

UNIVERSIDAD DE ALMERÍA



PROYECTO FIN DE CARRERA

Simulación de dinámicas eco-evolutivas: un modelo basado en individuos

Alumno:

Gabriel Barrionuevo Rosales

Directores:

Leocadio González Casado

Jordi Moya Laraño

*Este proyecto completa los requisitos necesarios
para la obtención del título de Ingeniero en Informática en la*

Escuela Politécnica Superior y Facultad de Ciencias Experimentales

1 de septiembre de 2014

Certificación en la que conste que el alumno está matriculado del proyecto."

Gabriel Barrionuevo Rosales

UNIVERSIDAD DE ALMERÍA

Resumen

Escuela Politécnica Superior y Facultad de Ciencias Experimentales

Ingeniero en Informática

Simulación de dinámicas eco-evolutivas: un modelo basado en individuos

por Gabriel Barrionuevo Rosales

Haciendo uso de las investigaciones realizadas y los resultados obtenidos en biología, se va a definir e implementar un modelo basado en individuos que represente un ecosistema real lo más fielmente posible, donde convivan diferentes especies de animales utilizando los recursos basales disponibles. La implementación servirá para observar patrones en la eco-evolución animal de modo que se puedan contrastar en las investigaciones de campo. En este tipo de experimentos se llevan a cabo una serie de observaciones en las cuales el seguimiento de cada individuo y la toma de datos es esencial para evaluar los efectos que se producen en la población.

Agradecimientos

Quiero expresar mi agradecimiento al profesor Leocadio González Casado por su participación y apuesta en el desarrollo de este proyecto, las cuales han enriquecido enormemente el trabajo realizado.

Agradecer también a Jordi Moya Laraño su función como guía y por todo lo aprendido, y sobre todo por transmitirme ese entusiasmo y pasión que pone a la hora de llevar a cabo todas las metas que se propone.

Agradezco también a José Román Bilbao Castro, por su constante apoyo y ánimos, que han hecho posible que esto saliera adelante incluso en los momentos más complicados.

Y por supuesto a mi familia, muchas gracias por apoyarme y confiar en mí durante todo este tiempo, no solamente con este proyecto sino durante el desarrollo de toda mi carrera.

Índice general

Resumen	II
Agradecimientos	III
Lista de Figuras	VII
Lista de Tablas	VIII
1. Introducción	1
1.1. Conceptos previos	1
1.1.1. Evolución	1
1.1.2. Ecología	1
1.1.3. Ecología evolutiva	2
1.1.4. Modelo basado en individuos	2
1.2. Planteamiento del problema	2
2. Plan de trabajo	4
2.1. Metodología de seguimiento	4
2.2. Planificación temporal	5
3. Análisis	6
3.1. Aspectos simulados en la experimentación	6
3.1.1. Genética y características del individuo	6
3.1.1.1. Rasgos	7
3.1.1.2. Variables de estado	8
3.1.2. Recursos basales	8
3.1.3. Movimiento e interacciones	9
3.1.4. Metabolismo	10
3.1.5. Reproducción	11
3.2. Variabilidad de parámetros	12

4. Diseño	13
4.1. Espacio y recursos basales	13
4.2. Rasgos genéticos y su base cuantitativa	14
4.3. Modularidad de los rasgos	19
4.4. Genética cuantitativa y reproducción	22
4.5. Movimiento adaptativo y experiencia previa	23
5. Implementación	25
5.1. Descripción general del proceso	25
5.2. Clases y métodos implementados	27
5.2.1. Animal	29
5.2.2. AnimalSpecies	32
5.2.3. Chromosome	35
5.2.4. ChromosomesGenerator	36
5.2.5. Fungus	37
5.2.6. FungusSpecies	38
5.2.7. Locus	39
5.2.8. Species	40
5.2.9. TerrainCell	40
5.2.10. World	44
6. Pruebas y resultados	47
6.1. Primeras pruebas de estabilidad	47
6.2. Ejemplo sencillo de simulación	48
6.3. Resumen de resultados eco-evolutivos obtenidos	54
6.3.1. Ejemplos de hipótesis	54
6.3.2. Conectancia y persistencia de la red trófica	55
6.3.3. Variación genética y persistencia de la red trófica	59
6.3.4. Distancia entre islas y persistencia de la red trófica	61
6.3.5. Control <i>top-down</i> de los depredadores sobre la diversidad de presas	63
6.3.6. Dinámicas multi-tróficas espacio-temporales durante una simulación de 500 días	65
6.3.7. Micro-evolución en 500 días de una red persistente	69
6.3.8. Dinámicas evolutivas de gusanos en presencia y ausencia de depredadores	72
7. Conclusiones y trabajo futuro	74
7.1. Conclusiones	74
7.2. Optimización para la estabilidad	74
7.3. Mejora del suelo	75
7.4. Paralelización y modelo de islas	75
7.5. Optimización de las salidas de datos	75

Apéndices	76
A. Manual de usuario de Weaver	77
A.1. Pasos para realizar una simulación	77
A.2. Estructura del archivo de configuración	79
A.3. Lista de archivos de salida generados	84
B. Ejemplo completo de archivo de configuración <i>run_params</i>	88
Bibliografía	122

Índice de figuras

2.1. Diagrama temporal de planificación de tareas.	5
3.1. Rasgos característicos y correlación entre ellos.	8
3.2. Organización del espacio y recursos basales.	9
5.1. Diagrama de clases principal	28
6.1. Ejemplo de sumario extendido que resume los datos sobre población durante la simulación	50
6.2. Ejemplo de salida diaria con información de los animales	52
6.3. Ejemplo de matriz de depredación entre especies	53
6.4. Redes tróficas y estructuras espaciales de metacomunidad incluidas en las simulaciones	56
6.5. Dinámicas de población de cada especie usando distintas conectancias	58
6.6. Dinámicas de población de cada especie al utilizar distintas variaciones genéticas	60
6.7. Dinámicas de población de cada especie con distintas distancias entre islas	62
6.8. Dinámicas de población de cada especie dependiendo de la presencia o ausencia de depredadores	64
6.9. Dinámicas del escenario más estable para 500 días	66
6.10. Fuerza de interacciones en una red persistente tras 500 días de simulación	68
6.11. Evolución de 14 rasgos de las 5 especies persistentes tras 500 días	70
6.12. Evolución de los rasgos de un gusano fungívoro (enc2) en presencia y ausencia de depredadores	73
A.1. Localización de los archivos previa ejecución de una simulación	78
A.2. Estado de la consola mientras se realiza una simulación	79

Índice de tablas

4.1. Módulos de rasgos implicados en la genética de los animales.	21
6.1. Resultados de las pruebas de estabilidad. Tiempos de ejecución según los parámetros indicados.	48

1 | Introducción

1.1. Conceptos previos

1.1.1. Evolución

La evolución pasada determina en gran medida los rasgos genéticos que caracterizan a los seres vivos, como son por ejemplo el tamaño del individuo, su fuerza, su velocidad o la eficacia de su metabolismo. Cada uno de estos rasgos emergen a partir de la información incluida en la genética del sujeto, que ha sido heredada generación tras generación. Al realizarse dicha transferencia, los genes terminan sufriendo ciertas modificaciones debidas al conocido proceso evolutivo, sea este adaptativo o azaroso [1]. En este proyecto se implementan una serie de procesos que simulan los patrones básicos de la genética y que son el punto clave para incorporar el mecanismo evolutivo dentro de las simulaciones que aquí se plantean.

1.1.2. Ecología

La ecología, por su parte, estudia otro tipo de propiedades que guardan más relación con el ecosistema que con el propio individuo. Esta ciencia se centra en la definición y descripción del ambiente, la observación y predicción de la cantidad y distribución de los animales en el espacio, y sobre todo en cómo afectan a estos elementos el comportamiento de los individuos y las interacciones que ocurren entre ellos [2]. En las simulaciones que se llevan a cabo en este trabajo se crea un ambiente bien definido con unas propiedades concretas.

Este ambiente puede variarse en cada simulación, de forma que se pueda analizar según convenga y de forma exhaustiva la influencia de los cambios en las poblaciones.

1.1.3. Ecología evolutiva

La ecología evolutiva abarca la estrecha relación entre ambas ciencias [1], y, en particular, el reciente y en auge estudio de las dinámicas eco-evolutivas que tiene como fin documentar cómo afectan las presiones de selección (factores ecológicos) a los cambios en las frecuencias génicas y cómo dichos cambios revierten recíprocamente en la dinámica ecológica de las poblaciones [3]. Sin embargo, esta tarea se vuelve muy compleja a la hora de llevar a cabo experimentos de campo, ya que para obtener algún resultado es necesario observar un número muy grande de individuos durante un intervalo de tiempo muy amplio, así como medir demasiados detalles que escapan a lo humanamente factible. Al simular estos experimentos se consigue obtener resultados de las hipótesis planteadas que luego deben corroborarse en el campo. La simulación permite un mayor nivel de detalle en los resultados obtenidos.

1.1.4. Modelo basado en individuos

La experimentación que se expone en este proyecto hace uso de un modelo basado en individuos (en adelante IBM: *Individual-Based Model*). Un IBM [4] es un modelo computacional que simula las acciones e interacciones de agentes autónomos, los cuales pueden estar englobados dentro de distintas especies, con el objetivo de evaluar los efectos generales que se producen en el sistema.

1.2. Planteamiento del problema

Se ha definido e implementado un IBM que representa un ecosistema real (el suelo de un bosque caducifolio) lo más fielmente posible. En este ecosistema se hace uso de las investigaciones de campo y los resultados obtenidos en biología [5] para crear un entorno en el que conviven diferentes especies de animales, los cuales hacen uso de los recursos

basales disponibles (hongos). Esta implementación se utiliza para observar patrones en la eco-evolución animal de modo que se puedan contrastar con las investigaciones de campo.

El programa desarrollado se llama Weaver y es un *framework* con el objetivo de salvar la frontera entre: las observaciones empíricas de redes tróficas basadas en individuos en espacios heterogéneos; y las predicciones teóricas provenientes de las dinámicas eco-evolutivas de metacomunidades multi-tróficas. Weaver es una extensión mejorada de la plataforma anterior (mini-Akira[5]) y permite la exploración de dinámicas eco-evolutivas basadas en individuos en redes tróficas multi-especie a lo largo del espacio. Este *framework* enlaza genes con ecosistemas a través del espacio, alcanzando un nivel de comprensión sin precedentes que proporciona una visión dentro de las dinámicas ecológicas y evolutivas en los niveles de genética, individuo, población y comunidad, en diferentes escalas del espacio. Todas estas dinámicas además pueden ser enlazadas a procesos que ocurren en los ecosistemas, como por ejemplo el control *top-down* que realizan los depredadores, induciendo cascadas tróficas al afectar a los recursos basales a diferentes temperaturas[5]. Otro ejemplo interesante es el papel que cumplen los depredadores en el mantenimiento de la biodiversidad, que puede ser analizado mediante el uso de diferentes redes tróficas en el mismo espacio, uso de islas o diferentes configuraciones de la genética. Además, muy lejos de ser una caja negra, el programa genera salidas muy detalladas que pueden ser utilizadas para entender los diferentes mecanismos que rigen las dinámicas eco-evolutivas. El investigador puede observar toda la información de cada gen o individuo durante la simulación. Hasta donde alcanza nuestro conocimiento, ningún otro *framework* es actualmente capaz de proveer tal grado de detalle.

