tanto en osteoblastos como en osteoclastos. Por un lado, favorece la diferenciación de OB_u a OB_p y la apoptosis de OC_a y por otro disminuye la diferenciación de OB_p a OB_a (Pivonka & Komarova, 2010).

2.3 Osteoporosis

2.3.1 Definición, tipos y factores de riesgo

La osteoporosis es una enfermedad ósea que consiste en el aumento de poros y por consiguiente en un empeoramiento de la función biomecánica, debido a que disminuye la masa y se produce una alteración de la microestructura ósea, resultando en fragilidad y en algunos casos, en fractura del hueso. La resistencia ósea es función de la calidad y de la densidad óseas. La primera hace referencia a la arquitectura, la remodelación, el daño acumulado y la mineralización, mientras que la segunda se entiende como la densidad mineral ósea que depende del pico de masa ósea y de la cantidad de hueso que se va perdiendo (Del Pino, 2010 y González, Vásquez & Molina, 2009).

Podemos clasificar la osteoporosis en primaria y secundaria. La primaria, a su vez, se divide en idiopática, que afecta a niños y a adultos jóvenes, e involutiva, debida al envejecimiento. Esta última se subdivide en involutiva de tipo I, debida a la postmenopausia, e involutiva de tipo II o senil. La secundaria es debida a alguna enfermedad o a la ingesta de fármacos, donde el consumo de glucocorticoides constituye la causa más común (Del Pino, 2010 y González et al., 2009).



Figuras 2.3.1-2.3.2. Jordán, Pachón, Ponce, Robainas y Moreno (2006). Osteoporosis: ¿Un problema de salud prevenible? La imagen de la izquierda corresponde a un hueso trabecular sano mientras que la de la derecha a uno con osteoporosis.

La patogénesis de la enfermedad puede estar relacionada con la disminución de hormonas sexuales, la tercera edad y el exceso de glucocorticoides (Manolagas, 2000). En primer lugar, con la deficiencia de dichas hormonas, se incrementa el proceso de remodelación, ya que hay una mayor producción tanto de osteoclastos como de osteoblastos. Sin embargo, hay un desequilibrio puesto que la esperanza de vida de los osteoclastos es mayor. En segundo lugar, durante la senectud la formación ósea es menor, sobre todo en el hueso trabecular, cuyas trabéculas se van volviendo más finas (Figuras 2.3.1-2.3.2). Y, finalmente, en tercer lugar los glucocorticoides inhiben la osteoblastogénesis en la médula ósea y favorecen la apoptosis de osteoblastos y osteocitos.

Los factores de riesgo para la osteoporosis son un pico de masa ósea bajo, una pérdida ósea acelerada tras alcanzarse dicho pico (Hansen, Overgaard, Riis & Christiansen, 1991), el tabaquismo, la cantidad de calcio y de vitamina D, una menopausia temprana y la disminución de masa ósea durante la menopausia (Aloia et al., 1984). La vitamina D es necesaria para la absorción del calcio y este para tener huesos fuertes y sanos. Durante la primera fase de la menopausia, se tiene un balance negativo de calcio. Alrededor de los 60 años, se produce una disminución de la absorción de dicho mineral. A medida que se avanza con los años, la presencia de la vitamina D en el intestino se reduce, lo que dificulta la absorción del calcio. Además con la edad se produce menos vitamina D (Tella & Gallagher, 2013).

2.3.2 Impacto y fracturas más frecuentes

Se trata de una enfermedad que afecta a 150 – 200 millones de personas en el mundo. Se estima que cerca del 33% de las mujeres mayores de 50 años van a tener osteoporosis. Los costes se dividen en directos, sanitarios, farmacéuticos, cuidados sociales, etc., e indirectos, como la pérdida de productividad o de calidad de vida (Del Pino, 2010). El coste sanitario por estas fracturas tiene una cuantía de casi 30 billones de euros en la Unión Europea. Además de las cifras monetarias, debemos hablar de la gran repercusión psicológica que la enfermedad acarrea, llevando a un estado de depresión, aislamiento y poca autoestima. En el año 2000, se produjeron 9 millones de fracturas por osteoporosis, de los cuales 1,6 se produjeron en la cadera, 1,7 en el antebrazo y 1,4 en vértebras (González et al., 2009).

De las posibles fracturas osteoporóticas, la de cadera es la más grave. El paciente osteoporótico puede fracturarse la cadera como consecuencia de una caída o, estando de pie, fracturarse la cadera y posteriormente caer al suelo, seguido de una estancia hospitalaria y de una disminución de la calidad de vida (Del Pino, 2010). Tras una caída, solo en el 1% de los casos se produce la fractura, es decir, en aquellas personas que caen de lado y se golpean en el trocánter mayor. El 80% de las fracturas de cadera ocurren en mujeres (González et al., 2009). Representa el 10% de las fracturas no vertebrales y a medida que nos adentramos en la tercera edad, dicho porcentaje sube hasta el 40% en los 80 años. El coste mundial de la fractura de cadera se estima que para 2050 sea de 131500 billones de dólares (Del Pino, 2012). En España, 3 millones de personas sufren osteoporosis y de cada 7 por 1000 habitantes, aproximadamente, sufrirán fractura de cadera (Nogués, Guerri, Solé y Díez-Pérez, 2012). En torno al 12 – 20% de las personas de la tercera edad con fractura de cadera fallecen, produciéndose 740000 muertes por fractura de cadera al año, favorecidas por factores como el género masculino, estado senil, enfermedad y pobre estado óseo (González et al., 2009) y, aproximadamente, el 20% se vuelven inválidos, a nivel mundial (Llerena, 1999).

La fractura vertebral es la más prevalente. En un tercio de los casos, se caracteriza por un dolor agudo en la zona afectada, mientras que en los dos tercios restantes, es asintomática. Se puede producir como consecuencia de un esfuerzo mecánico o por una osteoporosis avanzada. Se distinguen dos tipos: torácicas, que conducen a una cifosis o lumbares, que tienen como consecuencia un aplastamiento con su posterior pérdida de altura. Si se pierden alrededor de 3 cm en dos años, puede ser una señal de fractura vertebral (Del Pino, 2010). Cuenta con una incidencia cuatro veces superior que la de cadera, estando presente en el 12% de las personas de 60 años, en el 25% de las mujeres mayores de 75 años y en el 17% de los hombres mayores de 75 años (Nogués et al., 2012).

La fractura del antebrazo es la que tiene menor repercusión, aunque puede ser una señal de posible futura osteoporosis (Del Pino, 2010). Este caso no es muy frecuente en mujeres de la tercera edad debido a la pérdida de la coordinación neuromuscular, que hace que las mujeres caigan sobre la cadera en vez apoyar con la parte distal del antebrazo (González et al., 2009).

2.3.3 Osteoporosis postmenopáusica

La osteoporosis postmenopáusica es la más frecuente de todas y se debe tanto a la menopausia como al envejecimiento. La disminución de la función ovárica conlleva la reducción de los niveles de estrógenos y por consiguiente, una menor masa ósea. De ahí que los tratamientos con estrógenos disminuyan la osteoclastogénesis, influyan en el balance de calcio, regulando positivamente su absorción intestinal y negativamente su eliminación renal. Además, participan en el metabolismo de la vitamina D y disminuyen la cantidad de PTH (Del Pino, 2010).

Uno de los fármacos utilizados para tratar esta enfermedad es el denosumab, un anticuerpo monoclonal, es decir,

desciende de un solo tipo de célula madre del sistema inmunitario, que se une al RANKL para evitar que este se una al RANK e iniciarse así el proceso de remodelación ósea. Por lo tanto, la principal consecuencia de la administración del denosumab es el incremento de la densidad ósea. En mujeres que sufren la osteoporosis postmenopáusica se suele inyectar 60 mg cada 6 meses y se ha demostrado el aumento de la densidad ósea en vértebras lumbares y en la cadera, reduciendo así el riesgo de fractura (Eastell et al., 2011 y Martínez-Reina & Pivonka, 2019).

3. MODELO MATEMÁTICO

“Mantengo constantemente el tema de mi investigación ante mí, y espero hasta que el primer amanecer me abra gradualmente, poco a poco, una luz clara y plena”.

-Isaac Newton-

El objetivo de este apartado es conocer el modelo matemático que permite simular el proceso de remodelación ósea cuando la osteoporosis está presente. Para ello, vamos a comenzar con una descripción de dicho modelo, seguiremos con un análisis de sensibilidad para estudiar más en detalle cómo los diferentes parámetros influyen en el mismo y terminaremos describiendo el proceso de automatización de las simulaciones.

3.1 Descripción del modelo

La simulación matemática de cualquier proceso, en este caso biológico, requiere conocer en detalle todos los parámetros que van a intervenir en el modelo, tanto contantes como variables. En este caso, vamos a partir del estudio de Pivonka et al. (2008), de Buenzli, Pivonka y Smith (2011) y de Pivonka et al. (2013) para la remodelación ósea, de Martínez-Reina et al. (2008) para incluir el proceso de mineralización de los huesos y de Scheiner, Pivonka, Smith, Dunstan y Hellmich (2014) para estudiar el efecto de la osteoporosis y de su tratamiento. Dichos parámetros necesarios los hemos dividido en cantidades dinámicas (Tabla 3.1.1) y constantes (Tablas 3.1.2 y 3.1.3).

Tabla 3.1.1. Descripción de las cantidades dinámicas del modelo.

SÍMBOLOS	DESCRIPCIÓN
OC_p	Densidad de pre-osteoclastos, en pM
OC_a	Densidad de osteoclastos activos, en pM
OB_p	Densidad de pre-osteoblastos, en pM
OB_a	Densidad de osteoblastos activos, en pM
TGF_β	Concentración de TGF_β , en pM
$RANK$	Concentración de RANK, en pM

$RANKL$	Concentración de RANKL, en pM
OPG	Concentración de OPG, en pM
PTH	Concentración de PTH, en pM
f_{vas}	Porosidad vascular
S_V	Superficie específica
Ψ_{bm}	Densidad de energía de deformación, en MPa

Tabla 3.1.2. Descripción de las constantes del modelo.

Parámetro	Descripción
D_{OBu}	Tasa de diferenciación de OB_u a OB_p
D_{OBp}	Tasa de diferenciación de OB_p a OB_a
A_{OBa}	Tasa de apoptosis de OB_a
D_{OBa}	Tasa de diferenciación de OB_a a osteocitos
D_{OCu}	Tasa de diferenciación de OC_u a OC_p
D_{OCp}	Tasa de diferenciación de OC_p a OC_a
A_{OCa}	Tasa de apoptosis de OC_a
$K_{OBu,act}^{TGF-\beta}$	Coefficiente de activación de $TGF-\beta$ relacionado con la diferenciación de OB_u a OB_p
$K_{OBp,rep}^{TGF-\beta}$	Coefficiente de represión de $TGF-\beta$ relacionado con la diferenciación de OB_p a OB_a
$K_{OCa,act}^{TGF-\beta}$	Coefficiente de activación de $TGF-\beta$ relacionado con la apoptosis de OC_a
$\tilde{D}_{TGF-\beta}$	Tasa de degradación de $TGF-\beta$
$K_{[PTH-RANKL],act}$	Coefficiente de activación para la producción de RANKL debida a la unión de PTH
$K_{[PTH-RANKL],rep}$	Coefficiente de represión para la producción de RANKL debida a la unión de PTH
$K_{[RANKL-RANK],act}$	Coefficiente de activación relacionado con la unión de RANKL y RANK
N_{OCp}^{RANK}	Cantidad de RANK sobre la superficie de OC_p

$K_{OCu,act}^{MCSF}$	Coefficiente de activación relacionado con la unión de MCSF y OC _u
$P_{RANKL-d}$	Inyección externa de RANKL
P_{OPG-d}	Inyección externa de OPG
P_{PTH-d}	Inyección externa de PTH
$\tilde{D}_{PTH}$	Tasa de degradación de PTH
β_{PTH}	Tasa de síntesis de PTH sistémico
β_{OPG}	Tasa mínima de producción de OPG por célula
$\tilde{D}_{OPG}$	Tasa de degradación de OPG
$OPG_{m\acute{a}x}$	Máxima concentración posible de OPG
β_{RANKL}	Tasa de producción de RANKL por célula
R_{RANKL}	Máxima concentración de RANKL sobre cada superficie celular
$\tilde{D}_{RANKL}$	Tasa de degradación de RANKL
$K_{A1,[RANKL-OPG]}$	Constante de asociación RANKL - OPG
$K_{A2,[RANKL-RANK]}$	Constante de asociación RANKL - RANK
k_{res}	Tasa relativa de reabsorción ósea normalizada con respecto a la reabsorción ósea normal
$n_{TGF-\beta}^{bone}$	Contenido almacenado de TGF- β en la matriz ósea
$R1_{OBp}^{RANKL}$	Factor para la expresión de RANKL sobre OB _p
$R2_{OBa}^{RANKL}$	Factor para la expresión de RANKL sobre OB _a
$\beta1_{OBp}^{OPG}$	Factor para la expresión de OPG sobre OB _p
$\beta2_{OBa}^{OPG}$	Factor para la expresión de OPG sobre OB _a
σ	Tensión aplicada (compresión)
$\Delta\sigma$	Incremento de tensión aplicada
t_R	Tiempo de residencia del mineral
t_{mlt}	Longitud de la fase inicial de la mineralización
t_{prim}	Longitud de la fase primaria de la mineralización

κ	Tasa de deposición de mineral en la fase secundaria
v_{prim}	Volumen específico de mineral al final de la fase primaria
$v_{m\acute{a}x}$	Volumen específico de mineral correspondiente al máximo contenido de Ca
Δt	Incremento de tiempo
k_{OBu}	Exponente de la regulación geométrica. Diferenciación de OB _u a OB _p
k_{OBp}	Exponente de la regulación geométrica. Diferenciación de OB _p a OB _a
k_{OCu}	Exponente de la regulación geométrica. Diferenciación de OC _u a OC _p
k_{OCp}	Exponente de la regulación geométrica. Diferenciación de OC _p a OC _a
ϕ_{PMO}^{mech}	Capacidad de mecanorespuesta de los osteocitos
τ_{PMO}^{mech}	Constante de tiempo en la ecuación de la mecanorespuesta
ϕ_{PMO}^{RANKL}	Factor de atenuación de RANKL con PMO
τ_{PMO}^{RANKL}	Constante de tiempo en el factor de atenuación de RANKL con PMO
ζ	Constante de asociación RANKL - Denosumab
$C_{Denosumab}$	Concentración de Denosumab
f_{bm}	Lumbar Spine o Hip
<i>Tratamiento</i>	T_{02}, T_{04} ó T_{17}
P_{RANKL}^{PMO}	Tasa de producción de RANKL relacionada con PMO

Tabla 3.1.3. Constantes del modelo, valores numéricos y unidades.

Parámetro	Valor numérico	Unidad
D_{OBu}	$7 * 10^{-2}$	$d\acute{a}^{-1}$
D_{OBp}	$1,6570 * 10^{-1}$	$d\acute{a}^{-1}$
A_{OBa}	$2,1107 * 10^{-1}$	$d\acute{a}^{-1}$
D_{OBa}	$1 * 10^{-1}$	$d\acute{a}^{-1}$
D_{OCu}	$4,2 * 10^{-1}$	$d\acute{a}^{-1}$

D_{Ocp}	2,1	$día^{-1}$
A_{Oca}	5,6487	$día^{-1}$
$K_{OBu,act}^{TGF-\beta}$	$5,6328 * 10^{-4}$	pM
$K_{OBp,rep}^{TGF-\beta}$	$1,7543 * 10^{-4}$	pM
$K_{OCa,act}^{TGF-\beta}$	$5,6328 * 10^{-4}$	pM
$\tilde{D}_{TGF-\beta}$	2	$pM TGF - \beta * día^{-1}$
$K_{[PTH-RANKL],act}$	$1,5 * 10^2$	pM
$K_{[PTH-RANKL],rep}$	$2,2258 * 10^{-1}$	pM
$K_{[RANKL-RANK],act}$	5,6797	pM
N_{Ocp}^{RANK}	$1 * 10^4$	$pM RANK * (pM cell)^{-1}$
$K_{OCu,act}^{MCSF}$	$1 * 10^{-3}$	pM
$P_{RANKL-d}$	0	$pM * día^{-1}$
P_{OPG-d}	0	$pM * día^{-1}$
P_{PTH-d}	0	$pM * día^{-1}$
$\tilde{D}_{PTH}$	$8,6 * 10^1$	$pM PTH * día^{-1}$
β_{PTH}	$2,5 * 10^2$	$pM PTH * día^{-1}$
β_{OPG}	$1,6249 * 10^8$	$pM OPG * (pM cell)^{-1}$
$\tilde{D}_{OPG}$	$3,5 * 10^{-1}$	$pM OPG * día^{-1}$
$OPG_{máx}$	$2 * 10^8$	$pM OPG$
β_{RANKL}	$1,6842 * 10^2$	$pM RANKL * (pM cell)^{-1}$
R_{RANKL}	$2,7035 * 10^6$	-
$\tilde{D}_{RANKL}$	$1,01325 * 10^1$	$pM RANKL * día^{-1}$
$K_{A1,[RANKL-OPG]}$	$1 * 10^{-3}$	$(pM OPG)^{-1}$
$K_{A2,[RANKL-RANK]}$	$3,4118 * 10^{-2}$	$(pM RANKL)^{-1}$
k_{res}	200	%

$n_{TGF-\beta}^{bone}$	1	%
$R1_{OBp}^{RANKL}$	1	%
$R2_{OBa}^{RANKL}$	0	%
$\beta1_{OBp}^{OPG}$	0	%
$\beta2_{OBa}^{OPG}$	1	%
σ	-5	MPa
$\Delta\sigma$	0	MPa
t_R	1000	día
t_{mlt}	12	día
t_{prim}	10	día
κ	0,007	-
v_{prim}	$1,2054 * 10^{-1}$	-
$v_{m\acute{a}x}$	0,442	-
Δt	0,25	-
k_{OBu}	0,1	-
k_{OBp}	0	-
k_{OCu}	1	-
k_{OCp}	0	-
φ_{PMO}^{mech}	1	-
τ_{PMO}^{mech}	900	día
φ_{PMO}^{RANKL}	1	-
τ_{PMO}^{RANKL}	10	día
ζ	$1/3 * 0,007$	$(pM \text{ RANKL})^{-1}$
$C_{Denosumab}$	0	pM
f_{bm}	11 ó 25	%

<i>Tratamiento</i>	$T_{02}, T_{04} \text{ ó } T_{17}$	-
p_{RANKL}^{PMO}	4000	pM

Nota: el valor de $v_{m\acute{a}x}$ está tomado de Currey, 2004.

3.1.1 Porosidad vascular y superficie específica

Tras conocer todos los parámetros necesarios, nos encontramos en disposición de mostrar las distintas ecuaciones del modelo. Para lograr tal fin, es necesario comenzar por algunos conceptos. Por un lado, la porosidad vascular se define como el cociente entre:

$$f_{vas} = \frac{V_{vas}}{V_T} \quad (3.1)$$

$V_{vas} \equiv$ Volumen de poros vasculares.

$V_T \equiv$ Volumen de tejido óseo.

La fracción volumétrica de matriz ósea es el cociente entre:

$$f_{bm} = \frac{V_{bm}}{V_T} \quad (3.2)$$

$V_{bm} \equiv$ Volumen de matriz ósea.

Por otro lado, la superficie específica determina qué área tienen los osteoblastos y los osteoclastos para llevar a cabo sus funciones, en mm^2/mm^3 :

$$S_V = \frac{S_p}{V_T} \quad (3.3)$$

$S_p \equiv$ Superficie de poros.

Tras haber definido la porosidad vascular y la superficie específica podemos definir la relación entre ambas. Según Martin (1984) es la siguiente:

$$S_V(f_{vas}) = af_{vas} + bf_{vas}^2 + cf_{vas}^3 + df_{vas}^4 + ef_{vas}^5 \quad (3.4)$$

$$a = 32,3; b = -93,9; c = 134; d = -101 \text{ y } e = 28,8 \text{ en } \text{mm}^{-1}$$

3.1.2 Funciones de Hill

Una función sigmoideal, caracterizada por una evolución para valores bajos del eje horizontal y una saturación conforme avanzamos por el mismo, se suele utilizar para representar muchos procesos biológicos. Esta es la idea de las funciones de Hill, utilizadas para regular el comportamiento de las células óseas. La nomenclatura que vamos a seguir para caracterizarla se muestra a continuación:

$$\pi_{\left(\begin{smallmatrix} \text{activación} \\ o \\ \text{represión} \end{smallmatrix} \right)}^{(Ligando)}_{(molécula)}$$

Donde:

Ligando $\equiv$ Factor bioquímico.

Molécula $\equiv$ Célula ósea receptora.

Activación o represión $\equiv$ Dependiendo de si la unión del factor bioquímico y del receptor favorece o inhibe una determinada acción.

Matemáticamente, se definen como:

$$\pi_{act,A}^L = \frac{L}{L + K_{A,act}^L}$$

$\pi_{act,A}^L \equiv$ Representa la fracción de receptor sobre A que se une a L para favorecer un determinado proceso celular de A .

$$\pi_{rep,A}^L = 1 - \pi_{act,A}^L = \frac{1}{1 + \frac{L}{K_{A,rep}^L}}$$

$\pi_{rep,A}^L \equiv$ Representa la fracción de receptor sobre A que se une a L para reprimir un determinado proceso celular de A .

$K_A^L \equiv$ Constante de activación/represión de la unión del factor bioquímico y el receptor.

Las siguientes funciones de Hill surgen de la modelización matemática de las funciones biológicas de los factores bioquímicos explicadas en el Apartado 2.2.3:

Para el TGF- β :

$$\pi_{act,OB_u}^{TGF-\beta} = \frac{TGF - \beta}{TGF - \beta + K_{OB_u,act}^{TGF-\beta}} \quad (3.5)$$

$$\pi_{rep,OB_p}^{TGF-\beta} = \frac{1}{1 + \frac{TGF - \beta}{K_{OB_u,rep}^{TGF-\beta}}} \quad (3.6)$$

$$\pi_{act,OC_a}^{TGF-\beta} = \frac{TGF - \beta}{TGF - \beta + K_{OC_a,act}^{TGF-\beta}} \quad (3.7)$$

Para el RANKL:

$$\pi_{act,OC_u}^{RANKL} = \frac{RANKL}{RANKL + K_{OC_u,act}^{RANKL}} \quad (3.8)$$

$$\pi_{act,OC_p}^{RANKL} = \frac{RANKL}{RANKL + K_{OC_p,act}^{RANKL}} \quad (3.9)$$

$$\pi_{act,OC_u}^{RANKL} = \frac{RANKL}{RANKL + K_{OC_u,act}^{RANKL}} = \frac{RANKL}{RANKL + K_{OC_p,act}^{RANKL}} = \pi_{act,OC_p}^{RANKL} \quad (3.10)$$

En las Tablas 3.1.2 y 3.1.3, el coeficiente de activación correspondiente se denota como $K_{[RANKL-RANK],act}$.

Para el MCSF:

$$\pi_{act,OC_p}^{MCSF} = \frac{MCSF}{MCSF + K_{OC_u,act}^{MCSF}} \quad (3.11)$$

Para la PTH:

$$\pi_{act,OB_p}^{PTH} = \frac{PTH}{PTH + K_{OB_p,act}^{PTH}} \quad (3.12)$$

$$\pi_{act,OB_a}^{PTH} = \frac{PTH}{PTH + K_{OB_a,act}^{PTH}} \quad (3.13)$$

$$\pi_{act,OB_p}^{PTH} = \frac{PTH}{PTH + K_{OB_p,act}^{PTH}} = \frac{PTH}{PTH + K_{OB_a,act}^{PTH}} = \pi_{act,OB_a}^{PTH} \quad (3.14)$$

En las Tablas 3.1.2 y 3.1.3, el coeficiente de activación correspondiente se denota como $K_{[PTH-RANKL],act}$.

$$\pi_{rep,OB_p}^{PTH} = \frac{1}{1 + \frac{PTH}{K_{OB_p,rep}^{PTH}}} \quad (3.15)$$

$$\pi_{rep,OB_a}^{PTH} = \frac{1}{1 + \frac{PTH}{K_{OB_a,rep}^{PTH}}} \quad (3.16)$$

$$\pi_{rep,OB_p}^{PTH} = \frac{1}{1 + \frac{PTH}{K_{OB_p,rep}^{PTH}}} = \frac{1}{1 + \frac{PTH}{K_{OB_a,rep}^{PTH}}} = \pi_{rep,OB_a}^{PTH} \quad (3.17)$$

En las Tablas 3.1.2 y 3.1.3, el coeficiente de represión correspondiente se denota como $K_{[PTH-RANKL],rep}$.

3.1.3 Regulación geométrica

En este apartado tratamos de entender cómo la superficie ósea disponible es importante para determinados procesos celulares. Según Pivonka et al. (2013):

- En un nuevo inicio de remodelación ósea, la relación entre la superficie ósea disponible y el reclutamiento de los osteoclastos y osteoblastos precursores se tiene en cuenta mediante $g_{OBu}(S_v)$ y $g_{OCu}(S_v)$.
- Para la diferenciación de las células óseas precursoras a las activas también es necesario la superficie ósea y ello se modela mediante $g_{OBp}(S_v)$ y $g_{OCp}(S_v)$.

Matemáticamente, dichas funciones de regulación geométrica se definen como:

$$g_i(S_v) = \left(\frac{S_v}{S_v(t_0)} \right)^{k_i} \quad (3.18)$$

Donde:

$$k_i \geq 0$$

$$i = OB_u, OB_p, OC_u, OC_p$$

$$t_0 \equiv \text{Estado estacionario homeostático.}$$

3.1.4 Regulación mecánica

Además de la regulación geométrica, debemos tener en cuenta la mecánica, que trata de cuantificar cómo los osteoblastos precursores proliferan debido a las deformaciones de la matriz extravascular. La densidad de energía de deformación (SED), aquella que trata de cuantificar la adaptación del hueso a determinados niveles de carga, es función del tensor de tensiones macroscópico (Σ), del tensor de rigidez de la matriz ósea (c_{bm}) y de la porosidad vascular (f_{vas}):

$$\Psi_{bm} = \Psi_{bm}(\Sigma, c_{bm}, f_{vas})$$

De acuerdo con Scheiner, Pivonka y Hellmich (2013), la proliferación de RANKL debida a la SED y la función de activación que regula la proliferación de osteoblastos precursores se define matemáticamente como:

$$P_{RANKL}^{\Psi_{bm}} = \begin{cases} \kappa \left(1 - \frac{\Psi_{bm}}{\tilde{\Psi}_{bm}} \right) & \Psi_{bm} < \tilde{\Psi}_{bm} \\ 0 & \Psi_{bm} \geq \tilde{\Psi}_{bm} \end{cases} \quad (3.19)$$

$$\Pi_{act,OB_p}^{mech} = \begin{cases} \tilde{\Pi}_{act,OB_p}^{mech} & \Psi_{bm} \leq \tilde{\Psi}_{bm} \\ \tilde{\Pi}_{act,OB_p}^{mech} \left[1 + \lambda \left(\frac{\Psi_{bm}}{\tilde{\Psi}_{bm}} - 1 \right) \right] & \tilde{\Psi}_{bm} < \Psi_{bm} < \hat{\Psi}_{bm} \\ 1 & \hat{\Psi}_{bm} \leq \Psi_{bm} \end{cases} \quad (3.20)$$

Donde:

$\tilde{\Psi}_{bm} \equiv$ Umbral de densidad de energía de deformación.

$\hat{\Psi}_{bm} \equiv$ Límite superior de la densidad de energía de deformación. Se define como:

$$\hat{\Psi}_{bm} = \tilde{\Psi}_{bm} \left[1 - \tilde{\Pi}_{act,OB_p}^{mech} (1 - \lambda) \right] \frac{1}{\lambda \tilde{\Pi}_{act,OB_p}^{mech}}$$

$\tilde{\Pi}_{act,OB_p}^{mech} \equiv$ Mínimo de Π_{act,OB_p}^{mech} .

$\kappa \equiv$ Parámetro inhibitorio de producción de RANKL.

$\lambda \equiv$ Parámetro de fuerza anabólico.

3.1.5 Concentración de los factores bioquímicos

En apartados anteriores hemos definido las funciones de los distintos factores bioquímicos. A continuación, se muestra la expresión matemática de su concentración, las cuales se han tomado de Pivonka et al. (2008), Buenzli et al. (2011) y Pivonka et al. (2013):

$$TGF - \beta = \frac{n_{TGF\beta}^{Bone} \cdot k_{res} \cdot OC_a}{\tilde{D}_{TGF\beta}} \quad (3.21)$$

$$PTH = \frac{\beta_{PTH} + P_{PTH-d}}{\tilde{D}_{PTH}} \quad (3.22)$$

$$OPG_{eff} = \left(\beta 1_{OB_p}^{OPG} \cdot OB_p + \beta 2_{OB_a}^{OPG} \cdot OB_a \right) \cdot \pi_{rep,OB}^{PTH} \quad (3.23)$$

$$OPG = \frac{OPG_{eff} + P_{OPG-d}}{\frac{OPG_{eff}}{OPG_{máx}} + \tilde{D}_{OPG}} \quad (3.24)$$

$$\begin{aligned} RANKL_{eff} &= RANKL_{OB_p,eff} + RANKL_{OB_a,eff} \\ &= \left(R1_{OB_p}^{RANKL} \cdot OB_p + R2_{OB_a}^{RANKL} \cdot OB_a \right) \cdot \pi_{act,OB_a}^{PTH} \end{aligned} \quad (3.25)$$

$$\begin{aligned} RANKL_{tot} &= RANKL + [RANKL - OPG] + [RANKL - RANK] \\ &= RANK \cdot \left(1 + K_{A1,[RANKL-OPG]} \cdot OPG + K_{A2,[RANKL-RANK]} \cdot RANK \right) \end{aligned} \quad (3.26)$$

$$\begin{aligned} RANKL &= \frac{RANKL_{eff}}{1 + K_{A1,[RANKL-OPG]} \cdot OPG + K_{A2,[RANKL-RANK]} \cdot RANK} \\ &\quad \cdot \left(\frac{\beta_{RANKL} + P_{RANKL-d} (P_{RANKL}^{\Psi_{bm}})}{\beta_{RANKL} + RANKL_{eff} \cdot \tilde{D}_{RANKL}} \right) \end{aligned} \quad (3.27)$$

$$RANK = N_{OC_p}^{RANK} \cdot OC_p \quad (3.28)$$

3.1.6 Mineralización

Tras la retirada de hueso por parte de los osteoclastos, los osteoblastos depositan colágeno y agua. Dicha cantidad de agua representa un límite en el contenido de mineral y parte de ella se transformará en cristales de hidroxiapatita. El proceso de mineralización consta de tres etapas: primeramente, una fase inicial (en inglés, *Mineralization Lag Time*) que dura entre 6 y 22 días y donde no hay deposición de mineral, posteriormente una fase primaria, donde se alcanza el 70% de contenido de mineral y finalmente una fase secundaria, que se puede representar como una exponencial decreciente y que termina cuando se alcanza el 95% de mineral.

De acuerdo con Martínez-Reina et al. (2008), las ecuaciones que definen el proceso de mineralización se muestran a continuación:

En un determinado volumen de referencia (V_t), podemos encontrar hueso (V_b) y poros (V_p):

$$V_t = V_b + V_p \quad (3.29)$$

A su vez, el volumen de hueso (V_b) se compone de volumen de mineral (V_m), colágeno (V_o) y agua (V_w):

$$V_b = V_m + V_o + V_w \quad (3.30)$$

A continuación, se muestran los volúmenes específicos y la fracción de ceniza en función de estos:

$$v_i = \frac{V_i}{V_m + V_o + V_w} \quad i = m, o \text{ ó } w \quad (3.31)$$

$$v_o + v_m + v_w = 1 \quad (3.32)$$

$$\alpha = \frac{m_m}{m_m + m_o} = \frac{\rho_m V_m}{\rho_m V_m + \rho_o V_o} \rightarrow x \frac{1}{V_b} \rightarrow \left[\frac{\rho_m v_m}{\rho_m v_m + \rho_o v_o} \right] \quad (3.33)$$

Anteriormente, comentábamos que el proceso de mineralización consta de tres etapas. Cómo va variando el volumen de mineral en cada una de las tres etapas es una función definida a trozos que se muestra a continuación:

$$v_m(t) = \begin{cases} 0 & \text{si } t \leq t_{mlt} \\ v_{prim} \frac{t - t_{mlt}}{t_{prim}} & \text{si } t_{mlt} < t \leq t_{prim} + t_{mlt} \\ v_{m\acute{a}x} - (v_{m\acute{a}x} - v_{prim})e^{-\kappa(t - t_{prim} - t_{mlt})} & \text{si } t_{prim} + t_{mlt} < t \end{cases} \quad (3.34)$$

Otro aspecto importante es analizar cómo varía el volumen de mineral en cada incremento de tiempo. La siguiente expresión tiene tres términos. El primero hace referencia al mineral presente en un determinado instante de tiempo, el segundo al mineral reabsorbido de la matriz ósea y el tercero al incremento de mineral debido a la mineralización. La expresión matemática de este último se muestra en las ecuaciones 3.36 y 3.37:

$$v_m(t + \Delta t) = \frac{v_m(t)V_b(t) - v_m(t)\Delta V_r(t) + \Delta V_{min}(t + \Delta t)}{V_b(t + \Delta t)} \quad (3.35)$$

$$\Delta V_{min}(t + \Delta t) = \int_0^\infty \bar{V}_f(t, \tau) \frac{dv_m(\tau)}{d\tau} d\tau = \int_0^{t_R} \bar{V}_f(t, \tau) \frac{dv_m(\tau)}{d\tau} d\tau \quad (3.36)$$

$$\Delta V_{min}(t + \Delta t) \approx \sum_{i=0}^{t_R} \bar{V}_f(t, i) [v_m(i) - v_m(i-1)] \quad (3.37)$$

Por último, para calcular el volumen de tejido que se formó hace τ días, se utiliza un algoritmo de cola con metodología FIFO. Matemáticamente, se define con la siguiente expresión y se resume esquemáticamente en la Figura 3.1.1:

$$\bar{V}_f(t + 1, i + 1) = \bar{V}_f(t, i) - \Delta V_r(t) \frac{\bar{V}_f(t, i)}{V_b(t)} \quad (3.38)$$

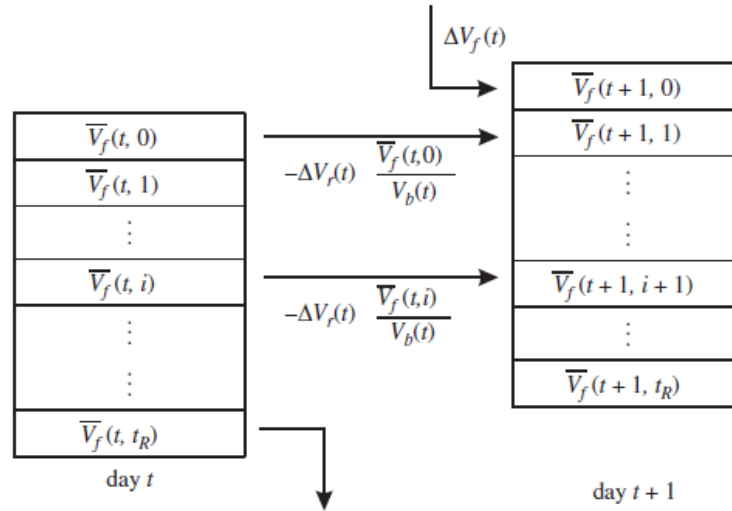


Figura 3.1.1. Martínez-Reina et al. (2008). On the role of bone damage in calcium homeostasis. Se utiliza un algoritmo de colas FIFO (del inglés, First In First Out) y se tienen en cuenta los volúmenes de tejidos reabsorbidos y formados en el día “t”.

3.1.7 Inclusión de la osteoporosis y del denosumab

En este apartado vamos a evaluar el efecto que tiene la PMO sobre la remodelación ósea sana y la inclusión del denosumab como tratamiento. Para ello, nos ayudaremos del trabajo llevado a cabo por Scheiner et al. (2014).

Por un lado, la PMO disminuye la cantidad de estrógenos lo que hace aumentar la concentración de osteoclastos y osteoblastos con respecto al estado de ausencia de la enfermedad. Sin embargo, la concentración de osteoclastos activos es superior a la de osteoblastos activos y, por lo tanto, nos encontramos en una situación de pérdida ósea.

En la PMO, se distinguen dos fases. Una primera en la que la concentración de RANKL es superior a la de OPG, que se traduce en una mayor actividad de los osteoclastos y una secundaria, en la que la pérdida ósea se modera y el exceso de RANKL se reduce. Dicho aumento inicial y disminución posterior se tiene en cuenta en las siguientes ecuaciones.

$$P_{RANKL}^{PMO} = P_{RANKL}^{PMO,ini} \cdot \varphi_{PMO}^{RANKL} \quad (3.39)$$

$$\varphi_{PMO}^{RANKL} = \frac{\xi^2}{\xi^2 + \left[\frac{t - t_{PMO,ini}}{\tau_{PMO}^{RANKL}} \right]^2} \quad (3.40)$$

Donde ξ y τ_{PMO}^{RANKL} tratan de ajustar la curva de Lorentz. Además, la reducción de estrógenos trae consigo una disminución en la capacidad de mecanorespuesta debido a que aumenta la apoptosis de los osteocitos. Dicho efecto se modela en la siguiente expresión:

$$\varphi_{PMO}^{mech} = \exp\left(\frac{t - t_{PMO,ini}}{\tau_{PMO}^{mech}}\right) \quad (3.41)$$

Tras comentar las consecuencias de la osteoporosis sobre el proceso de remodelación ósea es el turno de analizar el efecto del tratamiento que vamos a estudiar, el denosumab. Este anticuerpo monoclonal tiene gran afinidad por el RANK y evita así que el RANKL se una al RANK y se inicie el proceso de remodelación. Por lo tanto, el denosumab va a disminuir la concentración de RANKL de la siguiente forma:

$$\begin{aligned} RANKL &= \frac{RANKL_{eff}}{1 + K_{A1,[RANKL-OPG]} \cdot OPG + K_{A2,[RANKL-RANK]} \cdot RANK + \zeta \cdot K_{A3,Denosumab} \cdot C_{Denosumab}} \\ &\cdot \left(\frac{\beta_{RANKL} + P_{RANKL-d} (P_{RANKL}^{\Psi_{bm}})}{\beta_{RANKL} + RANKL_{eff} \cdot \tilde{D}_{RANKL}} \right) \end{aligned} \quad (3.42)$$

$$[RANKL - RANK] = K_{A2,[RANKL-RANK]} \cdot RANKL \cdot RANK \quad (3.43)$$

$$\pi_{act,OC_p}^{RANKL} = \frac{[RANKL - RANK]}{[RANKL - RANK] + K_{A3,Denosumab}} \quad (3.44)$$

Todo lo anteriormente mencionado se resume en la siguiente Figura:

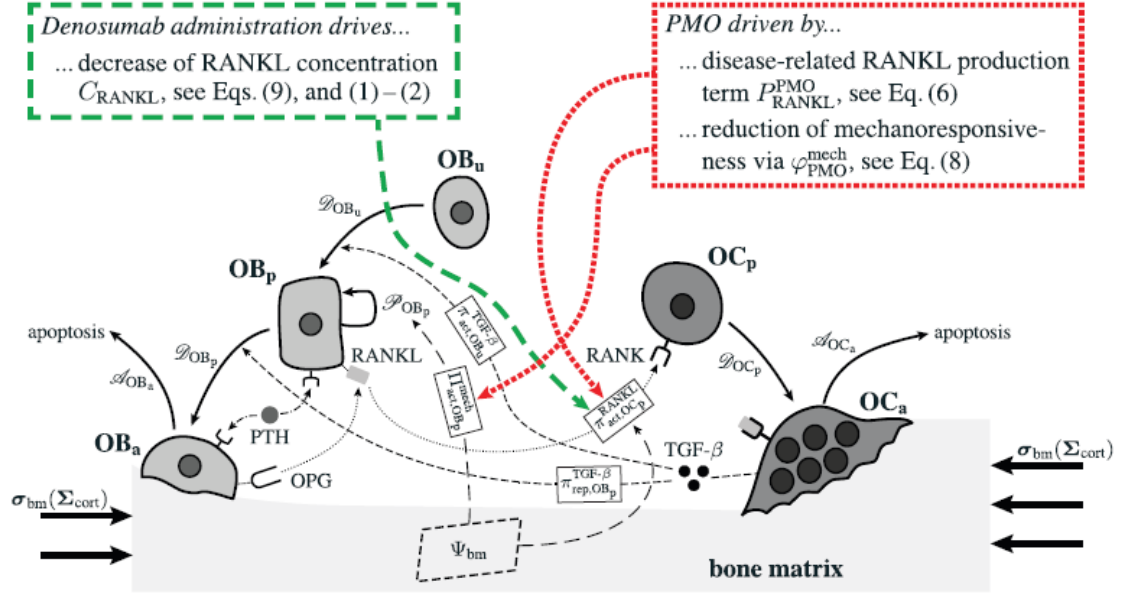


Figura 3.1.2. Scheiner et al. (2014). Mathematical modeling of postmenopausal osteoporosis and its treatment by the anticatabolic drug denosumab. Efecto de la PMO en la remodelación ósea y del tratamiento denosumab. Este se une al RANK para evitar la interacción RANKL-RANK y el consecuente inicio de la remodelación.

3.1.8 Sistema de ecuaciones diferenciales

A continuación, se detalla el sistema de ecuaciones diferenciales de Pivonka et al. (2013), donde se muestra la evolución con respecto al tiempo del número de células por unidad de volumen de osteoblastos y osteoclastos precursores y osteoblastos y osteoclastos activos. La concentración de las células no comprometidas se asume constante.

En las ecuaciones se encuentran todos los conceptos estudiados previamente: las funciones de Hill, la regulación geométrica y los procesos celulares de proliferación, diferenciación y apoptosis.

$$\begin{aligned} \frac{d}{dt} OB_p = & \left\{ g_{OB_u}(S_V) D_{OB_u} \pi_{act,OB_u}^{TGF\beta} \right\} OB_u + \left\{ P_{RANKL}^{\Psi_{bm}} \Pi_{act,OB_p}^{\Psi_{bm}} \right\} OB_p \\ & - \left\{ g_{OB_p}(S_V) D_{OB_p} \pi_{rep,OB_p}^{TGF\beta} \right\} OB_p \end{aligned} \quad (3.45)$$

$$\frac{d}{dt} OB_a = \left\{ g_{OB_p}(S_V) D_{OB_p} \pi_{rep,OB_p}^{TGF\beta} \right\} OB_p - A_{OB_a} OB_a \quad (3.46)$$

$$\frac{d}{dt} OC_p = \left\{ g_{OC_u}(S_V) D_{OC_u} \pi_{act,OC_u}^{MCSF} \pi_{act,OC_p}^{RANKL} \right\} OC_u - \left\{ g_{OC_p}(S_V) D_{OC_p} \pi_{act,OC_p}^{RANKL} \right\} OC_p \quad (3.47)$$

$$\frac{d}{dt} OC_a = \left\{ g_{OC_p}(S_V) D_{OC_p} \pi_{act,OC_p}^{RANKL} \right\} OC_p - \left\{ A_{OC_a} \pi_{act,OC_a}^{TGF\beta} \right\} OC_a \quad (3.48)$$

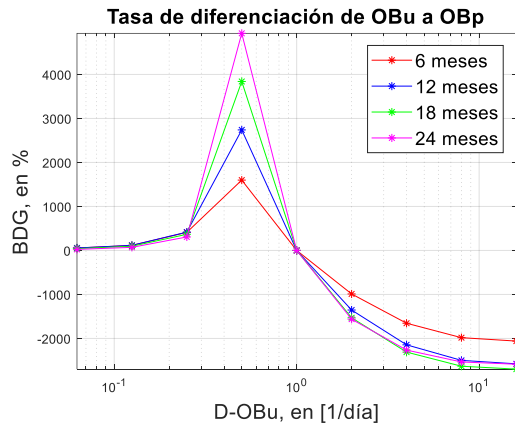
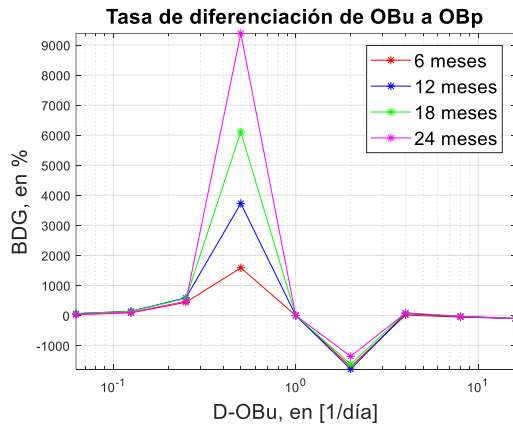
Además, la remodelación ósea también modifica la porosidad vascular o la fracción volumétrica de matriz ósea de la siguiente manera:

$$\frac{d}{dt} f_{vas} = -\frac{d}{dt} f_{bm} = -k_{form} OB_a + k_{res} OC_a \quad (3.49)$$

3.2 Análisis de sensibilidad

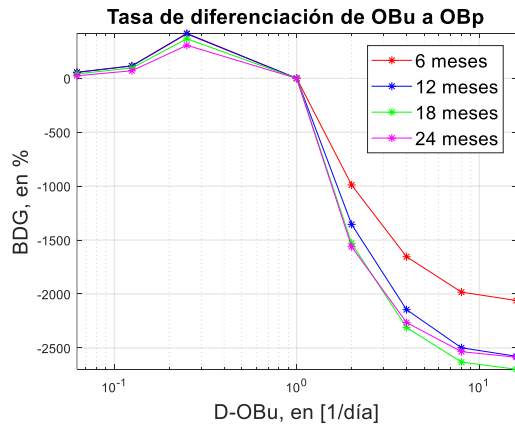
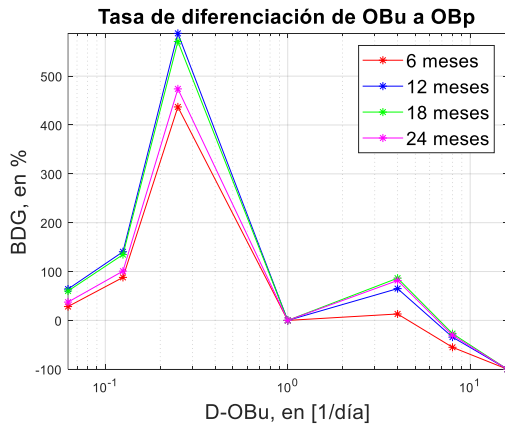
En este apartado vamos a realizar un análisis de sensibilidad de los distintos parámetros (ver Tablas 3.1.2 y 3.1.3), es decir, vamos a analizar cómo evoluciona el BDG (en inglés, *Bone Density Gain*) cuando variamos el valor de los distintos parámetros con respecto al valor nominal, con el objetivo de visualizar el efecto que los distintos parámetros tienen sobre la salida del modelo matemático. En el eje vertical de las gráficas se representa lo siguiente:

$$\Delta_{BDG} = \frac{BDG - BDG^{nom}}{BDG^{nom}} * 100 \quad (3.50)$$

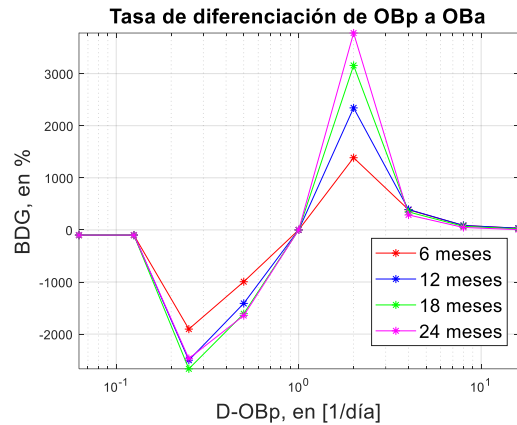
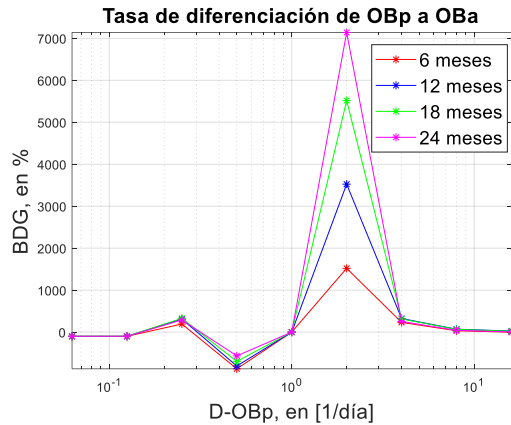


Figuras 3.2.1 y 3.2.2. Análisis de sensibilidad para la tasa de diferenciación de OBU a OBp. En ambas figuras, se representa en el eje horizontal, en escala logarítmica, el valor nominal de dicha variable [$x \frac{1}{16}$, $x \frac{1}{8}$, $x \frac{1}{4}$, $x \frac{1}{2}$, $x1$, $x2$, $x4$, $x8$, $x16$] y se evalúa el BDG en los meses 6, 12, 18 y 24, para la vértebra lumbar (izquierda) y la cadera (derecha).

Observando las gráficas superiores comprobamos que al multiplicar el valor nominal de D_{OBU} por ciertos coeficientes aparecen picos que evidencian que el sistema se vuelve inestable (ver en las Figuras 3.2.1 y 3.2.2 el valor que se llega a alcanzar en el eje vertical). Para evitarlos, obviamos dichos coeficientes, resultando las gráficas que se muestran a continuación:

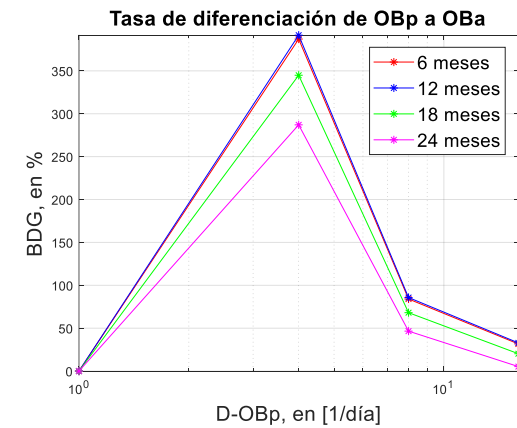
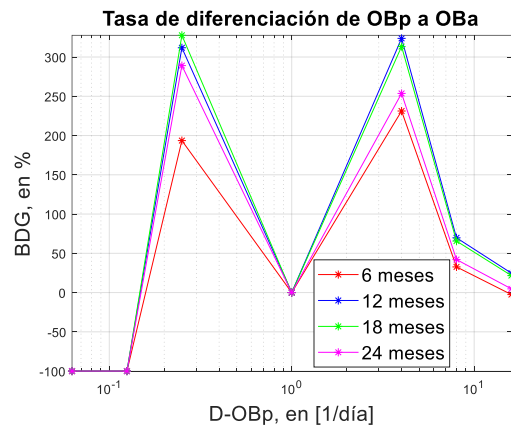


Figuras 3.2.3 y 3.2.4. Análisis de sensibilidad corregido para la tasa de diferenciación de OBU a OBp. Para la vértebra lumbar (izquierda), se ha multiplicado el valor nominal por [$x \frac{1}{16}$, $x \frac{1}{8}$, $x \frac{1}{4}$, $x1$, $x4$, $x8$, $x16$] y para la cadera (derecha) por [$x \frac{1}{16}$, $x \frac{1}{8}$, $x \frac{1}{4}$, $x1$, $x2$, $x4$, $x8$, $x16$], evaluándose el BDG en los meses 6, 12, 18 y 24.

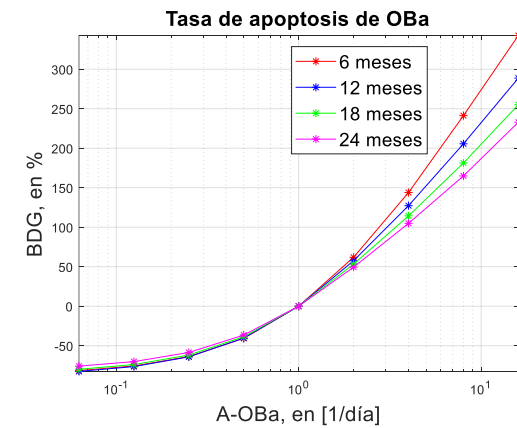
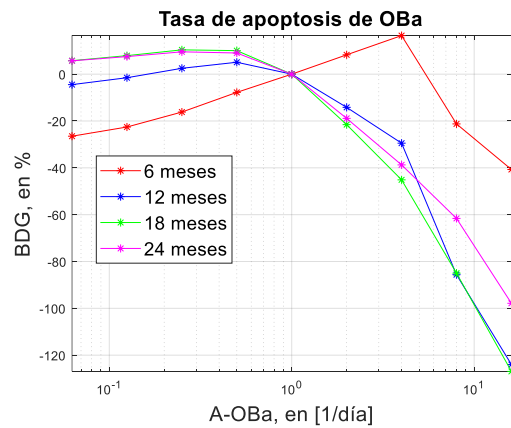


Figuras 3.2.5 y 3.2.6. Análisis de sensibilidad para la tasa de diferenciación de OBp a OBa. En ambas figuras, se representa en el eje horizontal, en escala logarítmica, el valor nominal de dicha variable [$x \frac{1}{16}, x \frac{1}{8}, x \frac{1}{4}, x \frac{1}{2}, x1, x2, x4, x8, x16$] y se evalúa el BDG en los meses 6, 12, 18 y 24, para la vértebra lumbar (izquierda) y la cadera (derecha).

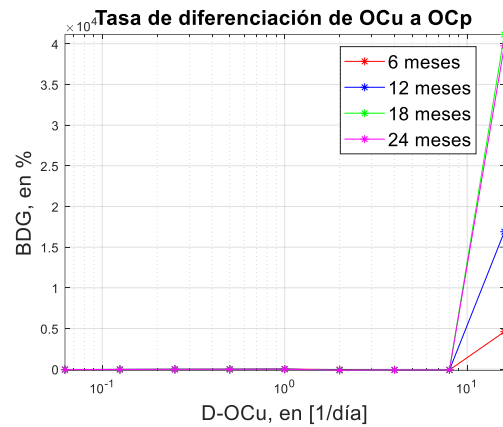
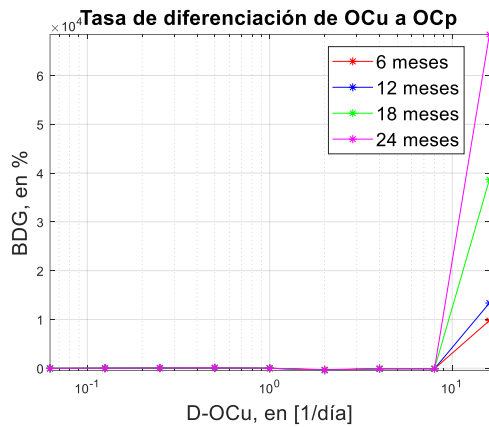
Al igual que pasaba en el caso anterior, para D_{OBp} también vuelven aparecer picos donde el modelo no responde bien. A continuación, se muestra el análisis de sensibilidad corregido:



Figuras 3.2.7 y 3.2.8. Análisis de sensibilidad corregido para la tasa de diferenciación de OBp a OBa. Para la vértebra lumbar (izquierda), se ha multiplicado el valor nominal por [$x \frac{1}{16}, x \frac{1}{8}, x \frac{1}{4}, x1, x4, x8, x16$] y para la cadera (derecha) por [$x1, x4, x8, x16$], evaluándose el BDG en los meses 6, 12, 18 y 24.

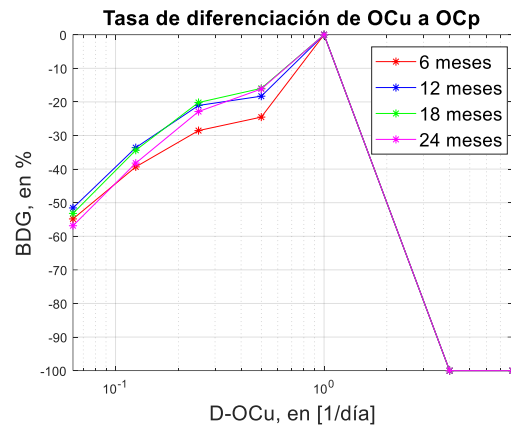
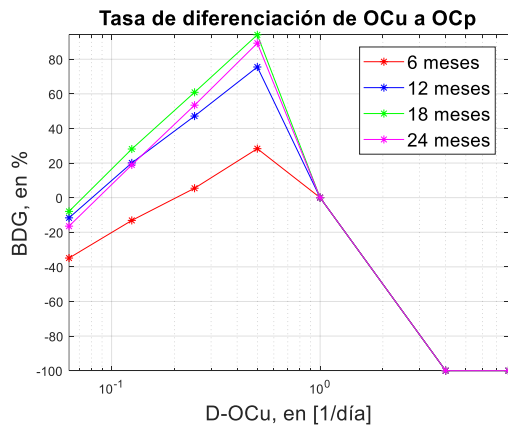


Figuras 3.2.9 y 3.2.10. Análisis de sensibilidad para la tasa de apoptosis de OBa. En ambas figuras, se representa en el eje horizontal, en escala logarítmica, el valor nominal de dicha variable [$x \frac{1}{16}, x \frac{1}{8}, x \frac{1}{4}, x \frac{1}{2}, x1, x2, x4, x8, x16$] y se evalúa el BDG en los meses 6, 12, 18 y 24, para la vértebra lumbar (izquierda) y la cadera (derecha).

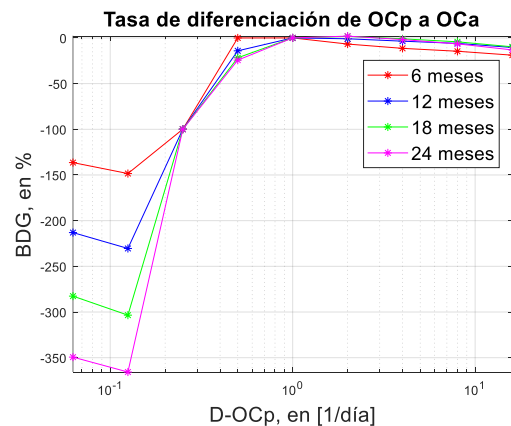
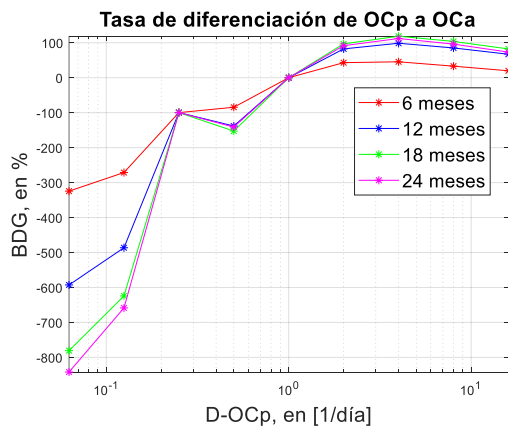


Figuras 3.2.11 y 3.2.12. Análisis de sensibilidad para la tasa de diferenciación de OCu a OCp. En ambas figuras, se representa en el eje horizontal, en escala logarítmica, el valor nominal de dicha variable [$x \frac{1}{16}$, $x \frac{1}{8}$, $x \frac{1}{4}$, $x \frac{1}{2}$, $x1$, $x2$, $x4$, $x8$, $x16$] y se evalúa el BDG en los meses 6, 12, 18 y 24, para la vértebra lumbar (izquierda) y la cadera (derecha).

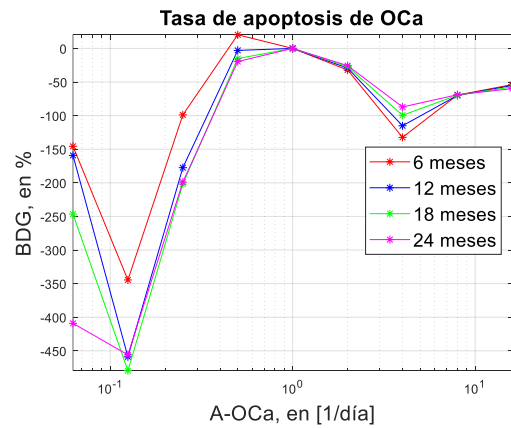
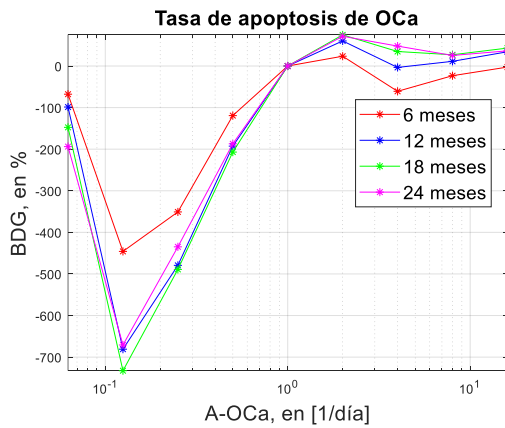
De nuevo, es necesario corregir el análisis de sensibilidad previo relativo a D_{OCu} ya que según las Figuras 3.2.11 y 3.2.12, el modelo matemático se vuelve inestable cuando multiplicamos el valor nominal de la variable considerada por 16. El resultado de los nuevos coeficientes considerados se muestra a continuación:



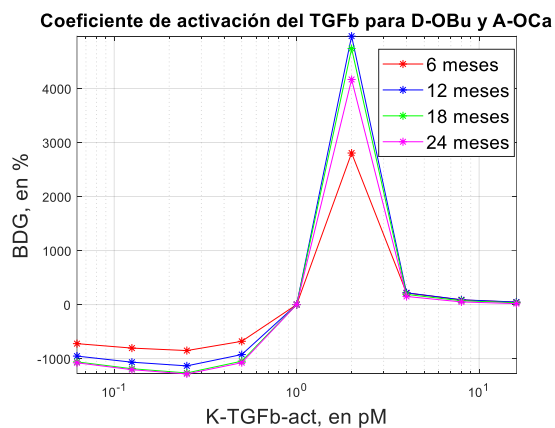
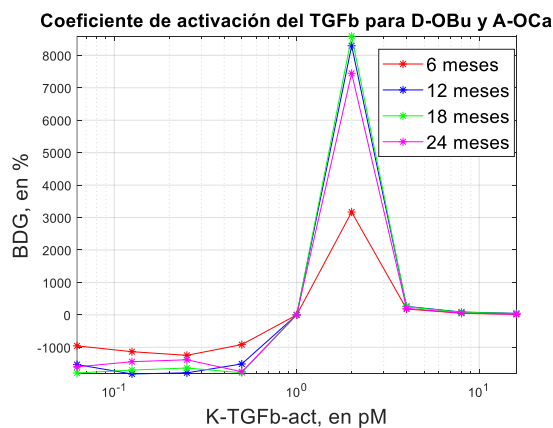
Figuras 3.2.13 y 3.2.14. Análisis de sensibilidad corregido para la tasa de diferenciación de OCu a OCp. Para la vértebra lumbar (izquierda), se ha multiplicado el valor nominal por [$x \frac{1}{16}$, $x \frac{1}{8}$, $x \frac{1}{4}$, $x \frac{1}{2}$, $x1$, $x4$, $x8$] y para la cadera (derecha) por [$x \frac{1}{16}$, $x \frac{1}{8}$, $x \frac{1}{4}$, $x \frac{1}{2}$, $x1$, $x4$, $x8$], evaluándose el BDG en los meses 6, 12, 18 y 24.



Figuras 3.2.15 y 3.2.16. Análisis de sensibilidad para la tasa de diferenciación de OCp a OCa. En ambas figuras, se representa en el eje horizontal, en escala logarítmica, el valor nominal de dicha variable [$x \frac{1}{16}, x \frac{1}{8}, x \frac{1}{4}, x \frac{1}{2}, x1, x2, x4, x8, x16$] y se evalúa el BDG en los meses 6, 12, 18 y 24, para la vértebra lumbar (izquierda) y la cadera (derecha).

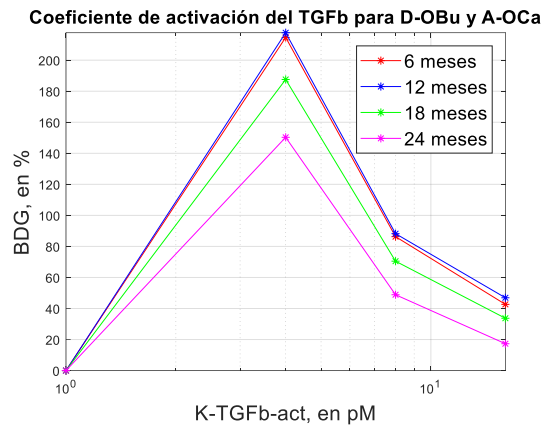
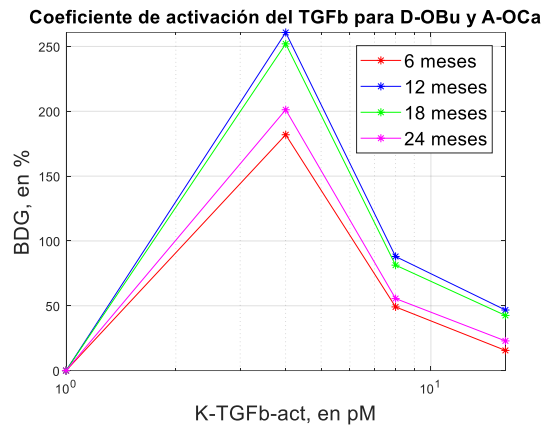


Figuras 3.2.17 y 3.2.18. Análisis de sensibilidad para la tasa de apoptosis de OCa. En ambas figuras, se representa en el eje horizontal, en escala logarítmica, el valor nominal de dicha variable [$x \frac{1}{16}, x \frac{1}{8}, x \frac{1}{4}, x \frac{1}{2}, x1, x2, x4, x8, x16$] y se evalúa el BDG en los meses 6, 12, 18 y 24, para la vértebra lumbar (izquierda) y la cadera (derecha).

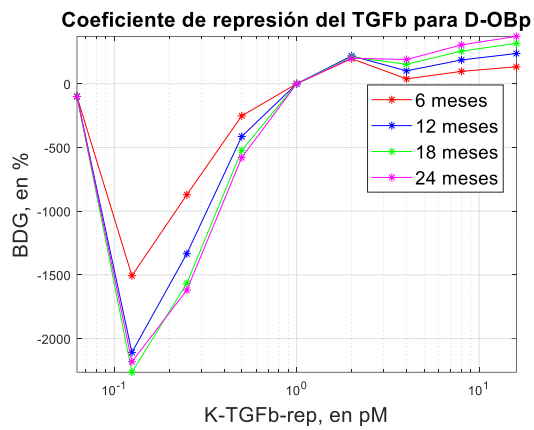
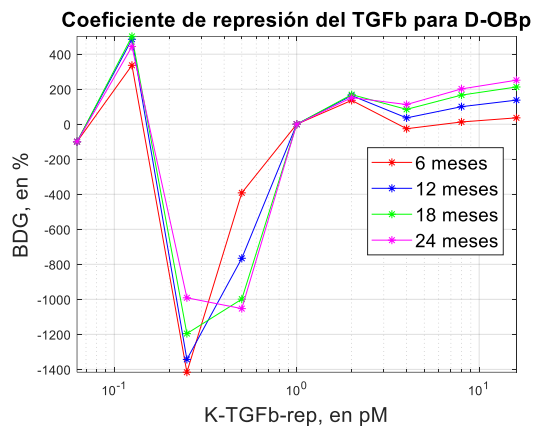


Figuras 3.2.19 y 3.2.20. Análisis de sensibilidad para el coeficiente de activación del TGF- β en los procesos celulares de diferenciación de OBu y apoptosis de OCa. En ambas figuras, se representa en el eje horizontal, en escala logarítmica, el valor nominal de dicha variable [$x \frac{1}{16}, x \frac{1}{8}, x \frac{1}{4}, x \frac{1}{2}, x1, x2, x4, x8, x16$] y se evalúa el BDG en los meses 6, 12, 18 y 24, para la vértebra lumbar (izquierda) y la cadera (derecha).

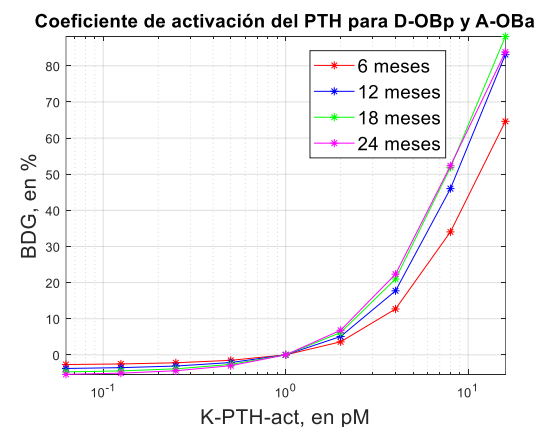
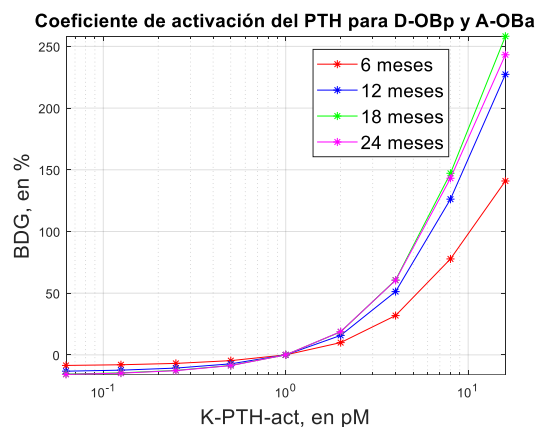
El coeficiente de activación del TGF- β es un parámetro que influye mucho en el modelo y cualquier variación en él puede llevar a resultados inesperados. Con el objetivo de suavizar los picos presentes en las gráficas superiores, se muestra el análisis de sensibilidad corregido:



Figuras 3.2.21 y 3.2.22. Análisis de sensibilidad corregido para el coeficiente de activación del TGF- β en los procesos celulares de diferenciación de OBu y apoptosis de OCa. Para la vértebra lumbar (izquierda), se ha multiplicado el valor nominal por $[x1, x4, x8 \text{ y } x16]$ y para la cadera (derecha) por $[x1, x4, x8, x16]$, evaluándose el BDG en los meses 6, 12, 18 y 24.

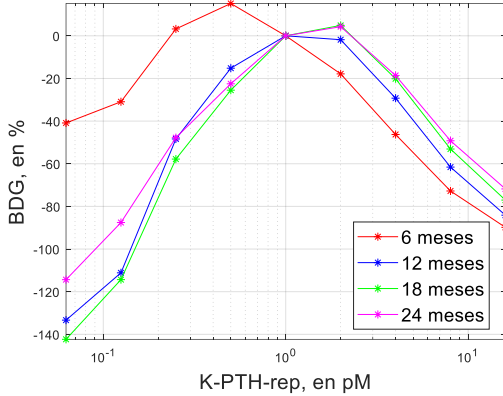


Figuras 3.2.23 y 3.2.24. Análisis de sensibilidad para el coeficiente de represión del TGF- β en la diferenciación de OBp. En ambas figuras, se representa en el eje horizontal, en escala logarítmica, el valor nominal de dicha variable $[x \frac{1}{16}, x \frac{1}{8}, x \frac{1}{4}, x \frac{1}{2}, x1, x2, x4, x8, x16]$ y se evalúa el BDG en los meses 6, 12, 18 y 24, para la vértebra lumbar (izquierda) y la cadera (derecha).

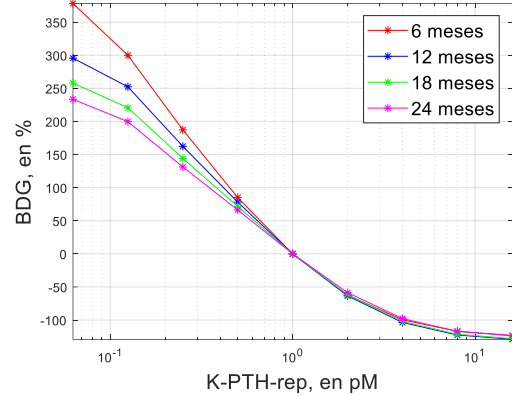


Figuras 3.2.25 y 3.2.26. Análisis de sensibilidad para el coeficiente de activación de la PTH en los procesos celulares de diferenciación de OBp y apoptosis de OBa. En ambas figuras, se representa en el eje horizontal, en escala logarítmica, el valor nominal de dicha variable $[x \frac{1}{16}, x \frac{1}{8}, x \frac{1}{4}, x \frac{1}{2}, x1, x2, x4, x8, x16]$ y se evalúa el BDG en los meses 6, 12, 18 y 24, para la vértebra lumbar (izquierda) y la cadera (derecha).

Coefficiente de represión del PTH para D-OBp y A-OBa

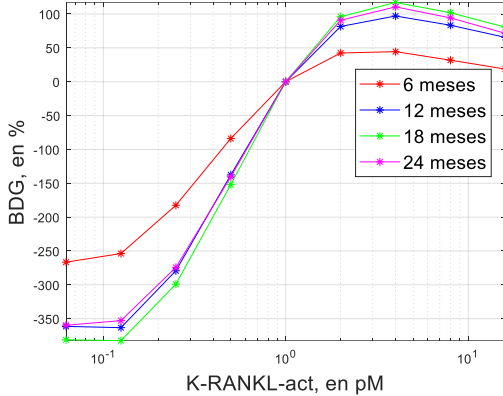


Coefficiente de represión del PTH para D-OBp y A-OBa

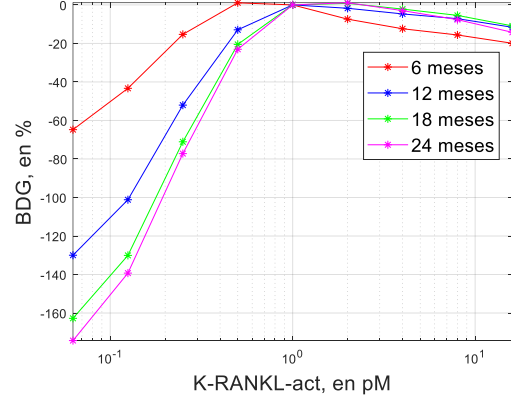


Figuras 3.2.27 y 3.2.28. Análisis de sensibilidad para el coeficiente de represión de la PTH en los procesos celulares de diferenciación de OBp y apoptosis de OBa. En ambas figuras, se representa en el eje horizontal, en escala logarítmica, el valor nominal de dicha variable [$x \frac{1}{16}$, $x \frac{1}{8}$, $x \frac{1}{4}$, $x \frac{1}{2}$, $x1$, $x2$, $x4$, $x8$, $x16$] y se evalúa el BDG en los meses 6, 12, 18 y 24, para la vértebra lumbar (izquierda) y la cadera (derecha).

Coefficiente de activación del RANKL para D-OCu y D-OCp

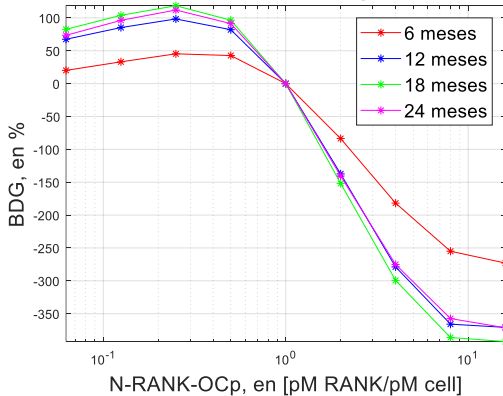


Coefficiente de activación del RANKL para D-OCu y D-OCp

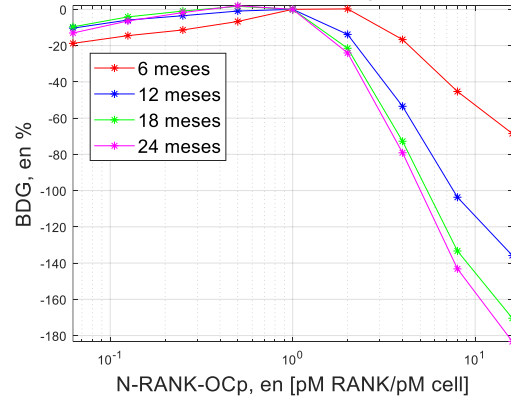


Figuras 3.2.29 y 3.2.30. Análisis de sensibilidad para el coeficiente de activación del RANKL en los procesos celulares de diferenciación de OCu y OCp. En ambas figuras, se representa en el eje horizontal, en escala logarítmica, el valor nominal de dicha variable [$x \frac{1}{16}$, $x \frac{1}{8}$, $x \frac{1}{4}$, $x \frac{1}{2}$, $x1$, $x2$, $x4$, $x8$, $x16$] y se evalúa el BDG en los meses 6, 12, 18 y 24, para la vértebra lumbar (izquierda) y la cadera (derecha).