En la sección 3 se analizan y se detallan cada uno de los aspectos simulados en la experimentación. En la sección 4 se describe la versión secuencial del algoritmo, explicando los diferentes pasos que lo componen. En la sección 5 se explica cómo se ha llevado a cabo la implementación. En la sección 6 se muestran las pruebas realizadas y los resultados obtenidos en la ejecución de las simulaciones. Finalmente, en la sección 7 se plantean una serie de propuestas para la continuación y mejora del proyecto.

2 | Plan de trabajo

2.1. Metodología de seguimiento

El desarrollo del problema que se plantea en este proyecto se ha realizado mediante una serie de reuniones con los expertos en la materia. Estas reuniones se han planificado de forma que proporcionen el máximo provecho a la hora de discutir los diferentes aspectos de la aplicación. Generalmente se han llevado a cabo en intervalos de dos semanas, y han servido para cumplir los siguientes objetivos:

1. **Revisar** (y corregir, en su caso) el trabajo realizado hasta la fecha, comprobando que los algoritmos implementados proporcionan las respuestas deseadas. En este proceso es clave la intervención del director experto en biología.
2. **Compartir** ideas acerca de las posibles mejoras aplicables tanto al trabajo ya realizado como al pendiente. Esto permite volver a pensar en los mecanismos implementados en los procedimientos, teniendo en cuenta el estado actual de desarrollo, lo que puede incrementar la fiabilidad y semejanza del ecosistema generado con el ecosistema real observado, así como obtener un mayor rendimiento computacional al revisar la eficiencia del código.
3. **Planificar** los siguientes pasos inmediatos. Se trabaja con una metodología de cumplimiento de objetivos, o hitos, marcando aproximadamente el punto que hay que alcanzar hasta la siguiente reunión.

2.2. Planificación temporal

El proyecto se ha dividido en cinco etapas diferenciadas que no presentan necesariamente una continuidad en el tiempo. Esto significa que, aplicando la metodología de seguimiento que se ha descrito en la sección anterior, las etapas se van realizando de forma ordenada pero realizando varios ciclos de forma consecutiva. En cada reunión comienza una fase de análisis y se avanza hacia las demás fases hasta la documentación, volviendo a empezar otra fase de análisis en la siguiente reunión, y así sucesivamente. La duración total de las etapas es de 90 horas. En el diagrama de Gantt de la figura 2.1 se muestra el porcentaje de tiempo estimado para cada etapa. A continuación se enumeran y describen las fases que se desarrollan dentro de cada ciclo.

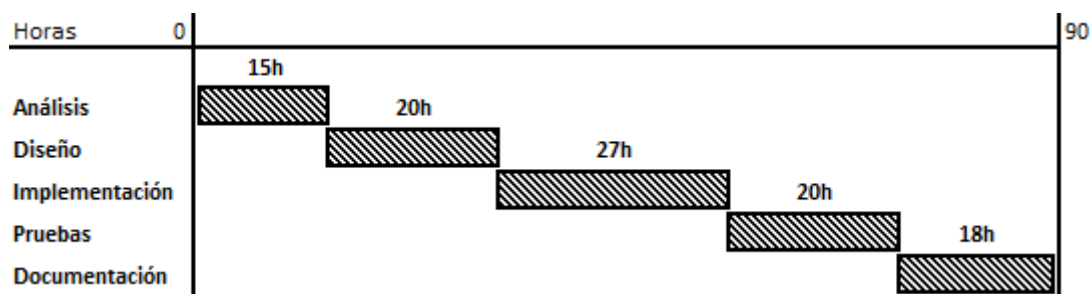


FIGURA 2.1: Diagrama temporal de planificación de tareas.

1. *Análisis*. En la primera etapa se analizan y estudian los algoritmos para el desarrollo del IBM: crecimiento de los recursos basales, metabolismo de los individuos, interacción entre ellos, genética, reproducción, etc (15 horas).
2. *Diseño*. Se estudian con detenimiento los procedimientos necesarios para conseguir el desarrollo del IBM. Se eligen las estructuras de datos más adecuadas para la implementación del mismo y se modela el sistema (20 horas).
3. *Implementación*. Se escribe el código siguiendo las decisiones tomadas (27 horas).
4. *Pruebas*. Se revisan los resultados para comprobar que se cumplen los requisitos descritos en el análisis y diseño del sistema (20 horas).
5. *Documentación*. En esta última etapa se escribirá una memoria describiendo el trabajo realizado y los resultados obtenidos (18 horas).

3 | Análisis

El análisis del problema planteado se ha basado principalmente en las reuniones descritas en la sección 2.1. En ellas se han tratado paso a paso y minuciosamente cada uno de los aspectos descritos a continuación.

3.1. Aspectos simulados en la experimentación

En la simulación existen una serie de conceptos que es necesario describir para entender la complejidad del problema. En esta sección se van a exponer y son los siguientes:

- Creación de la genética y rasgos de los seres.
- Desarrollo de los recursos basales.
- Movimiento e interacciones entre individuos.
- Gestión del metabolismo.
- Mecanismo de reproducción.

3.1.1. Genética y características del individuo

Dentro de este IBM, los agentes se dividen en dos tipos especies: los depredadores, por ejemplo los que pertenecen a la especie de los ácaros mesostigmata; y las presas, por ejemplo la especie de los colémbolos. Por el momento no existe en las simulaciones ninguna

especie del tipo estrictamente omnívora, es decir, que se alimente tanto de animales como de hongos.

Todo tipo de individuo presente en la simulación, ya sea depredador o presa, comparte los mismos rasgos característicos, aunque los valores de éstos son diferentes para cada individuo y los rangos numéricos varían dependiendo de la especie. Además cada tipo de especie se rige por un comportamiento distinto: las presas huirán de los depredadores buscando alimento (hongos), y los depredadores buscarán presas (huyendo de los depredadores superiores en la medida de lo posible, o incluso evitando a los de su misma especie si éstos son caníbales).

Independientemente de la especie a la que pertenezca, cada individuo está compuesto por trece rasgos que son definidos a partir de trece pares de cromosomas. Cada cromosoma está compuesto por veinte alelos ubicados en las veinte posiciones o loci disponibles, aunque número es susceptible de ser modificado. Estos alelos, sumados junto a los de su cromosoma par correspondiente, generan el valor inicial asociado al rasgo que se esté considerando. Sin embargo, los rasgos pueden estar correlacionados entre sí en mayor o menor medida, positiva o negativamente, con lo cual existe una dependencia directa entre ellos. Esto se explica con más detalle en la siguiente sección.

3.1.1.1. Rasgos

Cada individuo posee trece rasgos característicos que lo definen. La figura 3.1 representa estos rasgos y la correlación que existe entre ellos. Si un rasgo es dominante dentro de su módulo, significa que influye positiva o negativamente en los demás rasgos del módulo. Este efecto aumenta o disminuye dependiendo del coeficiente de correlación ρ seleccionado para ese módulo. Es importante destacar que este coeficiente de correlación es modificable para cada ejecución y cada módulo, con el objetivo de poder estudiar las diferentes respuestas del sistema según los diferentes casos.

Cada rasgo está cuantificado dentro de un rango determinado. Este rango puede variar de una especie a otra, pero siempre estará definido dentro de unos límites evolutivos que han sido calculados a través de la experimentación [5]. En la sección 4.2 se enumeran todos los rasgos y sus correspondientes rangos y límites, se describe qué función cumplen, cómo es la correlación que hay entre ellos y se indican los aspectos más relevantes de cada uno.

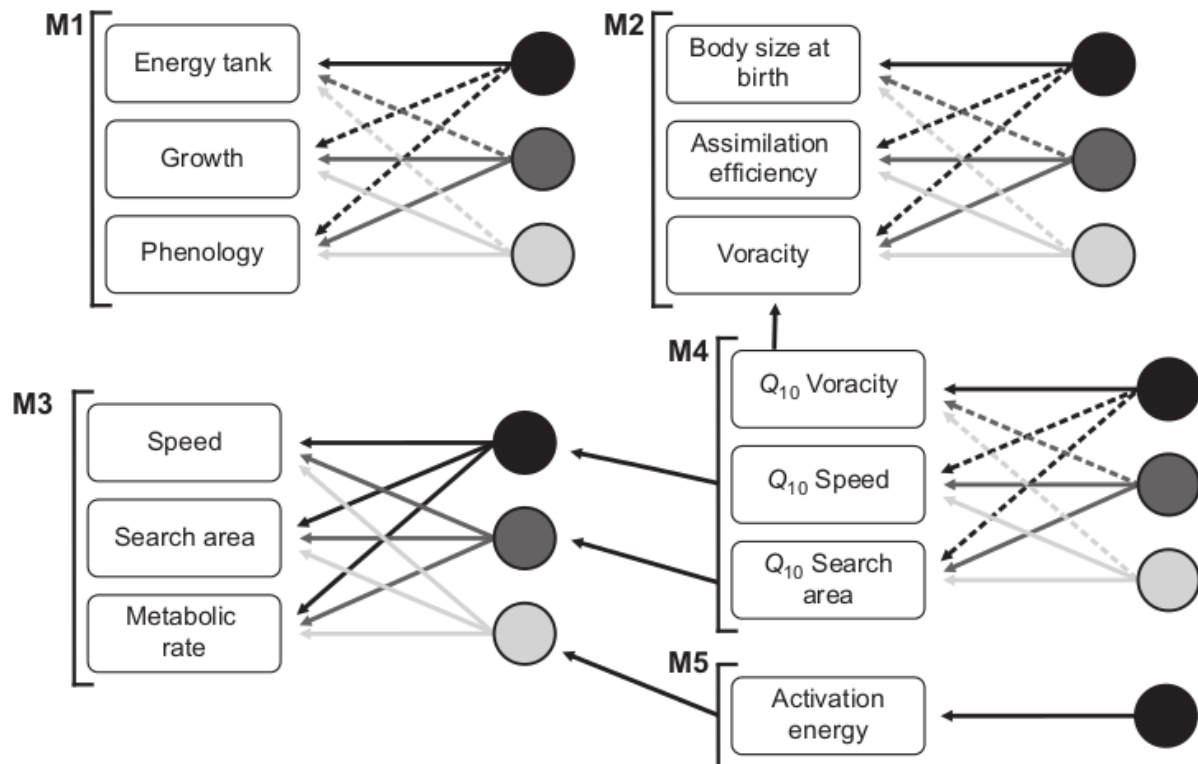


FIGURA 3.1: Rasgos característicos y correlación entre ellos.

3.1.1.2. Variables de estado

Además de sus rasgos característicos, cada ser vivo almacena una serie de variables de estado que deben ser actualizadas cada día a medida que se producen las interacciones con el entorno y con otros individuos.

Entre estas variables se encuentran: la masa del individuo, si se ha movido o no durante el día actual, la cantidad de pasos recorridos durante la etapa de movimiento, la cantidad de encuentros que ha tenido un individuo con los demás, etc.

3.1.2. Recursos basales

En el diseño inicial y por simplicidad, el espacio en el que se desarrolla este ecosistema tiene una sola dimensión. Dicho espacio está dividido en celdas contiguas, y, como se

puede observar en la figura 3.2, se distribuye de forma que la celda inicial se conecta con la final, formando un anillo.



FIGURA 3.2: Organización del espacio y recursos basales.

En la versión actual es posible seleccionar las dimensiones deseadas para el espacio (ver sección A.2 sobre la estructura del archivo de configuración). Además, en vez de usar una configuración en anillo, el espacio es cerrado y los animales consideran los límites del espacio como paredes sólidas. Esto puede ser aún mejorado, tal y como se explica en la sección 7.3 como trabajo futuro.

En cada una de las celdas existe una cantidad determinada de recursos basales. Esta propiedad es inherente a las celdas y su comportamiento simula el crecimiento de una variedad de hongos en el suelo. Al comienzo de la simulación se inicia la cantidad de hongo en cada celda, siguiendo un patrón establecido. Los hongos crecen cada día dependiendo de la capacidad total de la celda y de la cantidad actual de hongo [6].

Estos recursos basales son consumidos por un tipo de especie: las presas. Esto provoca que la cantidad de hongo varíe cada día, pudiendo ocurrir que llegue a extinguirse dentro de la celda e incluso en todo el espacio. Para minimizar la ocurrencia de este último suceso se ha establecido un umbral que, una vez superado, provoca la propagación de esporas hacia la celda colindante que contenga menor cantidad de recursos basales.

3.1.3. Movimiento e interacciones

El movimiento y las interacciones de unos seres vivos con otros es la parte más compleja del modelo. Para simplificar, se ha establecido que en cada celda pueden estar situados en el mismo momento cualquier número de individuos de cualquier especie. Estos individuos

se mueven por las celdas caminando un número de pasos que depende del rasgo *área-de-búsqueda*. Cada individuo solo tiene percepción de la celda actual y las dos colindantes. En caso de no encontrar alimento en estas celdas (ya sean hongos o presas), el individuo puede realizar un salto aleatorio cuya longitud máxima está definida por el valor de *área-de-búsqueda*. Este procedimiento es común para todos los seres vivos. Sin embargo, cada especie se mueve atendiendo a una serie de variables diferentes:

- **Presas** – Se mueven por el espacio buscando las celdas que contienen más hongos. En la presencia de depredadores la decisión no es tan básica, ya que también hay que tener en cuenta el riesgo de depredación.
- **Depredadores** – Caminan hacia las celdas que contienen un mayor número de presas. También consideran el número de depredadores existentes en una celda, ya que pueden cazar y ser cazados por otros individuos de su misma especie. Al igual que las presas, pueden quedar expuestos a los demás depredadores en cada movimiento.

Los encuentros entre depredador y presa (o bien depredador y depredador) tienen lugar cuando ambos se encuentran en la misma celda y cuando se active una variable aleatoria que depende del *tamaño* y la *voracidad* de ambos seres vivos: el cazador y el cazado. Si el encuentro se produce, entonces la víctima tendrá la oportunidad de escapar, dependiendo de su *voracidad*, *tamaño*, y *velocidad*. Durante el proceso se actualizan las variables de estado que almacenan el número de encuentros por día y los globales. De esta forma se lleva un registro minucioso de las interacciones entre todos los individuos.

Por último, según se ha documentado en animales caníbales [7], si la víctima es de la especie depredadora y escapa satisfactoriamente, puede llevar a cabo un contraataque si se siente segura, es decir, solamente contraataca cuando es atacada por un depredador externo que invade su espacio vital.

3.1.4. Metabolismo

Las dos especies de seres vivos presentes en la simulación deben metabolizar el alimento ingerido, convirtiendo éste en *tanque-de-energía* y, llegado a un determinado umbral,

este *tanque-de-energía* se utiliza para mudar y crecer en *tamaño*. Una vez alcanzado un determinado número de mudas y se tiene la suficiente energía, el individuo está listo para reproducirse, y buscará para ello algún individuo con el que aparearse ese mismo día.

Sin embargo el metabolismo va más allá, pues no sólo hay que considerar el crecimiento sino también el gasto del *tanque-de-energía*. Este gasto se debe a tres razones principalmente: el mantenimiento propio del individuo (metabolismo basal), la distancia que haya recorrido durante el día y el número de encuentros con depredadores que haya sufrido. Si el balance del gasto acumulado a lo largo de los días produce que el *tanque-de-energía* alcance un mínimo crítico, el animal morirá por falta de sustento.

3.1.5. Reproducción

En la simulación hay que tener en cuenta la reproducción como un suceso que hará escalar enormemente la cantidad de información y de procesamiento requeridos. Como ya hemos dicho antes, todos los individuos que alcanzan la madurez sexual pasan a estar preparados para reproducirse. Además, si consiguen reproducirse varias veces perecerán por causa de senescencia reproductiva (envejecimiento por el coste de la reproducción).

En el proceso de la reproducción se realiza la mezcla de genes mediante el proceso de la meiosis y la recombinación. De esta forma, la evolución de la genética en los individuos es estudiada a lo largo de la simulación con el fin de entender aspectos relevantes y novedosos desde el punto de vista eco-evolutivo.

Cuando un individuo está listo para reproducirse, simplemente busca aleatoriamente otro compañero que también lo esté. Dado que los individuos son hermafroditas recíprocos (tienen los dos sexos), el proceso se realiza dos veces intercambiando el orden de los actores, para conservar así las distintas combinaciones del material genético. Este proceso genera nueva descendencia que en algunas ocasiones puede superar la decena de nuevos individuos en cada evento reproductivo.

3.2. Variabilidad de parámetros

Cada simulación ejecuta todos los procesos detallados anteriormente, y para ello es necesario que un gran número de parámetros sean especificados al comienzo. Estos parámetros son tales como el tamaño del espacio, el número de alelos que componen el genoma de un individuo, los rangos de valores entre los que se mueven los rasgos de cada especie, entre otros.

Para fijar estos parámetros se utiliza un archivo de entrada escrito en formato JSON, que contiene el nombre del parámetro y el valor asociado al mismo, de forma que la modificación de los mismos sea intuitiva si se conocen los detalles de las simulaciones. Estos valores pueden ser modificados en cada simulación para realizar el conjunto de pruebas que sea necesario.

4 | Diseño

En las siguientes secciones se describen con mayor detalle cada uno de los procesos implicados en la simulación y que se describieron en la sección 3.1.

4.1. Espacio y recursos basales

El espacio en el que se mueven los animales en la simulación consta de una sola dimensión, tal y como se describe en la sección 3.1.2 de análisis. Por lo tanto cada individuo solamente podrá moverse en dos posibles direcciones. Recordemos que los individuos pueden moverse desde la última celda hacia la primera, y viceversa. La escala del espacio no es fija y viene determinada por el tamaño de los individuos de una simulación concreta. En la configuración usada, la distribución de los recursos basales está correlacionada con el espacio, aunque dicha distribución puede variarse en cualquier caso. El hongo se sitúa a lo largo de todo el espacio de forma que la mayor cantidad se localiza en la celda central, decreciendo linealmente hacia los "bordes" del espacio (la primera y la última celdas, aunque realmente se tiene una estructura cerrada en anillo).

La productividad de hongo está asociada a la celda y es la encargada de regular que éste crezca más o menos deprisa en diferentes zonas del espacio. El recurso basal crece cada día de acuerdo a una función de crecimiento que viene determinada por el siguiente algoritmo[5]:

$$M_t = M_{t-1} + r_T M_{t-1} \left(1 - \frac{M_{t-1}}{K}\right) \quad (4.1)$$

donde M_t y M_{t-1} representan la biomasa total de hongo en la celda en el instante t y $t - 1$, respectivamente; K indica la capacidad de carga de hongo de la celda, y r_T es el ratio de incremento, que depende directamente de la temperatura de la siguiente forma [6]:

$$r_t = \frac{e^{b-E(1/kT)}}{M_{t-1}^{1/4}} \quad (4.2)$$

donde $b = 25,98$ es una constante de normalización, y $E = -0,68$ es la energía de activación calculada sobre los organismos[6]; k es la constante de Boltzmann ($8,62 \times 10^{-5} eV/K$), y T es la temperatura ambiental en grados Kelvin. En este punto se asume que el total de biomasa en la celda (M_{t-1}) pertenece a un solo organismo o tipo de hongo, así que r_T se actualiza dinámicamente cada día dependiendo de M_{t-1} . Cuando una celda está casi saturada (es decir cuando la cantidad de hongo aumenta tanto que se acerca a la capacidad de carga de la celda), las esporas colonizan las celdas contiguas permitiendo así que el hongo se reintegre en zonas donde se había extinguido. En la configuración usada y para asegurar que las simulaciones siguen su curso para varias generaciones de animales, la biomasa inicial M_0 se ha establecido al 99% de K en todas las celdas.

4.2. Rasgos genéticos y su base cuantitativa

La escala de los animales en la configuración por defecto es homogénea, ya que todos los individuos pesan menos de 1mg cuando son adultos (por ello se usan ácaros y colémbolos como ejemplo). Al incluir más especies es necesario reescalar y reparametrizar los valores asociados a los rasgos de cada especie. Los 13 rasgos que se comentan en la sección 3.1.1.1 tienen un límite evolutivo, a partir del cual la población no puede evolucionar por causas fisiológicas. Teniendo esto en cuenta, se ha establecido una variabilidad en los rasgos que depende de la genética y que trabaja dentro de esos límites evolutivos. Por lo tanto, para cada rasgo X se establece un rango de valores como el que se muestra a continuación:

$$l_X = L_X + \varphi \left(\frac{U_X - L_X}{2} \right) \quad (4.3)$$

$$u_X = U_X - \varphi \left(\frac{U_X - L_X}{2} \right) \quad (4.4)$$

donde l_X y u_X representan los límites inferior y superior del rango de posibles valores utilizado para el rasgo X en la simulación, respectivamente. L_X y U_X definen los límites del rango estándar para el rasgo y φ es un coeficiente de restricción entre 0 y 1 que determina qué porcentaje de la mitad del rango estándar es usado para calcular el rango final de posibles valores utilizado (l_X, u_X). Por lo tanto, un mayor φ conlleva menos variabilidad en el rasgo. Además, tal y como se describe anteriormente, existen los límites evolutivos (Π_X, K_X) a partir de los cuáles la población no puede evolucionar. Lógicamente, se fuerza a que $U_X < K_X$ y que $L_X > \Pi_X$ de modo que se asegura una variabilidad en los rasgos lo suficientemente grande para que puedan evolucionar nuevos especímenes (debido a la variación genética implícita), además de proporcionar un umbral (L_X, U_X) bastante alejado de los límites evolutivos (Π_X, K_X). En la configuración por defecto se usa $\varphi = 0,5$, pero este valor puede modificarse para obtener una mayor o menor variación genética, según se necesite para la simulación planteada.