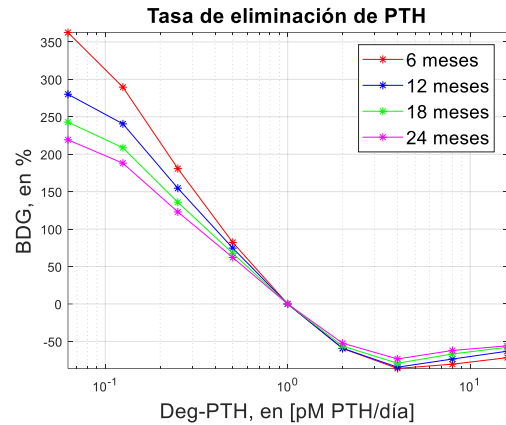
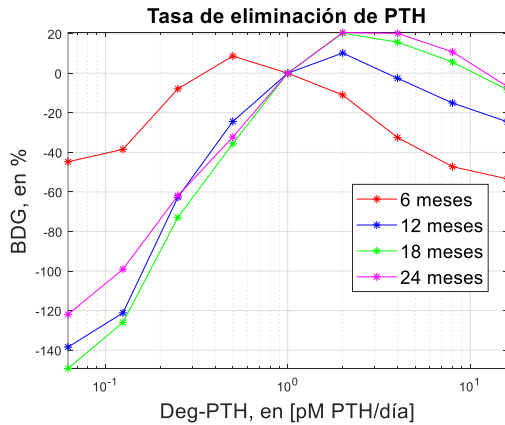
Cantidad de RANK sobre la superficie de OCp



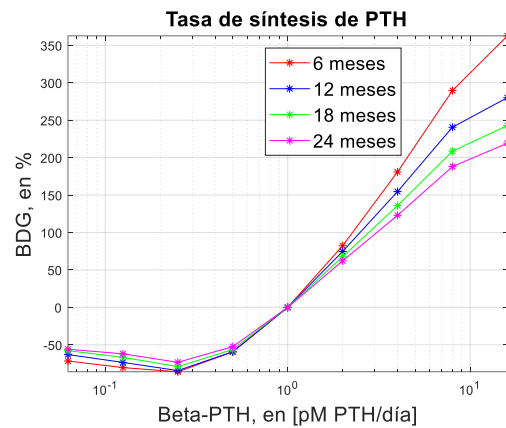
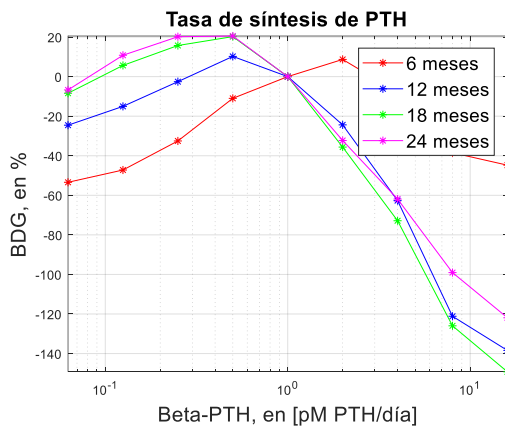
Cantidad de RANK sobre la superficie de OCp



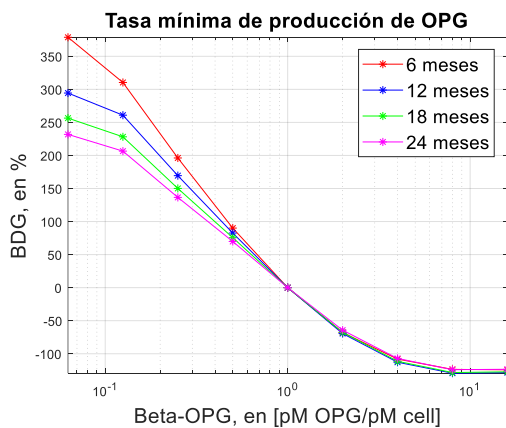
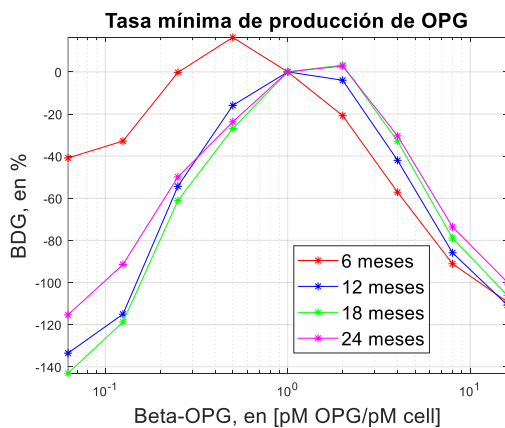
Figuras 3.2.31 y 3.2.32. Análisis de sensibilidad para la cantidad de RANK sobre la superficie de OCp. En ambas figuras, se representa en el eje horizontal, en escala logarítmica, el valor nominal de dicha variable [$x \frac{1}{16}$, $x \frac{1}{8}$, $x \frac{1}{4}$, $x \frac{1}{2}$, $x1$, $x2$, $x4$, $x8$, $x16$] y se evalúa el BDG en los meses 6, 12, 18 y 24, para la vértebra lumbar (izquierda) y la cadera (derecha).



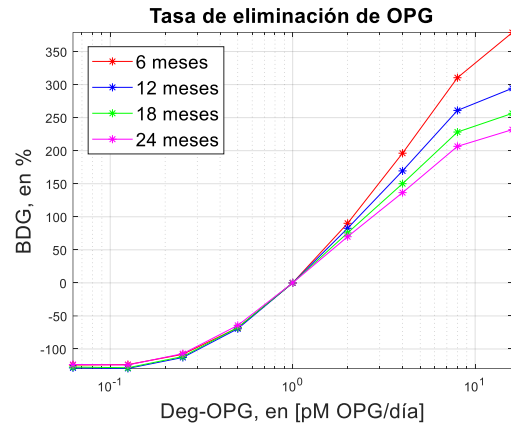
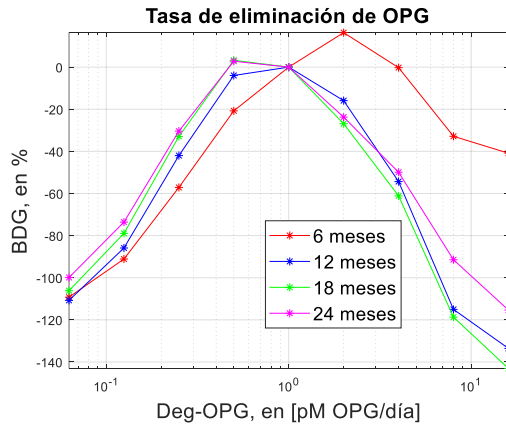
Figuras 3.2.33 y 3.2.34. Análisis de sensibilidad para la tasa de eliminación de PTH. En ambas figuras, se representa en el eje horizontal, en escala logarítmica, el valor nominal de dicha variable [$x_{\frac{1}{16}}$, $x_{\frac{1}{8}}$, $x_{\frac{1}{4}}$, $x_{\frac{1}{2}}$, x_1 , x_2 , x_4 , x_8 , x_{16}] y se evalúa el BDG en los meses 6, 12, 18 y 24, para la vértebra lumbar (izquierda) y la cadera (derecha).



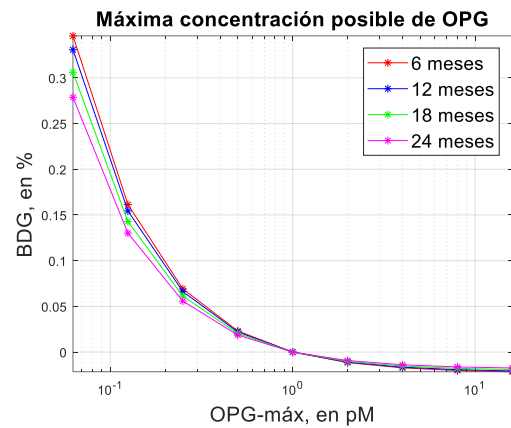
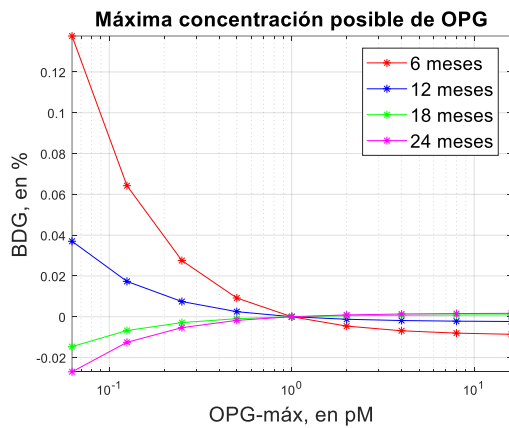
Figuras 3.2.35 y 3.2.36. Análisis de sensibilidad para la tasa de síntesis de PTH. En ambas figuras, se representa en el eje horizontal, en escala logarítmica, el valor nominal de dicha variable [$x_{\frac{1}{16}}$, $x_{\frac{1}{8}}$, $x_{\frac{1}{4}}$, $x_{\frac{1}{2}}$, x_1 , x_2 , x_4 , x_8 , x_{16}] y se evalúa el BDG en los meses 6, 12, 18 y 24, para la vértebra lumbar (izquierda) y la cadera (derecha).



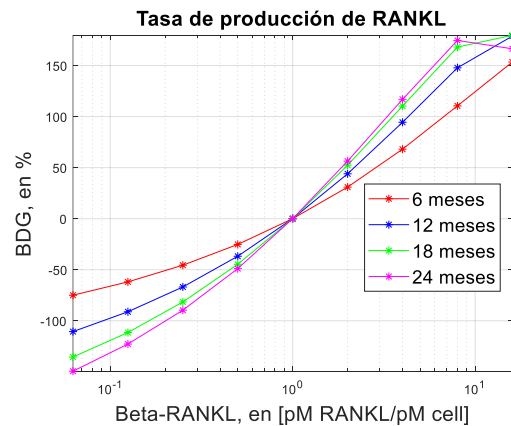
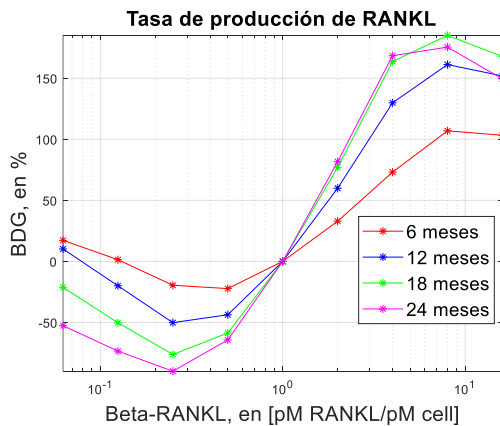
Figuras 3.2.37 y 3.2.38. Análisis de sensibilidad para la tasa de producción de OPG. En ambas figuras, se representa en el eje horizontal, en escala logarítmica, el valor nominal de dicha variable [$x_{\frac{1}{16}}$, $x_{\frac{1}{8}}$, $x_{\frac{1}{4}}$, $x_{\frac{1}{2}}$, x_1 , x_2 , x_4 , x_8 , x_{16}] y se evalúa el BDG en los meses 6, 12, 18 y 24, para la vértebra lumbar (izquierda) y la cadera (derecha).



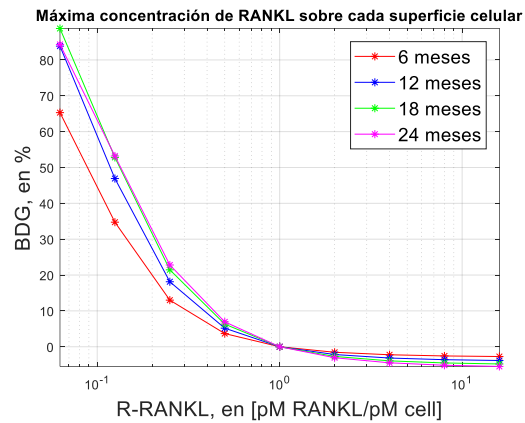
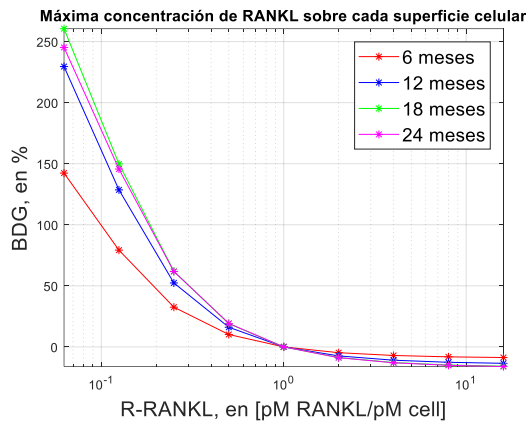
Figuras 3.2.39 y 3.2.40. Análisis de sensibilidad para la tasa de eliminación de OPG. En ambas figuras, se representa en el eje horizontal, en escala logarítmica, el valor nominal de dicha variable [$x \frac{1}{16}$, $x \frac{1}{8}$, $x \frac{1}{4}$, $x \frac{1}{2}$, $x1$, $x2$, $x4$, $x8$, $x16$] y se evalúa el BDG en los meses 6, 12, 18 y 24, para la vértebra lumbar (izquierda) y la cadera (derecha).



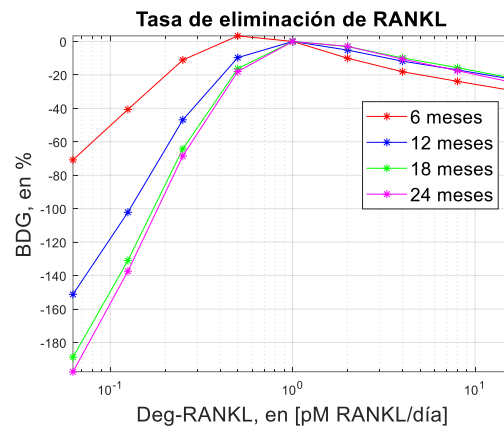
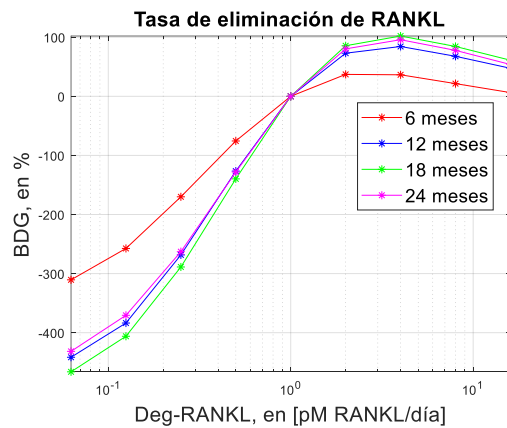
Figuras 3.2.41 y 3.2.42. Análisis de sensibilidad para la máxima concentración de OPG. En ambas figuras, se representa en el eje horizontal, en escala logarítmica, el valor nominal de dicha variable [$x \frac{1}{16}$, $x \frac{1}{8}$, $x \frac{1}{4}$, $x \frac{1}{2}$, $x1$, $x2$, $x4$, $x8$, $x16$] y se evalúa el BDG en los meses 6, 12, 18 y 24, para la vértebra lumbar (izquierda) y la cadera (derecha).



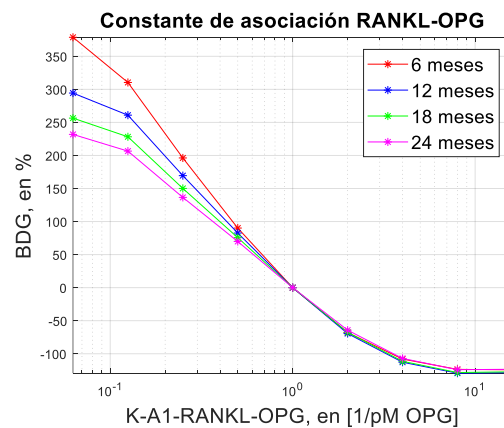
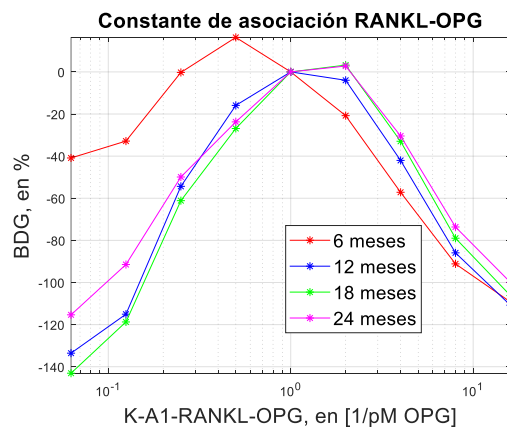
Figuras 3.2.43 y 3.2.44. Análisis de sensibilidad para la tasa de producción de RANKL. En ambas figuras, se representa en el eje horizontal, en escala logarítmica, el valor nominal de dicha variable [$x \frac{1}{16}$, $x \frac{1}{8}$, $x \frac{1}{4}$, $x \frac{1}{2}$, $x1$, $x2$, $x4$, $x8$, $x16$] y se evalúa el BDG en los meses 6, 12, 18 y 24, para la vértebra lumbar (izquierda) y la cadera (derecha).



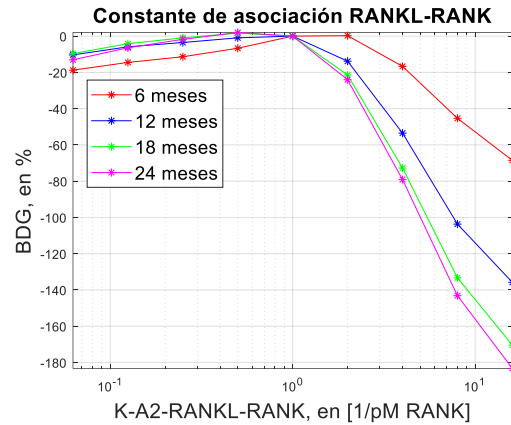
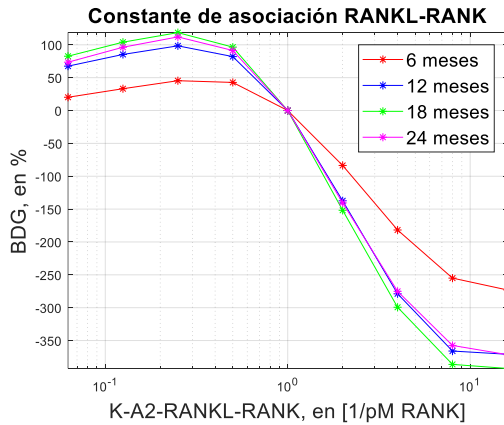
Figuras 3.2.45 y 3.2.46. Análisis de sensibilidad para la máxima concentración de RANKL sobre cada superficie celular. En ambas figuras, se representa en el eje horizontal, en escala logarítmica, el valor nominal de dicha variable [$x_{\frac{1}{16}}, x_{\frac{1}{8}}, x_{\frac{1}{4}}, x_{\frac{1}{2}}, x_1, x_2, x_4, x_8, x_{16}$] y se evalúa el BDG en los meses 6, 12, 18 y 24, para la vértebra lumbar (izquierda) y la cadera (derecha).



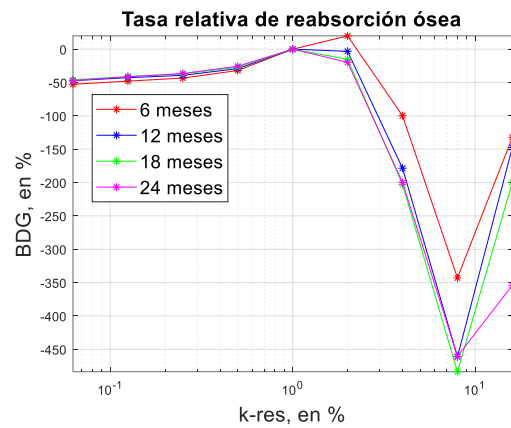
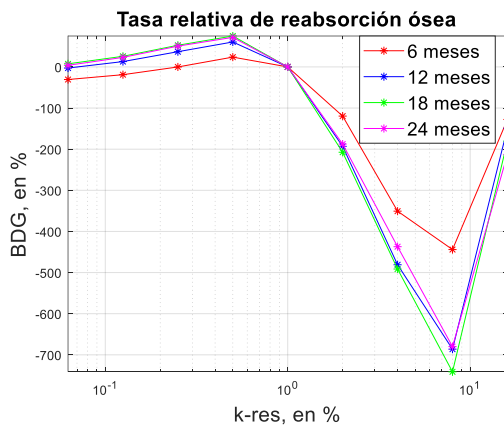
Figuras 3.2.47 y 3.2.48. Análisis de sensibilidad para la tasa eliminación de RANKL. En ambas figuras, se representa en el eje horizontal, en escala logarítmica, el valor nominal de dicha variable [$x_{\frac{1}{16}}, x_{\frac{1}{8}}, x_{\frac{1}{4}}, x_{\frac{1}{2}}, x_1, x_2, x_4, x_8, x_{16}$] y se evalúa el BDG en los meses 6, 12, 18 y 24, para la vértebra lumbar (izquierda) y la cadera (derecha).



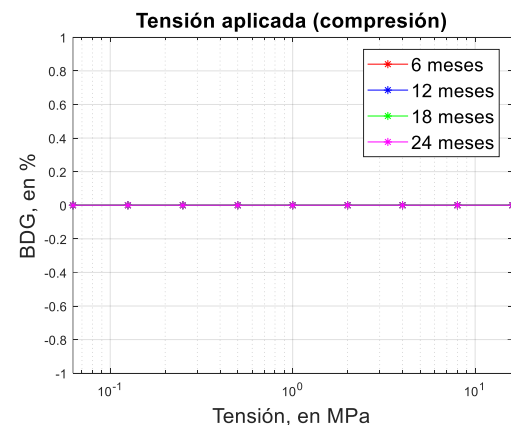
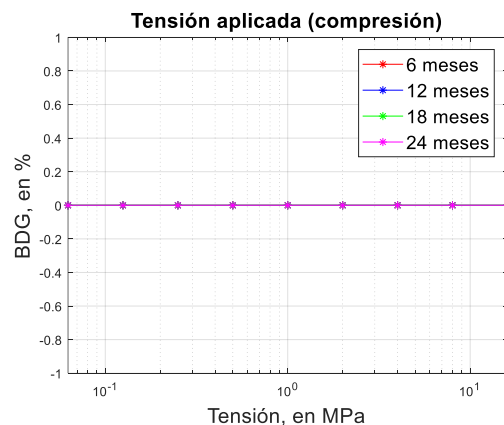
Figuras 3.2.49 y 3.2.50. Análisis de sensibilidad para la constante de asociación RANKL-OPG. En ambas figuras, se representa en el eje horizontal, en escala logarítmica, el valor nominal de dicha variable [$x_{\frac{1}{16}}, x_{\frac{1}{8}}, x_{\frac{1}{4}}, x_{\frac{1}{2}}, x_1, x_2, x_4, x_8, x_{16}$] y se evalúa el BDG en los meses 6, 12, 18 y 24, para la vértebra lumbar (izquierda) y la cadera (derecha).



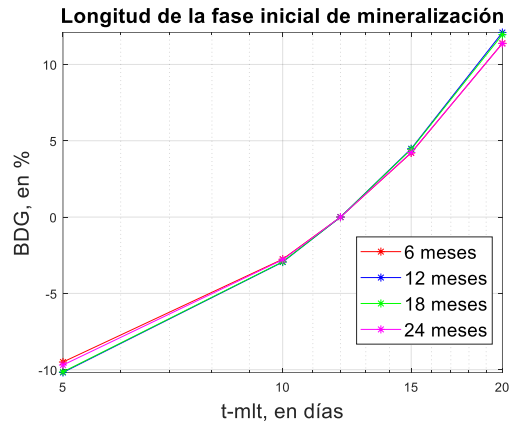
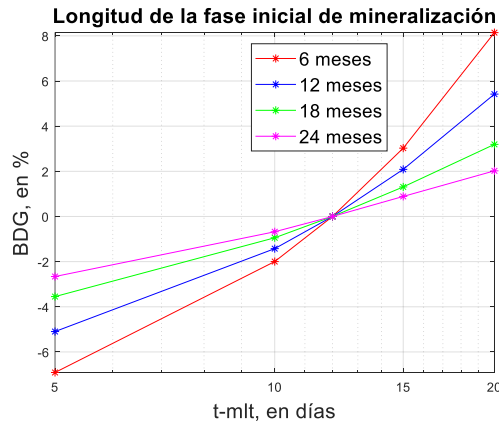
Figuras 3.2.51 y 3.2.52. Análisis de sensibilidad para la constante de asociación RANKL-RANK. En ambas figuras, se representa en el eje horizontal, en escala logarítmica, el valor nominal de dicha variable [$x_{\frac{1}{16}}$, $x_{\frac{1}{8}}$, $x_{\frac{1}{4}}$, $x_{\frac{1}{2}}$, x_1 , x_2 , x_4 , x_8 , x_{16}] y se evalúa el BDG en los meses 6, 12, 18 y 24, para la vértebra lumbar (izquierda) y la cadera (derecha).



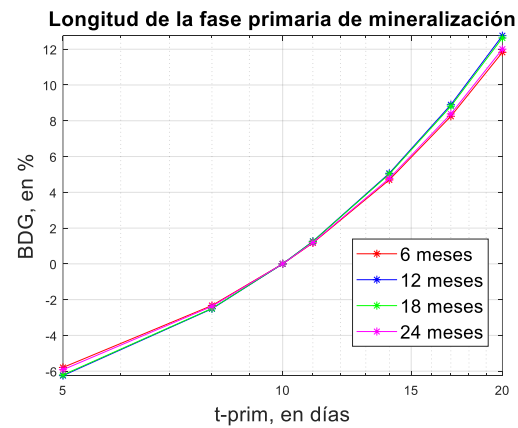
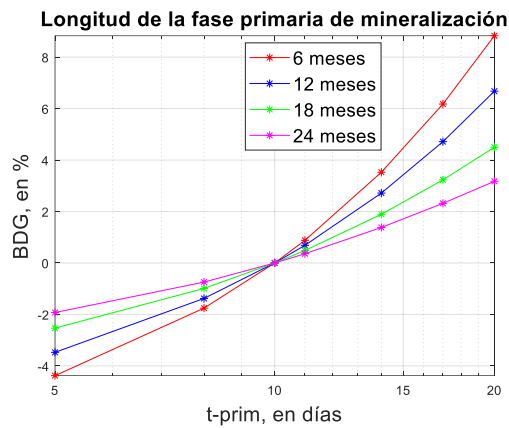
Figuras 3.2.53 y 3.2.54. Análisis de sensibilidad para la tasa relativa de reabsorción ósea. En ambas figuras, se representa en el eje horizontal, en escala logarítmica, el valor nominal de dicha variable [$x_{\frac{1}{16}}$, $x_{\frac{1}{8}}$, $x_{\frac{1}{4}}$, $x_{\frac{1}{2}}$, x_1 , x_2 , x_4 , x_8 , x_{16}] y se evalúa el BDG en los meses 6, 12, 18 y 24, para la vértebra lumbar (izquierda) y la cadera (derecha).



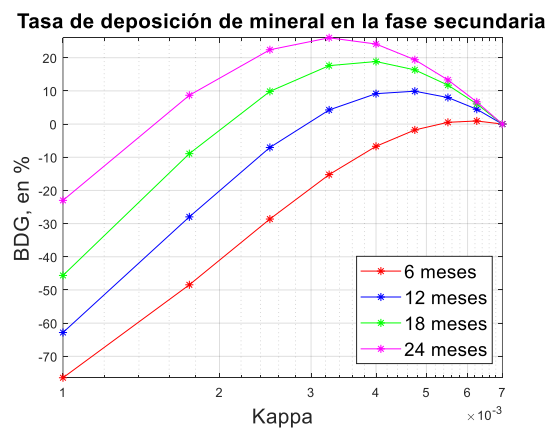
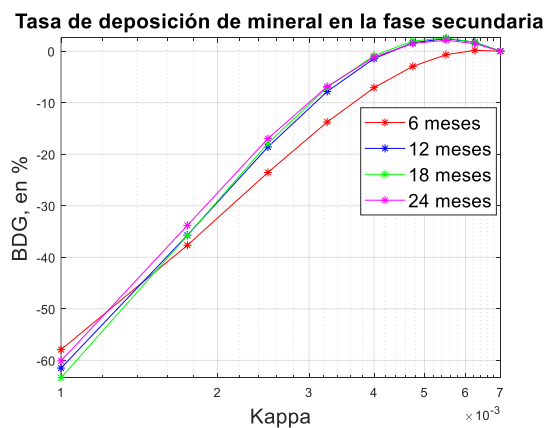
Figuras 3.2.55 y 3.2.56. Análisis de sensibilidad para la tensión aplicada de compresión. En ambas figuras, se representa en el eje horizontal, en escala logarítmica, el valor nominal de dicha variable [$x_{\frac{1}{16}}$, $x_{\frac{1}{8}}$, $x_{\frac{1}{4}}$, $x_{\frac{1}{2}}$, x_1 , x_2 , x_4 , x_8 , x_{16}] y se evalúa el BDG en los meses 6, 12, 18 y 24, para la vértebra lumbar (izquierda) y la cadera (derecha).



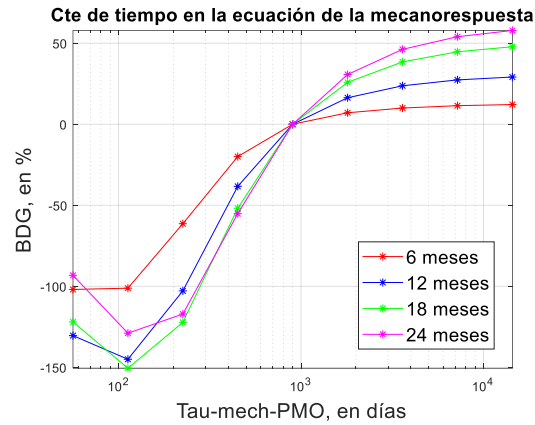
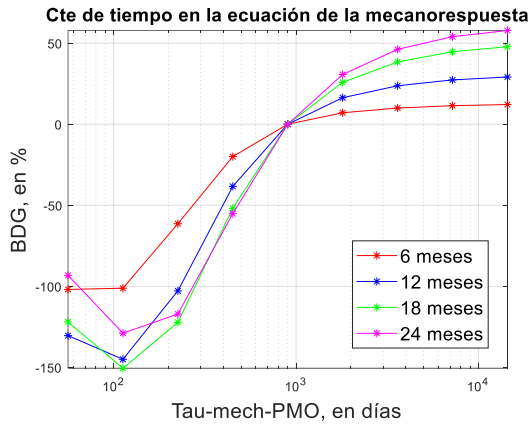
Figuras 3.2.57 y 3.2.58. Análisis de sensibilidad para la longitud de la fase inicial de mineralización. En ambas figuras, se representa en el eje horizontal, en escala logarítmica, los valores [0, 5, 10, 15, 20] y se evalúa el BDG en los meses 6, 12, 18 y 24, para la vértebra lumbar (izquierda) y la cadera (derecha).



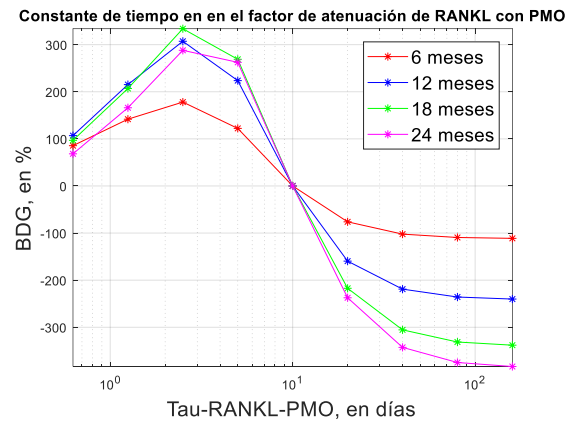
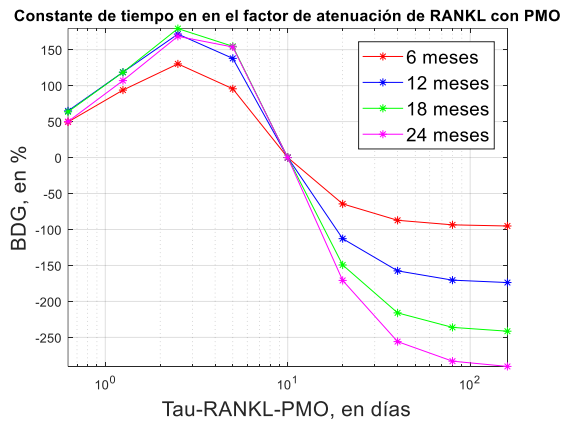
Figuras 3.2.59 y 3.2.60. Análisis de sensibilidad para la longitud de la fase primaria de mineralización. En ambas figuras, se representa en el eje horizontal, en escala logarítmica, los valores [5, 8, 11, 14, 17, 20] y se evalúa el BDG en los meses 6, 12, 18 y 24, para la vértebra lumbar (izquierda) y la cadera (derecha).



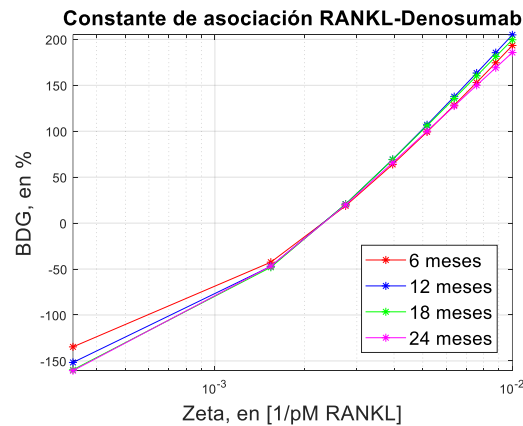
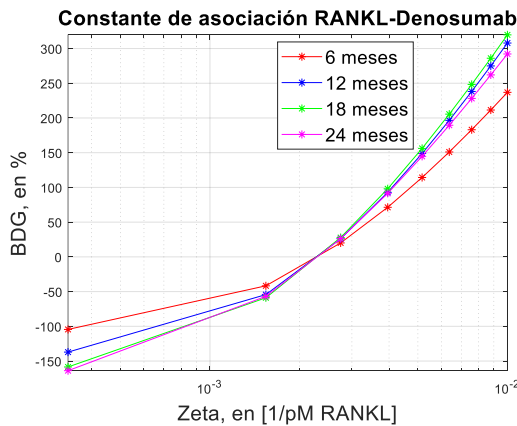
Figuras 3.2.61 y 3.2.62. Análisis de sensibilidad para la tasa de deposición de mineral en la fase secundaria de mineralización. En ambas figuras, se representa en el eje horizontal, en escala logarítmica, un rango de valores entre [0,001; 0,007] y se evalúa el BDG en los meses 6, 12, 18 y 24, para la vértebra lumbar (izquierda) y la cadera (derecha).



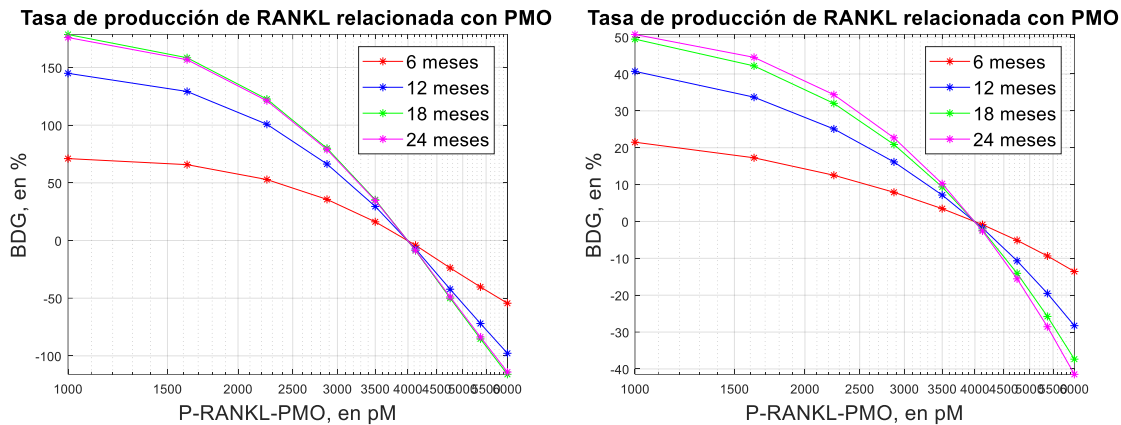
Figuras 3.2.63 y 3.2.64. Análisis de sensibilidad para la constante de tiempo en la ecuación de la mecanorespuesta. En ambas figuras, se representa en el eje horizontal, en escala logarítmica, el valor nominal de dicha variable [$x \frac{1}{16}$, $x \frac{1}{8}$, $x \frac{1}{4}$, $x \frac{1}{2}$, $x1$, $x2$, $x4$, $x8$, $x16$] y se evalúa el BDG en los meses 6, 12, 18 y 24, para la vértebra lumbar (izquierda) y la cadera (derecha).



Figuras 3.2.65 y 3.2.66. Análisis de sensibilidad para la constante de tiempo en el factor de atenuación de RANKL con PMO. En ambas figuras, se representa en el eje horizontal, en escala logarítmica, el valor nominal de dicha variable [$x \frac{1}{16}$, $x \frac{1}{8}$, $x \frac{1}{4}$, $x \frac{1}{2}$, $x1$, $x2$, $x4$, $x8$, $x16$] y se evalúa el BDG en los meses 6, 12, 18 y 24, para la vértebra lumbar (izquierda) y la cadera (derecha).



Figuras 3.2.67 y 3.2.68. Análisis de sensibilidad para la constante de asociación RANKL-Denosumab. En ambas figuras, se representa en el eje horizontal, en escala logarítmica, un rango de valores entre $[0,001; 0,03]x \frac{1}{3}$ y se evalúa el BDG en los meses 6, 12, 18 y 24, para la vértebra lumbar (izquierda) y la cadera (derecha).



Figuras 3.2.69 y 3.2.70. Análisis de sensibilidad para la tasa de producción de RANKL relacionado con PMO. En ambas figuras, se representa en el eje horizontal, en escala logarítmica, un rango de valores entre [1000; 6000] y se evalúa el BDG en los meses 6, 12, 18 y 24, para la vértebra lumbar (izquierda) y la cadera (derecha).

3.3 Proceso de automatización

Para la realización de este trabajo de fin de máster, hemos tenido como punto de partida un conjunto de funciones de Matlab, unos tratamientos y un programa principal a partir del cual el modelo matemático recibe una serie de entradas y ofrece las salidas correspondientes. Como iba a ser necesario realizar una gran cantidad de simulaciones, el objetivo inicial era automatizar dicho proceso, como se detalla a continuación.

Del total de funciones de Matlab, en “fu_modelparameters4” se recoge todos los parámetros del modelo con unos determinados valores que van a ir variando con las distintas simulaciones. El primer paso ha sido construir una matriz de entrada al modelo donde cada fila representa un conjunto de entrada para dicho modelo y cada columna es un determinado parámetro.

Posteriormente, hemos identificado qué funciones de Matlab llaman a “fu_modelparameters4”. Son las siguientes: “fu_bonecelldynamics4”, “fu_loadcase4” y “fu_steadystate4”. También, el programa principal “run_bonemodel_v4”.