Los trece rasgos incluidos en las simulaciones y los límites de sus rangos estándar y evolutivos son los siguientes:

- ***tanque-de-energía-inicial*** – Representa el porcentaje de masa dedicado al mantenimiento y al futuro crecimiento. La *masa corporal* real del individuo (M) es la suma del *tamaño* más el *tanque-de-energía*, y ambas son variables de estado. Límites del rango de datos = [25 %, 50 %] del tamaño del individuo; límites evolutivos = [0 %, 100 %].
- ***crecimiento*** – Este rasgo determina cuánto crece un individuo cada vez que muda, es decir, cuánta energía es utilizada para pasar al siguiente estado de desarrollo, también llamado instar. Este sistema de mudas es propio de los artrópodos que se están simulando. Una fracción del *tanque-de-energía* (variable de estado) existente en el momento de la muda será mantenida aún después del proceso de muda. El *crecimiento* es en realidad el ratio en que se relacionan el nuevo instar relativo al instar previo. El número de instar que puede alcanzar un individuo depende de la especie. Límites del rango de datos: depredadores = [1,01, 1,2]; presas = [1,01, 1,1]; Π_X, K_X : 1,01 – 1,2.

- **fenología** – Define el día de nacimiento, ya sea a partir del comienzo de la simulación o desde la puesta de huevos correspondiente. El individuo empieza a estar activo cuando se alcanza este día de nacimiento en la simulación. Este rasgo también podría considerarse como el tiempo de desarrollo del huevo. Además, la fecha final de nacimiento varía dependiendo de la temperatura del ambiente, y para dicho cálculo se ha añadido a la ecuación un valor Q_{10} [8] calculado a partir del rango de masa corporal de los animales de la simulación. Límites del rango de datos: depredadores = [3, 11]; presas = [2, 10]; límites evolutivos = [1, 100].
- **tamaño-inicial** – Representa la masa estructural del individuo al nacer. Los rangos han sido establecidos para simular los datos observados de los colémbolos y los ácaros. Límites del rango de datos: depredadores = [0,01425, 0,02025] mg; presas = [0,00725, 0,01325] mg; límites evolutivos = [0,05, 0,022].
- **asimilación** – La eficiencia de asimilación indica qué cantidad de la comida ingerida será convertida en masa propia al realizarse la etapa de metabolismo. Siguiendo trabajos previos en fauna del suelo, se asume que este valor ronda el 85 % [9]. Límites del rango de datos = [0,7, 0,9]; límites evolutivos = [0, 1].
- **ratio-de-voracidad** – Indica el ratio máximo de consumo que puede alcanzar un animal en un día. Este valor se ha implementado como un coeficiente v que hace escalar a la *voracidad* con la *masa corporal* del individuo, de la forma $0,1M^v$ [10]. Si un depredador supera este ratio al ingerir alimento, quedará en estado inactivo durante varios días mientras lo digiere. Límites del rango de datos = [0,6, 0,7]; límites evolutivos = [0,55, 0,75].
- **ratio-de-velocidad** – Representa la velocidad explosiva (en cm/s) cuando un depredador se encuentra con una presa (o con otro depredador), intentando el primero cazar al segundo, y éste escapar de él. Se ha implementado como un coeficiente s que hace escalar a la *velocidad* con la *masa corporal* del individuo, de la forma αM^s . En esta fórmula α es una constante de normalización entre 1 – 7, cuyo valor depende linealmente del propio s . Esto es así para cubrir todas las posibles velocidades observadas desde los ácaros más pequeños hasta los más grandes [11], cubriendo un rango de masas de 0,03 – 465 mg. Además, los colémbolos pueden escapar de la depredación realizando un salto, lo cual es un mecanismo común y que haría que se moviesen a una velocidad de varios órdenes de magnitud mayor.

Sin embargo, para simplificar el proceso de caza y depredación, no se consideran estos saltos en esta fórmula concreta. Límites del rango de datos = $[0,1, 0,3]$; límites evolutivos = $[0,05, 0,35]$.

- **área-de-búsqueda** – Representa la movilidad del individuo, que es muy diferente de la velocidad. La *velocidad*, como se ha descrito, indica la facilidad de escape o de caza cuando se producen los encuentros entre depredador y presa. Sin embargo, el *área-de-búsqueda* refleja la movilidad del animal indicando cuánta distancia es capaz de recorrer para buscar recursos o celdas seguras. Cuanto mayor sea este valor mayor número de celdas podrá viajar por día. Al no disponer de mejores datos, se ha establecido que el área cubierta en un día escala con la *masa corporal* en un factor m , de una forma similar que la velocidad: M^m . Ya que la *masa corporal* es siempre < 1 mg en toda la simulación, un valor mayor del coeficiente m implica menos movilidad para la misma masa corporal. Para incluir depredadores eficientes en las simulaciones, éstos tienen mayor movilidad que las presas. Para traducir esta movilidad calculada a un número de celdas real dentro de la simulación, se utiliza una interpolación lineal que asocia al mínimo M^m la distancia de 1 celda por día, y al máximo posible (la del depredador más grande y a mayor temperatura) la distancia de 10 celdas por día. Límites del rango de datos: depredadores = $[0,1, 0,3]$; presas = $[0,2, 0,4]$; límites evolutivos = $[0,05, 0,5]$.
- **ratio-metabólico** – Cuantifica la pérdida de energía debida al metabolismo. Este proceso se ha implementado siguiendo la MTE (Metabolic Theory of Ecology)[12] y estimaciones recientes en fauna del suelo, emulando los efectos en el ratio metabólico de la temperatura, energía de activación y masa corporal[13]:

$$\ln I = \ln I_0 + a \ln M - E \left(\frac{1}{kT} \right) \quad (4.5)$$

donde I es el *ratio-metabólico* (J/h), I_0 es una constante de normalización, a es un coeficiente que relaciona el *tamaño* del individuo con su *ratio-metabólico*, E es la *energía-de-activación* (descrita en el siguiente rasgo, está en electronvoltios eV), k es la constante de Boltzmann ($8,62 \times 10^{-5} eV/K$) y T es la temperatura ambiental en grados Kelvin. Todos estos parámetros han sido incluidos tal y como se ha observado para los ácaros y los colémbolos[13]. Además de la pérdida de energía debida al metabolismo, se ha añadido una fracción que incluye la pérdida debida

al estrés por los encuentros con depredadores[14]. Dicho estrés también influye en el estado actual de la *voracidad* del animal y en su *área-de-búsqueda*, ya que si han tenido recientemente malas experiencias, serán menos propensos a moverse y buscar recursos. Límites del rango de datos: depredadores = $[0,6, 0,7]$; presas = $[0,7, 0,8]$; límites evolutivos = $[0,55, 0,85]$.

- **energía-de-activación** – Este un parámetro (E) que fija la energía necesaria para activar el metabolismo del animal. Es decir, para una temperatura dada, este rasgo indica el grado en que ocurren todos los procesos bioquímicos y energéticos basales que implican gasto energético o metabólico. Está implicado en la ecuación (4.5) y permitirá estudiar la evolución adaptativa mediante la sensibilidad termal del *ratio-metabólico*. Los límites de los rangos se han fijado según los coeficientes publicados para los ácaros y los colémbolos[13]. Límites del rango de datos: depredadores = $[0,3, 0,5]$; presas = $[0,55, 0,75]$; límites evolutivos = $[0,3, 0,75]$.

Además de los diez rasgos descritos, se han añadido otros tres rasgos adicionales que representan la variabilidad en la plasticidad debida a la temperatura (Q_{10}) para tres variables: *voracidad*, *velocidad* y *área-de-búsqueda*. Estos parámetros Q_{10} cuantifican en qué medida aumenta una variable cuando la temperatura se ve incrementada 10°C . De esta forma se establece una temperatura mínima de 15°C y máxima de 25°C , y estos rasgos harán que esas variables vean modificado su valor dependiendo de la temperatura actual del ambiente, que puede ser diferente y debe ser fijada en cada simulación. Idealmente los animales se benefician más cuando son más activos, esto es, cuando la temperatura es mayor. Sin embargo esto puede ser también perjudicial para ellos, ya que una mayor actividad implica una mayor exposición ante los depredadores[15]. Los límites de los rangos para los tres rasgos que influyen en la plasticidad Q_{10} son los siguientes:

- **Q_{10} -en-voracidad** – Determina el cambio que sufre la *voracidad* del individuo al producirse una variación en la temperatura del entorno de 10°C . Límites del rango de datos = $[2, 4]$; límites evolutivos = $[1, 6]$
- **Q_{10} -en-velocidad** – Determina el cambio que sufre la *velocidad* del individuo al producirse una variación en la temperatura del entorno de 10°C . Límites del rango de datos = $[1,5, 2,5]$; límites evolutivos = $[1, 3]$

- **Q10-en-área-de-búsqueda** – Determina el cambio que sufre el *área-de-búsqueda* del individuo al producirse una variación en la temperatura del entorno de 10 °C. Límites del rango de datos = [1,5, 2,5]; límites evolutivos = [1, 3]

Para calcular la influencia de estos rasgos en las variables mencionadas anteriormente, se ha utilizado una interpolación lineal entre el mínimo y el máximo de las temperaturas posibles en las simulaciones (15-25°C). Así, para simulaciones a una temperatura intermedia se utiliza el valor interpolado, por ejemplo, a 21°C se estaría usando un valor intermedio entre el correspondiente a 15°C y 25°C.

4.3. Modularidad de los rasgos

La modularidad e integración fenotípica explica cómo están inter-relacionados los rasgos cuantitativos a lo largo de los individuos de una comunidad. Así, los organismos más integrados son aquellos cuyos rasgos (ambos genéticos y funcionales) están muy fuertemente correlacionados entre sí, considerando todos los individuos. Lo contrario sería decir que un organismo poco integrado es aquel que presenta una modularidad baja, con lo cual sus rasgos no están correlacionados. Por ejemplo, si unos animales que son agresivos porque su genética lo ha definido así además tienen ratios de crecimiento altos, se podría decir que estos animales estarían fenotípicamente integrados para los rasgos de agresividad y crecimiento. Un módulo puede ser definido como un conjunto de rasgos intercorrelacionados que es independiente de otros conjuntos de rasgos intercorrelacionados. Los 13 rasgos que se han explicado anteriormente han sido inicialmente incluidos en 5 módulos: 4 módulos con 3 rasgos cada uno, y un 5º módulo con solamente un rasgo. Cada uno de los módulos de 3 rasgos están formados, bien por tres correlaciones genéticas positivas, o bien por dos negativas y una positiva entre rasgos, reflejando así los *tradeoffs* genéticos. Sin embargo, hay que resaltar que estas correlaciones entre rasgos están implementadas como un ejemplo para presentar este marco eco-evolutivo, pero estos módulos no tienen necesariamente que estar organizados de esta forma en la naturaleza. Se van a representar los módulos de tres rasgos mediante el nombre del rasgo seguido de dos signos, uno referido a la dirección de la correlación con el primer rasgo (leído de izquierda a derecha) y el otro referido a la correlación con el segundo rasgo. Por ejemplo, en el módulo 1 tenemos:

- *tanque-de-energía-inicial*(-, -).
- *crecimiento*(-, +).
- *fenología*(-, +).

Esto significa que los animales que han nacido con más reservas invierten menos en crecimiento; es decir, al crecer mudan a tamaños menores e incrementan su ratio de crecimiento, y también tardan menos en nacer. Este patrón es una restricción ecológica ya que, para dos animales idénticos, crecer a un tamaño mayor conlleva más tiempo. Así, la restricción genética (correlación negativa) imita a la restricción ecológica.