Llegados a este punto, tenemos la matriz de entrada al modelo donde cada fila contiene un conjunto de parámetros con unos determinados valores que constituyen una entrada para el modelo y además, hemos localizado las funciones y el programa que necesitan dicha entrada. Un aspecto importante de este proceso es seguir un orden, conocer la fila, es decir, el número de simulación en la que nos encontramos.

El siguiente paso consiste en sincronizar todo lo anterior para dar forma a todo el proceso de automatización. Para ello, nos construimos un archivo llamado “Call_Model” que al ejecutarse, realiza lo siguiente:

- Guarda en un archivo “.mat” el número de simulación en la que nos encontramos.
- Un bucle “for” donde se llama al programa principal “run_bonemodel_v4” y se actualiza el número de simulación.
 - Dentro de “run_bonemodel_v4” se llama a “fu_bonecelldynamics4”, “fu_loadcase4” y “fu_steadystate4” y gracias al índice que contiene el número de simulación, que se carga del archivo “.mat”, dicha posición se conoce en todo momento.
- Al final de “run_bonemodel_v4”, la salida del modelo se va guardando por filas en una matriz de salida donde el número de fila viene dado por el número de simulación en la que nos encontramos.

Esquemáticamente, se resume en la siguiente Figura:

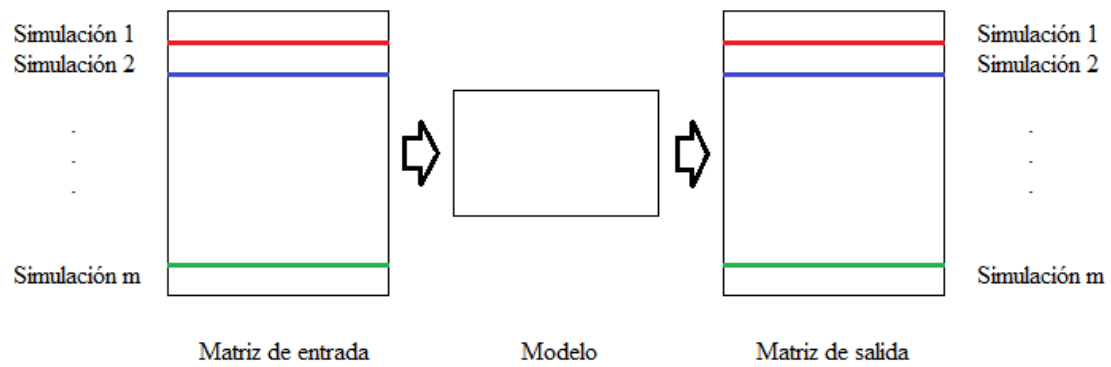


Figura 3.3.1. Proceso de automatización. Tanto los datos de entrada como los de salida del modelo se agrupan en matrices. Al realizarse el proceso de automatización, el modelo toma una fila de la matriz de entrada de índice “i”, realiza los cálculos correspondientes y la salida la almacena en una matriz en la posición “i”.

4. REDES NEURONALES

“Si entiendes cómo funciona el universo, en cierto modo, lo controlas”.

-Stephen Hawking-

Nodos, conexiones, flujo de información... son conceptos que estamos acostumbrados a escuchar en ingeniería del transporte y seguro que a todos le recuerdan al esquema de una red neuronal. Esto está relacionado con el hecho de que todas las ramas de la ingeniería las podemos encontrar en nuestro interior. En las últimas décadas, se ha estado investigando mucho en el desarrollo de modelos matemáticos que intentan simular el comportamiento de nuestro cerebro y gracias a ello hoy en día podemos hablar de las Redes Neuronales Artificiales (en inglés, *Artificial Neural Network*) y de sus aplicaciones en diagnósticos médicos, predicción del tiempo, reconocimiento de imágenes, etc.

4.1 Neurobiología

Ante todo, vamos a comenzar con un breve repaso de la biología del sistema nervioso (Navarro, 2014). Este está formado por neuronas y vías nerviosas, que inervan el músculo del corazón, el músculo liso de todos los órganos y las glándulas exocrinas y endocrinas, con el objetivo de mantener al organismo en un estado de homeostasis y responder ante posibles cambios externos o internos. Se compone de tres partes: el sistema simpático, el parasimpático y el entérico. Este extraordinario sistema de cálculo está compuesto por billones de neuronas interconectadas y gracias a ello, podemos crear pensamientos, albergar recuerdos, sentir qué ocurre en nuestro interior y entorno, entre otras cosas.

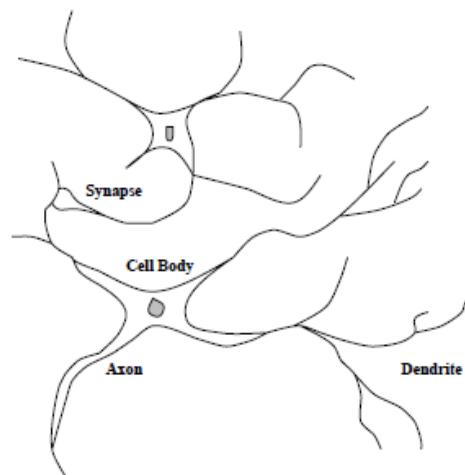


Figura 4.1.1. Azam (2000). *Biologically Inspired Modular Neural Networks*. Distintas partes de una neurona: soma, axón y dendritas. Sinapsis entre dos neuronas.

Una neurona se compone de un cuerpo celular o soma, el axón que representa una línea de transmisión y las dendritas que son ramificaciones que les permiten recibir información (ver Figura 4.1.1). La interconexión que se da entre dos neuronas se denomina sinapsis y en ella intervienen el botón sináptico de la neurona anterior y la neurona posterior (Basogain, 1998). Podemos diferenciar entre sinapsis axón-soma, axón-dendrita o axón-axón. Si hiciéramos un zoom, veríamos algo parecido a la siguiente Figura:

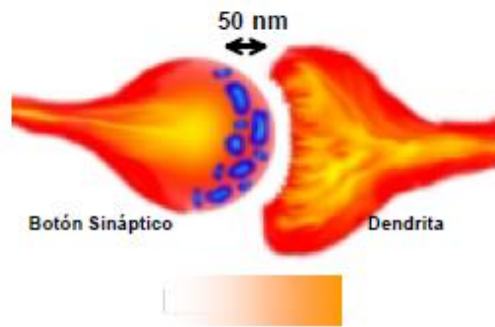


Figura 4.1.2. Izaurieta y Saavedra. (2000). *Redes neuronales artificiales. Botón sináptico y neurotransmisores en la sinapsis de dos neuronas.*

Antes de entrar en las redes neuronales artificiales, debemos entender cómo se produce la transmisión de información en las neuronas: al producirse la sinapsis, las dendritas de la neurona posterior reciben unas sustancias químicas llamadas neurotransmisores. Las diferentes fuentes de entrada se suman en el cuerpo neuronal y si todo el nivel energético supera un cierto umbral, se produce el flujo de información a través del axón (Izaurieta & Saavedra, 2000).

4.2 Red Neuronal Artificial

Una vez entendido el problema biológico, nos encontramos en disposición de abordar el modelado ingenieril. Una red neuronal artificial está formada por capas de entrada, capas ocultas y capas de salida. Sin embargo, para que resulte más fácil su entendimiento vamos a comenzar por un elemento procesador como se muestra en la Figura 4.2.1. Como comentábamos en el párrafo anterior, se compone de (Izaurieta & Saavedra, 2000 y Pal & Srimani, 1996):

- Señales de entrada (en la figura, “x”), donde cada una de ellas va a tener una importancia diferente y ello se cuantifica a través de los pesos sinápticos (en la figura, “w”). Todas confluyen en el cuerpo neuronal.
- Función de activación. Sobre el total de las entradas se aplica una función de activación, que puede ser un escalón o una sigmoidea, entre otras (en la figura, “ φ ”).
- Salidas. Si se supera el umbral de activación, tiene lugar la transmisión de las salidas (en la figura, v).

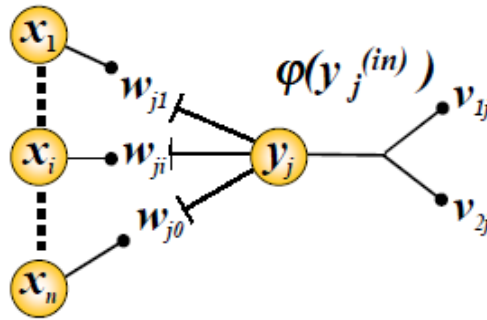


Figura 4.2.1. Izaurieta y Saavedra. (2000). Redes neuronales artificiales. Modelo de una red neuronal artificial: entradas, pesos, función de activación y salidas.

El aprendizaje de la red neuronal se denomina entrenamiento y puede ser supervisado o no supervisado (Salas, 2004). En el primero se entrena la red con datos de entrada y de salida para que esta sea capaz de ajustar los pesos, de manera que la salida que ofrezca sea lo más parecida a la salida objetivo, mientras que, en el segundo la red cuenta con un conjunto de datos y trata de encontrar alguna configuración en ellos.

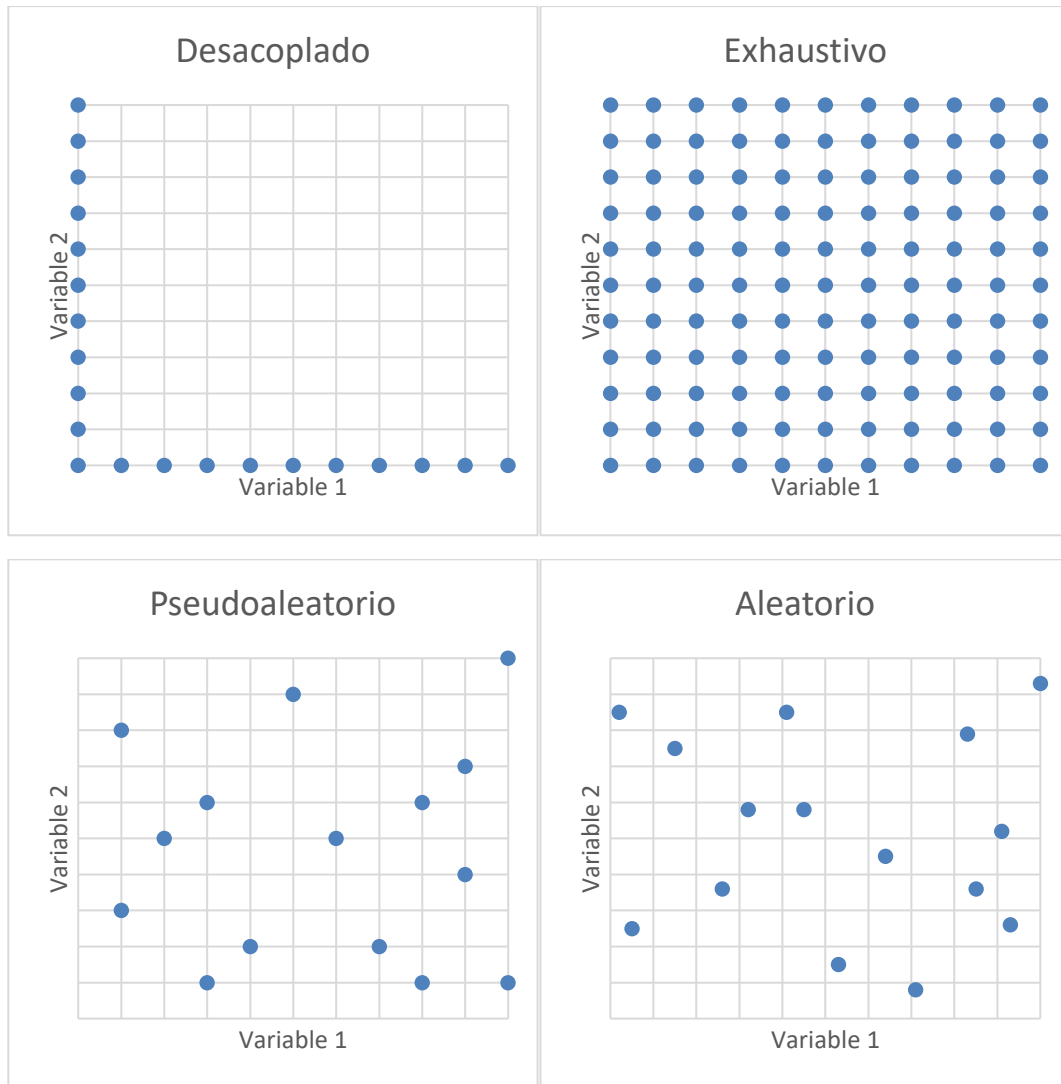
4.3 Tipos de entrenamiento

En este apartado, vamos a definir los distintos tipos de entrenamiento posibles para la red neuronal. Para que sea más fácil su entendimiento, vamos a suponer que tenemos dos variables con sus respectivos rangos:

$$Variable_1 = [var_{1_0} \quad var_{1_{fin}}]$$

$$Variable_2 = [var_{2_0} \quad var_{2_{fin}}]$$

El valor nominal de dichas variables es el valor medio del intervalo.



Figuras 4.3.1-4.3.4. Distintos tipos de entrenamiento para la red neuronal.

Por un lado, en el entrenamiento desacoplado, el origen del sistema de referencia es el valor nominal y las variables van tomando valores a lo largo de los respectivos ejes, tal y como se muestra en la Figura 4.3.1.

Por otro lado, en el entrenamiento exhaustivo, los casos forman una especie de cuadrícula, como se puede observar en la Figura 4.3.2. Si lo comparamos con el desacoplado, vemos que el número de casos aumenta considerablemente.

Si seleccionamos una muestra del total de casos posibles, es decir, del exhaustivo, tenemos un entrenamiento pseudoaleatorio, de tal forma que los puntos tomados coinciden con los de la cuadrícula, como observamos en la Figura 4.3.3.

Finalmente, en el entrenamiento aleatorio seleccionamos al azar un número determinado de puntos de entrenamiento. En este caso, los valores no tienen por qué coincidir con los de la cuadrícula. En el ejemplo, Figura 4.3.4, se han tomado 15 valores.

Posteriormente, tras obtener los resultados, analizaremos las ventajas y los inconvenientes de cada uno de ellos con el objetivo de obtener el mejor entrenamiento para la red neuronal.

4.4 Validación cruzada

La validación cruzada (del inglés, *K-Fold Cross Validation*) es una herramienta estadística mediante la cual se divide el conjunto de datos en K subconjuntos (en nuestro caso, 10) con el objetivo de establecer dos partes diferenciadas, una para entrenamiento, compuesta por 9 subconjuntos y otra para validación, formada por 1 (West, 2000 y Kohavi, 1995).

Como la finalidad de este trabajo es el diseño de redes neuronales, esta técnica resulta muy apropiada cuando lo que queremos conseguir es que los datos de entrenamiento estén lo más distribuidos posible (Fiszelew & García-Martínez, 2002). La idea, se resume en el siguiente esquema (Figura 4.4.1):



Figura 4.4.1. Validación cruzada. El total de datos se divide en diez, se deja un subconjunto para validación y los restantes para el aprendizaje de la red neuronal. Este proceso se repite diez veces y en cada iteración se tienen subconjuntos diferentes.

4.4.1 Datos necesarios

En este apartado vamos a definir los datos necesarios para que la red neuronal lleve a cabo la validación cruzada. Por un lado, necesitamos los datos de entrada a la red, los cuales coinciden con los datos de entrada al modelo y por otro lado, necesitamos los datos objetivo de la red que coinciden con la salida del modelo. Ambos conjuntos de datos se encuentran agrupados en matrices:

- Matriz de entrada a la red:
 - Filas: los parámetros del modelo.
 - Columnas: representan las distintas combinaciones numéricas de los parámetros.
- Matriz objetivo de la red:
 - Filas: los distintos instantes de tiempo donde se evalúa el BDG.
 - Columnas: evaluación en los distintos instantes de tiempo de las diferentes combinaciones de parámetros.

Para que la red pueda comenzar su proceso de aprendizaje es importante que el número de columnas de la matriz de entrada coincida con el número de columnas de la matriz objetivo, es decir, el número de casos de entrenamiento tiene que coincidir.

Llegados a este punto, vamos a diferenciar entre casos de entrenamiento no aleatorios y aleatorios. Para el primero de ellos, antes de comenzar con la validación cruzada, modificamos el orden de las columnas de las matrices de entrada y objetivo de la misma manera (ver Figura 4.4.2), para que la red aprenda con datos lo más distribuidos posibles. El hecho de que el desorden sea el mismo es importante para saber qué combinaciones de parámetros ofrecen un determinado BDG. Cuando tenemos datos aleatorios, esto no es necesario.

Matriz de entrada a la RNA					Matriz objetivo de la RNA				
C	C	C		C	C	C		C	
O	O	O		O	O	O		O	
L	L	L		L	L	L		L	
U	U	U		U	U	U		U	
M	M	M	.	M	M	M	.	M	
N	N	N	.	N	N	N	.	N	
A	A	A		A	A	A		A	
1	2	3		N	1	2	3	N	

Figura 4.4.2. En los datos de entrada y objetivo de la red neuronal, el orden de los casos de entrenamiento (columnas) tiene que coincidir.

4.4.2 Configuración de la red

A continuación, vamos a entrar más en detalle en la configuración de la red neuronal. De entre todas las funciones de entrenamiento posibles, nos hemos centrado en dos: Levenberg-Marquardt y Bayesian Regularization, ya que de acuerdo con algunos artículos científicos, son los que ofrecen menor error. Por un lado, se dice que el primero es capaz de converger más rápidamente, mientras que el segundo representa una mejor opción para problemas que presentan relaciones más complejas (Kayri, 2016).

En cuanto al número de capas ocultas hemos hecho pruebas con una, dos y tres, con 10 y 20 neuronas en cada una de ellas. Un ejemplo de cómo quedaría la estructura se presenta en la siguiente Figura:

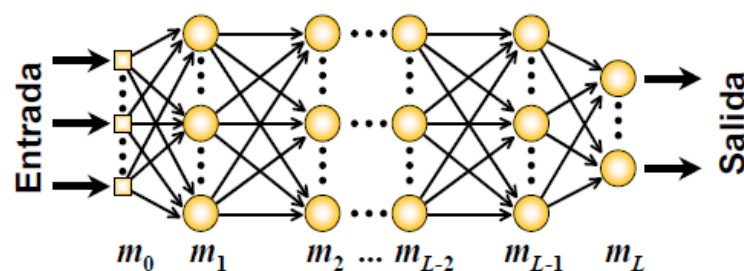


Figura 4.4.3. Izaurieta y Saavedra (2000). Redes neuronales artificiales. Esquema de una red neuronal artificial con una capa de entrada, una de salida y varias capas intermedias.

Finalmente, un detalle importante para tener en cuenta en la programación ha sido el reparto de datos para *Training*, *Validation* and *Testing*, puesto que queremos que la red nos entrene con el 100% del subconjunto que le damos para entrenamiento, ya que por defecto, la red en Matlab lo hace con el 70% de los datos.

4.4.3 Entrenamiento y test

En cada una de las 10 iteraciones, la red previamente configurada aprende con los subconjuntos de entrenamiento correspondientes, es decir, se necesita por una parte los subconjuntos de entrenamiento correspondientes a los datos de entrada (ver Figuras 4.4.4-4.4.5, subconjuntos azules) y por otra parte, los subconjuntos de entrenamiento correspondientes a los datos objetivo de la red (ver Figuras 4.4.4-4.4.5, subconjuntos azules).

Datos de entrada a la red neuronal									
T	Tr	Tr	Tr	Tr	Tr	Tr	Tr	Tr	Tr

Figura 4.4.4. Datos de entrada. De los diez subconjuntos, nueve son para entrenamiento y uno para validación.

Datos objetivo de la red neuronal									
T	Tr	Tr	Tr	Tr	Tr	Tr	Tr	Tr	Tr

Figura 4.4.5. Datos objetivo. La distribución de los subconjuntos en los datos de entrada y objetivo de la red coinciden.

Tras el entreno, es el turno de probar la red. Para ello, a la red se le da como entrada el subconjunto de prueba correspondiente a los datos de entrada (ver Figuras 4.4.4-4.4.5, subconjunto rojo) y se evalúa. Tras esto, se comparan a través del coeficiente de variación, el subconjunto de prueba correspondiente a los datos objetivo (ver Figuras 4.4.4-4.4.5, subconjunto rojo) y la salida ofrecida por la red mediante la siguiente fórmula:

$$\left[\sum_{i=1}^{N_1} \sum_{j=1}^{N_2} \left\| \frac{Target(i,j) - Salida(i,j)}{Target(i,j)} \right\|^2 * \frac{1}{N_1 N_2} \right]^{1/2} * 100$$

Donde:

$N_1 \equiv$ número de instantes en que se evalúa el BDG.

$N_2 \equiv$ número de casos de entrenamiento.

Todo este proceso se realiza 10 veces (ver Figura 4.4.1), con el objetivo de obtener la red neuronal que ofrezca menor coeficiente de variación (Krogh & Vedelsby, 1995).

5. RESULTADOS

“Trabajar es vivir y a mí me encanta vivir”.

-Charles Chaplin-

Del análisis de sensibilidad previamente realizado, hemos seleccionado determinadas variables que tienen mucha influencia y cuyo estudio resulta muy interesante desde el punto de vista del modelo. Vamos a comenzar realizando todo el proceso con dos variables (Kappa, “ κ ” y Zeta, “ ζ ”), para seleccionar el tipo de entrenamiento que ofrezca menor coeficiente de variación. Una vez que lo tengamos, probaremos con tres (Kappa, Zeta y P_{RANKL}^{PMO}), luego con cuatro (Kappa, Zeta, P_{RANKL}^{PMO} y N_{OCp}^{RANK}) y finalmente con cinco variables (Kappa, Zeta, P_{RANKL}^{PMO} , N_{OCp}^{RANK} y β_{RANKL}), puesto que nuestro objetivo es encontrar una relación entre el número de casos de entrenamiento y el número de variables. Los rangos utilizados se muestran a continuación:

$$Kappa = [0,001; 0,007]$$

$$Zeta = [0,001; 0,03] * \frac{1}{3} \quad (pM \ RANKL)^{-1}$$

$$P_{RANKL}^{PMO} = [1000; 6000] \quad pM$$

$$N_{OCp}^{RANK} = \left[10^{-4} * \frac{1}{16}; 10^{-4} * 16 \right] \quad pM \ RANK * (pM \ cell)^{-1}$$

$$\beta_{RANKL} = \left[1,6842 * 10^2 * \frac{1}{4}; 1,6842 * 10^2 * 4 \right] \quad pM \ RANKL * (pM \ cell)^{-1}$$

5.1 Procedimiento de obtención de los resultados

Cada uno de los diferentes tipos de entrenamientos estudiados (desacoplado, exhaustivo, pseudoaleatorio y aleatorio), van a formar un conjunto de datos de entrada para el modelo. Gracias al proceso de automatización de las simulaciones, se obtienen los datos de salida, que como ya sabemos, coinciden con los datos objetivo de la red neuronal.

Con cada conjunto de datos entrada-salida se crea una red neuronal correspondiente, a la que se le realiza la validación cruzada. Posteriormente, seleccionamos 10 puntos de manera que estén distribuidos por el dominio considerado y que vamos a utilizar para validar las distintas redes.

A lo largo de este estudio, se han realizado tanto análisis gráficos cualitativos, para visualizar cómo la red es capaz de aprender como cuantitativos, que nos ofrecen un valor numérico del error producido en dicho aprendizaje.

5.2 Dos variables

5.2.1 Entrenamiento desacoplado

La distribución de los casos de entrenamiento se muestra en la Figura 5.2.1. En total, se han tomado 42 puntos que son el resultado de 13 valores de Kappa más 30 valores de Zeta menos 1, para evitar que el punto correspondiente al valor nominal de las variables estuviera duplicado. En la figura, además, se representan los puntos de validación tomados y así comprobamos que los de entrenamiento no coinciden con los de validación.

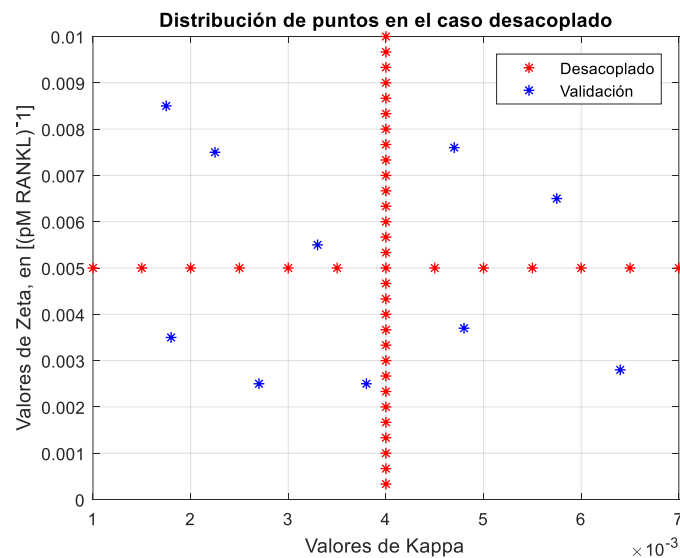


Figura 5.2.1. Distribución de los puntos de entrenamiento en todo el rango de Kappa-Zeta, caso desacoplado, dos variables.

En las Figuras 5.2.2-5.2.4, observamos a qué distancia se encuentra la salida que ofrece la red neuronal de la del modelo. Menor distancia equivale a mejor aproximación, mientras que mayor distancia se traduce en peor aproximación. Con este tipo de entrenamiento, en los puntos de validación más cercanos a los ejes, la red ofrece un mejor resultado.

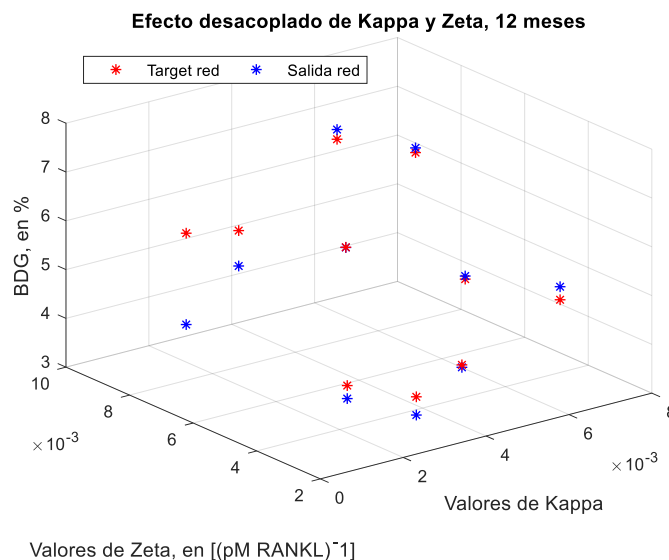


Figura 5.2.2. Caso desacoplado, comparación entre la salida del modelo y la de la red.

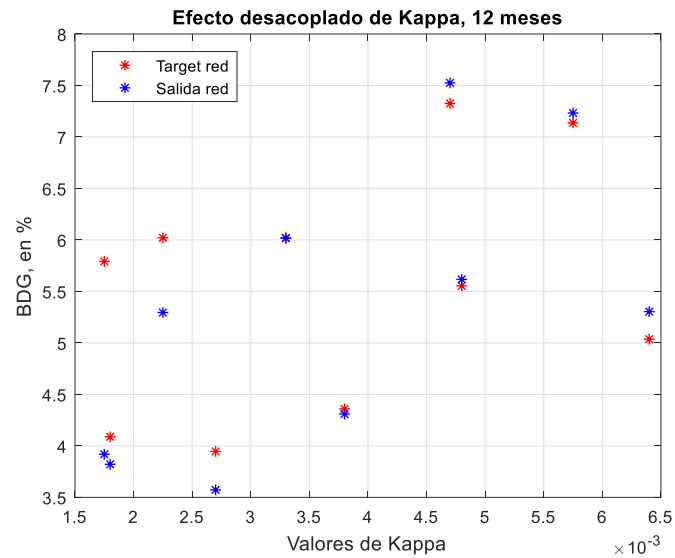


Figura 5.2.3. Efecto de Kappa sobre el aprendizaje de la red neuronal, caso desacoplado, dos variables.

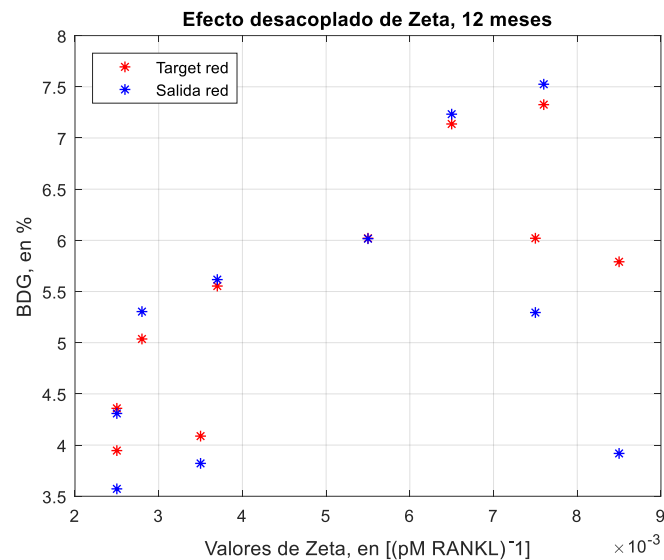


Figura 5.2.4. Efecto de Zeta sobre el aprendizaje de la red neuronal, caso desacoplado, dos variables.

5.2.2 Entrenamiento exhaustivo

En este tipo de entrenamiento, el número de puntos aumenta considerablemente. Ahora tenemos 13 valores de Kappa por 30 valores de Zeta casos de entrenamiento, cuya distribución se muestra en la Figura 5.2.5, donde también se representan los puntos de validación.

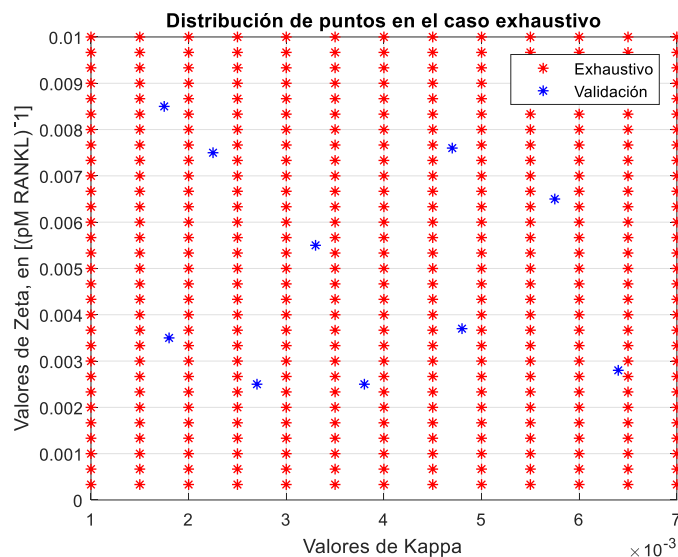


Figura 5.2.5. Distribución de los puntos de entrenamiento en todo el rango de Kappa-Zeta, caso exhaustivo, dos variables.

Por otro lado, en las Figuras 5.2.6-5.2.8 comprobamos que la red aproxima muy bien los datos de salida del modelo, puesto que los puntos rojos y azules prácticamente coinciden. Como en este apartado estamos entrenando la red con un mayor número de casos, la probabilidad de que los puntos de validación estén cercanos a los de entrenamiento es mayor que en el caso desacoplado.

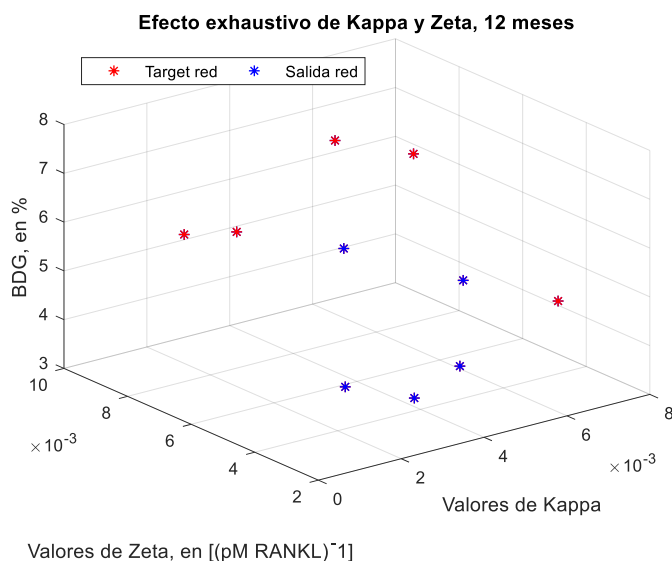


Figura 5.2.6. Caso exhaustivo, comparación entre la salida del modelo y la de la red.

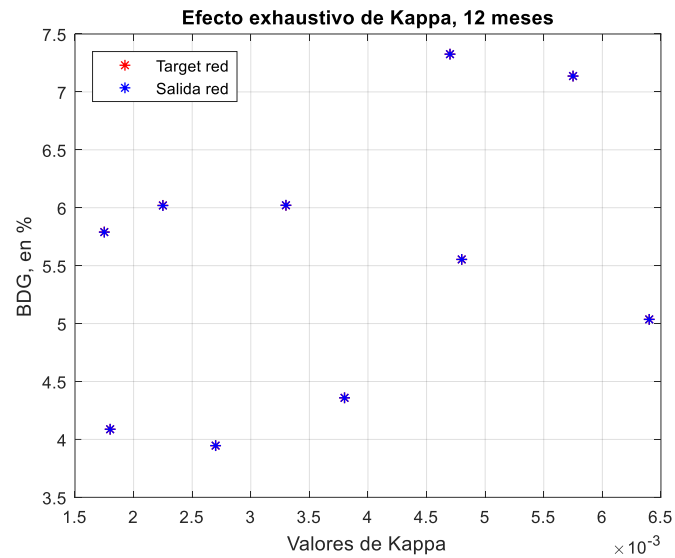


Figura 5.2.7. Efecto de Kappa sobre el aprendizaje de la red neuronal, caso exhaustivo, dos variables.

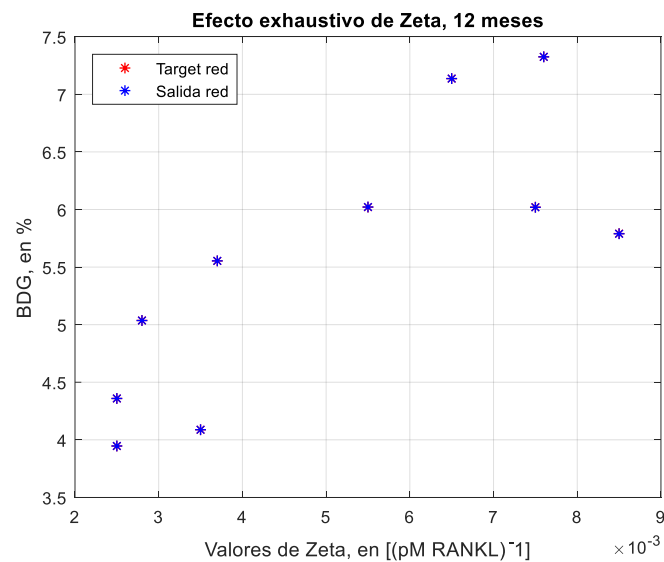


Figura 5.2.8. Efecto de Zeta sobre el aprendizaje de la red neuronal, caso exhaustivo, dos variables.

5.2.3 Entrenamiento pseudoaleatorio

Este tipo de entrenamiento es un paso intermedio entre el exhaustivo y el aleatorio. En este apartado, decidimos tomar 160 puntos elegidos al azar de la cuadrícula de la Figura 5.2.5 (exhaustivo), de manera que ninguno coincidiera con los puntos de validación. La distribución se presenta en la siguiente Figura:

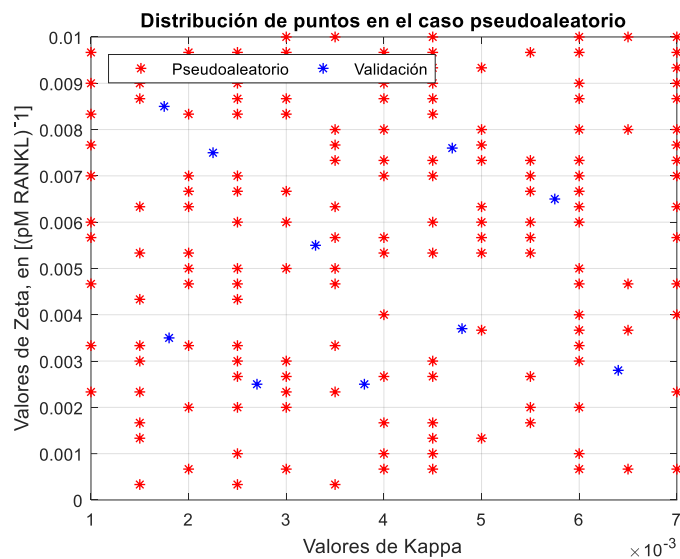


Figura 5.2.9. Distribución de los puntos de entrenamiento en todo el rango de Kappa-Zeta, caso pseudoaleatorio, dos variables.

Dado que los casos de entrenamiento están distribuidos por todo el dominio, los puntos de validación se encuentran relativamente cercanos a ellos y tal y como pasaba en el caso anterior, los puntos de salida del modelo y los de salida de la red se aproximan bastante bien, Figuras 5.2.10-5.2.12:

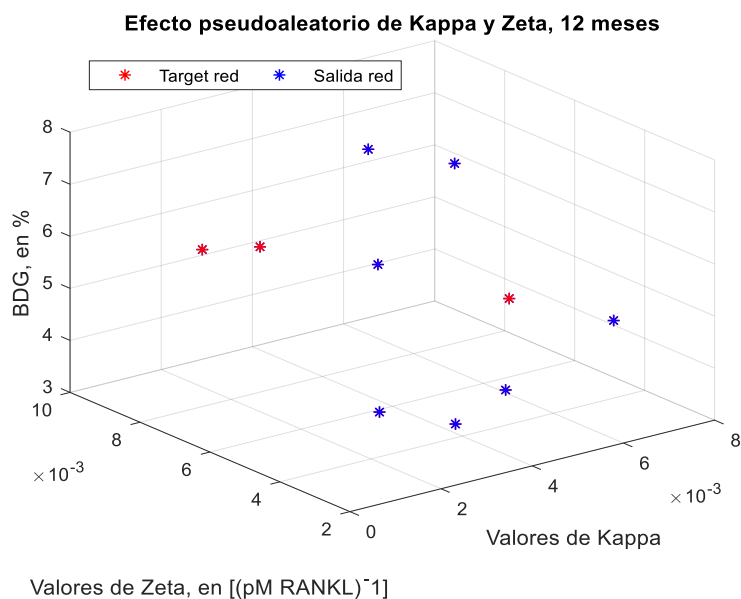


Figura 5.2.10. Caso pseudoaleatorio, comparación entre la salida del modelo y la de la red.

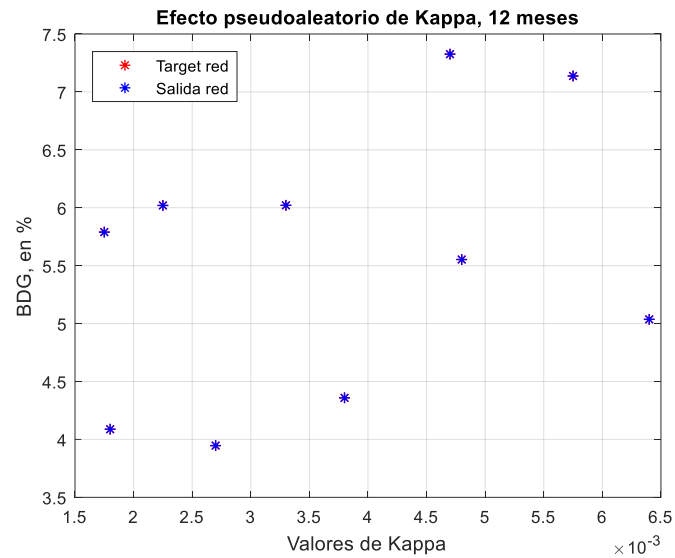


Figura 5.2.11. Efecto de Kappa sobre el aprendizaje de la red neuronal, caso pseudoaleatorio, dos variables.

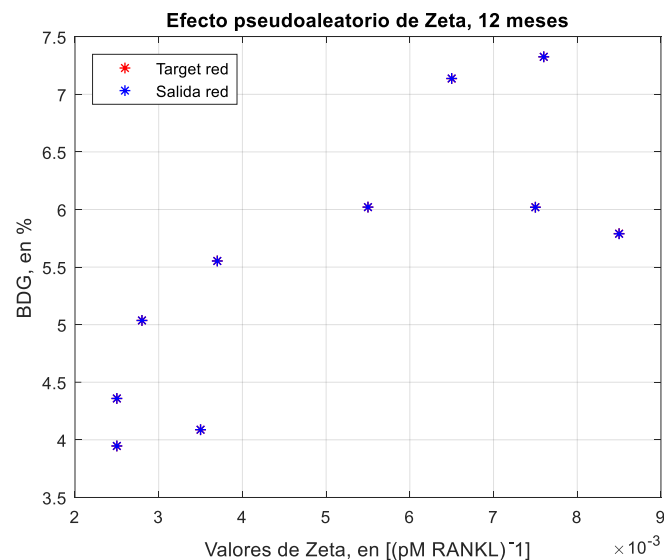


Figura 5.2.12. Efecto de Zeta sobre el aprendizaje de la red neuronal, caso pseudoaleatorio, dos variables.

5.2.4 Entrenamiento aleatorio

En esta sección, hemos seleccionado al azar 160 puntos pertenecientes al rango de las variables. Si comparamos la Figura 5.2.5 con la Figura 5.2.13, comprobamos que los puntos ya no están ordenados como en una cuadrícula. Además, ninguno coincide con los de validación.

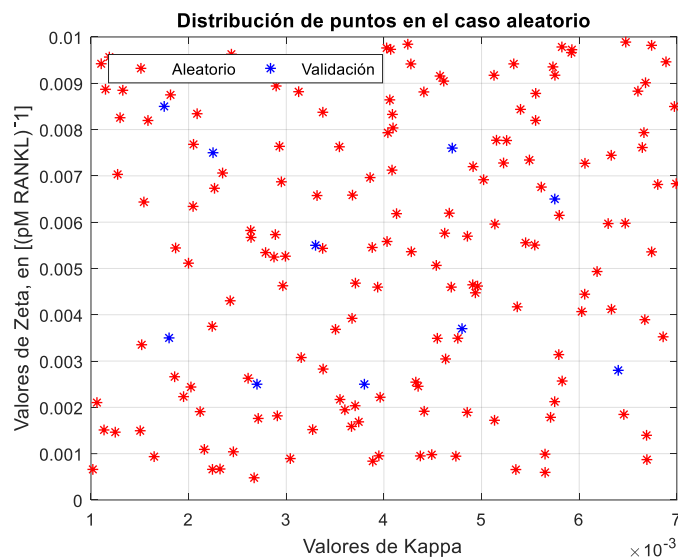


Figura 5.2.13. Distribución de los puntos de entrenamiento en todo el rango de Kappa-Zeta, caso aleatorio, dos variables.

Con las Figuras 5.2.14-5.2.16 que se muestran a continuación, comprobamos que con el entrenamiento aleatorio la red neuronal vuelve a aproximar bastante bien la salida del modelo. Llegados a este punto, hemos analizado cómo la red es capaz de aprender para los distintos tipos de entrenamiento y con dos variables. En lo que sigue realizaremos un análisis del coeficiente de variación para justificar cuantitativamente el tipo de entrenamiento que vamos a seleccionar.

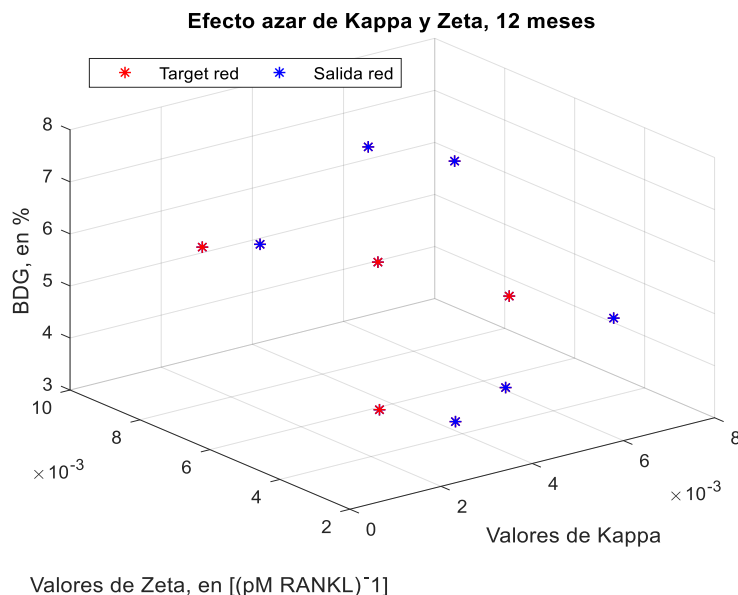


Figura 5.2.14. Caso aleatorio, comparación entre la salida del modelo y la de la red.

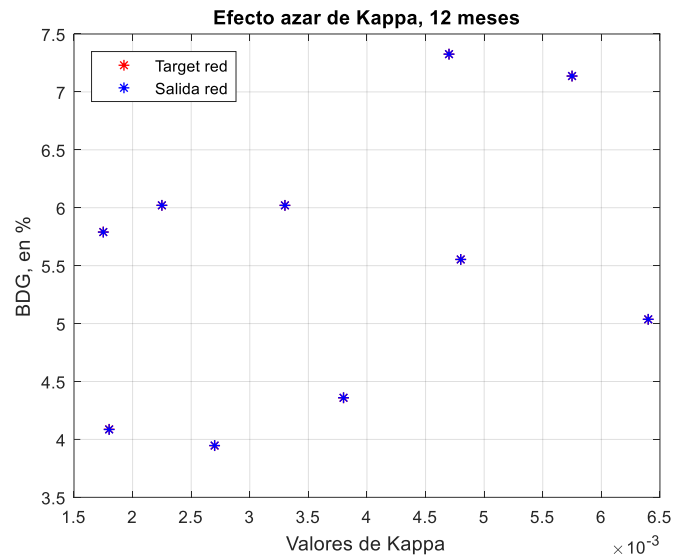


Figura 5.2.15. Efecto de Kappa sobre el aprendizaje de la red neuronal, caso aleatorio, dos variables.

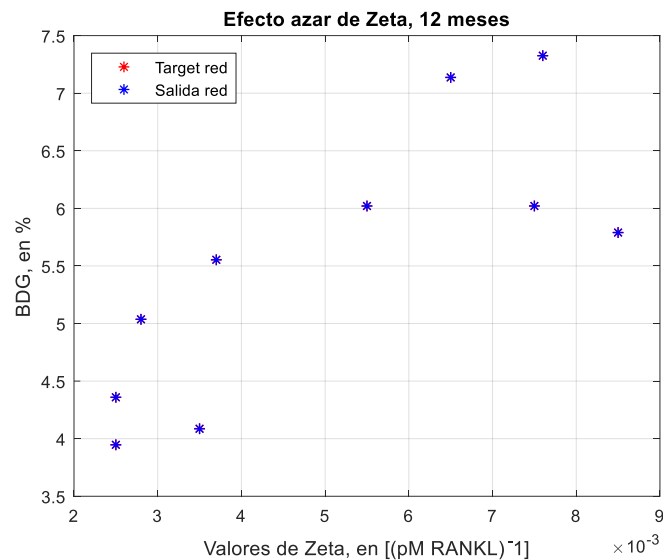


Figura 5.2.16. Efecto de Zeta sobre el aprendizaje de la red neuronal, caso aleatorio, dos variables.

5.2.5 Elección del tipo de entrenamiento

Nos encontramos en un escenario en el que estamos analizando cómo la red neuronal es capaz de aprender el efecto que dos variables tienen sobre la salida del modelo con diferentes tipos de entrenamiento. Tras el análisis cualitativo previo, toca cuantificar dicho aprendizaje a través del coeficiente de variación.

En la Figura 5.2.17, se representa dicho coeficiente frente al número de puntos de entrenamiento para el caso desacoplado (42 puntos), aleatorio (160 puntos), pseudoaleatorio (160 puntos) y exhaustivo (390 puntos). Se observa claramente una gran diferencia entre el desacoplado y los demás, por lo que descartamos este tipo de entrenamiento.

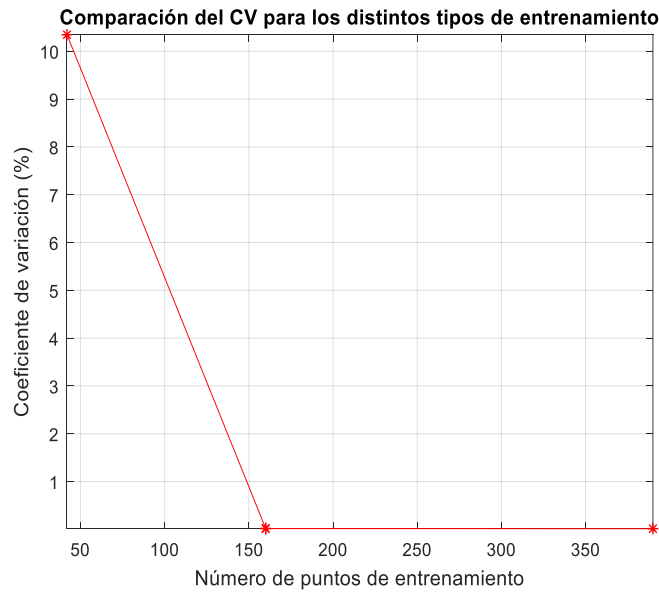


Figura 5.2.17. Coeficiente de variación para los entrenamientos desacoplado (42 puntos), aleatorio (160 puntos), pseudoaleatorio (160 puntos) y exhaustivo (390 puntos).

Si ampliamos la gráfica superior para analizar más en detalle los tipos de entrenamiento aleatorio, pseudoaleatorio y exhaustivo, comprobamos en la Figura 5.2.18 que prácticamente no hay diferencia en el coeficiente de variación, 0,019, 0,0179 y 0,0151 respectivamente, mientras que sí la hay en el número de puntos de entrenamiento, 160, 160 y 390 respectivamente.

Además, tiene sentido los distintos valores obtenidos del coeficiente de variación, que el del aleatorio sea mayor que el del pseudoaleatorio y que del exhaustivo, puesto que si recordamos, los puntos de entrenamiento en estos dos últimos estaban ordenados en una especie de cuadrícula y debido a ese orden la red aprende mejor.

Para ahorrar coste computacional a la hora de realizar las simulaciones en el modelo y para evitar el efecto del ordenamiento de los puntos de entrenamiento en el aprendizaje de la red, vamos a seleccionar el entrenamiento aleatorio. En lo que sigue, aumentaremos el número de variables y realizaremos los análisis cualitativos y cuantitativos correspondientes.

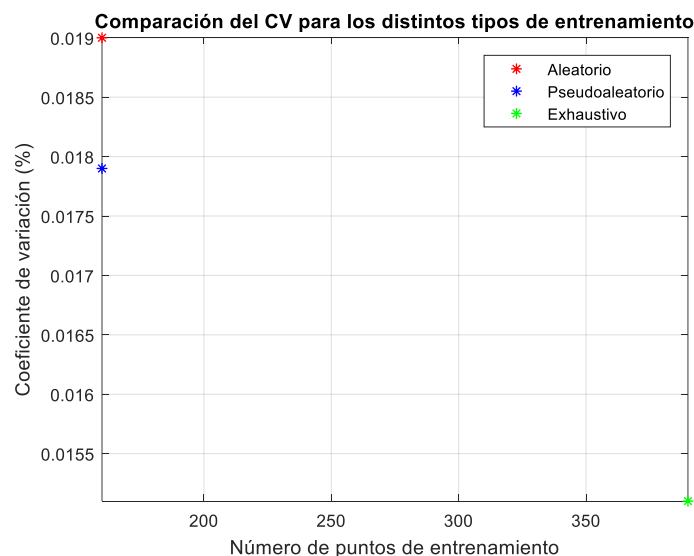


Figura 5.2.18. Coeficiente de variación para los entrenamientos aleatorio, pseudoaleatorio y exhaustivo.

5.3 Tres variables

En este apartado, analizamos el aprendizaje de la red neuronal al añadir una tercera variable, el P_{RANKL}^{PMO} . Ahora, los puntos se organizan en un cubo como en el de la Figura 5.3.1 y para que resulte más sencillo ver cómo se distribuyen los puntos, se muestran las diferentes vistas en las Figuras 5.3.2-5.3.4:

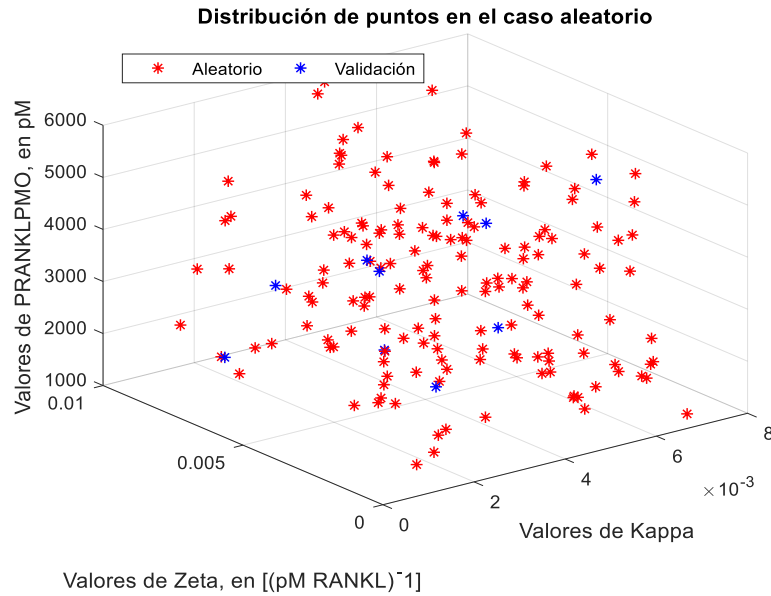


Figura 5.3.1. Caso aleatorio, tres variables.

En las siguientes figuras tenemos las distintas vistas del cubo, X-Y, X-Z, Y-Z. De nuevo, los puntos de validación se han elegido para que estén lo más distribuidos posible por todo el dominio y hemos vuelto a seleccionar los 160 puntos de manera aleatoria, comprobando computacionalmente que ninguno coincida.

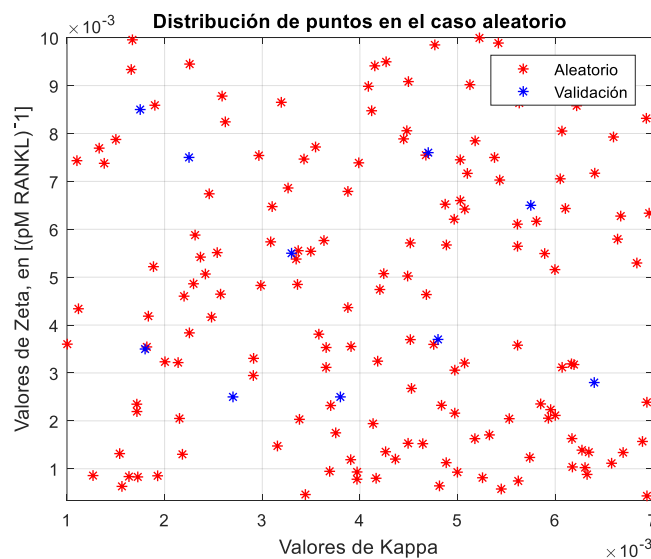


Figura 5.3.2. Distribución de los puntos de entrenamiento en todo el rango de Kappa-Zeta, tres variables.

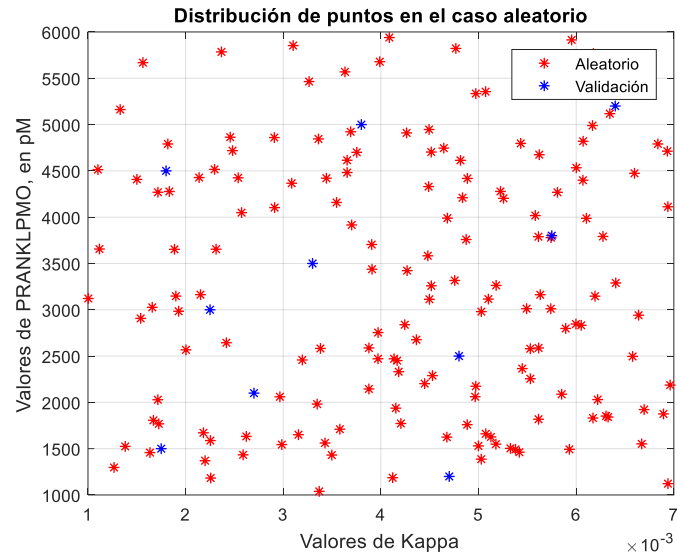


Figura 5.3.3. Distribución de los puntos de entrenamiento en todo el rango de variables $Kappa$ - P_{RANKL}^{PMO} , tres variables.

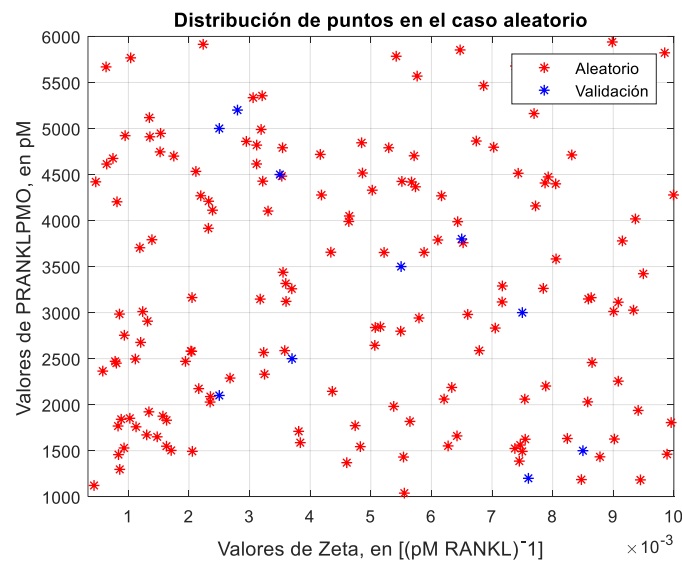


Figura 5.3.4. Distribución de los puntos de entrenamiento en todo el rango de variables $Zeta$ - P_{RANKL}^{PMO} , tres variables.

Como queremos visualizar el efecto de tres variables de entrada al modelo/red sobre el BDG que ofrece el modelo/red es necesario ver el efecto de cada variable por separado, como se muestra en las Figuras 5.3.5-5.3.7. Se aprecia que con este tipo de entrenamiento y este número de variables la red aprende bastante bien.

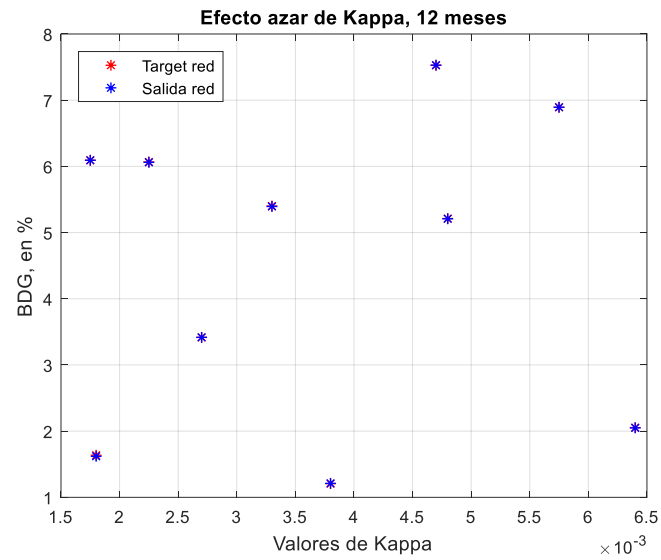


Figura 5.3.5. Efecto de Kappa sobre el aprendizaje de la red neuronal, tres variables.

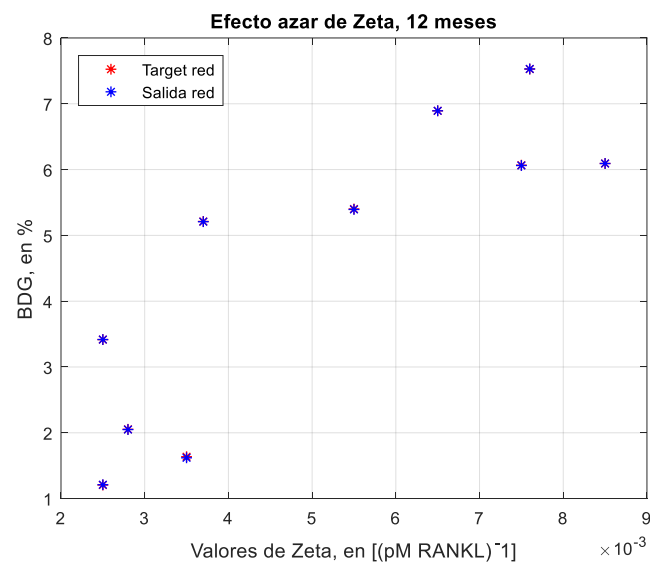


Figura 5.3.6. Efecto de Zeta sobre el aprendizaje de la red neuronal, tres variables.

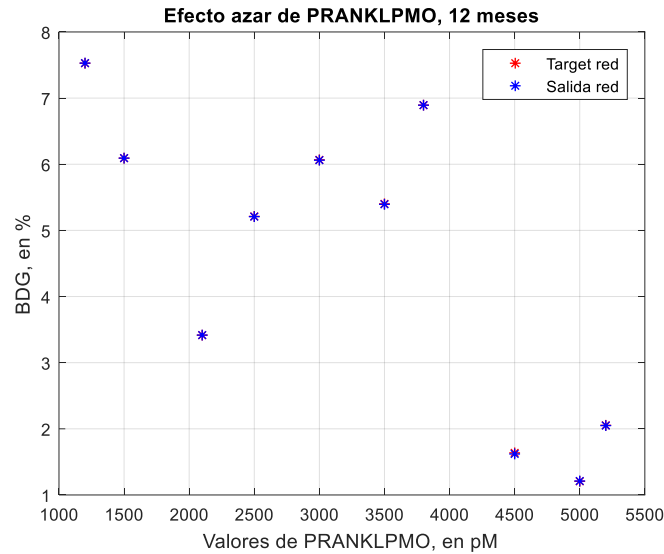


Figura 5.3.7. Efecto de P_{RANKL}^{PMO} sobre el aprendizaje de la red neuronal, tres variables.

5.4 Cuatro variables

Tras la adición del P_{RANKL}^{PMO} , ahora añadimos una nueva variable, el N_{OCp}^{RANK} . Con esta nueva incorporación, tenemos cuatro dimensiones, por lo que parece razonable que aumentemos el número de puntos de entrenamiento. Añadiremos 140 más para tener un total de 300 simulaciones. Las vistas se representan en las Figuras 5.4.1-5.4.6, donde observamos los 300 casos de entrenamiento junto con los puntos de validación.

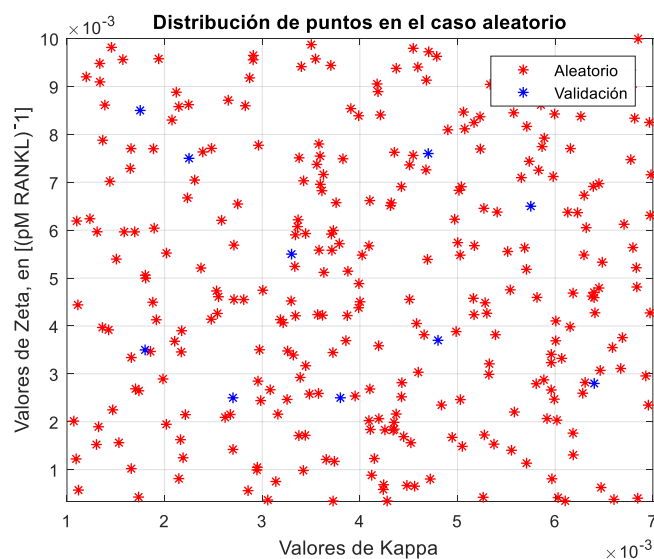


Figura 5.4.1. Distribución de los puntos de entrenamiento en todo el rango de Kappa-Zeta, cuatro variables.

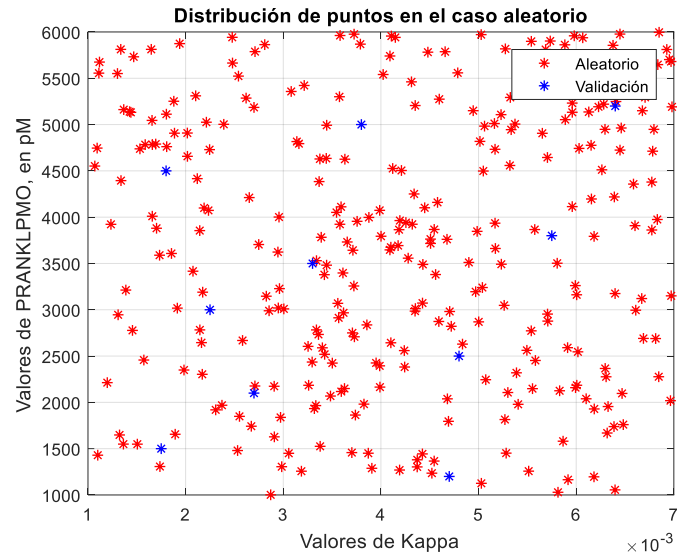


Figura 5.4.2. Distribución de los puntos de entrenamiento en todo el rango de variables $Kappa-P_{RANKL}^{PMO}$, cuatro variables.

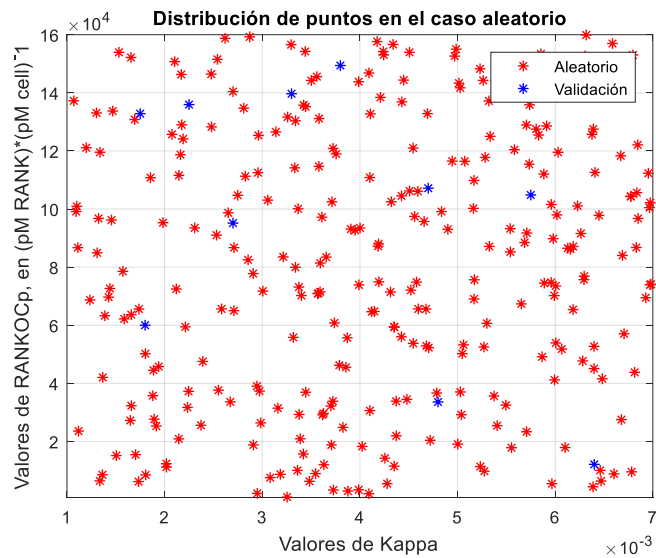


Figura 5.4.3. Distribución de los puntos de entrenamiento en todo el rango de variables $Kappa-N_{OCp}^{RANK}$, cuatro variables.

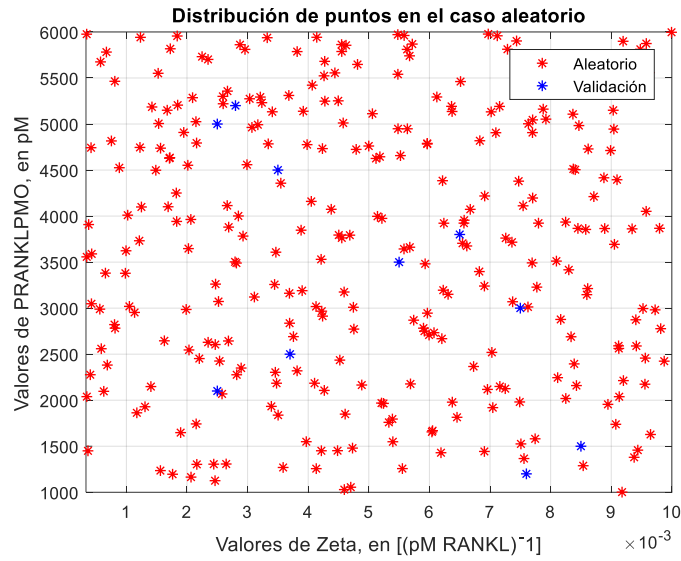


Figura 5.4.4. Distribución de los puntos de entrenamiento en todo el rango de variables Zeta- P_{RANKL}^{PMO} , cuatro variables.

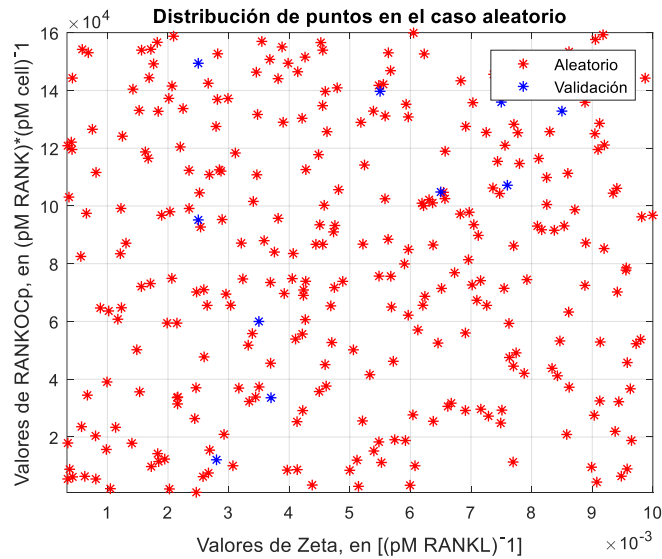


Figura 5.4.5. Distribución de los puntos de entrenamiento en todo el rango de variables Zeta- N_{OCp}^{RANK} , cuatro variables.

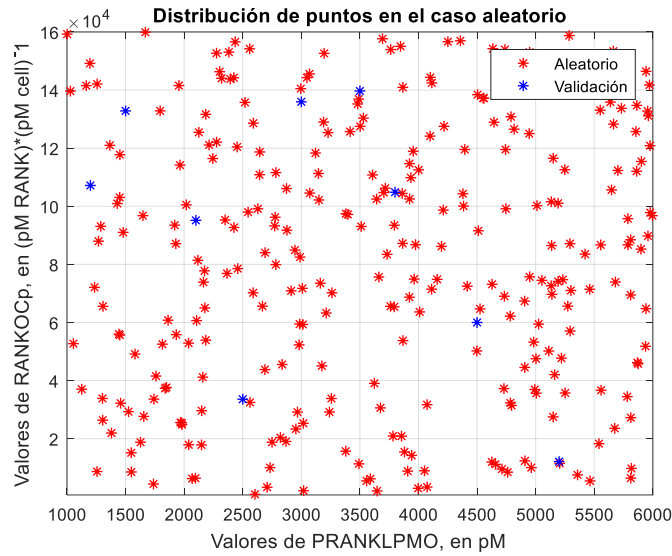


Figura 5.4.6. Distribución de los puntos de entrenamiento en todo el rango de variables P_{RANKL}^{PMO} - N_{OCp}^{RANK} , cuatro variables.

Tal y como ocurría con tres variables, con cuatro también se hace necesario separar el efecto de cada una de ellas sobre el BDG. Podemos observar en las Figuras 5.4.7-5.4.10, que en determinados casos la red aprende bastante bien pero para otros sí hay diferencia entre la salida del modelo y la de la red. En apartados posteriores cuantificaremos esa diferencia con el coeficiente de variación.

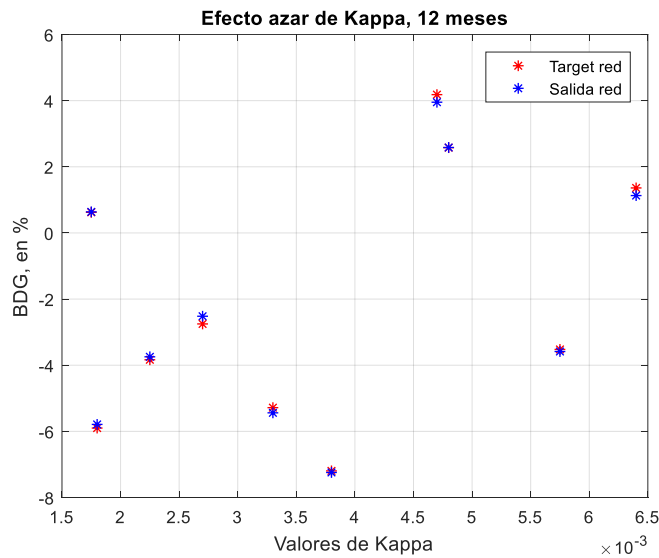


Figura 5.4.7. Efecto de Kappa sobre el aprendizaje de la red neuronal, cuatro variables.

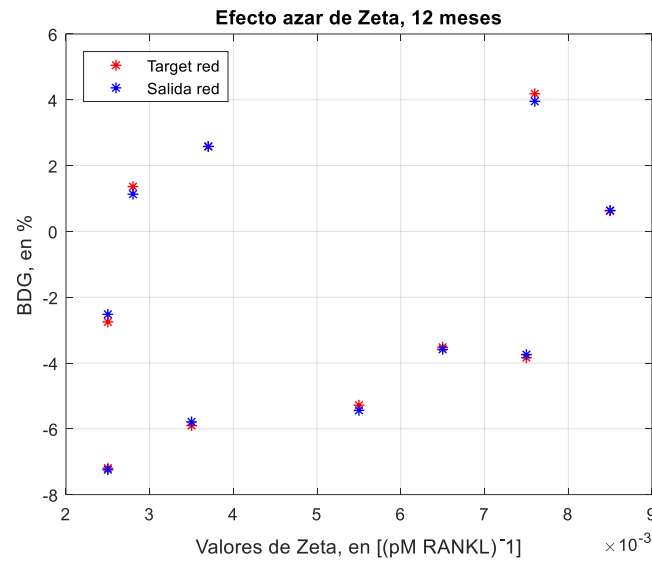


Figura 5.4.8. Efecto de Zeta sobre el aprendizaje de la red neuronal, cuatro variables.

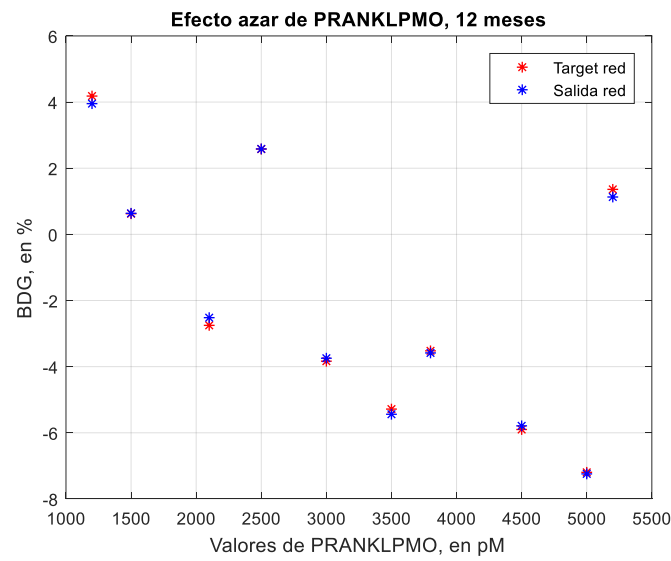


Figura 5.4.9. Efecto de P_{RANKL}^{PMO} sobre el aprendizaje de la red neuronal, cuatro variables.

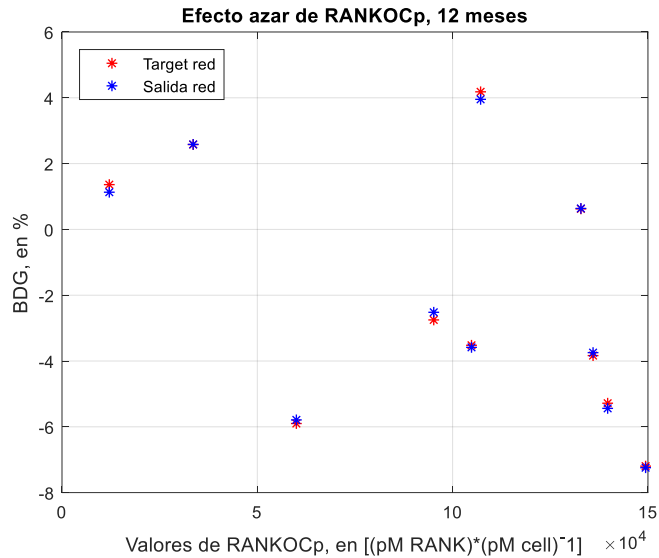


Figura 5.4.10. Efecto de N_{OCp}^{RANK} sobre el aprendizaje de la red neuronal, cuatro variables.