El módulo 2 contiene los siguientes tres rasgos:

- *ratio-de-velocidad*(+, +).
- *ratio-metabólico*(+, +).
- *área-de-búsqueda*(+, +).

En el módulo 3 se encuentran estos tres rasgos:

- *tamaño-inicial*(-, -).
- *asimilación*(-, +).
- *ratio-de-voracidad*(-, +).

El módulo 4 contiene los tres rasgos siguientes:

- *Q10-en-voracidad*(-, -).
- *Q10-en-velocidad*(-, +).
- *Q10-en-área-de-búsqueda*(-, +).

En la tabla 4.1 se representan dichos módulos junto a su tipo de correlación dentro del módulo. Así, una relación negativa entre el *tamaño-inicial* y el *ratio-de-voracidad* implica que los animales más grandes tienden a ser proporcionalmente más voraces, ya que un *ratio-de-voracidad* menor conlleva una mayor voracidad (recordemos que se aplica $0,1M^v$ donde v representa este *ratio-de-voracidad* y M está siempre entre 0 y 1). De igual forma, un coeficiente de *asimilación* mayor correlacionado positivamente con el *ratio-de-voracidad* significa que los animales más voraces son los menos eficientes al asimilar la comida, reflejando un tradeoff potencial entre las eficiencias de asimilación y de voracidad. Finalmente, el modulo 4 refleja los tradeoffs de adaptación termal para la movilidad, añadiendo la plasticidad derivada del aumento de la temperatura, que causa un incremento en la voracidad de los animales (haciendo efecto así también en la velocidad y el área de búsqueda). Esto permite un acercamiento a la evolución adaptativa de los rasgos cuantitativos en el contexto de las interacciones bióticas y el cambio climático. Sin embargo, ha sido descubierto recientemente que el signo y la magnitud de estas correlaciones genéticas cambia dependiendo del entorno, lo cual sugiere que podrían restringir la evolución adaptativa en menor grado de lo que se pensaba.

TABLA 4.1: Módulos de rasgos implicados en la genética de los animales.

	Nombre del rasgo	Papel en la correlación
Módulo 1	<i>tanque-de-energía-inicial</i>	(-, -)
	<i>crecimiento</i>	(-, +)
	<i>fenología</i>	(-, +)
Módulo 2	<i>ratio-de-velocidad</i>	(+, +)
	<i>área-de-búsqueda</i>	(+, +)
	<i>ratio-de-metabolismo</i>	(+, +)
Módulo 3	<i>tamaño-inicial</i>	(-, -)
	<i>asimilación</i>	(-, +)
	<i>ratio-de-voracidad</i>	(-, +)
Módulo 4	<i>Q10-en-voracidad</i>	(-, -)
	<i>Q10-en-velocidad</i>	(-, +)
	<i>Q10-en-área-de-búsqueda</i>	(-, +)
Módulo 5	<i>energía-de-activación</i>	

4.4. Genética cuantitativa y reproducción

Para asignar a los rasgos una base cuantitativa genética, se asume que un rasgo está definido mediante un par de cromosomas. Por lo tanto, cada individuo tiene 13 pares de cromosomas. Por simplicidad, en las primeras versiones de la aplicación todos los animales eran hermafroditas. Sin embargo esto se ha mejorado y ahora existen ambos individuos machos y hembra de cada especie, aunque los cromosomas siguen siendo autosomas todos ellos. Cada rasgo está determinado por 20 loci (o posiciones) en cada cromosoma de los que forman su par, y cada locus (posición) obtiene su valor a partir de 10 posibles alelos cuya frecuencia se obtiene a partir de una distribución uniforme. Para inducir la correlación genética entre rasgos debido a efectos pleiotrópicos [16], a un número concreto de loci se les habilita el tener efecto en más de un rasgo (hasta en tres rasgos diferentes dentro del mismo módulo). Así, se pueden obtener correlaciones genéticas más fuertes aumentando el número de loci compartidos por los rasgos dentro de un módulo.

Se ha definido el parámetro ρ para establecer el número de loci comunes para el efecto pleiotrópico [5] tal y como se muestra a continuación: $\rho = S_L/N_L$, donde N_L es el número de loci total del rasgo, y S_L es el número de loci pleiotrópicos (es decir, los que son compartidos con otro rasgo). Por lo tanto, un valor mayor para ρ significa que el rasgo se determina mediante un número menor de loci exclusivos y utilizando más loci compartidos o que afectan a otros rasgos. Por ejemplo, $\rho = 0,9$ significa que un rasgo de 20 loci se determina por 18 loci pleiotrópicos (compartidos con otro rasgo) y solamente con 2 loci exclusivos. Un valor de $\rho = 0$ implica que todos los loci son exclusivos y que el rasgo está completamente desacoplado de cualquier otro rasgo, tal y como es el caso del rasgo *energía-de-activación*.

Se ha asignado el parámetro ρ con el motivo de obtener una aproximación a la correlación genética, permitiendo que se puedan simular diferentes arquitecturas genéticas. Para calcular el efecto pleiotrópico total de los alelos, cada alelo suma una pequeña cantidad numérica a un rasgo, obtenida de una distribución uniforme (0-1), para lo cual se asume codominancia exacta. Por lo tanto, sea cual sea la identidad del alelo, todos los alelos suman de la misma forma su cantidad para formar el valor total fenotípico del rasgo. Las correlaciones positivas entre rasgos se inducen sumando ambos valores fenotípicos tanto de los loci exclusivos como de los loci pleiotrópicos que determinan un rasgo concreto.

Las correlaciones negativas entre rasgos se inducen restando a 1 cada valor fenotípico de los loci pleiotrópicos y añadiendo esta diferencia a la suma de valores fenotípicos de los loci exclusivos. Tras realizar estas sumas, para transformar estos valores fenotípicos arbitrarios en valores fenotípicos que tengan sentido ecológicamente, se usa una interpolación lineal para realizar un cambio entre la escala arbitraria generada y la escala que tiene sentido ecológico. Es decir, se convierten esos valores arbitrarios de forma que se transfieren a la escala de valores cuyo rango de datos se ha explicado para cada rasgo en la sección 4.2 de rasgos genéticos y base cuantitativa.

Este procedimiento ha permitido incorporar satisfactoriamente la cantidad deseada de correlaciones fenotípicas entre rasgos, que se determinan mediante la herencia Mendeliana de varios genes, cada uno con un efecto relativamente pequeño, resultando en la imitación de una genética cuantitativa. Además, como hay cierto número de loci que no se expresan nunca (en los rasgos pleiotrópicos, algunos loci no se expresan porque su valor fenotípico es asignado desde los loci de otro cromosoma), estos loci pueden usarse para seguir el destino de los alelos neutrales (deriva genética) durante la simulación. Igual que en los cromosomas reales, cuánto más lejos se encuentran los loci neutrales de los funcionales en el cromosoma, más neutralmente se comportarán. Por otro lado, los loci cercanos a los genes funcionales formarán parte indirectamente de la selección debido a la posibilidad de enlace por proximidad a los genes seleccionados durante la recombinación, como es el caso de los microsatélites [17] [18].

4.5. Movimiento adaptativo y experiencia previa

Ya que en este desarrollo las especies depredadoras pueden tener depredadores superiores (o simplemente ser caníbales), ambos depredadores y presas se mueven de celda a celda exhibiendo un comportamiento adaptativo antidepredadores, es decir, evitando las zonas donde se localizan más depredadores. También se considera la disponibilidad de recursos. Por lo tanto, cuando un individuo tiene que decidir si quedarse en la celda actual o moverse a alguna de las colindantes, el animal evalúa un ratio de depredador-recursos y elige la celda con el ratio más bajo. Cuando se ejecutan simulaciones sin depredadores, las presas simplemente se mueven a las celdas cuya productividad de hongo sea mayor. Además, cuando los recursos se agotan tanto en la celda actual como en las colindantes,

el animal *salta* un número de celdas con una dirección aleatoria y cuya distancia depende de su rasgo de movilidad *área-de-búsqueda* (ver sección [4.2](#) de rasgos genéticos).

5 | Implementación

5.1. Descripción general del proceso

Hasta este punto nos hemos centrado en explicar los conceptos que se utilizan en cada etapa. Estos conceptos se han explicado en un orden lógico para facilitar la comprensión del proceso y proporcionar a la vez los detalles necesarios para entender cada fase. El Algoritmo 1 muestra el pseudocódigo secuencial desde el que se parte para explicar cómo está implementado el simulador.

Como podemos ver en los pasos 1 y 2 del algoritmo, en primer lugar se crea el mundo inicializando las celdas y los animales. Este proceso establece una cantidad de hongo para cada celda y genera los cromosomas para cada individuo, analizándolos y extrayendo de ellos los rasgos característicos. Cada ser vivo se coloca en una celda aleatoriamente.

La simulación entonces comienza desde el día 0 hasta el número de días máximo indicado, repitiendo los pasos desde el 3 al 14. Cada día empieza activando los individuos que están preparados para nacer, en el paso 3. Entonces se ejecuta el algoritmo de crecimiento y propagación del hongo (pasos 4 y 5, respectivamente).

Antes de realizar el movimiento, en el paso 6 se modifican algunos rasgos de cada individuo basándose en los encuentros que ha podido sufrir en el día anterior. En el paso 7 las presas caminarán hacia las celdas más convenientes, considerando depredadores, si éstos existen en la simulación, y quedando expuestos a ser devorados por ellos en su caso.

Algoritmo 1 IBM secuencial

1. Inicializar celdas	<i>También los hongos</i>
2. Inicializar animales	<i>Genes y primera generación</i>
for <i>dia = 0 to DIAS</i>	
3. Activar animales	<i>Nacen si están preparados</i>
4. Crecer hongos	<i>En todas las celdas</i>
5. Propagar hongos	<i>Si hay exceso</i>
6. Tunear rasgos	<i>Depende de los encuentros</i>
7. Mover presas	
8. Alimentar presas	<i>Comen hongos</i>
9. Metabolizar y crecer presas	<i>Depende del alimento y la distancia</i>
10. Morir depredadores por ambiente	<i>Ayuda a la estabilidad</i>
11. Mover depredadores	
12. Asimilar alimento depredadores	
13. Metabolizar y crecer depredadores	<i>Depende del alimento y la distancia</i>
14. Reproducir animales	<i>Ambos presas y depredadores preparados</i>

Las presas concluyen entonces su actividad alimentándose en el paso 8 y evaluando su balance energético (relativo al metabolismo) en el paso 9. Aún no se reproducirán pues pueden ser devorados por los depredadores en los siguientes pasos.

Los depredadores pueden morir aleatoriamente por causa del ambiente en el paso 10. Esta característica se incluye para ayudar a la estabilidad del sistema. Si sobreviven, se mueven y cazan en el paso 11 asimilando el alimento ingerido en el paso 12. Su balance energético (relativo al metabolismo) se evalúa en el paso 13.

Por último, los individuos preparados de ambas especies se reproducen buscando aleatoriamente un compañero. Este proceso se realiza al final, en el paso 14, para asegurar que los individuos implicados son sólo aquellos que han sobrevivido al día actual.