5.5 Cinco variables

Finalmente, tras la adición del N_{OCp}^{RANK} , añadimos la quinta variable, el β_{RANKL} . Al igual que en el apartado anterior, volvemos a tener 300 puntos de entrenamiento y en las Figuras 5.5.1-5.5.10 se recogen las vistas correspondientes a las cinco dimensiones con los puntos de validación:

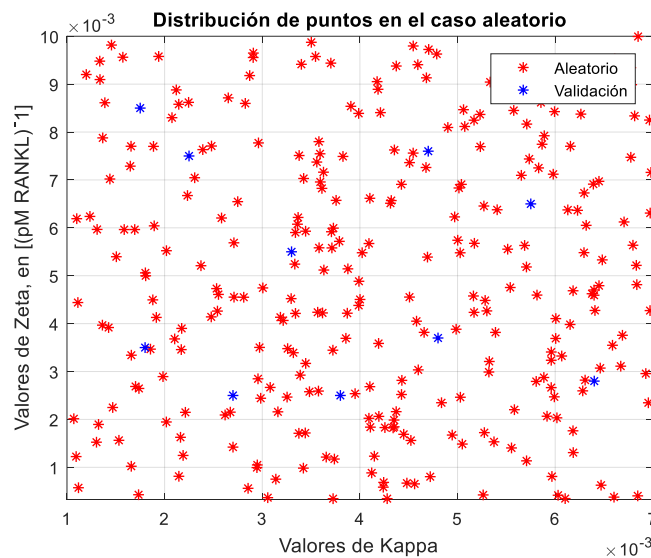


Figura 5.5.1. Distribución de los puntos de entrenamiento en todo el rango de Kappa-Zeta, cinco variables.

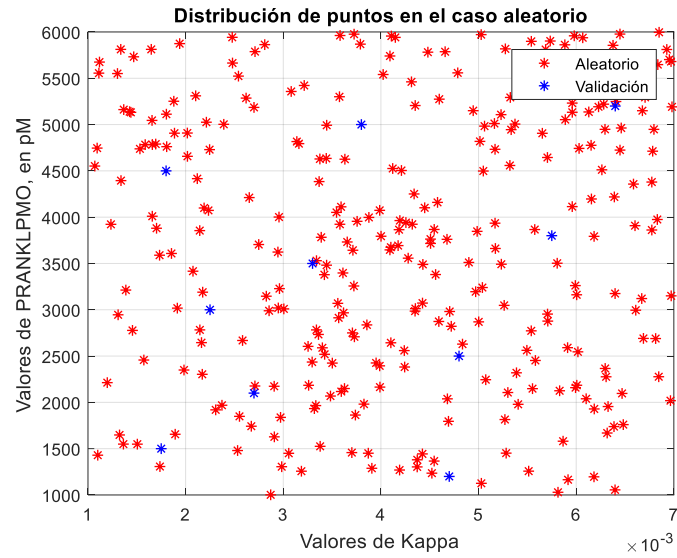


Figura 5.5.2. Distribución de los puntos de entrenamiento en todo el rango de variables $Kappa-PPMO_{RANKL}$, cinco variables.

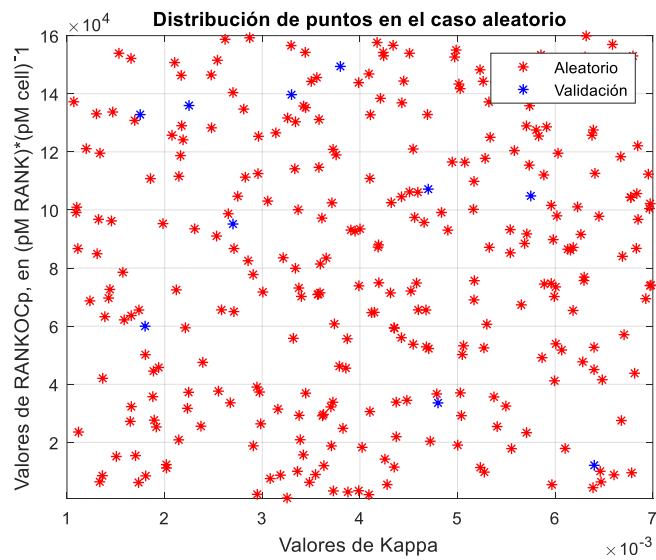


Figura 5.5.3. Distribución de los puntos de entrenamiento en todo el rango de variables $Kappa-N_{OC_p}^{RANK}$, cinco variables.

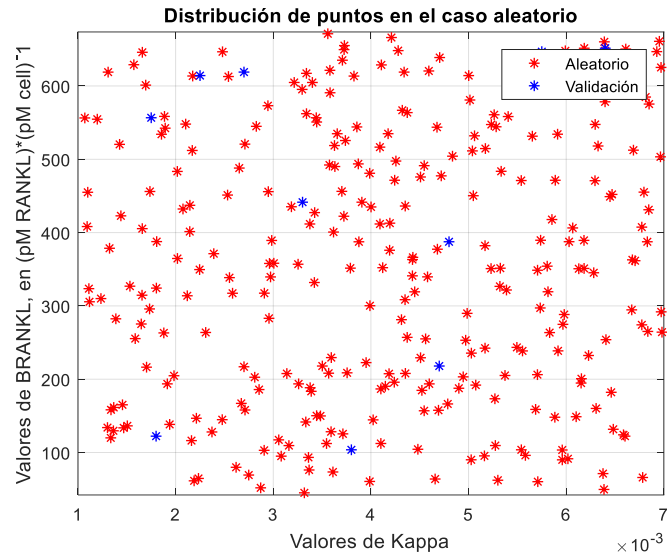


Figura 5.5.4. Distribución de los puntos de entrenamiento en todo el rango de variables $Kappa-\beta_{RANKL}$, cinco variables.

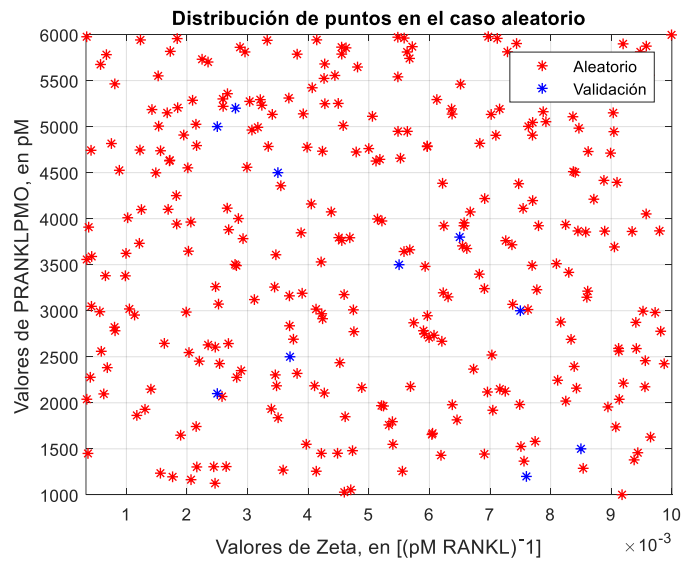


Figura 5.5.5. Distribución de los puntos de entrenamiento en todo el rango de variables $Zeta-P_{RANKL}^{PMO}$, cinco variables.

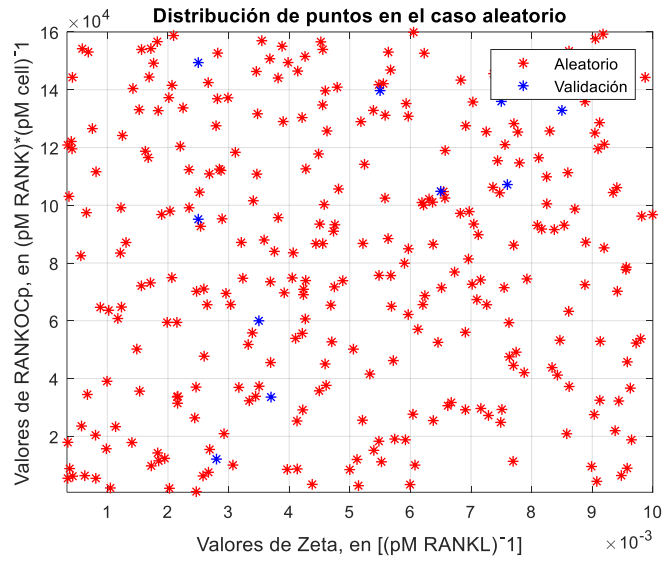


Figura 5.5.6. Distribución de los puntos de entrenamiento en todo el rango de variables Zeta - N_{OCp}^{RANK} , cinco variables.

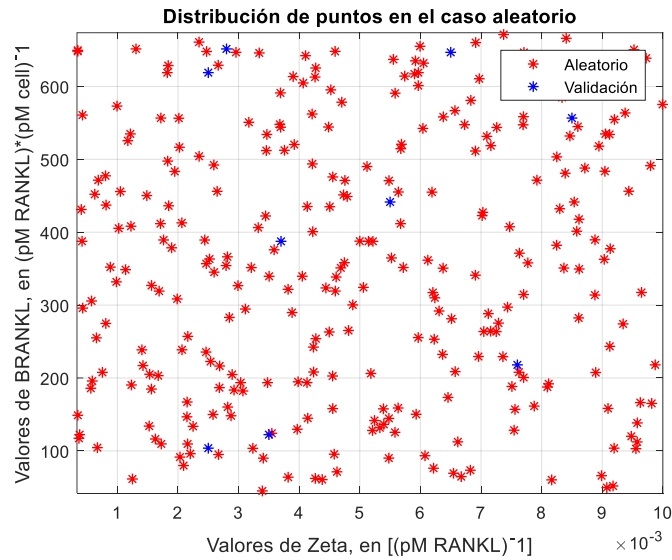


Figura 5.5.7. Distribución de los puntos de entrenamiento en todo el rango de variables Zeta - β_{RANKL} , cinco variables.

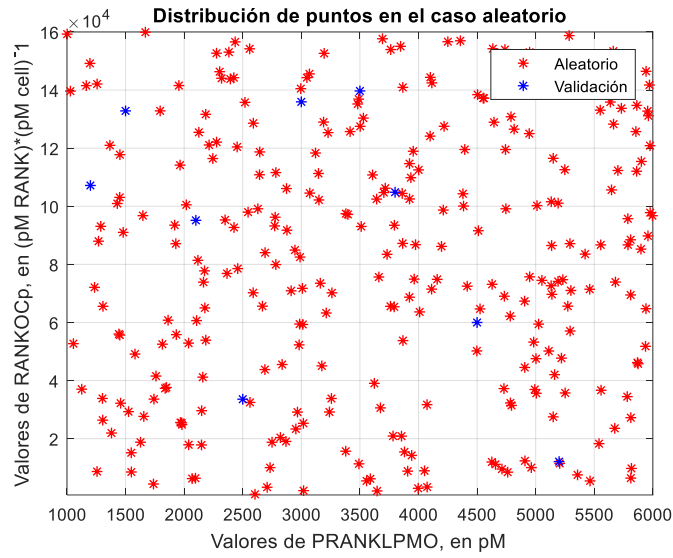


Figura 5.5.8. Distribución de los puntos de entrenamiento en todo el rango de variables $P_{RANKL}^{PMO} - N_{OC_p}^{RANK}$, cinco variables.

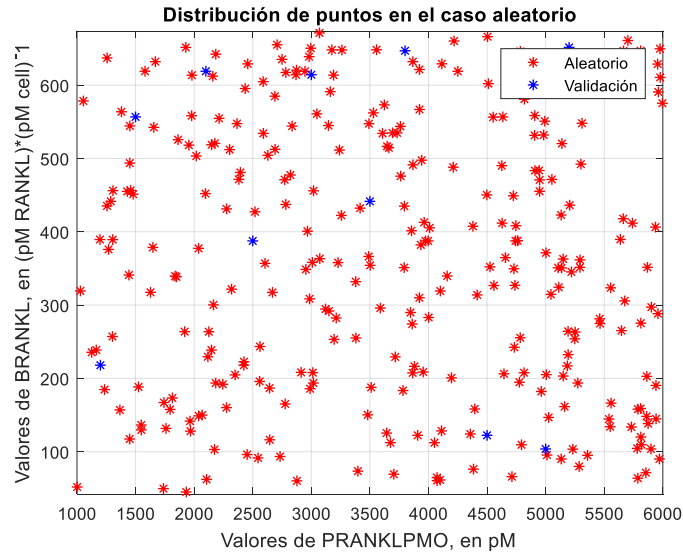


Figura 5.5.9. Distribución de los puntos de entrenamiento en todo el rango de variables $P_{RANKL}^{PMO} - \beta_{RANKL}$, cinco variables.

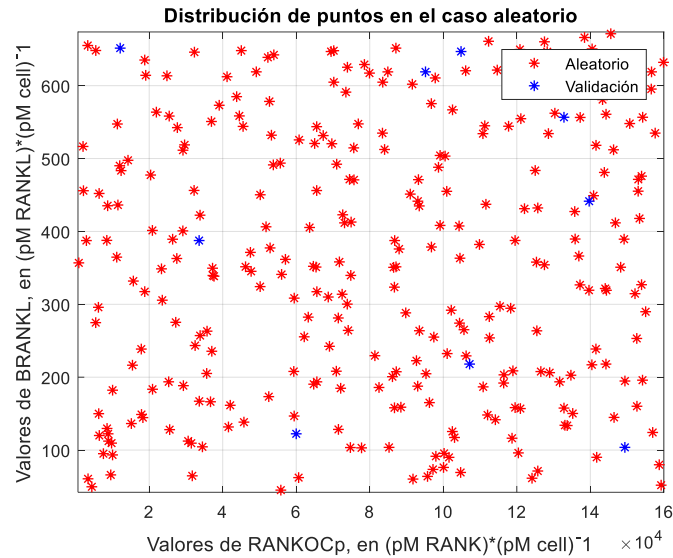


Figura 5.5.10. Distribución de los puntos de entrenamiento en todo el rango de variables $N_{OCp}^{RANK} - \beta_{RANKL}$, cinco variables.

De nuevo, en las Figuras 5.5.11-5.5.15, se muestra el efecto que cada variable tiene por separado sobre el BDG. El resultado es parecido al apartado anterior, ya que para algunos casos la red es capaz de aproximar bien la salida del modelo mientras que para otros hay más distancia entre los puntos. El siguiente paso es realizar el análisis cuantitativo.

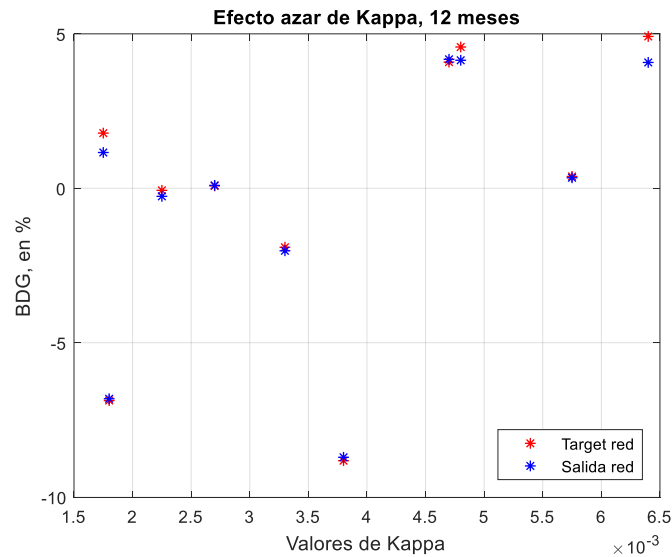


Figura 5.5.11. Efecto de Kappa sobre el aprendizaje de la red neuronal, cinco variables.

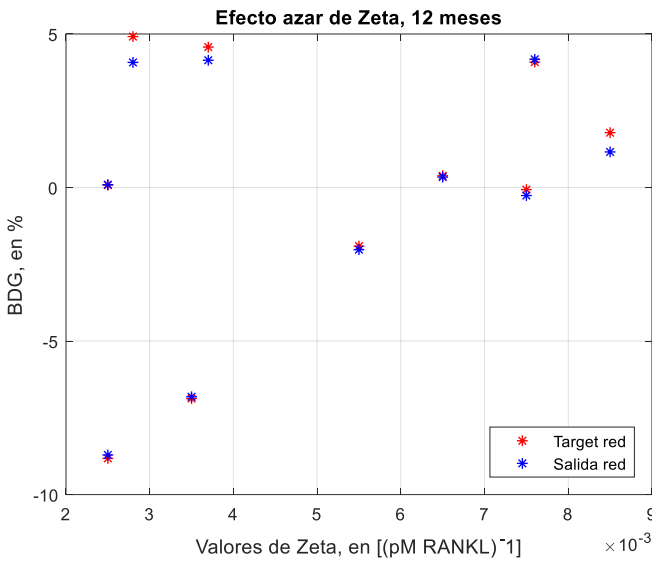


Figura 5.5.12. Efecto de Zeta sobre el aprendizaje de la red neuronal, cinco variables.

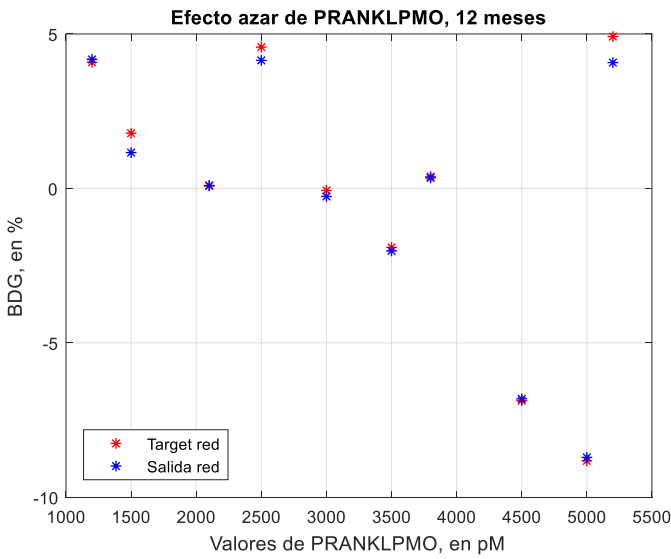


Figura 5.5.13. Efecto de P_{RANKL}^{PMO} sobre el aprendizaje de la red neuronal, cinco variables.

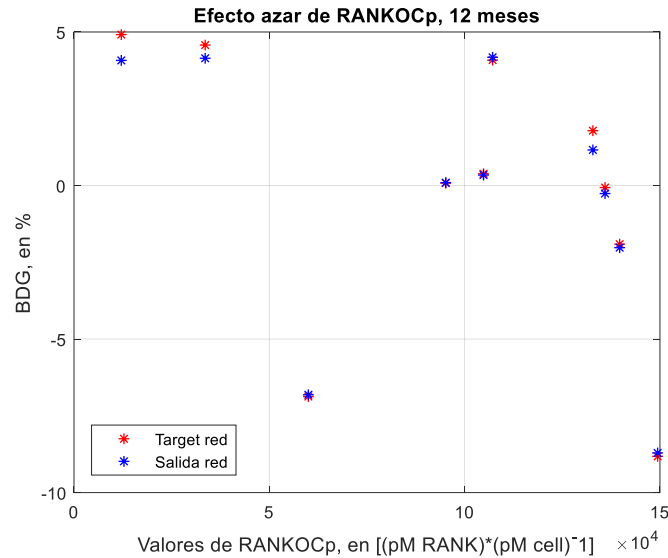


Figura 5.5.14. Efecto de N_{OCp}^{RANK} sobre el aprendizaje de la red neuronal, cinco variables.

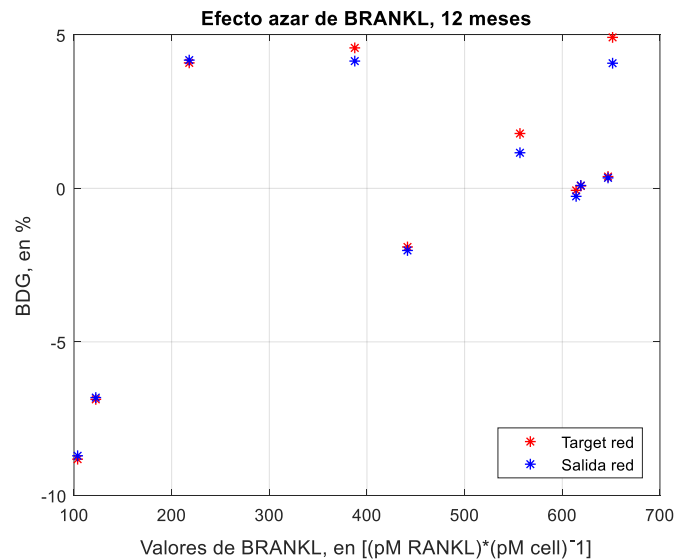


Figura 5.5.15. Efecto de β_{RANKL} sobre el aprendizaje de la red neuronal, cinco variables.

5.6 Análisis del coeficiente de variación para dos, tres, cuatro y cinco variables

En apartados anteriores hemos utilizado el coeficiente de variación para comparar los distintos tipos de entrenamiento posibles para la red neuronal y para elegir el más idóneo, todo ello con dos variables. En esta sección vamos a analizar cómo varía dicho coeficiente cuando disminuimos el número de puntos para entrenamientos aleatorios con varias variables, que según la Figura 5.6.1, dicha relación tiene forma de exponencial negativa.

Por un lado, con dos variables observamos que el salto se produce alrededor de 30 puntos de entrenamiento y que a partir de ese valor, el coeficiente de variación tiende a un valor bastante bajo. Por otro lado, con tres variables dicho salto se produce en torno a 40 puntos. Al aumentar el número de variables, vamos aumentando

el espacio y tiene sentido que sea necesario aumentar el número de puntos.

Sin embargo, para cuatro y cinco variables el espacio aumenta mucho y por esa razón, se realizaron 300 simulaciones en cada uno de ellos, en vez de las 160 realizadas con dos y tres variables. En estos casos, es interesante analizar la tendencia del coeficiente de variación al ir disminuyendo los puntos de entrenamiento y modificando los parámetros de la red.

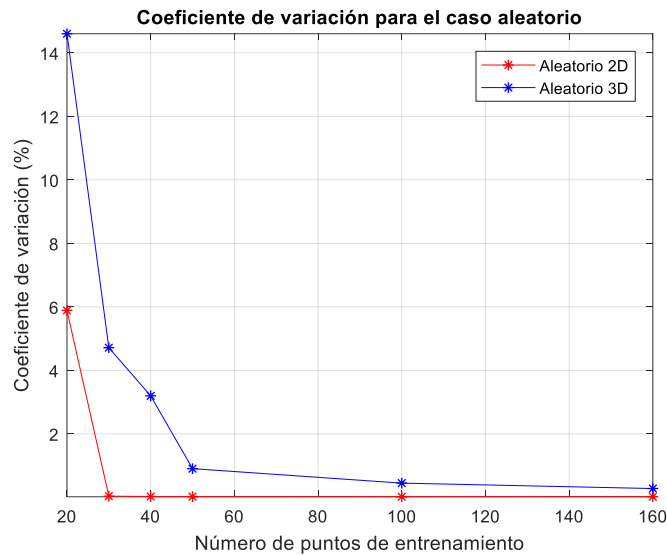


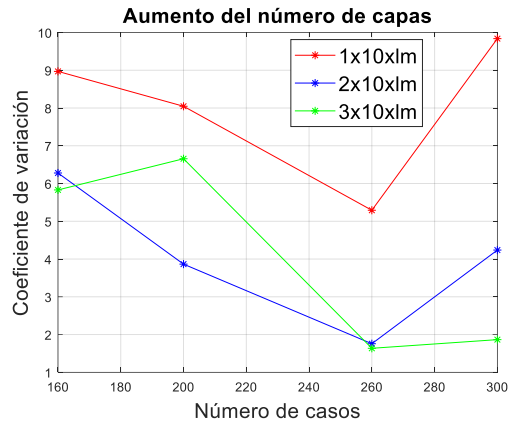
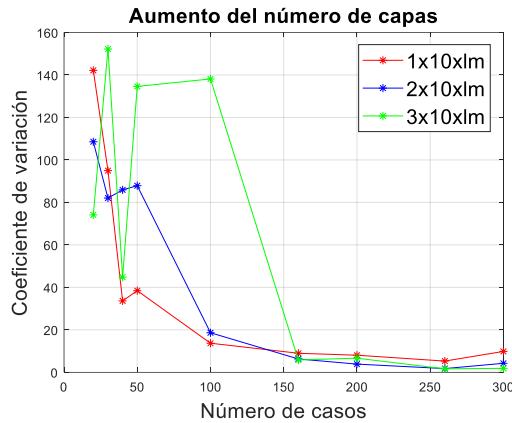
Figura 5.6.1. Coeficiente de variación frente al número de puntos de entrenamiento en el caso aleatorio.

5.6.1 Parámetros de la red

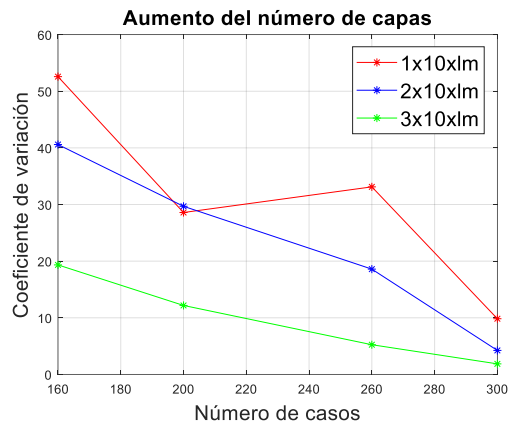
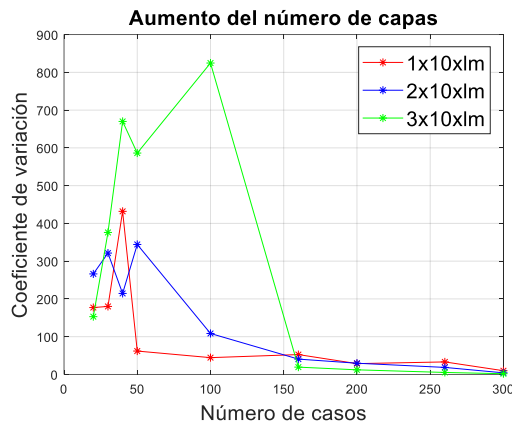
Con el objetivo de analizar más en detalle y de mejorar el coeficiente de variación hemos realizado cinco pruebas para 20, 30, 40, 50, 100, 160, 200, 260 y una prueba para 300 puntos de entrenamiento, y en cada una de ellas hemos probado las siguientes configuraciones de la red: una, dos y tres capas de 10 neuronas (Figuras 5.6.2-5.6.5), una, dos y tres capas de 20 neuronas (Figuras 5.6.6-5.6.9) y finalmente, para las funciones de entrenamiento Levenberg-Marquardt y Bayesian Regularization (Figuras 5.6.10-5.6.13). Todo esto representa un total de 287 redes neuronales, tanto para cuatro como para cinco variables, y para cada configuración nos hemos quedado con la que ofrece menor coeficiente de variación.

5.6.1.1 Influencia del número de capas

Analizando las Figuras 5.6.2-5.6.5, podemos deducir que para este escenario la tendencia se da a partir de los 100-150 casos de entrenamiento y que antes se produce una especie de transitorio que pone de manifiesto la necesidad del aumento del número de casos para cuatro y cinco variables. Si observamos la zona de los 260-300 casos comprobamos que el hecho de aumentar las capas neuronales mejora el coeficiente de variación.



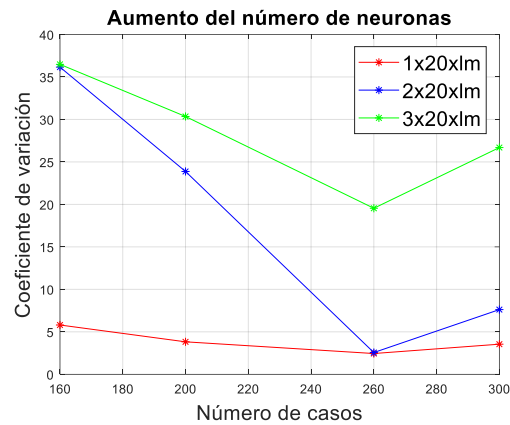
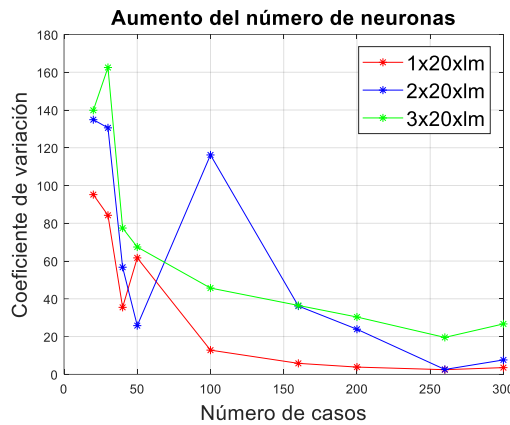
Figuras 5.6.2-5.6.3. Coeficiente de variación frente al número de puntos de entrenamiento para una, dos y tres capas con diez neuronas en cada una de ellas y cuatro variables.



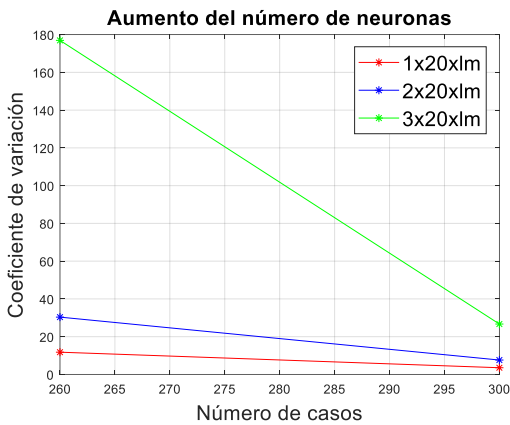
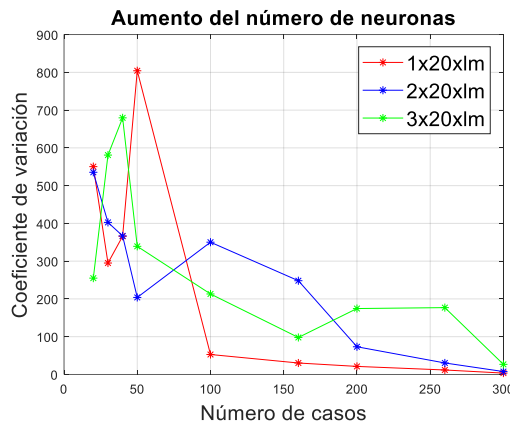
Figuras 5.6.4-5.6.5. Coeficiente de variación frente al número de puntos de entrenamiento para una, dos y tres capas con diez neuronas en cada una de ellas y cinco variables.

5.6.1.2 Influencia del número de neuronas

Si observamos las Figuras 5.6.6-5.6.9, podemos comprobar que aumentar el número de neuronas, complica el problema y no mejora el coeficiente de variación. Tanto para cuatro como para cinco variables, el coeficiente de variación va aumentando a medida que aumentan las capas y las neuronas. Para una capa y 20 neuronas el salto se vuelve a producir entre los 100-150 casos de entrenamiento.



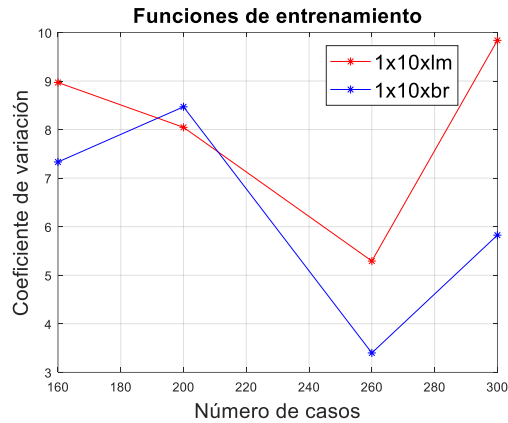
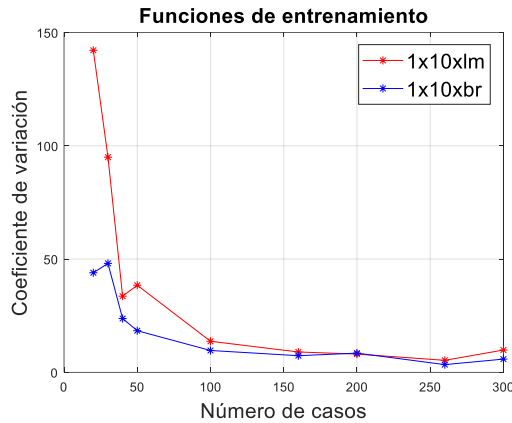
Figuras 5.6.6-5.6.7. Coeficiente de variación frente al número de puntos de entrenamiento para una, dos y tres capas con veinte neuronas en cada una de ellas y cuatro variables.



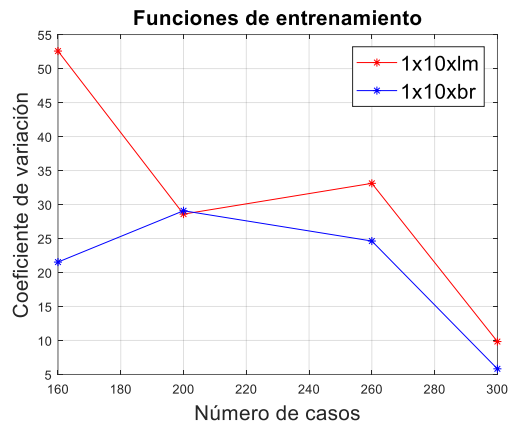
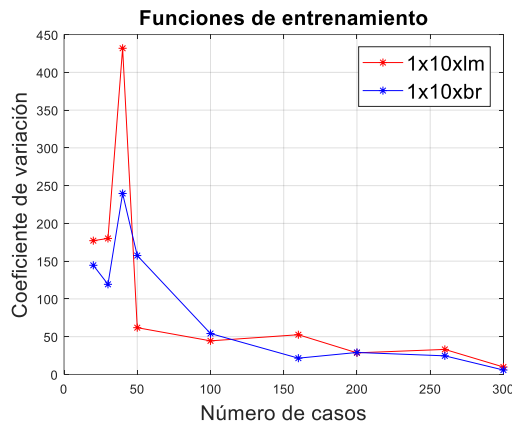
Figuras 5.6.8-5.6.9. Coeficiente de variación frente al número de puntos de entrenamiento para una, dos y tres capas con veinte neuronas en cada una de ellas y cinco variables.

5.6.1.3 Influencia de la función de entrenamiento

Por último nos queda estudiar el efecto de la función de entrenamiento, Figuras 5.6.10-5.6.13. De acuerdo con la literatura, la *Bayesian Regularization* es más adecuada para problemas cambiantes. De hecho, en esa etapa transitoria inicial ofrece coeficientes de variación menores que la *Levenberg-Marquardt*. Sin embargo, conforme se aumenta el número de casos hay poca diferencia entre ambas.



Figuras 5.6.10-5.6.11. Coeficiente de variación frente al número de puntos de entrenamiento para una capa con diez neuronas, funciones de entrenamiento Levenberg-Marquardt y Bayesian Regularization y cuatro variables.



Figuras 5.6.12-5.6.13. Coeficiente de variación frente al número de puntos de entrenamiento para una capa con diez neuronas, funciones de entrenamiento Levenberg-Marquardt y Bayesian Regularization y cinco variables.

5.6.1.4 Selección de la mejor opción

Tras el estudio previo realizado de los parámetros de la red en cuanto a número de capas, de neuronas y función de entrenamiento podemos concluir que la mejor configuración es la correspondiente a una capa, diez neuronas y función de entrenamiento Bayesian Regularization, puesto que si analizamos las gráficas superiores podemos comprobar que es la más estable, la que ofrece menos picos y tiene una tendencia más suave, tanto para cuatro como para cinco variables.

Ahora es el turno de establecer un criterio de selección, teniendo en cuenta el valor numérico del coeficiente de variación, para determinar cuántos puntos de entrenamiento son necesarios para un correcto aprendizaje de la red neuronal en cada una de las dimensiones estudiadas.

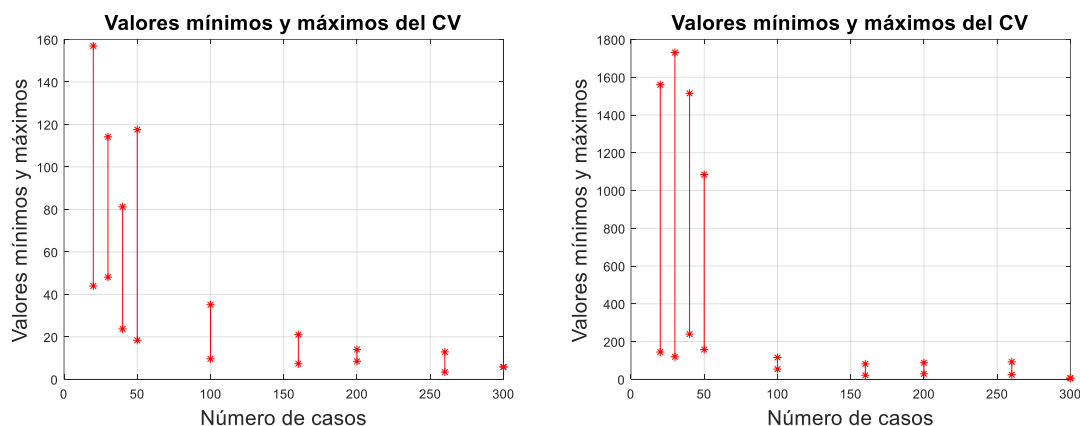
Tabla 5.6.1. Número de puntos de entrenamiento necesarios para conseguir un determinado coeficiente de variación entre la salida del modelo y la salida de la red neuronal, para cuatro y cinco variables.

	Coeficiente de Variación			
Dimensiones	20 – 15%	15 – 10%	10 – 5%	<5%
4D	45 – 70 pts	70 – 100 pts	100 – 300 pts	+300 pts
5D	270 – 280 pts	280 – 290 pts	290 – 300 pts	+300 pts

Teniendo en cuenta el coeficiente de variación en función del número de puntos de entrenamiento para la configuración seleccionada, en las Figuras 5.6.2-5.6.13 se aprecia que las curvas son relativamente parecidas. Sin embargo, hay mucha diferencia con el valor numérico del coeficiente de variación reflejado en el eje vertical y ello se comprueba en la Tabla 5.6.1, poniendo de manifiesto que a medida que se van añadiendo variables al problema se necesitan más casos de entrenamiento para lograr un coeficiente de variación aceptable.

Recordemos que acabamos de realizar un estudio de los parámetros de la red para analizar más en detalle el coeficiente de variación. De cada una de las cinco pruebas realizadas a cada conjunto de casos (20, 30, 40, 50, 100, 160, 200 y 260) nos quedamos con la que ofrece menos coeficiente de variación, en el escenario de entrenamiento aleatorio, para cuatro y cinco variables. Sin embargo, un aspecto interesante es estudiar cómo va siendo esta variación al aumentar el número de casos.

En las Figuras 5.6.14-5.6.15, se representa la variación para una capa, 10 neuronas y función de entrenamiento *Bayesian Regularization*, para cuatro y cinco variables, respectivamente. Cuando tenemos pocos puntos de entrenamiento la variación es bastante considerable. No obstante, al aumentar los casos dicha variación se va reduciendo y se puede apreciar una tendencia. Si por el contrario, al tener más puntos hubiera habido mucha variación, las gráficas obtenidas anteriormente (Figuras 5.6.2-5.6.13) no serían fiables.



Figuras 5.6.14-5.6.15. Coeficiente de variación frente al número de casos de entrenamiento para los valores máximos y mínimos de cada conjunto de casos, para cuatro y cinco variables.

5.7 Importancia de la distribución de los puntos de entrenamiento

No podemos terminar este bloque de resultados sin aclarar un aspecto importante. Para que la curva del coeficiente de variación se parezca a una exponencial negativa, los puntos de entrenamiento deben estar bien distribuidos por todo el dominio, al igual que los puntos de validación. En caso contrario, aparecen picos. Por ejemplo, para un entrenamiento aleatorio, con dos variables y 30 puntos de entrenamiento, en la Figura 5.7.1 los puntos se encuentran distribuidos por el dominio, sin embargo, en la Figura 5.7.2 los puntos están más localizados por las zonas superior e inferior.

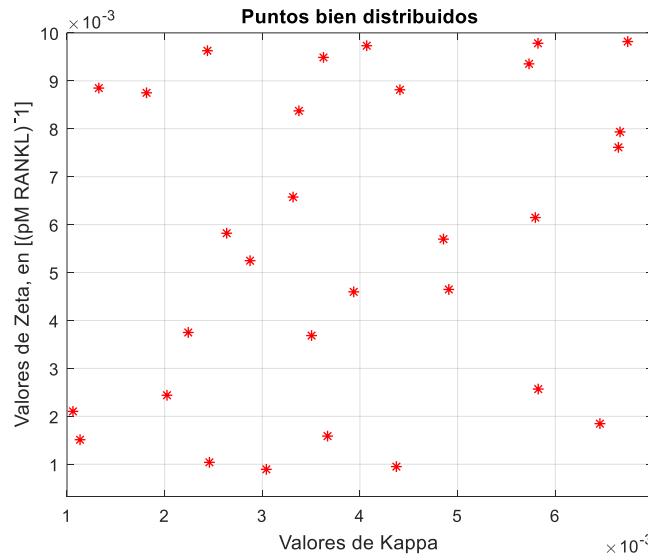


Figura 5.7.1. Entrenamiento aleatorio, dos variables, 30 puntos de entrenamiento bien distribuidos.

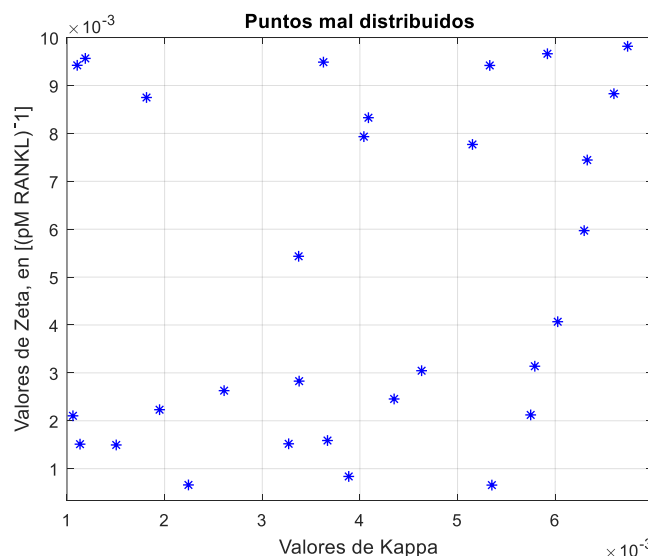


Figura 5.7.2. Entrenamiento aleatorio, dos variables, 30 puntos de entrenamiento más localizados en la zona superior e inferior.

Este hecho, por supuesto influye en el coeficiente de variación, puesto que mientras mejor distribuidos estén los puntos de entrenamiento por todo el dominio, la red neuronal recibe más rango de entrenamiento y es capaz de aproximar mejor la salida del modelo. En la Figura 5.7.3, para 30 puntos, el punto rojo tiene asociado la Figura

5.7.1, mientras que el punto azul la Figura 5.7.2.

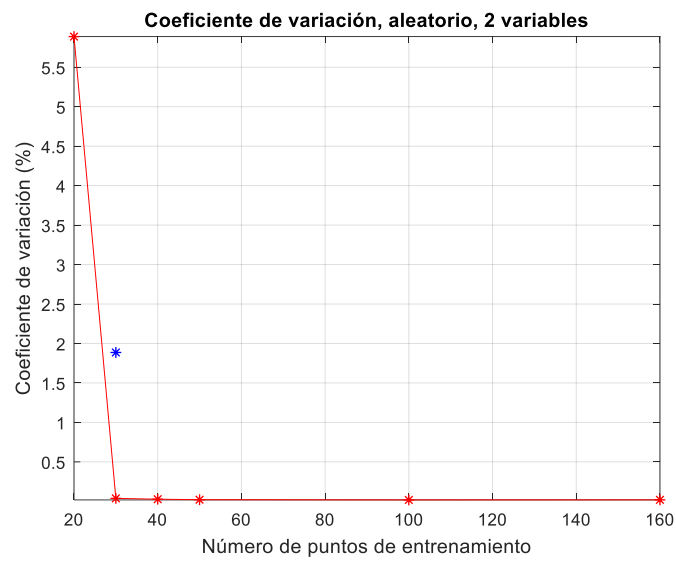


Figura 5.7.3. Entrenamiento aleatorio, 2 variables. El punto azul corresponde a 30 puntos de entrenamiento que no están bien distribuidos por todo el espacio.

6. CONCLUSIONES Y TRABAJOS FUTUROS

“La vida es una preparación para el futuro y la mejor preparación para el futuro es vivir como si no hubiera ninguno”.

-Albert Einstein-

La red neuronal artificial es una buena herramienta de cálculo que trata de simular el comportamiento de nuestro cerebro, en la que el aprendizaje tiene una relación directamente proporcional con el número de datos.

De entre los tipos de entrenamiento estudiados, desacoplado, exhaustivo, pseudoaleatorio y aleatorio, cada uno presenta sus ventajas y desventajas correspondientes y su elección debe depender del tipo de aplicación. En nuestro caso, para evitar el ordenamiento de los puntos de entrenamiento y ahorrar en coste computacional, el entrenamiento aleatorio representa la mejor opción.

En este trabajo, hemos estudiado cómo la red es capaz de aprender cuando tenemos dos, tres, cuatro y cinco variables. Para un determinado coeficiente de variación entre el objetivo de la red y la salida de la red, con más variables se necesitan más puntos de entrenamiento. Si se quiere obtener un error bajo, la relación entre el número de variables y los casos de entrenamiento es exponencial. El coeficiente de variación deseado dependerá del tipo de aplicación.

Número de neuronas en cada capa. En cuanto a la configuración de los parámetros de la red neuronal, el hecho de aumentar el número de capas y el número de neuronas en cada capa, no mejora el coeficiente de variación. El motivo puede ser que el problema se complica y que se necesitan más datos de entrenamiento. Por otro lado, con respecto a las funciones de entrenamiento estudiadas, *Levenberg-Marquardt* y *Bayesian Regularization*, hemos comprobado que esta última es más adecuada cuando tenemos un problema con datos aleatorios que no siguen un determinado patrón.

Un aspecto importante que hemos tenido en cuenta es que los resultados obtenidos corresponden con los que esperamos obtener si los casos de entrenamiento se encuentran bien distribuidos por todo el dominio considerado.

Llegados a este punto, podemos decir que este trabajo representa una base en el estudio de las redes neuronales artificiales, es decir, hemos presentado la herramienta, hemos probado distintos tipos de entrenamiento y hemos analizado las diferentes configuraciones de los parámetros de la red. Por supuesto, en posibles trabajos futuros, se podrá mejorar y ampliar, quizás con la adición de nuevas variables u optimizando el proceso de automatización. Además, sería interesante realizar algún código de Matlab que proporcione una distribución lo más repartida posible de los puntos de entrenamiento.

7. BIBLIOGRAFÍA

- [1] Aloia, J. F., Cohn, S. H., Vaswani, A., Yeh, J. K., Yuen, K., & Ellis, K. (1985). Risk factors for postmenopausal osteoporosis. *The American journal of medicine*, 78(1), 95-100.
- [2] Azam, F. (2000). *Biologically inspired modular neural networks* (Doctoral dissertation, Virginia Tech).
- [3] Basheer, I. A., & Hajmeer, M. (2000). Artificial neural networks: fundamentals, computing, design, and application. *Journal of microbiological methods*, 43(1), 3-31.
- [4] Bilbao, J. N., Sánchez, A. C., & Gil-Antuñano, S. P. (2011). Regulación del metabolismo óseo a través del sistema RANK-RANKL-OPG. *Revista de Osteoporosis y Metabolismo Mineral*, 3(2), 105-112.
- [5] Boskey, A. L. (2013). Bone composition: relationship to bone fragility and antiosteoporotic drug effects. *BoneKEy reports*, 2.
- [6] Boyce, B. F., & Xing, L. (2008). Functions of RANKL/RANK/OPG in bone modeling and remodeling. *Archives of biochemistry and biophysics*, 473(2), 139-146.
- [7] Buenzli, P. R., Pivonka, P., & Smith, D. W. (2011). Spatio-temporal structure of cell distribution in cortical bone multicellular units: a mathematical model. *Bone*, 48(4), 918-926.
- [8] Buenzli, P. R., Pivonka, P., Gardiner, B. S., & Smith, D. W. (2012). Modelling the anabolic response of bone using a cell population model. *Journal of theoretical biology*, 307, 42-52.
- [9] Currey, J. D. (1979). Mechanical properties of bone tissues with greatly differing functions. *Journal of biomechanics*, 12(4), 313-319.
- [10] Del Pino Montes, J. (2010). Osteoporosis: Concepto e importancia. Cuadro clínico. *Revista de Osteoporosis y Metabolismo Mineral*, 2(4), 15-20.
- [11] Del Pino Montes, J. (2012). Coste de la osteoporosis postmenopáusica. *Rev Osteoporos Metab Miner*, 4, 17-21.
- [12] Eastell, R., Christiansen, C., Grauer, A., Kutilek, S., Libanati, C., McClung, M. R., ... & Wang, A. (2011). Effects of denosumab on bone turnover markers in postmenopausal osteoporosis. *Journal of Bone and Mineral Research*, 26(3), 530-537.
- [13] Evia, J. B. (2011). Marcadores del Remodelado Óseo y Osteoporosis. *Revista Mexicana Patologia Clinica*, 58(3), 113-137.
- [14] Fiszelew, A. B. E. L., García-Martínez, R., & de Buenos Aires, T. (2002). Generación automática de redes neuronales con ajuste de parámetros basado en algoritmos genéticos. *Revista del Instituto Tecnológico de Buenos Aires*, 26, 76-101.
- [15] Fuller, K., Wong, B., Fox, S., Choi, Y., & Chambers, T. J. (1998). TRANCE is necessary and sufficient for osteoblast-mediated activation of bone resorption in osteoclasts. *The Journal of experimental medicine*, 188(5), 997-1001.
- [16] González, L. A., Vásquez, G. M., & Molina, J. F. (2009). Epidemiología de la osteoporosis. *Revista Colombiana de Reumatología*, 16(1), 61-75.

- [17] Hansen, M. A., Overgaard, K., Riis, B. J., & Christiansen, C. (1991). Role of peak bone mass and bone loss in postmenopausal osteoporosis: 12 year study. *British medical journal*, 303(6808), 961-964.
- [18] Izaurieta, F., & Saavedra, C. (2000). Redes neuronales artificiales. *Departamento de Física, Universidad de Concepción Chile*.
- [19] Kayri, M. (2016). Predictive abilities of bayesian regularization and Levenberg–Marquardt algorithms in artificial neural networks: a comparative empirical study on social data. *Mathematical and Computational Applications*, 21(2), 20.
- [20] Kohavi, R. (1995, August). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai* (Vol. 14, No. 2, pp. 1137-1145).
- [21] Komarova, S. V., Smith, R. J., Dixon, S. J., Sims, S. M., & Wahl, L. M. (2003). Mathematical model predicts a critical role for osteoclast autocrine regulation in the control of bone remodeling. *Bone*, 33(2), 206-215.
- [22] Krogh, A., & Vedelsby, J. (1995). Neural network ensembles, cross validation, and active learning. In *Advances in neural information processing systems* (pp. 231-238).
- [23] Llerena, G. A. R. (1999). Osteoporosis: impacto socioeconómico e importancia de establecer programas de prevención e intervención a nivel mundial. *Revista Cubana de Reumatología*, 1(1), 4-9.
- [24] Llusá Pérez, M., Merí Vived, A., & Ruano Gil, D. (2004). *Manual y atlas fotográfico de anatomía del aparato locomotor*. Médica Panamericana.
- [25] Manolagas, S. C. (2000). Birth and death of bone cells: basic regulatory mechanisms and implications for the pathogenesis and treatment of osteoporosis. *Endocrine reviews*, 21(2), 115-137.
- [26] Martin, R. B. (1984). Porosity and specific surface of bone. *Critical reviews in biomedical engineering*, 10(3), 179-222.
- [27] Martínez-Reina, J., & Pivonka, P. (2019). Effects of long-term treatment of denosumab on bone mineral density: insights from an in-silico model of bone mineralization. *Bone*, 125, 87-95.
- [28] Martínez-Reina, J., García-Aznar, J. M., Domínguez, J., & Doblaré, M. (2009). A bone remodelling model including the directional activity of BMUs. *Biomechanics and modeling in mechanobiology*, 8(2), 111-127.
- [29] Martínez-Reina, J., García-Aznar, J. M., Domínguez, J., & Doblaré, M. (2008). On the role of bone damage in calcium homeostasis. *Journal of theoretical biology*, 254(3), 704-712.
- [30] Martinez-Reina, J., Ojeda, J., & Mayo, J. (2016). On the use of bone remodelling models to estimate the density distribution of bones. Uniqueness of the solution. *Plos one*, 11(2).
- [31] Matsuo, K., & Irie, N. (2008). Osteoclast–osteoblast communication. *Archives of biochemistry and biophysics*, 473(2), 201-209.
- [32] Navarro, X. (2002). Fisiología del sistema nervioso autónomo. *Revista Neurológica*, 35(6), 553-562.
- [33] Noble, B. S. (2008). The osteocyte lineage. *Archives of biochemistry and biophysics*, 473(2), 106-111.
- [34] Nogués Solán, X., Guerri, R., Solé, E., & Díez-Pérez, A. (2010). Impacto socioeconómico de la osteoporosis. *Rev Osteoporos Metab Miner*, 2, S8-S11.
- [35] Olabe, X. B. (1998). Redes neuronales artificiales y sus aplicaciones. *Publicaciones de la Escuela de Ingenieros*.
- [36] Padrón, M. J., González, L. P., de León, R. P., Fiallo, I. R., & Kim, S. E. M. (2006). Osteoporosis:¿ Un problema de salud prevenible? *Revista Médica Electrónica*, 28(5), 452-459.
- [37] Pal, S. K., & Srimani, P. K. (2006). Neurocomputing motivation, models, and hybridization.

- [38] Parfitt, A. M. (1994). Osteonal and hemi-osteonal remodeling: the spatial and temporal framework for signal traffic in adult human bone. *Journal of cellular biochemistry*, 55(3), 273-286.
- [39] Parfitt, A. M. (2009). Osteocytes: The link between strain, structure and strenght. In *ASBMR. 26 th Annual Meeting Webcast. Plenary Symposium II: New insights into bone strenght: Harold M. Frost memorial session*". <http://app2.capitalreach.com/esp1204/servlet/tc>.
- [40] Pivonka, P., & Komarova, S. V. (2010). Mathematical modeling in bone biology: From intracellular signaling to tissue mechanics. *Bone*, 47(2), 181-189.
- [41] Pivonka, P., Buenzli, P. R., Scheiner, S., Hellmich, C., & Dunstan, C. R. (2013). The influence of bone surface availability in bone remodelling—a mathematical model including coupled geometrical and biomechanical regulations of bone cells. *Engineering Structures*, 47, 134-147.
- [42] Pivonka, P., Zimak, J., Smith, D. W., Gardiner, B. S., Dunstan, C. R., Sims, N. A., ... & Mundy, G. R. (2008). Model structure and control of bone remodeling: a theoretical study. *Bone*, 43(2), 249-263.
- [43] Pivonka, P., Zimak, J., Smith, D. W., Gardiner, B. S., Dunstan, C. R., Sims, N. A., ... & Mundy, G. R. (2010). Theoretical investigation of the role of the RANK–RANKL–OPG system in bone remodeling. *Journal of Theoretical Biology*, 262(2), 306-316.
- [44] Quarles, L. D. (2008). Endocrine functions of bone in mineral metabolism regulation. *The Journal of clinical investigation*, 118(12), 3820-3828.
- [45] Riancho, J. A., & Delgado-Calle, J. (2011). Mecanismos de interacción osteoblasto-osteoclasto. *Reumatología Clínica*, 7, 1-4.
- [46] Salas, R. (2004). Redes neuronales artificiales. *Universidad de Valparaíso. Departamento de Computación*, 1.
- [47] Scheiner, S., Pivonka, P., & Hellmich, C. (2013). Coupling systems biology with multiscale mechanics, for computer simulations of bone remodeling. *Computer Methods in Applied Mechanics and Engineering*, 254, 181-196.
- [48] Scheiner, S., Pivonka, P., Smith, D. W., Dunstan, C. R., & Hellmich, C. (2014). Mathematical modeling of postmenopausal osteoporosis and its treatment by the anti-catabolic drug denosumab. *International journal for numerical methods in biomedical engineering*, 30(1), 1-27.
- [49] Tella, S. H., & Gallagher, J. C. (2014). Prevention and treatment of postmenopausal osteoporosis. *The Journal of steroid biochemistry and molecular biology*, 142, 155-170.
- [50] West, D. (2000). Neural network credit scoring models. *Computers & Operations Research*, 27(11-12), 1131-1152.

8. ANEXOS

En este apartado, se recogen los programas principales de Matlab que se han necesitado para la realización de este trabajo. Ante todo, ha sido necesario construir las matrices de entrada al modelo matemático, Apartado 8.1. Posteriormente, tras realizar el proceso de automatización de las simulaciones, se ejecuta el programa principal, Apartados 8.2 y 8.3, obteniéndose así la matriz de salida. Llegados a este punto, ya podemos construir las matrices de entrada y objetivo de la red neuronal, Apartado 8.4, que son las del modelo pero simplificadas. Para crear las distintas submatrices de casos de entrenamiento, se necesitan los Apartados 8.5 y 8.6. Con todos los apartados anteriores finalizados, se lleva a cabo la validación cruzada, Apartado 8.7 y, finalmente, se representa el coeficiente de variación, Apartado 8.8.

8.1 Creación de la matriz de entrada al modelo matemático

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Selección de puntos al azar %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear, clc

% Nos construimos la Matriz de Entrada Aleatoria 5D a partir de la Aleatoria 4D

Casos = 300;

% Kappain = 0.001;
% Kappafin = 0.007;
%
% KA = Kappain + (Kappafin-Kappain).*rand(Casos,1);
% % KappaA = sort(KA);
%
% Zetain = 0.001*1/3;
% Zetafin = 0.03*1/3;
%
% ZA = Zetain + (Zetafin-Zetain).*rand(Casos,1);
```

```
% % ZetaA = sort(ZA);
%
% P_RANKL_PMOin = 1000;
% P_RANKL_PMOfin = 6000;
%
% PA = P_RANKL_PMOin + (P_RANKL_PMOfin-P_RANKL_PMOin).*rand(Casos,1);
% % P_RANKL_PMOA = sort(PA);
%
% RANK_OCpin = 1e4*1/16;
% RANK_OCpfin = 1e4*16;
%
% RA = RANK_OCpin + (RANK_OCpfin-RANK_OCpin).*rand(Casos,1);
% % RANK_OCpA = sort(RA);

B_RANKLin = (1.6842*1e2)*1/4;
B_RANKLfin = (1.6842*1e2)*4;

BA = B_RANKLin + (B_RANKLfin-B_RANKLin).*rand(Casos,1);
% % B_RANKLA = sort(BA);

load ('InModel_A4_300.mat', 'InModel')
Tamano = size (InModel);

InModel(:,23) = BA(:);

save ('InModel_A5_300.mat', 'InModel')

%%

load ('InModel_A5_300.mat', 'InModel')

C = unique(InModel,'rows');

%%
```

```
Kappain = 0.001;
Kappafin = 0.007;

Zetafin = 0.03*1/3;

P_RANKL_PMOin = 1000;
P_RANKL_PMOfin = 6000;

RANK_OCpin = 1e4*1/16;
RANK_OCpfin = 1e4*16;

B_RANKLin = (1.6842*1e2)*1/4;
B_RANKLfin = (1.6842*1e2)*4;

load ('InModel_A5_300.mat', 'InModel')

RA = InModel(:,13);
BA = InModel(:,23);
KA = InModel(:,39);
ZA = InModel(:,51);
PA = InModel(:,55);

load ('InModel_A5_10.mat', 'InModel')

RP = InModel(:,13);
BP = InModel(:,23);
KP = InModel(:,39);
ZP = InModel(:,51);
PP = InModel(:,55);

figure (1)
plot (KA, ZA, 'r*', KP, ZP, 'b*')
axis ([Kappain Kappafin Zetafin Zetafin])
legend ('Aleatorio', 'Validación')
grid
```

```

xlabel ('Valores de Kappa')
ylabel ('Valores de Zeta, en [(pM RANKL)^-1]')
title ('Distribución de puntos en el caso aleatorio')

figure (2)
plot (KA, PA, 'r*', KP, PP, 'b*')
axis ([Kappain Kappafin P_RANKL_PMOin P_RANKL_PMOfin])
legend ('Aleatorio', 'Validación')
grid
xlabel ('Valores de Kappa')
ylabel ('Valores de PRANKLPMO, en pM')
title ('Distribución de puntos en el caso aleatorio')

figure (3)
plot (KA, RA, 'r*', KP, RP, 'b*')
axis ([Kappain Kappafin RANK_OCpin RANK_OCpfin])
legend ('Aleatorio', 'Validación')
grid
xlabel ('Valores de Kappa')
ylabel ('Valores de RANKOCp, en (pM RANK)*(pM cell)^-1')
title ('Distribución de puntos en el caso aleatorio')

figure (4)
plot (KA, BA, 'r*', KP, BP, 'b*')
axis ([Kappain Kappafin B_RANKLin B_RANKLfin])
legend ('Aleatorio', 'Validación')
grid
xlabel ('Valores de Kappa')
ylabel ('Valores de BRANKL, en (pM RANKL)*(pM cell)^-1')
title ('Distribución de puntos en el caso aleatorio')

figure (5)
plot (ZA, PA, 'r*', ZP, PP, 'b*')
axis ([Zetaain Zetafin P_RANKL_PMOin P_RANKL_PMOfin])
legend ('Aleatorio', 'Validación')
grid
xlabel ('Valores de Zeta, en [(pM RANKL)^-1]')

```

```
ylabel ('Valores de PRANKLPMO, en pM')
title ('Distribución de puntos en el caso aleatorio')
```

```
figure (6)
plot (ZA, RA, 'r*', ZP, RP, 'b*')
axis ([Zetafin Zetafin RANK_OCpin RANK_OCpin])
legend ('Aleatorio', 'Validación')
grid
xlabel ('Valores de Zeta, en [(pM RANKL)^-1]')
ylabel ('Valores de RANKOCp, en (pM RANK)*(pM cell)^-1')
title ('Distribución de puntos en el caso aleatorio')
```

```
figure (7)
plot (ZA, BA, 'r*', ZP, BP, 'b*')
axis ([Zetafin Zetafin B_RANKLin B_RANKLfin])
legend ('Aleatorio', 'Validación')
grid
xlabel ('Valores de Zeta, en [(pM RANKL)^-1]')
ylabel ('Valores de BRANKL, en (pM RANKL)*(pM cell)^-1')
title ('Distribución de puntos en el caso aleatorio')
```

```
figure (8)
plot (PA, RA, 'r*', PP, RP, 'b*')
axis ([P_RANKL_PMOin P_RANKL_PMOfin RANK_OCpin RANK_OCpin])
legend ('Aleatorio', 'Validación')
grid
xlabel ('Valores de PRANKLPMO, en pM')
ylabel ('Valores de RANKOCp, en (pM RANK)*(pM cell)^-1')
title ('Distribución de puntos en el caso aleatorio')
```

```
figure (9)
plot (PA, BA, 'r*', PP, BP, 'b*')
axis ([P_RANKL_PMOin P_RANKL_PMOfin B_RANKLin B_RANKLfin])
legend ('Aleatorio', 'Validación')
grid
xlabel ('Valores de PRANKLPMO, en pM')
ylabel ('Valores de BRANKL, en (pM RANKL)*(pM cell)^-1')
```

```
title ('Distribución de puntos en el caso aleatorio')

figure (10)

plot (RA, BA, 'r*', RP, BP, 'b*')
axis ([RANK_OCpin RANK_OCpfin B_RANKLin B_RANKLfin])
legend ('Aleatorio', 'Validación')
grid
xlabel ('Valores de RANKOCp, en  $(pM \text{ RANK}) * (pM \text{ cell})^{-1}$ ')
ylabel ('Valores de BRANKL, en  $(pM \text{ RANKL}) * (pM \text{ cell})^{-1}$ ')
title ('Distribución de puntos en el caso aleatorio')
```

8.2 Programa principal

```
% run_BONEREMODELING
%
% bone remodeling model based on the BCPM of Pivonka et al. 2008
% extended towards mechanical feedback and consideration of PMO
% and treatment with Denosumab
%
% Javier Martínez-Reina, 2017. Inclusion of mineralisation
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Difference with run_bonemodel_v3c:
% What is new in this routine is the treatment of PMO with Denosumab
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% run file for mechanostat theory
%
% as input parameters a proper loading case LC needs to be chosen
% currently a uniaxial compressive stress state is set as the
% load case at sig = - 30 MPa; this could be representative for bending in
% a long bone;
%
% mechanical load cases:
% LC=14 ... underloading sig_new = - 25 MPa
% LC=15 ... overloading sig_new = - 35 MPa
%
```

```
%  
% the mechanostat needs to work under the following load cases  
%  
% (1) PTH continuous administration:  
%     split up of DOBu*OBu*pi_TGFBe - into a diff & prolif term is very  
%     sensitive - check that the continuous PTH case works  
%  
% (2) Mechanical loading - gain in bone mass after increase of mech load  
%  
% (3) Mechanical unloading - loss of bone (low turnover)  
%  
% (4) case of enforcing osteocyste apoptosis  
%  
%  
clear all  
close all  
  
% run BONEREMODELING  
% addpath('C:\FEMLAB\femlab-peter\bone models\model - 1')  
% addpath('D:\FEMLAB\femlab-peter\bone models\model - 1')  
% runboneremodeling  
%  
% delete initial data files  
%  
delete *.dat  
  
%  
% to run this program a proper loading case LC needs to be chosen  
%  
%  
%  
%-----%  
%                                     %  
% Parameters used for bone remodeling model                                     %  
%                                     %  
%-----%  
% Refer to model paramters file                                     %
```

```
%-----%
%
% read model input parameters
%
% ModelParameter=fu_modelparameters4();

load ('Simulación.mat', 'Simu')
load ('InModel_A5_300.mat', 'InModel')

for sim=Simu

% Vector_in=fu_matriz_in(sim);

D_OBu      = InModel(sim,1);
D_OBp      = InModel(sim,2);
A_OBa      = InModel(sim,3);
D_OBa      = InModel(sim,4);
D_OCu      = InModel(sim,5);
D_OCp      = InModel(sim,6);
A_OCa      = InModel(sim,7);

KD_TGF_beta_activate = InModel(sim,8);
KD_TGF_beta_repress  = InModel(sim,9);
KD_PTH_activate      = InModel(sim,10);
KD_PTH_repress       = InModel(sim,11);
KD_RANKL_activate    = InModel(sim,12);
RANK                 = InModel(sim,13);
KD_MCSF_activate     = InModel(sim,14);

P_RANKL_d   = InModel(sim,15);
P_OPG_d     = InModel(sim,16);
P_PTH_d     = InModel(sim,17);

Deg_PTH     = InModel(sim,18);
Beta_PTH    = InModel(sim,19);
```

```
Beta_OPG      = InModel(sim,20);
Deg_OPG       = InModel(sim,21);
OPG_max       = InModel(sim,22);

Beta_RANKL    = InModel(sim,23);
R_RANKL       = InModel(sim,24);
Deg_RANKL     = InModel(sim,25);
KA1_RANKL     = InModel(sim,26);
KA2_RANKL     = InModel(sim,27);

K_res         = InModel(sim,28);
Alpha         = InModel(sim,29);

R1_OBp_RANKL  = InModel(sim,30);
R2_OBa_RANKL  = InModel(sim,31);
Beta1_OBp_OPG = InModel(sim,32);
Beta2_OBa_OPG = InModel(sim,33);

sig_macro_t0  = InModel(sim,34);
de_sig_macro  = InModel(sim,35);

tresid        = InModel(sim,36);
DT            = InModel(sim,37);
MINP          = InModel(sim,38);
XKAPPA        = InModel(sim,39);
VMPRIM        = InModel(sim,40);
VMMAX         = InModel(sim,41);
DeltaT        = InModel(sim,42);

k_OBu         = InModel(sim,43);
k_OBp         = InModel(sim,44);
k_OCu         = InModel(sim,45);
k_OCp         = InModel(sim,46);
```

```
%set global variables
```

```
% global  OBp_Scaling_Factor  OBa_Scaling_Factor  Multiplying_Factor
ParameterWrite

%setappdata(0, 'version',version);

%

% define 9 loadcases (see paper Lemaire)

%

% LoadCase= 3 ... continuous PTH (Lemaire)

%

% LoadCase= 10 ... steady state (no loading)

%           I_PTH = 0

%

% LoadCase= 11 ...  I_PTH = 0 & A_OBa < A_OBa_t0

%

%

% LoadCase= 13 ...  I_PTH = 0 & pd_OBa > 0 (just to see difference to LC 12)

%

%

% LoadCase= 14 ...  |sig_macro| < |sig_macro_t0| (simulation of decreased
mechanical usage)

%

%

% LoadCase= 15 ...  |sig_macro| > |sig_macro_t0| (simulation of increased
mechanical usage)

%

%

LoadCase=35;

ParameterWrite = 0;

setappdata(0,'ParameterWrite',ParameterWrite);

Multiplying_Factor = 0.00;

error = 1e-6;

%

% define drug administration interval

% t_0 ... ODE t_start
```

```
% t_end ... ODE t_end
% delta_t ... ODE step size
% t_int1 ... start injecting
% t_int2 ... stop injecting
TimeInitial = 0;
TimeEnd = 11*365/DeltaT; %10*365;
TimeInterval_1 = 0;
TimeInterval_2 = 1*365/DeltaT; % 0.5;
TimeInterval_3 = 11*365/DeltaT;

%Initialization of CellsVector
CellsVector = zeros(TimeInterval_3 , 7);
%Initialization of VFPREV
VFPREV = zeros(tresid,1);

% Import the temporal evolution of Denosumab with the corresponding dose
%if (sim>=1) && (sim<=(N*N))
    load('treatment_02REV.mat') % The array Denosumab is imported
%elseif (sim>=(N*N+1)) && (sim<=(2*N*N))
    %load('treatment_14REV.mat') % The array Denosumab is imported
%elseif (sim>=(2*N*N+1)) && (sim<=(3*N*N))
    %load('treatment_17REV.mat') % The array Denosumab is imported
%end

setappdata(0,'LoadCase',LoadCase);
setappdata(0,'TimeInterval_1',TimeInterval_1);
setappdata(0,'TimeInterval_2',TimeInterval_2);
setappdata(0,'TimeInterval_3',TimeInterval_3);

%
% default values

% call individual loadcase fu
% loadcase is a function of the loadcase and loading times.
LoadCaseParameter=fu_loadcase4(LoadCase,Denosumab,TimeInterval_1,TimeInterval_2,
TimeInterval_3,TimeInitial);
```

```
% different possibilities for the load cases.

D_OBu          = LoadCaseParameter(1);
D_OBp          = LoadCaseParameter(2);
D_OBa          = LoadCaseParameter(3);
A_OBa          = LoadCaseParameter(4);
D_OCp          = LoadCaseParameter(5);
D_OCp          = LoadCaseParameter(6);
A_OCa          = LoadCaseParameter(7);


pd_OBp         = LoadCaseParameter(8);
pd_OBa         = LoadCaseParameter(9);
pd_OCY         = LoadCaseParameter(10);
pd_OCp         = LoadCaseParameter(11);
pd_OCa         = LoadCaseParameter(12);


P_RANKL_d      = LoadCaseParameter(13);
P_OPG_d        = LoadCaseParameter(14);
P_PTH_d        = LoadCaseParameter(15);


Beta_OPG       = LoadCaseParameter(16);
RANK           = LoadCaseParameter(17);
R_RANKL        = LoadCaseParameter(18);
KD_RANKL_activate = LoadCaseParameter(19);


de_sig_macro   = LoadCaseParameter(21);      % MPa


Denos          = LoadCaseParameter(24);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%    Steady State Calculation
%
%    constants:
%    - precursor cell numbers
%    - initial bone volume
%    - external force
```

```

    OBU_t0 = 1.0e-2; % pM
    OCu_t0 = 1.0e-2; % pM
    OCY_t0 = 1.0e-2; % pM
    fbm_t0 = InModel(sim,53); % [%] %Valor nominal
%
% provide steady-state parameters for
% boneremodeling subroutine
setappdata(0, 'OBU_t0', OBU_t0);
setappdata(0, 'OCY_t0', OCY_t0);
setappdata(0, 'OCu_t0', OCu_t0);
setappdata(0, 'fbm_t0', fbm_t0);
setappdata(0, 'sig_macro_t0', sig_macro_t0);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%% Steady State Calculation

%
% newly calibrated cell numbers
%
% [OBp ; OBa; OCp; OCa]
% OBp ~ 2 x OBa
% OBa ~ 10 x OCa (consider OCa has 10 nuclei)
% OCp ~ 10 x OCa
%
% old cell numbers
% cells_old = [6.196390627918603e-004,...
%             5.583931899482344e-004,...
%             5.000000000000000e-003,...
%             8.069635262731931e-004];
%
cells_0 = [1.00e-003,...
          5.00e-004,...
          1.00e-003,...
          1.00e-004];

```

```
options=optimset('TolFun',1e-14,'TolX',1e-14); % Option tolerances

[cells_t0] = fsolve(@fu_steadystate4,cells_0,options);
cells_t0

% NOTE: The same function fu_steadystate4 is used since in the steady state
% the geometrical feedback has no effect --->  $g_i = (S_v/S_{v\_t0})^{k_i}$ . The
% quotient is equal to 1.
%
% state variables
%
OBp_t0 = cells_t0(1);
OBa_t0 = cells_t0(2);
OCp_t0 = cells_t0(3);
OCa_t0 = cells_t0(4);

%
% some of the model parameters are not independent:
% (1)K_form ... determined by K_res and dBV_t0
% (2)A_OCY ... determined by D_OBa and dOCY_t0
%
% compute resorption/formation ratio

% resorption/formation rate from original parameter set
K_form = K_res*OCa_t0/OBa_t0; %
%
% compute OCY apoptosis rate
A_OCY = D_OBa*OBa_t0/OCa_t0; %

% provide steady-state parameters for
% boneremodeling subroutine
setappdata(0, 'OBa_t0',OBa_t0);
setappdata(0, 'OCa_t0',OCa_t0);
setappdata(0, 'K_form',K_form);
setappdata(0, 'K_res',K_res);
```

```
setappdata(0, 'D_OBa',D_OBa);
setappdata(0, 'A_OCY',A_OCY);

% Calculate the initial ash fraction (ash_fraction0)
% as a function of the initial turnover rate

% First, the queue VFPREV is initialized
% as a function of DVF and DVR, the volumes of tissue formed and resorbed
% at the equilibrium

DVR = OCa_t0*K_res*DeltaT/100;
DVF = DVR;           % Remodelling equilibrium is assumed
p    = 1-fbm_t0/100;  % Initial porosity in [0,1]

i=0;
while i<length(VFPREV)
    VFPREV = fu_updateVFPREV(p,DVR,DVF,VFPREV);
    i=i+1;
end

% Then, the initial ash fraction is calculated

VM_t0 = 0;

for i=1:length(VFPREV)
    VM_t0= VM_t0+VFPREV(i)*fu_calmineral(i*DeltaT,DT,MINP,XKAPPA,VMPRIM,VMMAX);
end

VM_t0=VM_t0/(1-p);

ash_t0 = 3.2*VM_t0/(3.2*VM_t0+1.1*3.0/7.0);
setappdata(0, 'ash_t0',ash_t0);
%
% define the initial conditions for the ODE solver
%
%x_t0 = [OBp_t0,OBa_t0,OCY_t0,OCp_t0,OCa_t0,fbm_t0];
```

```

CellsVector(1,:)=[OBp_t0,OBa_t0,OCY_t0,OCp_t0,OCa_t0,fbm_t0,VM_t0];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% ODE Calculations

ParameterWrite = 0;

for i=2:length(CellsVector(:,1))
%   for i=2:2

% Integration of ODEs
    xdot = fu_bonecelldynamics4(i,CellsVector(i-1,:),Denosumab);
    for j=1:6
        CellsVector(i,j) = CellsVector(i-1,j) + xdot(j) * DeltaT;
        if CellsVector(i,j) < 0
            CellsVector(i,j) = 0;
        end
    end

% Checking that fbm lies within [1,99]
    if CellsVector(i,6) > 99
        CellsVector(i,6) = 99;
    elseif CellsVector(i,6) < 1
        CellsVector(i,6) = 1;
    end

    OBa = CellsVector(i,2);
    OCa = CellsVector(i,5);
    fvas = 100 - CellsVector(i,6);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Mineralization
% Update the queue VFPREV
    aux1 = OCa*K_res*DeltaT/100;
    aux2 = OBa*K_form*DeltaT/100;
    VFPREV = fu_updateVFPREV(fvas/100,aux1,aux2,VFPREV);

```

```
% Calculate the mineral content
VM = 0;

for j=1:length(VFPREV)
VM = VM+VFPREV(j)*fu_calmineral(j*DeltaT,DT,MINP,XKAPPA,VMPRIM,VMMAX);
end

VM=VM/(1-fvas/100);

ash = 3.2*VM/(3.2*VM+1.1*3.0/7.0);

% Uncomment the following lines to switch off mineralization
% ash = ash_t0;
% VM = VM_t0;

% End of mineralization
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
CellsVector(i,7) = VM;

end

TimeVector = 1:length(CellsVector(:,1));

%options = odeset('RelTol',error,'AbsTol',error);
%[TimeVector,CellsVector] = ode23s('fu_bonecelldynamics2b',[TimeInitial
TimeEnd],x_t0,options);

ParameterWrite = 1;
setappdata(0,'ParameterWrite',ParameterWrite);

for i=1:length(CellsVector(:,1))
ParameterVector(i,:) = fu_bonecelldynamics4(i,CellsVector(i,:),Denosumab)';
end