5.2. Clases y métodos implementados

El sistema se ha desarrollado a partir de una versión previa ya existente pero que era menos completa y menos eficiente. La versión anterior está implementada en el lenguaje de programación R, el cual ofrece unas características de rendimiento bastante pobres en comparación a otros lenguajes. Además la inclusión de cualquier nueva característica en este código de R requería un gran esfuerzo por parte del programador, ya que las funciones dentro del código están muy dispersas y son difíciles de seguir, suponiendo un gran trabajo de abstracción mental.

El código de partida escrito en R realiza tareas similares a las detalladas en la sección 3 de análisis, pero no implementa todos los métodos asociados de forma tan clara y precisa. Además, las estructuras de datos utilizadas en R se componen principalmente de matrices que albergan información de diversos tipos. Por ejemplo, los individuos se almacenan en cada fila y una posición de dicha fila contiene la información sobre un rasgo o una variable del animal en cuestión, incluida la posición en la que se encuentra. Las celdas se almacenan en cada fila de su matriz correspondiente y cada posición de la fila alberga información acerca del contenido de la celda. La genética de cada animal se almacena en cada fila de su matriz correspondiente y cada posición de la fila contiene información sobre un valor genético o alelo.

La nueva implementación se ha llevado a cabo bajo el lenguaje de programación *C++*, el cual ha permitido una estructura más uniforme en el código siendo posible identificar más fácilmente cada una de las funciones implicadas en la simulación. Se ha seguido un modelo orientado a objetos para obtener así un código modular y extensible de cara a futuras modificaciones: adición de nuevas especies, inclusión de nuevas dinámicas en los animales, mejora de las características del suelo, etc. Todas estas propuestas de trabajo futuro pueden ser consultadas en la sección 7.

A continuación se van a mostrar todas las clases que participan en la implementación, y se van a explicar uno a uno los métodos que las componen. No obstante, para facilitar la lectura, se van a pasar por alto los métodos que realizan algún trabajo sencillo y se van a detallar los que realmente realizan tareas que son importantes para la ejecución de las simulaciones. La figura 5.1 muestra el diagrama de clases principal, las cuales van a ser detalladas en las siguientes secciones.

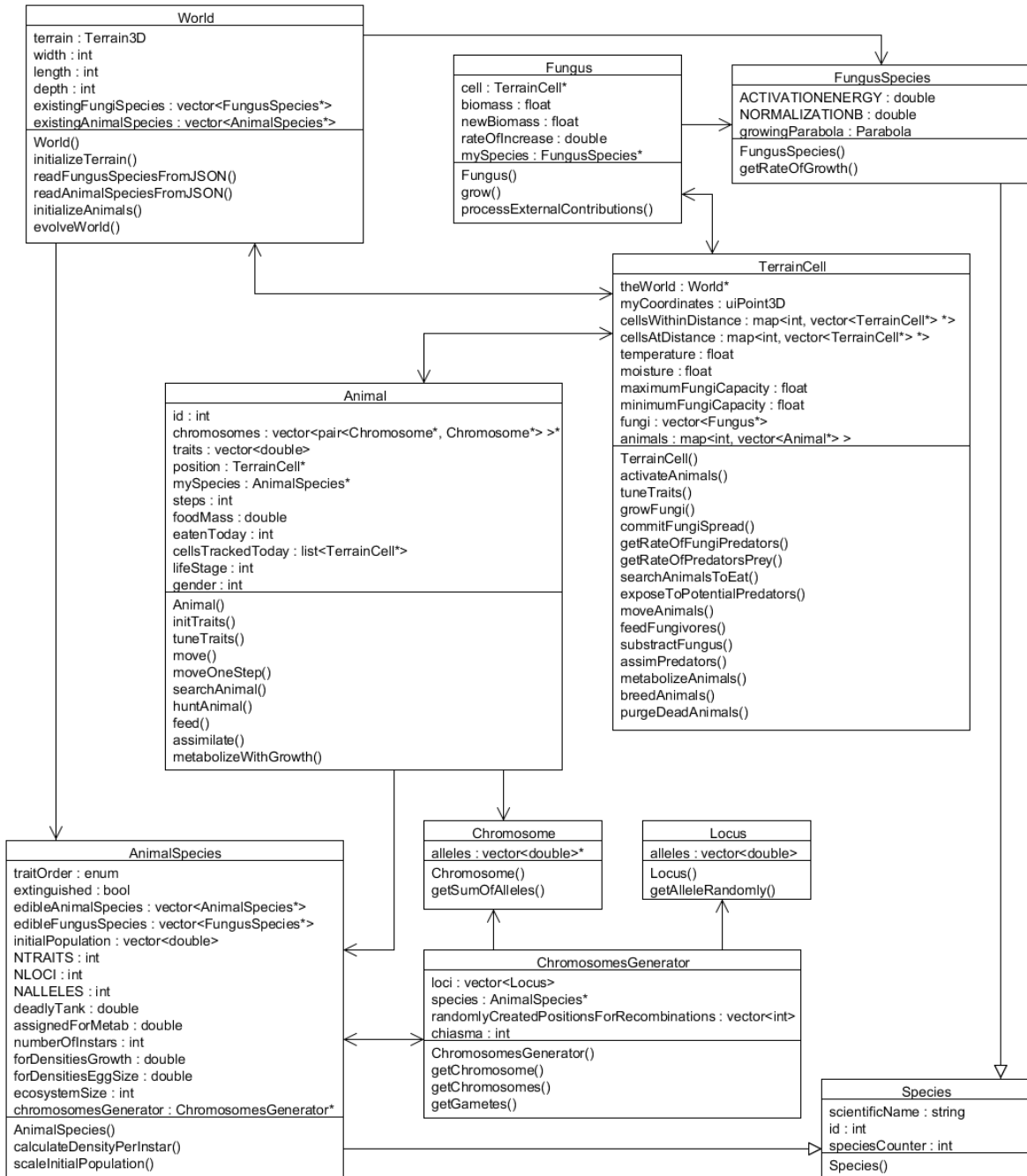


FIGURA 5.1: Diagrama de clases principal

5.2.1. Animal

La clase `Animal` contiene la especificación para cada individuo presente en la simulación. Un `Animal` tiene las siguientes características:

- ***id*** – Cada individuo tiene un identificador único entero que lo diferencia del resto. Este identificador sigue un orden global, es decir, todos los individuos tienen uno diferente incluso si son de especies diferentes.
- ***chromosomes*** – Un vector de pares de cromosomas (ver clase `Chromosome` en 5.2.3). Cada par de cromosomas contenido en este array contiene el material genético asociado a uno de los 13 rasgos del individuo.
- ***traits*** – Un vector de *doubles* que contiene los valores asociados a los 13 rasgos del individuo (obtenidos tal y como se explica en la sección 4.4 de genética).
- ***position*** – Un puntero a la dirección de la celda en que se encuentra actualmente el individuo (ver clase `TerrainCell` en 5.2.9).
- ***mySpecies*** – Un puntero a la dirección de la especie a la que pertenece el individuo (ver clase `AnimalSpecies` en 5.2.2).
- ***steps*** – Un entero indicando el número de pasos que ha caminado el individuo durante el día actual, es decir, el número de celdas que ha recorrido.
- ***foodMass*** – Un *double* que indica la cantidad de masa ingerida durante el día actual. Esta masa puede provenir tanto de las presas ingeridas, como del hongo ingerido, dependiendo de la especie a la que pertenezca el individuo.
- ***eatenToday*** – Un entero indicando el número de presas ingeridas durante el día actual.
- ***cellsTrackedToday*** – Una lista de punteros a las direcciones de las celdas que el individuo ya ha visitado durante el día actual. Esto es de utilidad a la hora de realizar la función de movimiento del individuo, ya que lógicamente no es de esperar que un individuo revise más de una vez la misma celda durante el mismo día.

- **lifeStage** – Un entero indicando el estado del animal: no nacido, activo, muerto por inanición, muerto por depredación, reproduciéndose, muerto por senescencia. Para facilitar la comprensión del programador, se han listado estos diferentes estados en un *enum* dentro de la misma clase *Animal*.
- **gender** – Un entero indicando el género del animal: macho, hembra o hermafrodita. Estos tres también han sido listados en un *enum* dentro de la misma clase *Animal*.

Además de estos parámetros, también se almacenan datos como la fecha de muerte del animal, la generación a la que pertenece, su fecundidad hasta la fecha, el tamaño al nacer, la velocidad al nacer y el tanque de energía al nacer.

Los métodos que se implementan en esta clase son los siguientes:

- **Animal()** – Es el constructor de la clase, que toma como entrada un vector de pares de cromosomas, la posición en la que se encontrará el animal al aparecer en la simulación y la especie a la que pertenece (además de otras variables menos importantes como el número de la generación a la que pertenece, su género y su día de nacimiento). Este método crea el animal con todos los rasgos por defecto a 0 y posteriormente realiza una llamada a *initTraits()* para inicializarlos a sus valores correctos.
- **initTraits()** – Este método utiliza la información recogida en el vector de pares de cromosomas para inicializar los valores de los rasgos del individuo, tal y como se explica en la sección 4.4 de genética.
- **tuneTraits()** – Esta función modifica el *ratio-de-voracidad* y el *área-de-búsqueda* de acuerdo a la condición del animal, es decir, a cuánta energía tiene almacenada. Si la condición es mejor, el individuo se moverá menos y se alimentará menos, y viceversa.
- **move()** – Esta función se encarga de realizar el movimiento del individuo, en base a su *área-de-búsqueda*. El animal intentará realizar tantos pasos como indique su *área de búsqueda* y además quedará expuesto ante los depredadores al visitar una nueva celda. Después, si el animal es un depredador, busca presas potenciales en esta nueva celda e intenta cazarlas. Además, se realiza aquí el movimiento de los

animales que están en estado de reproducción, que evaluarán las celdas cercanas en busca de una pareja para reproducirse.

- ***moveOneStep()*** – Esta función es llamada desde la función *move()* anterior cada vez que el individuo intenta realizar un paso (intentará como máximo *área-de-búsqueda* pasos). Se evalúan las celdas contiguas a la actual y se decide cuál será la de destino dependiendo del número de presas y depredadores que haya en ellas. Si no existe alimento en las celdas contiguas, el animal puede realizar un salto aleatorio a una posición cercana.
- ***searchAnimal()*** – Esta función realiza el proceso de encuentro entre dos individuos dentro de una misma celda. Dependiendo de la diferencia entre tamaños y de la voracidad de ambos individuos, se calcula la probabilidad base de que se produzca un encuentro entre ellos. Después se genera un número aleatorio que se contrasta con esta probabilidad para determinar si se produce realmente el encuentro o no. Si el encuentro se produce, se procede a realizar el intento de depredación mediante la función *huntAnimal()*.
- ***huntAnimal()*** – Esta función realiza el proceso de depredación entre dos individuos dentro de una misma celda, entre los que se ha realizado un encuentro, según la función *searchAnimal()*. Dependiendo de la velocidad de ambos individuos y de la voracidad del atacante, se determina una probabilidad base de depredación. Después se genera un número aleatorio que se compara con esta probabilidad para determinar si se produce finalmente la depredación o no. Si la depredación se produce, se añade la presa a la lista de animales ingeridos del depredador y se marca como muerta por depredación. Además, la presa se convierte a masa efectiva que se acumula en el contador *foodMass* del depredador para su posterior asimilación.
- ***feed()*** – Esta función se ejecuta solo por las presas y después de haber realizado el movimiento del individuo, en cuyo caso comerá de la celda la cantidad de hongo que está dispuesta a ingerir (inferida mediante su voracidad). La búsqueda de hongo dentro de la celda se realiza recorriendo el vector *fungi* y comprobando qué fuentes de hongo son comestibles para el individuo en cuestión (véase la clase *TerrainCell* en la subsección 5.2.9). Si consigue alimentarse a partir de las fuentes de hongo de la celda, se procede en esta misma función a asimilar dicha masa ingerida y convertirla en energía, en cuyo proceso participa el rasgo de *asimilación*.

- ***assimilate()*** – El individuo asimila la masa obtenida debido a la ingesta de presas, en cuyo proceso participa el rasgo de *asimilación*. Este mecanismo por lo tanto supone un aumento de la energía del animal. Solo realizan esta función los depredadores (pues las presas realizan la asimilación directamente en la función *feed()*), que además restauran aquí a cero su contador de *eatToday* finalizando su ciclo alimenticio diario.
- ***metabolize WithGrowth()*** – El individuo incurre en una pérdida de energía debida a su metabolismo y también debida al número de pasos que haya recorrido el día actual. Además, tras eliminar dicha cantidad de energía, se comprueba si el animal tiene suficiente energía para seguir activo, de lo contrario pasaría al estado muerto por inanición. Si sigue activo, se compara la energía necesaria para mudar al siguiente instar con la energía actual del individuo, y si es suficiente se realiza el crecimiento y la muda. Si el animal ha alcanzado el máximo instar posible para su especie, pasa a estado de reproducción.

5.2.2. AnimalSpecies

Esta clase contiene todas las variables asociadas a cada especie de animal. En principio, actúa como una clase que solamente almacena dichas variables para su uso en cada uno de los individuos, pero también implementa algunos métodos relativos a la densidad de población del sistema.

A continuación se listan las variables más importantes que contiene la clase *AnimalSpecies*:

- ***traitOrder*** – Es un *enum* que lista los nombres de los 13 rasgos implicados en todas las especies. Se utiliza para facilitar el acceso a los datos del vector de rasgos de cada Animal (véase la clase *Animal* en la subsección 5.2.1) mejorando la organización del código.
- ***extinguished*** – Un valor booleano que indica si la especie se ha extinguido o no en la simulación actual. Se utiliza para comprobar cuántas especies de animales quedan activas en la simulación.

- ***edibleAnimalSpecies*** – Es un vector de punteros a instancias de *AnimalSpecies* que indica las especies de animal que son comestibles para esta especie concreta. Se utiliza a la hora de comparar celdas óptimas en la función de movimiento (véase la clase *Animal* en la subsección 5.2.1), así como en la función de búsqueda de animales (véase la clase *TerrainCell* en la subsección 5.2.9).
- ***edibleFungusSpecies*** – Es un vector de punteros a instancias de *FungusSpecies* que indica las especies de hongo que son comestibles para esta especie de animal concreta. Se utiliza a la hora de comparar celdas óptimas en la función de movimiento (véase la clase *Animal* en la subsección 5.2.1), así como en la función de alimentar a la presas (véase la clase *TerrainCell* en la subsección 5.2.9).
- ***initialPopulation*** – Es un vector de *doubles* que almacena la población inicial de individuos de esta especie en concreto para cada uno de los instars posibles de la especie. Es decir, al comienzo de la simulación existirá una cantidad determinada de individuos en cada instar de crecimiento. Para calcular cada uno de estos valores se utilizan las funciones *calculateDensityPerInstar()* y *scaleInitialPopulation()* descritas más adelante.
- ***NTRAITs*** – Es un entero que indica el número de rasgos de los individuos de las especies implicadas en la simulación. En principio este valor es fijo y vale 13, pero se almacena aquí para permitir otros valores en futuras implementaciones.
- ***NLOCI*** – Es un entero que indica el número de loci de los cromosomas de los individuos implicados en la simulación. En principio este valor es fijo y vale 20, pero se almacena aquí para permitir otros valores en futuras implementaciones.
- ***NALLELES*** – Es un entero que indica el número de alelos posibles que pueden ocupar un locus concreto dentro de un cromosoma. En principio este valor es fijo y vale 10, pero se almacena aquí para permitir otros valores en futuras implementaciones.
- ***deadlyTank*** – Es un *double* que indica para esta especie el ratio mínimo de energía/tamaño que puede alcanzar un individuo como mínimo antes de morir por inanición.

- ***assignedForMetab*** – Es un *double* que indica para esta especie el porcentaje de masa que un individuo necesita para realizar su crecimiento y mudar al siguiente instar.
- ***numberOfInstars*** – Es un entero que indica el número máximo de instar que puede alcanzar un individuo de esta especie. Cuando un individuo alcanza este instar máximo, pasa a estado de reproducción (ver función *metabolizeWithGrowth()* de la clase *Animal* en la subsección 5.2.1).
- ***forDensitiesGrowth*** – Es un *double* que indica el ratio de crecimiento que se aplica en la función *calculateDensityPerInstar()* detallada más adelante. Este valor sirve para calcular la masa aproximada que tienen los individuos de esta especie en cada instar y calcular su densidad de población adecuada.
- ***forDensitiesEggSize*** – Es un *double* que indica el tamaño aproximado de un huevo de esta especie. Se utiliza en la función *calculateDensityPerInstar()* como punto de partida para calcular la masa de los instars siguientes, y poder hallar la densidad de población adecuada para cada instar.
- ***ecosystemSize*** – Es un entero que indica la población total deseada para la simulación actual. Aunque se ha incluido en esta clase para su uso directo, el valor indica la población conjunta deseada de todas las especies de la simulación. Se utiliza en la función *scaleInitialPopulation()* para escalar las densidades de población calculadas anteriormente y generar un número de individuos total que iguale o se aproxime al valor establecido.
- ***chromosomesGenerator*** – Es un puntero a una instancia de la clase *ChromosomesGenerator* (véase la clase en la subsección 5.2.4). Cada especie animal podría tener sus propias características genéticas, y por eso esta variable se encuentra en esta clase.

Los métodos que se implementan en la clase *AnimalSpecies* son los siguientes:

- ***AnimalSpecies()*** – Es el constructor de la clase, que toma como entrada el nombre de la especie y todos los valores relacionados con la genética de la misma (*NTRAITTS*, *NLOCI*, *NALLELES* vistos anteriormente). Este método crea la especie animal e

instancia el generador de cromosomas para usarlo posteriormente en la creación de nuevos individuos de la especie (véase la clase `ChromosomesGenerator` en la subsección 5.2.4).

- ***calculateDensityPerInstar()*** – Es una función que toma como entrada un entero, el número de instar, y calcula la densidad de población que debe existir en el sistema para ese instar concreto, a partir del tamaño de huevo inicial *forDensitiesEggSize* y haciendo uso del ratio de crecimiento *forDensitiesGrowth* que ya se había indicado.
- ***scaleInitialPopulation()*** – Esta función toma como entrada la densidad total de población del sistema, y calcula para esta especie la población exacta inicial para cada instar, a partir del tamaño deseado del ecosistema *ecosystemSize*.

5.2.3. Chromosome

Esta clase contiene la especificación para un cromosoma, el cual estará formado por una serie de alelos, tal y como se explica en la sección 4.4 de genética. Por lo tanto, la clase es bastante simple y contiene solamente la siguiente variable:

- ***alleles*** – Un puntero a un vector de *doubles* que contienen los valores asociados a cada alelo de cada posición del cromosoma, o locus.

Además, la clase contiene los métodos necesarios para recoger la información de este vector de alelos o bien sumar los mismos a la hora de obtener los valores de los rasgos:

- ***Chromosome()*** – Es el constructor de la clase, que toma como entrada un vector de alelos y que debe ser instanciado por `ChromosomesGenerator` (véase la subsección 5.2.4).
- ***getSumOfAlleles()*** – Es una función que obtiene el valor de la suma de los alelos del cromosoma. Es utilizada por la clase `Animal` a la hora de inicializar los rasgos del individuo (véase la clase en la subsección 5.2.1).

5.2.4. ChromosomesGenerator

Esta clase contiene la especificación de un generador de cromosomas. Se utiliza siempre que se quiera crear un nuevo animal, ya sea en la inicialización de la simulación o bien mediante la reproducción entre dos individuos. La clase contiene las siguientes variables:

- ***loci*** – Es un vector de Locus (véase la clase en la subsección 5.2.7). A partir de él se obtienen los valores de los alelos para crear los nuevos cromosomas.
- ***species*** – Es un puntero a la instancia de AnimalSpecies de la cual forma parte este generador de cromosomas. Se utiliza para tener fácil acceso a las variables necesarias en la creación de cromosomas, como son NTRAITS, NLOCI y NALLELES (véase la clase AnimalSpecies en la subsección 5.2.2).
- ***randomlyCreatedPositionsForRecombinations*** – Es un vector de enteros que se utiliza a la hora de realizar la recombinación en el proceso de reproducción. Contiene una serie de posiciones para los alelos que se generan de forma aleatoria y que permanecen constantes para la misma especie.
- ***chiasma*** – Es un entero que contiene el índice que se usará como punto de corte o quiasma en el proceso de recombinación. Este valor es el mismo para todos los individuos de la misma especie, pero puede variar entre especies.

La clase ChromosomesGenerator implementa además los siguientes métodos, que se ocupan de tratar la genética y generar los cromosomas para los nuevos individuos:

- ***ChromosomesGenerator()*** – Es el constructor de la clase, que toma como entrada un puntero a la instancia de AnimalSpecies a la cual pertenece el generador de cromosomas. En esta función se inicializa el *chiasma* y el vector de *randomlyCreatedPositionsForRecombinations*. Además, se instancian los Locus incluidos en el vector *loci*, que contendrán todos los posibles alelos que pueden aparecer en cada locus (véase la clase Locus en la subsección 5.2.7).
- ***getChromosome()*** – Esta función proporciona un nuevo Chromosome. Los valores de este nuevo cromosoma son elegidos aleatoriamente de entre los disponibles para cada Locus.

- ***getChromosomes()*** – Esta función hace uso de la anterior, y se encarga de proporcionar un vector completo de pares de cromosomas, que serán los que constituyan la genética de un nuevo animal.
- ***getGametes()*** – Es una función que toma como entrada ambos vectores de pares de cromosomas de los dos individuos implicados en la reproducción, y realiza las recombinaciones necesarias para devolver otro vector de pares de cromosomas, que constituirá la base genética del nuevo descendiente (véase la sección 4.4). Hace uso de las variables *chiasma* y *randomlyCreatedPositionsForRecombinations* que se habían inicializado en el constructor de la clase.