ParameterWrite = 0;
```

```
setappdata(0,'ParameterWrite',ParameterWrite);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Plotting Output
absolute = 1;
length2 = length(TimeVector);

if absolute == 1
% OBp concentration time plot
figure;
plot(TimeVector,CellsVector(:,1),'b--','LineWidth',2);
hold on;
plot(TimeVector,CellsVector(:,2),'b-','LineWidth',2);
plot(TimeVector,CellsVector(:,4),'r--','LineWidth',2);
plot(TimeVector,CellsVector(:,5),'r-','LineWidth',2);
%title('Time evolution of OBp concentrations', 'FontSize', 12);
xlabel('Time [days]', 'FontSize', 12);
ylabel('Cell levels [pM]', 'FontSize', 12);

% OBa concentration time plot
%figure;
%plot(TimeVector,CellsVector(:,2),'k-','LineWidth',2);
%hold on;
%title('Time evolution of OBa concentrations', 'FontSize', 12);
%xlabel('Time [days]', 'FontSize', 12);
%ylabel('OBa levels', 'FontSize', 12);

% OCY concentration time plot
%figure;
%plot(TimeVector,CellsVector(:,3),'k-','LineWidth',2);
%hold on;
%title('Time evolution of OCa concentrations', 'FontSize', 12);
%xlabel('Time [days]', 'FontSize', 12);
%ylabel('OCY levels', 'FontSize', 12);
```

```
% OCp concentration time plot
%figure;
%plot(TimeVector,CellsVector(:,4),'k-','LineWidth',2);
%hold on;
%title('Time evolution of OCa concentrations', 'FontSize', 12);
%xlabel('Time [days]', 'FontSize', 12);
%ylabel('OCp levels', 'FontSize', 12);

% OCa concentration time plot
%figure;
%plot(TimeVector,CellsVector(:,5),'k-','LineWidth',2);
%hold on;
%title('Time evolution of OCa concentrations', 'FontSize', 12);
%xlabel('Time [days]', 'FontSize', 12);
%ylabel('OCa levels', 'FontSize', 12);

else

% OBp concentration time plot
figure;
plot(TimeVector(1:length2),CellsVector(1:length2,1)./OBp_t0,'k-','LineWidth',2);
hold on;
%title('Time evolution of OBp concentrations', 'FontSize', 12);
xlabel('Time [days]', 'FontSize', 12);
ylabel('Normalized OBp levels', 'FontSize', 12);

% OBa concentration time plot
figure;
plot(TimeVector(1:length2),CellsVector(1:length2,2)./OBa_t0,'k-','LineWidth',2);
hold on;
%title('Time evolution of OBa concentrations', 'FontSize', 12);
xlabel('Time [days]', 'FontSize', 12);
```

```
ylabel('Normalized OBa levels', 'FontSize', 12);

% OCY concentration time plot
figure;
plot(TimeVector(1:length2),CellsVector(1:length2,3)./OCY_t0,'k-','LineWidth',2);
hold on;
%title('Time evolution of OCa concentrations', 'FontSize', 12);
xlabel('Time [days]', 'FontSize', 12);
ylabel('Normalized OCY levels', 'FontSize', 12);

% OCp concentration time plot
figure;
plot(TimeVector(1:length2),CellsVector(1:length2,4)./OCp_t0,'k-','LineWidth',2);
hold on;
%title('Time evolution of OCa concentrations', 'FontSize', 12);
xlabel('Time [days]', 'FontSize', 12);
ylabel('Normalized OCp levels', 'FontSize', 12);

% OCa concentration time plot
figure;
plot(TimeVector(1:length2),CellsVector(1:length2,5)./OCa_t0,'k-','LineWidth',2);
hold on;
%title('Time evolution of OCa concentrations', 'FontSize', 12);
xlabel('Time [days]', 'FontSize', 12);
ylabel('Normalized OCa levels', 'FontSize', 12);

% all cell # concentration time plot
figure;
plot(TimeVector(1:length2),CellsVector(1:length2,1)./OBp_t0,'b--',
'LineWidth',2);
hold on;
%title('Time evolution of OBa concentrations', 'FontSize', 12);
xlabel('Time [days]', 'FontSize', 12);
ylabel('Normalized cell levels [-]', 'FontSize', 12);
plot(TimeVector,CellsVector(1:length2,2)./OBa_t0,'b-','LineWidth',2);
plot(TimeVector,CellsVector(1:length2,5)./OCa_t0,'r-','LineWidth',2);
```

```
%h = legend('OB_p','OB_a','OC_a',1);

end

%
% calculate bone mass ballance
figure;
plot(TimeVector(1:length2),CellsVector(1:length2,6),'k-','LineWidth',2);
hold on;
%title('Time evolution of change in bone volume for different version',
'FontSize', 12);
xlabel('Time [days]', 'FontSize', 12);
ylabel('bone matrix fraction f_{bm} [%]', 'FontSize', 12);

%
% calculate bone mass ballance
figure;
plot(TimeVector(1:length2),100-CellsVector(1:length2,6),'k-','LineWidth',2);
hold on;
%title('Time evolution of change in bone volume for different version',
'FontSize', 12);
xlabel('Time [days]', 'FontSize', 12);
ylabel('vascular pore fraction f_{vas} [%]', 'FontSize', 12);

%
% calculate bone mass ballance
figure;
plot(TimeVector(1:length2),ParameterVector(1:length2,25),'k-','LineWidth',2);
hold on;
%title('Time evolution of change in apparent density', 'FontSize', 12);
xlabel('Time [days]', 'FontSize', 12);
ylabel('apparent density [g/cm^3]', 'FontSize', 12);

%
% calculate bone mass ballance
```

```
figure;
plot(TimeVector(1:length2),ParameterVector(1:length2,24),'k-','LineWidth',2);
hold on;
%title('Time evolution of change in apparent density', 'FontSize', 12);
xlabel('Time [days]', 'FontSize', 12);
ylabel('ash fraction', 'FontSize', 12);

%
% calculate mechanical Force
%figure;
%plot(TimeVector(1:length2),CellsVector(:,7),'k-','LineWidth',2);
%hold on;
%title('Time evolution of change in bone volume for different version',
'FontSize', 12);
%xlabel('Time [days]', 'FontSize', 12);
%ylabel('macro stress [MPa]', 'FontSize', 12);

%
% calculate mechanical stress
%figure;
%plot(TimeVector(1:length2),CellsVector(:,7)./CellsVector(:,6),'k-
','LineWidth',2);
%hold on;
%title('Time evolution of change in bone volume for different version',
'FontSize', 12);
%xlabel('Time [days]', 'FontSize', 12);
%ylabel('stress [N]', 'FontSize', 12);

%E_bone          = 1.00e+02;
%eps              = (CellsVector(:,7)./CellsVector(:,6))/E_bone;
%
% calculate mechanical stress
figure;
plot(TimeVector(1:length2),ParameterVector(1:length2,16),'k-','LineWidth',2);
hold on
%title('Time evolution of change in bone volume for different version',
'FontSize', 12);
xlabel('Time [days]', 'FontSize', 12);
```

```
ylabel('macro stress [MPa]', 'FontSize', 12);

%
% calculate SED_bm0 & SED_bm
figure;
plot(TimeVector(1:length2),ParameterVector(1:length2,17),'k-','LineWidth',2);
hold on
plot(TimeVector,ParameterVector(:,18),'k-','LineWidth',2);
%title('Time evolution of change in bone volume for different version',
'FontSize', 12);
xlabel('Time [days]', 'FontSize', 12);
ylabel('SED_{ebm} [-]', 'FontSize', 12);

%
% calculate Pi_eps: anabolic control fu
figure;
plot(TimeVector(1:length2),ParameterVector(1:length2,19),'k-','LineWidth',2);
hold on
%title('Time evolution of change in bone volume for different version',
'FontSize', 12);
xlabel('Time [days]', 'FontSize', 12);
ylabel('\pi_{ebm} [-]', 'FontSize', 12);

%
% calculate P_eps_RANKL: catabolic control fu
figure;
plot(TimeVector(1:length2),ParameterVector(1:length2,20),'k-','LineWidth',2);
hold on
%title('Time evolution of change in bone volume for different version',
'FontSize', 12);
xlabel('Time [days]', 'FontSize', 12);
ylabel('P_{ebm,RANKL} [pM/day]', 'FontSize', 12);

%
% calculate P_eps_RANKL: catabolic control fu
figure;
plot(TimeVector(1:length2),ParameterVector(1:length2,14),'k-','LineWidth',2);
hold on
```

```
%title('Time evolution of change in bone volume for different version',
'FontSize', 12);

xlabel('Time [days]', 'FontSize', 12);
ylabel('\pi_{RANKL} [pM/day]', 'FontSize', 12);

%
% calculate specific surface area
%figure;
%plot(TimeVector(1:length2),ParameterVector(1:length2,19),'k-','LineWidth',2);
%hold on

%title('Time evolution of change in bone volume for different version',
'FontSize', 12);
%xlabel('Time [days]', 'FontSize', 12);
%ylabel('S_{v} [1/mm]', 'FontSize', 12);

%
% calculate specific surface area
%figure;
%plot(ParameterVector(1:length2,20),ParameterVector(1:length2,19),'k-
','LineWidth',2);
%hold on

%title('Time evolution of change in bone volume for different version',
'FontSize', 12);
%xlabel('f_{vas} [-]', 'FontSize', 12);
%ylabel('S_{v} [1/mm]', 'FontSize', 12);

% calculate mechanical set point strain
%figure;
%plot(TimeVector(1:length2),ParameterVector(1:length2,18),'k-','LineWidth',2);
%hold on;

%title('Time evolution of change in bone volume for different version',
'FontSize', 12);
%xlabel('Time [days]', 'FontSize', 12);
%ylabel('set point strain \epsilon_{sp}[-]', 'FontSize', 12);

% calculate change in bone stiffness
%figure;
```

```
%plot(TimeVector(1:length2),ParameterVector(1:length2,19),'k-','LineWidth',2);
%hold on;
%title('Time evolution of change in bone volume for different version',
'FontSize', 12);
%xlabel('Time [days]', 'FontSize', 12);
%ylabel('E_{bone} [N/mm^2]', 'FontSize', 12);

% calculate difference between actual strain and set point strain
%figure;
%plot(TimeVector(1:length2),ParameterVector(1:length2,18) -
ParameterVector(:,17),'k-','LineWidth',2);
%hold on;
%title('Time evolution of change in bone volume for different version',
'FontSize', 12);
%xlabel('Time [days]', 'FontSize', 12);
%ylabel('(\epsilon_{sp}- \epsilon) [-]', 'FontSize', 12);

% calculate difference between actual strain and set point strain
%figure;
%plot(TimeVector(1:length2),ParameterVector(1:length2,15),'k-','LineWidth',2);
%hold on;
%title('Time evolution of change in bone volume for different version',
'FontSize', 12);
%xlabel('Time [days]', 'FontSize', 12);
%ylabel('\pi_{Wnt} [-]', 'FontSize', 12);

if absolute == 1

% % plot OPG, RANKL, RANK concentrations

%% plot of Normalized OPG
figure;
plot(TimeVector(1:length2),ParameterVector(1:length2,3),'k-','LineWidth',2);
hold on;
title('OPG levels [pM]', 'FontSize', 12);
xlabel('t [days]');
```

```
ylabel('OPG [pM]');

% Normalized RANKL
figure;
plot(TimeVector(1:length2),ParameterVector(1:length2,4),'k-','LineWidth',2);
hold on;
title('RANK levels', 'FontSize', 12);
xlabel('Time [days]', 'FontSize', 12);
ylabel('RANK1 [pM]', 'FontSize', 12);

%
figure;
plot(TimeVector(1:length2),ParameterVector(1:length2,7),'k-','LineWidth',2);
hold on;
%title('Plot of RANKL concentration', 'FontSize', 12);
xlabel('Time [days]', 'FontSize', 12);
ylabel('RANKL levels [pM]', 'FontSize', 12);

else
    % % plot OPG, RANKL, RANK concentrations

    %%% plot of Normalized OPG
    figure;
    plot(TimeVector(1:length2),ParameterVector(1:length2,3)./ParameterVector(1,2),'k-','LineWidth',2);
    hold on;
    %title('Plot of OPG concentration', 'FontSize', 12);
    xlabel('t [days]');
    ylabel('OPG [pM]');

    % Normalized RANKL
    figure;
    plot(TimeVector(1:length2),ParameterVector(1:length2,3)./ParameterVector(1,3),'k-','LineWidth',2);
    hold on;
    % title('Normalized RANKL levels', 'FontSize', 12);
    xlabel('Time [days]', 'FontSize', 12);
    ylabel('RANK1 [pM]', 'FontSize', 12);
```

```
%  
figure;  
plot(TimeVector(1:length2),ParameterVector(1:length2,4)./ParameterVector(1,4),'k-','LineWidth',2);  
hold on;  
%title('Plot of RANKL concentration', 'FontSize', 12);  
xlabel('Time [days]', 'FontSize', 12);  
ylabel('Normalized RANKL levels', 'FontSize', 12);  
  
%  
figure;  
plot(TimeVector(1:length2),ParameterVector(1:length2,11),'k-','LineWidth',2);  
hold on;  
%title('Plot of RANKL concentration', 'FontSize', 12);  
xlabel('Time [days]', 'FontSize', 12);  
ylabel('Pi_TGFbeta_OBu_act', 'FontSize', 12);  
  
%  
figure;  
plot(TimeVector(1:length2),ParameterVector(1:length2,12),'k-','LineWidth',2);  
hold on;  
%title('Plot of RANKL concentration', 'FontSize', 12);  
xlabel('Time [days]', 'FontSize', 12);  
ylabel('Pi_TGFbeta_OCa_act', 'FontSize', 12);  
  
%  
figure;  
plot(TimeVector(1:length2),ParameterVector(1:length2,13),'k-','LineWidth',2);  
hold on;  
%title('Plot of RANKL concentration', 'FontSize', 12);  
xlabel('Time [days]', 'FontSize', 12);  
ylabel('Pi_TGFbeta_OBp_rep', 'FontSize', 12);  
  
end
```

```

write_steady_state = 1;

if write_steady_state == 1

    OBp_t0 = CellsVector(length(CellsVector(:,1)),1);
    OBa_t0 = CellsVector(length(CellsVector(:,1)),2);
    OCY_t0 = CellsVector(length(CellsVector(:,1)),3);
    OCp_t0 = CellsVector(length(CellsVector(:,1)),4);
    OCa_t0 = CellsVector(length(CellsVector(:,1)),5);
    BV_t0  = CellsVector(length(CellsVector(:,1)),6);
    ash_t0 = CellsVector(length(CellsVector(:,1)),7);

    PTH_tot      = ParameterVector(length(CellsVector(:,1)),1);
    OPG_eff_t0   = ParameterVector(length(CellsVector(:,1)),2);
    OPG_t0       = ParameterVector(length(CellsVector(:,1)),3);
    RANK_t0      = ParameterVector(length(CellsVector(:,1)),4);
    RANKL_eff_t0 = ParameterVector(length(CellsVector(:,1)),5);
    RANKL_tot_t0 = ParameterVector(length(CellsVector(:,1)),6);
    RANKL_t0     = ParameterVector(length(CellsVector(:,1)),7);
    RANKL_RANK_t0 = ParameterVector(length(CellsVector(:,1)),8);
    Pi_PTH_act   = ParameterVector(length(CellsVector(:,1)),9);
    Pi_PTH_rep   = ParameterVector(length(CellsVector(:,1)),10);
    Pi_TGFbeta_OBu_act = ParameterVector(length(CellsVector(:,1)),11);
    Pi_TGFbeta_OCa_act = ParameterVector(length(CellsVector(:,1)),12);
    Pi_TGFbeta_OBp_rep = ParameterVector(length(CellsVector(:,1)),13);
    Pi_RANK_act   = ParameterVector(length(CellsVector(:,1)),14);
    Pi_MCSF_act   = ParameterVector(length(CellsVector(:,1)),15);
    sig_macro     = ParameterVector(length(CellsVector(:,1)),16);
    SED_bm0       = ParameterVector(length(CellsVector(:,1)),17);
    SED_bm        = ParameterVector(length(CellsVector(:,1)),18);
    Pi_eps        = ParameterVector(length(CellsVector(:,1)),19);

end

out1      = TimeVector(1:length2);
out2      = ParameterVector(1:length2,25);
out3      = CellsVector(1:length2,7);

```

```
fbm = CellsVector(1:length2,6);
ash = 3.2*out3 ./ (3.2*out3 + 1.1*3.0/7.0);

%save('treatment_17REV_vb_0.15_zeta=0.02_kappa=0.001.mat','out1','out2','ash','fbm')

%Nuevo

timep = out1'*0.25; % El vector de pasos de integración se
convierte a tiempo, multiplicando por Delta_t
dens60 = out2; % Se le da otro nombre a la densidad
aparente

% Se definen los instantes de tiempo donde se van a guardar los resultados.
% No todos se compararán con los de Miller
int = 30*4;
t1 = 1*int; % Resultados tras 1 mes de tratamiento
t2 = 3*int; % Resultados tras 3 meses de tratamiento
t3 = 6*int;
t4 = 12*int;
t5 = 18*int;
t6 = 24*int;
t7 = 36*int;
t8 = 48*int;

years = 365 * 4; %Step en el que comienza el tratamiento (después de 1
año de PMO)
total = length(timep);
timep_2 = timep(years:total,1)-timep(years,1);
BDG = (dens60(years:total,1)-dens60(years)) / dens60(years) *100;
% BDG en %
ash_2 = ash(years:total,1);

timep_vector = [0 timep_2(t1,1) timep_2(t2,1) timep_2(t3,1) timep_2(t4,1)
timep_2(t5,1)...
timep_2(t6,1) timep_2(t7,1) timep_2(t8,1)] / 30; % Expresado en meses
BDG_vector = [0 BDG(t1,1) BDG(t2,1) BDG(t3,1) BDG(t4,1) BDG(t5,1)
BDG(t6,1) BDG(t7,1) BDG(t8,1)];

%
```

```
load ('OutModel_A5_300.mat', 'OutModel')

Resultado_LS          = OutModel;

Resultado_LS(sim,:) = BDG_vector;

OutModel              = Resultado_LS;

save ('OutModel_A5_300.mat', 'OutModel')

end
```

8.3 Llamada al programa principal

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Llamada al proceso de automatización de las simulaciones %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear, clc

% m = ; %Número de filas de la matriz de salida del modelo
% n = ; %Número de columnas de la matriz de salida del modelo

OutModel = zeros(m,n);

save ('OutModel_A5_300.mat', 'OutModel')
load ('InModel_A5_300.mat', 'InModel')

Tamano = size(InModel);

% Cont = 0;
% Filas_no_nulas = 0;
%
% for i = 1:Tamano(1)
```

```
%  
%     if InModel(i,:) == 0  
%  
%         Cont = Cont+1;  
%         Filas_no_nulas = Tamano(1)-Cont;  
%  
%     end  
%  
% end  
  
tic  
  
Simu = 1;  
save ('Simulación.mat', 'Simu')  
  
iterin = 1;  
iterfin = Tamano(1);  
  
for iter = iterin:iterfin  
  
    run_bonemodel_v4;  
  
    load ('Simulación.mat', 'Simu')  
  
    Simu = Simu+1;  
  
    save ('Simulación.mat', 'Simu')  
  
end  
  
toc
```

8.4 Creación de las matrices de entrada y objetivo de la red neuronal

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Matrices I/T de la Red Neuronal caso Aleatorio 5D %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear, clc

%% Matriz de Entrada a la Red Neuronal

load ('InModel_A5_300.mat', 'InModel')

Param = 7;

Tamano_In = size(InModel);

% Cont = 0;
% Filas_no_nulas = 0;
%
% for i=1:Tamano_InLS(1)
%
%     if InMatrixLS(i,:) == 0
%
%         Cont = Cont+1;
%         Filas_no_nulas = Tamano_InLS(1)-Cont;
%
%     end
%
% end

% InNN_aux      = zeros(Filas_no_nulas*2,Tamano_InLS(2));
% EntradaRN_aux = zeros(Filas_no_nulas*2,Param);

InNN_aux      = zeros(Tamano_In(1), Tamano_In(2));
EntradaRN_aux = zeros(Tamano_In(1), Param);

for i = 1:Tamano_In(1)
```

```
InNN_aux(i,:) = InModel(i,:);

end

% Creamos la matriz simplificada de entrada a la RN
EntradaRN_aux(:,1) = InNN_aux(:,13); % RANK_OCp
EntradaRN_aux(:,2) = InNN_aux(:,23); % B_RANKL
EntradaRN_aux(:,3) = InNN_aux(:,39); % Kappa
EntradaRN_aux(:,4) = InNN_aux(:,51); % (1/3)*Zeta
EntradaRN_aux(:,5) = InNN_aux(:,53); % fbm
EntradaRN_aux(:,6) = InNN_aux(:,54); % Tratamiento
EntradaRN_aux(:,7) = InNN_aux(:,55); % P_RANKL_PMO

% Comprobamos que no hay puntos repetidos
A = [EntradaRN_aux(:,1) EntradaRN_aux(:,2) EntradaRN_aux(:,3) EntradaRN_aux(:,4)
EntradaRN_aux(:,7)];
[C,ia,ic] = unique(A(:,:),'rows','stable');

% En "C" están los valores únicos de A
% "ia" indica la posición de las filas únicas en A
% "ic" indica la posición de las filas de A en C

% No hay :)

% Pasamos de datos por fila a datos por columna
% InNN=InNN_aux';
EntradaRN = EntradaRN_aux';

%%

%% Matriz de Salida a la Red Neuronal

load ('OutModel_A5_300.mat', 'OutModel')

Tamano_Out = size(OutModel);
```

```
OutNN_aux      = zeros(Tamano_Out(1), Tamano_Out(2));
SalidaRN_aux = zeros(Tamano_Out(1), 5); %Consideramos 5 instantes de tiempo

for i = 1:Tamano_Out(1)

    OutNN_aux(i,:) = OutModel(i,:);

end

% Creamos la matriz simplificada de salida de la RN
SalidaRN_aux(:,1) = OutNN_aux(:,3);
SalidaRN_aux(:,2) = OutNN_aux(:,4);
SalidaRN_aux(:,3) = OutNN_aux(:,5);
SalidaRN_aux(:,4) = OutNN_aux(:,6);
SalidaRN_aux(:,5) = OutNN_aux(:,7);

% Pasamos de datos por fila a datos por columna
% OutNN=OutNN_aux';
SalidaRN = SalidaRN_aux';

InNNA300      = EntradaRN;
TargetNNA300 = SalidaRN;

save ('Matrices_NN_A5_300.mat', 'InNNA300', 'TargetNNA300')
```

8.5 Creación de submatrices

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Matrices de entrada caso Aleatorio 2D %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

close all
clear, clc

% Puntos del caso aleatorio
load ('Matrices_NN_A5_300.mat', 'InNNA300')

TamanoA300 = size (InNNA300);

Kappa = InNNA300(1,:);
Zeta = InNNA300(2,:);

% Seleccionamos los casos aleatorios
% Filas = ;
% N = ;

Columnas = 1:TamanoA300(2);

Conjunto = randsample(Columnas, N);
ConjuntoO = sort(Conjunto);

InNNA = zeros (Filas, N);

for i = 1:N

    InNNA(:,i) = InNNA300(:, ConjuntoO(i));

end

KappaA = InNNA(1,:);
ZetaA = InNNA(2,:);
```

```
Kappain = 0.001;
Kappafin = 0.007;

Zetaain = 0.001*1/3;
Zetafin = 0.03*1/3;

% Dibujamos los puntos del caso aleatorio
figure (1)
plot (KappaA, ZetaA, 'r*')
axis ([Kappain Kappafin Zetaain Zetafin])
grid
xlabel ('Valores de Kappa')
ylabel ('Valores de Zeta, en [(pM RANKL)^-1]')
title ('Puntos del caso aleatorio')

save ('Matrices_NN_A5_N.mat', 'InNNA')
```

8.6 Búsqueda de puntos de entrenamiento

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Búsqueda de puntos para el Aleatorio 2D %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear, clc

load ('Matrices.mat', 'InNNA')
load ('Matrices_NN_A5_300.mat', 'InNNA300', 'TargetNNA300')

TamanoA = size (InNNA);
TamanoA300 = size (InNNA300);

% Filas = ;
```

```
TargetNNA = zeros (Filas, TamanoA(2));

limit = 0.0000000001;

for i = 1:TamanoA(2)

    for j = 1:TamanoA300(2)

        Dif = InNNA(:,i) - InNNA300(:,j);

        if all (abs(Dif) < limit)

            TargetNNA(:,i) = TargetNNA300(:,j);

        end

    end

end

end

save ('Matrices_NN_A5_N.mat', 'InNNA', 'TargetNNA')
```

8.7 Validación cruzada

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Validación Cruzada %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear, clc

% Cargamos las matrices necesarias
load ('Matrices_NN_A5_N.mat', 'InNNA', 'TargetNNA')

x = InNNA;
t = TargetNNA;

% % Ante todo, adimensionalizamos las matrices de entrada y salida a la Red
Neuronal
% [xa,ta] = Adimensional(x,t);

% % Posteriormente, desordenamos las muestras (columnas) de las matrices de
entrada y salida a la Red Neuronal
% [xd,td] = Desorden(x,t);

xd = x;
td = t;

% save ('Matrices_X.mat', 'xd', 'td')
% load ('Matrices_X.mat', 'xd', 'td')

TamanoIn = size(xd);
TamanoT = size(td);

% Choose a Training Function
% For a list of all training functions type: help nntrain
% 'trainlm' is usually fastest.
% 'trainbr' takes longer but may be better for challenging problems.
% 'trainscg' uses less memory. Suitable in low memory situations.
trainFcn = 'trainlm'; % Levenberg-Marquardt backpropagation.
```

```
% Create a Fitting Network
% hiddenLayerSize = 10;
% net = fitnet(hiddenLayerSize,trainFcn);
net = fitnet([10],trainFcn);

% Setup Division of Data for Training, Validation, Testing
net.divideParam.trainRatio = 100/100; %Así conseguimos que entrene con el 100%
del subconjunto que le decimos
net.divideParam.valRatio    = 0/100;
net.divideParam.testRatio   = 0/100;

[trainInd,valInd,testInd] =
dividerand(x,net.divideParam.trainRatio,net.divideParam.valRatio,net.divideParam
.testRatio) Para comprobar

% Definimos los parámetros necesarios
NC      = TamanoIn(2); %Número de columnas
KFold   = 10;          %Dividimos el total de columnas entre KFold
NCFold  = NC/KFold;    %Número de columnas por cada subconjunto

Columntest = zeros(1,NCFold);

redes = cell(1,KFold);

CV = zeros(1,KFold);

%%%Cross Validation

i0    = 1;
ifin  = NCFold;

Paso = 0;

tic

for iter = 1:KFold %Repetimos el proceso KFold veces
```

```
testsetIn = zeros(TamanoIn(1),NCFold*1);
testsetT  = zeros(TamanoT(1), NCFold*1);

for i = i0:ifin

    testsetIn(:,i) = xd(:,i+Paso);
    testsetT(:,i)  = td(:,i+Paso); %"testsetT" son los Targets de la RN

    Columntest(i) = i+Paso; %Guardamos las posiciones de las columnas que se
toma para el conjunto de prueba

end

%%%%

LB = Columntest(i0); %Valor del límite inferior del vector
UB = Columntest(ifin); %Valor del límite superior del vector

trainsetIn = zeros(TamanoIn(1),NCFold*(KFold-1));
trainsetT  = zeros(TamanoT(1), NCFold*(KFold-1));

k = 1;

for j = 1:NC

    if j>=1 && (j<LB || j>UB)

        trainsetIn(:,k) = xd(:,j);
        trainsetT(:,k)  = td(:,j);

        k = k+1;

    end

end

end
```

```
%%%%

% Train the Network
[net,tr] = train(net,trainsetIn,trainsetT); %Devuelve la red "net" entrenada

redes{iter} = net; %Vamos guardando las redes neuronales

% Test the Network
y = net(testsetIn); %La red "net" entrenada la prueba. "y" es la salida de
la RN entrenada

%%%%

% Cálculo del coeficiente de variación "CV"

CV(iter) = CoefVar(testsetT,y);

%%%%

Paso = Paso+NCFold;

end

toc

% Nos quedamos con la red de menor coeficiente de variación

[CVmin,Pos] = min(CV); %Obtenemos el CV mínimo y su posición

RedNeuronal = redes{Pos}; %Nos quedamos con la red de menor CV

save ('RN_N_1x10_lm.mat', 'RedNeuronal', 'CVmin')

% View the Network
%view(net)
```

```
% Plot
% Uncomment these lines to enable various plots.
%figure, plotperform(tr)
%figure, plottrainstate(tr)
%figure, ploterrhist(e)
%figure, plotregression(t,y)
%figure, plotfit(net,x,t)
```

8.8 Coeficiente de variación

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Coeficiente de variación para 4D %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

close all
clear, clc

X = [20 30 40 50 100 160 200 260 300];
X2 = [160 200 260 300];

load ('CV_300.mat', 'CV300_1x10_lm', 'CV300_2x10_lm', 'CV300_3x10_lm',
'CV300_1x20_lm', 'CV300_2x20_lm', 'CV300_3x20_lm', 'CV300_1x10_br')

load ('CV_1x10.mat', 'CV20_1x10_lm', 'CV30_1x10_lm', 'CV40_1x10_lm',
'CV50_1x10_lm', 'CV100_1x10_lm', 'CV160_1x10_lm', 'CV200_1x10_lm',
'CV260_1x10_lm')

CV_1x10 = [CV20_1x10_lm CV30_1x10_lm CV40_1x10_lm CV50_1x10_lm CV100_1x10_lm
CV160_1x10_lm CV200_1x10_lm CV260_1x10_lm CV300_1x10_lm];
CV_1x10_2 = [CV160_1x10_lm CV200_1x10_lm CV260_1x10_lm CV300_1x10_lm];

load ('CV_2x10.mat', 'CV20_2x10_lm', 'CV30_2x10_lm', 'CV40_2x10_lm',
'CV50_2x10_lm', 'CV100_2x10_lm', 'CV160_2x10_lm', 'CV200_2x10_lm',
'CV260_2x10_lm')

CV_2x10 = [CV20_2x10_lm CV30_2x10_lm CV40_2x10_lm CV50_2x10_lm CV100_2x10_lm
CV160_2x10_lm CV200_2x10_lm CV260_2x10_lm CV300_2x10_lm];
```

```
CV_2x10_2 = [CV160_2x10_lm CV200_2x10_lm CV260_2x10_lm CV300_2x10_lm];

load ('CV_3x10.mat', 'CV20_3x10_lm', 'CV30_3x10_lm', 'CV40_3x10_lm',
'CV50_3x10_lm', 'CV100_3x10_lm', 'CV160_3x10_lm', 'CV200_3x10_lm',
'CV260_3x10_lm')
CV_3x10 = [CV20_3x10_lm CV30_3x10_lm CV40_3x10_lm CV50_3x10_lm CV100_3x10_lm
CV160_3x10_lm CV200_3x10_lm CV260_3x10_lm CV300_3x10_lm];
CV_3x10_2 = [CV160_3x10_lm CV200_3x10_lm CV260_3x10_lm CV300_3x10_lm];

load ('CV_1x20.mat', 'CV20_1x20_lm', 'CV30_1x20_lm', 'CV40_1x20_lm',
'CV50_1x20_lm', 'CV100_1x20_lm', 'CV160_1x20_lm', 'CV200_1x20_lm',
'CV260_1x20_lm')
CV_1x20 = [CV20_1x20_lm CV30_1x20_lm CV40_1x20_lm CV50_1x20_lm CV100_1x20_lm
CV160_1x20_lm CV200_1x20_lm CV260_1x20_lm CV300_1x20_lm];
CV_1x20_2 = [CV160_1x20_lm CV200_1x20_lm CV260_1x20_lm CV300_1x20_lm];

load ('CV_2x20.mat', 'CV20_2x20_lm', 'CV30_2x20_lm', 'CV40_2x20_lm',
'CV50_2x20_lm', 'CV100_2x20_lm', 'CV160_2x20_lm', 'CV200_2x20_lm',
'CV260_2x20_lm')
CV_2x20 = [CV20_2x20_lm CV30_2x20_lm CV40_2x20_lm CV50_2x20_lm CV100_2x20_lm
CV160_2x20_lm CV200_2x20_lm CV260_2x20_lm CV300_2x20_lm];
CV_2x20_2 = [CV160_2x20_lm CV200_2x20_lm CV260_2x20_lm CV300_2x20_lm];

load ('CV_3x20.mat', 'CV20_3x20_lm', 'CV30_3x20_lm', 'CV40_3x20_lm',
'CV50_3x20_lm', 'CV100_3x20_lm', 'CV160_3x20_lm', 'CV200_3x20_lm',
'CV260_3x20_lm')
CV_3x20 = [CV20_3x20_lm CV30_3x20_lm CV40_3x20_lm CV50_3x20_lm CV100_3x20_lm
CV160_3x20_lm CV200_3x20_lm CV260_3x20_lm CV300_3x20_lm];
CV_3x20_2 = [CV160_3x20_lm CV200_3x20_lm CV260_3x20_lm CV300_3x20_lm];

load ('CV_1x10_br.mat', 'CV20_1x10_br', 'CV30_1x10_br', 'CV40_1x10_br',
'CV50_1x10_br', 'CV100_1x10_br', 'CV160_1x10_br', 'CV200_1x10_br',
'CV260_1x10_br')
CV_1x10_br = [CV20_1x10_br CV30_1x10_br CV40_1x10_br CV50_1x10_br CV100_1x10_br
CV160_1x10_br CV200_1x10_br CV260_1x10_br CV300_1x10_br];
CV_1x10_br_2 = [CV160_1x10_br CV200_1x10_br CV260_1x10_br CV300_1x10_br];

load ('CV_1x10_br_max.mat', 'CV20_1x10_br_max', 'CV30_1x10_br_max',
'CV40_1x10_br_max', 'CV50_1x10_br_max', 'CV100_1x10_br_max',
'CV160_1x10_br_max', 'CV200_1x10_br_max', 'CV260_1x10_br_max')
CV_1x10_br_max = [CV20_1x10_br_max CV30_1x10_br_max CV40_1x10_br_max
```

```
CV50_1x10_br_max CV100_1x10_br_max CV160_1x10_br_max CV200_1x10_br_max  
CV260_1x10_br_max];
```

```
X20 = [20 20];  
X30 = [30 30];  
X40 = [40 40];  
X50 = [50 50];  
X100 = [100 100];  
X160 = [160 160];  
X200 = [200 200];  
X260 = [260 260];  
X300 = [300 300];
```

```
CV20_1x10_br_I = [CV20_1x10_br CV20_1x10_br_max];  
CV30_1x10_br_I = [CV30_1x10_br CV30_1x10_br_max];  
CV40_1x10_br_I = [CV40_1x10_br CV40_1x10_br_max];  
CV50_1x10_br_I = [CV50_1x10_br CV50_1x10_br_max];  
CV100_1x10_br_I = [CV100_1x10_br CV100_1x10_br_max];  
CV160_1x10_br_I = [CV160_1x10_br CV160_1x10_br_max];  
CV200_1x10_br_I = [CV200_1x10_br CV200_1x10_br_max];  
CV260_1x10_br_I = [CV260_1x10_br CV260_1x10_br_max];  
CV300_1x10_br_I = [CV300_1x10_br CV300_1x10_br];
```

```
figure (1)  
plot (X, CV_1x10, 'r*- ', X, CV_2x10, 'b*- ', X, CV_3x10, 'g*- ')  
grid  
legend ('1x10xlm', '2x10xlm', '3x10xlm')  
set (legend, 'FontSize', 16)  
xlabel ('Número de casos', 'FontSize', 16)  
ylabel ('Coeficiente de variación', 'FontSize', 16)  
title ('Aumento del número de capas', 'FontSize', 16)
```

```
figure (2)  
plot (X2, CV_1x10_2, 'r*- ', X2, CV_2x10_2, 'b*- ', X2, CV_3x10_2, 'g*- ')  
grid
```

```
legend ('1x10xlm', '2x10xlm', '3x10xlm')
set      (legend, 'FontSize', 16)
xlabel   ('Número de casos', 'FontSize', 16)
ylabel   ('Coeficiente de variación', 'FontSize', 16)
title    ('Aumento del número de capas', 'FontSize', 16)

figure (3)
plot     (X, CV_1x20, 'r*- ', X, CV_2x20, 'b*- ', X, CV_3x20, 'g*- ')
grid
legend   ('1x20xlm', '2x20xlm', '3x20xlm')
set      (legend, 'FontSize', 16)
xlabel   ('Número de casos', 'FontSize', 16)
ylabel   ('Coeficiente de variación', 'FontSize', 16)
title    ('Aumento del número de neuronas', 'FontSize', 16)

figure (4)
plot     (X2, CV_1x20_2, 'r*- ', X2, CV_2x20_2, 'b*- ', X2, CV_3x20_2, 'g*- ')
grid
legend   ('1x20xlm', '2x20xlm', '3x20xlm')
set      (legend, 'FontSize', 16)
xlabel   ('Número de casos', 'FontSize', 16)
ylabel   ('Coeficiente de variación', 'FontSize', 16)
title    ('Aumento del número de neuronas', 'FontSize', 16)

figure (5)
plot     (X, CV_1x10, 'r*- ', X, CV_1x10_br, 'b*- ')
grid
legend   ('1x10xlm', '1x10xbr')
set      (legend, 'FontSize', 16)
xlabel   ('Número de casos', 'FontSize', 16)
ylabel   ('Coeficiente de variación', 'FontSize', 16)
title    ('Funciones de entrenamiento', 'FontSize', 16)

figure (6)
plot     (X2, CV_1x10_2, 'r*- ', X2, CV_1x10_br_2, 'b*- ')
grid
legend   ('1x10xlm', '1x10xbr')
```

```
set      (legend, 'FontSize', 16)
xlabel  ('Número de casos', 'FontSize', 16)
ylabel  ('Coeficiente de variación', 'FontSize', 16)
title   ('Funciones de entrenamiento', 'FontSize', 16)

figure (7)

plot     (X20, CV20_1x10_br_I, 'r*-', X30, CV30_1x10_br_I, 'r*-', X40,
CV40_1x10_br_I, 'r*-', X50, CV50_1x10_br_I, 'r*-', X100, CV100_1x10_br_I, 'r*-',
X160, CV160_1x10_br_I, 'r*-', X200, CV200_1x10_br_I, 'r*-', X260,
CV260_1x10_br_I, 'r*-', X300, CV300_1x10_br_I, 'r*-')
grid
xlabel  ('Número de casos', 'FontSize', 16)
ylabel  ('Valores mínimos y máximos', 'FontSize', 16)
title   ('Valores mínimos y máximos del CV', 'FontSize', 16)
```