5.2.5. Fungus

La clase Fungus contiene la especificación para una fuente de hongo. Se utiliza para crear instancias de un tipo de hongo determinado, y para controlar la cantidad de biomasa disponible en cada momento. La clase contiene las siguientes variables:

- ***cell*** – Es un puntero a la instancia de TerrainCell en la cual se ubica este hongo.
- ***biomass*** – Es un valor en punto flotante que indica la cantidad de biomasa disponible en esta fuente de hongo. Esta cantidad disminuye o aumenta dependiendo de diversos factores: ser consumido por animales cercanos, crecimiento propio del hongo a lo largo del tiempo, propagación desde celdas contiguas.
- ***newBiomass*** – Es un valor en punto flotante que indica la cantidad de biomasa que se ha obtenido mediante fuentes externas, debido al crecimiento del hongo (véase función *processExternalContributions()* de esta misma clase).
- ***rateOfIncrease*** – Es un *double* que contiene el valor del ratio de incremento del hongo. Este valor se modifica dependiendo de la biomasa disponible, tal y como se explica en la sección 4.1.
- ***mySpecies*** – Es un puntero a la dirección de la especie a la que pertenece esta fuente de hongo (véase la clase FungusSpecies en la subsección 5.2.6).

La clase `Fungus` implementa además las siguientes funciones que sirven para gestionar la biomasa existente en cada fuente:

- **`Fungus()`** – Es el constructor de la clase, que toma como entrada un puntero a la instancia de `FungusSpecies` a la cual pertenece esta fuente de hongo, así como la cantidad de biomasa inicial.
- **`grow()`** – Es una función que implementa el crecimiento del hongo y devuelve la cantidad en exceso, si la hubiera. Utiliza el ratio de crecimiento *rateOfIncrease* así como varios valores de la celda *cell* en la que se encuentra (véase la clase `TerrainCell` en la subsección 5.2.9).
- **`processExternalContributions()`** – Se encarga de añadir a esta fuente de hongo la cantidad de biomasa obtenida mediante contribuciones externas de otras celdas, si fuese posible sin alcanzar el máximo de capacidad de la celda.

5.2.6. `FungusSpecies`

Esta clase contiene todas las variables asociadas a cada especie de hongo. Actúa como una clase que solamente almacena dichas variables para su uso en cada una de las fuentes de hongo `Fungus`. También realiza los cálculos relacionados con el ratio de crecimiento de cada fuente `Fungus` de esta especie.

A continuación se listan las variables más importantes que contiene la clase `FungusSpecies`:

- **`ACTIVATIONENERGY`** – Es un *double* que indica el valor de la energía de activación para esta especie de hongo, necesaria para realizar los cálculos relativos al ratio de crecimiento (véase la sección 4.1).
- **`NORMALIZATIONB`** – Es un *double* que indica el valor de la constante de normalización *b* para esta especie de hongo, necesaria para realizar los cálculos relativos al ratio de crecimiento (véase la sección 4.1).
- **`growingParabola`** – Es una instancia de una clase que contiene la información sobre la parábola que representa el ratio de crecimiento de la especie. Se utiliza en los cálculos del ratio de crecimiento en *getRateOfGrowth*.

Además de las funciones para obtener y escribir las variables, la clase contiene el siguiente método a destacar:

- ***FungusSpecies()*** – Es el constructor de la clase, que toma como entrada el nombre de la especie y todos los valores relacionados con el crecimiento de la misma (*ACTIVATIONENERGY* y *NORMALIZATIONB* vistos anteriormente). Este método crea la especie de hongo.
- ***getRateOfGrowth()*** – Es una función que toma como valores de entrada la biomasa actual de la fuente de hongo, la temperatura, humedad y capacidad de la celda. Realiza el cálculo del ratio de crecimiento del hongo a partir de las variables de esta clase (*ACTIVATIONENERGY* y *NORMALIZATIONB*). El cálculo que se realiza se ha explicado en la sección 4.1.

5.2.7. Locus

Esta clase contiene la especificación para un locus, el cual forma parte de un cromosoma. Cada locus contiene en su implementación un grupo de posibles alelos que pueden ser elegidos para ocupar dicha posición en el cromosoma. Por lo tanto, la clase contiene la siguiente variable:

- ***alleles*** – Es un vector de *doubles* que representan los posibles valores para los alelos que pueden ser elegidos para ocupar la posición de este locus. Al crear un cromosoma, se elige aleatoriamente uno de estos alelos y se copia su valor dentro del cromosoma concreto.

La clase Locus además implementa los siguientes métodos:

- ***Locus()*** – Es el constructor de la clase, que toma como entrada el número de alelos de la especie de animal a la que pertenece este locus. Este método crea la instancia del locus e inicializa el vector de alelos *alleles* a una serie de valores aleatorios uniformemente distribuidos.

- ***getAlleleRandomly()*** – Es una función que selecciona aleatoriamente uno de los posibles alelos contenidos en el vector *alleles*, y lo devuelve para ocupar dicha posición en el cromosoma que se está tratando de construir (véase la clase *Chromosome* en la subsección 5.2.3).

5.2.8. Species

Esta clase contiene las variables asociadas a cualquier especie ya sea animal u hongo. Las clases *AnimalSpecies* (5.2.2) y *FungusSpecies* (5.2.6) heredan de esta clase. Contiene algunas variables que debe tener cualquier especie:

- ***scientificName*** – Es una cadena de texto que indica el nombre científico de la especie instanciada.
- ***id*** – Es un entero que sirve como identificador para la especie instanciada.
- ***speciesCounter*** – Es un entero que funciona como contador del número de especies. Cada vez que se crea una especie nueva, ya sea animal u hongo, se le asigna como identificador y el contador se incrementa.

Es una clase muy básica que solo implementa los métodos de acceso a las variables y el constructor:

- ***Species()*** – Es el constructor de la clase, que toma como entrada el nombre científico de la especie. Este método crea la instancia de la especie y le asigna un identificador *id* a partir de *speciesCounter*. También incrementa dicho contador de especies.

5.2.9. TerrainCell

Esta clase contiene la especificación para una celda del terreno sobre el cual actúan los animales y hongos. Su función es la de servir como conector entre el mundo y los individuos que lo habitan, así que contiene toda la información sobre los animales y

hongos que se localizan en la celda concreta. En la clase `TerrainCell` se encuentran las siguientes variables:

- ***theWorld*** – Es un puntero a una instancia de la clase `World` a la que pertenece la celda actual. Se utiliza para tener acceso a las dimensiones del mundo y para poder construir la red de celdas vecinas (*cellsWithinDistance* y *cellsAtDistance*) manteniendo un fácil acceso a las mismas.
- ***myCoordinates*** – Es un conjunto de tres enteros que indica las coordenadas de la celda respecto a su posición en el mundo.
- ***cellsWithinDistance*** – Es un mapa que almacena punteros a instancias del tipo `TerrainCell`, y cuya clave es un entero que representa la distancia entre la que se encuentran dichas celdas. Se utiliza para obtener las celdas colindantes a la actual, considerando la capacidad de percepción del individuo. En el caso de estas simulaciones la percepción se ve limitada a una celda, pero se pueden presentar escenarios en los cuales los individuos posean mayor distancia de percepción del ambiente.
- ***cellsAtDistance*** – Es un mapa que almacena punteros a instancias del tipo `TerrainCell`, y cuya clave es un entero que representa la distancia exacta a la que se encuentran dichas celdas. Este mapa se utiliza para almacenar y obtener de una forma eficiente las celdas hacia las cuales un individuo puede realizar un salto. Así, solamente se consideran aquí las celdas posicionadas a una distancia fija, para realizar el salto hacia alguna de ellas.
- ***temperature*** – Es un valor en punto flotante que indica la temperatura actual de la celda. Esta temperatura no varía en las simulaciones actuales, sino que permanece fija durante todos los días. Se utiliza como parámetro al calcular el ratio de crecimiento del hongo cada día (véase la clase `FungusSpecies` en la subsección 5.2.6) y también para ajustar o *tunear* los rasgos de los individuos (véase la clase `Animal` en la subsección 5.2.1).
- ***moisture*** – Es un valor en punto flotante que indica la humedad actual de la celda. Se utiliza como parámetro al calcular el ratio de crecimiento del hongo cada día (véase la clase `FungusSpecies` en la subsección 5.2.6).

- ***maximumFungiCapacity*** – Es un valor en punto flotante que indica la capacidad máxima de hongo que puede existir en la celda.
- ***minimumFungiCapacity*** – Es un valor en punto flotante que indica la capacidad mínima de hongo que puede existir en la celda.
- ***fungi*** – Es un vector de punteros a instancias de la clase Fungus. Contiene todas las fuentes de las diferentes especies de hongo que coexisten en la celda. Se usa para tener acceso a la biomasa de hongo almacenada en cada celda y disminuir o aumentar su cantidad dependiendo del crecimiento y el consumo de la misma.
- ***animals*** – Es un mapa que contiene punteros a instancias de la clase Animal, y que tiene como clave un entero que indica el estado de vida actual del animal (véase la clase Animal en la subsección 5.2.1). Se usa para localizar a los animales que existen en la celda y así poder realizar las distintas funciones de su ciclo diario.

La clase TerrainCell implementa además las siguientes funciones, que se encargan de activar el ciclo diario de los animales y controlar el crecimiento y consumo de la biomasa de hongo:

- ***TerrainCell()*** – Es el constructor de la clase, que toma como entrada el mundo al que pertenece la nueva celda y sus coordenadas. Este método crea la instancia de la especie e inicializa a cero las variables relacionadas con el agua y el hongo.
- ***activateAnimals()*** – Es una función que toma como entrada el día actual de la simulación. Se encarga de establecer el estado activo para los animales que han alcanzado el día indicado en su *fenología* (véase la sección 4.2).
- ***tuneTraits()*** – Esta función llama a *tuneTraits()* de cada animal de la celda (véase la clase Animal en la subsección 5.2.1). Para cada caso se modifica el *ratio-de-ivoracidad* y el *área-de-búsqueda* de acuerdo a la condición del animal.
- ***growFungi()*** – Esta función llama a *grow()* de cada fuente de hongo Fungus de la celda, activando su crecimiento (véase la clase Fungus en la subsección 5.2.5).
- ***commitFungiSpread()*** – Esta función llama a *processExternalContributions()* de cada fuente de hongo Fungus de la celda, para procesar las contribuciones de

biomasa provenientes de las celdas contiguas (véase la clase `Fungus` en la subsección 5.2.5).

- ***getRateOfFungiPredators()*** – Es una función que toma como entrada un puntero a una instancia de `Animal`, el cual está evaluando la celda. Se utiliza para obtener el ratio entre hongo comestible para dicho animal y depredadores potenciales que lo amenazan. Este ratio se utiliza para decidir el destino en la función de movimiento *move()* (véase la clase `Animal` en la subsección 5.2.1).
- ***getRateOfPredatorsPrey()*** – Es una función que toma como entrada un puntero a una instancia de `Animal`, el cual está evaluando la celda. Se utiliza para obtener el ratio entre depredadores potenciales que amenazan al animal dentro de la celda y presas comestibles para el mismo. Este ratio se utiliza para decidir el destino en la función de movimiento *move()* (véase la clase `Animal` en la subsección 5.2.1).
- ***searchAnimalsToEat()*** – Es una función que toma como entrada un puntero a una instancia de `Animal`, el cual es un depredador que está buscando presas para cazar. El animal recorre la celda en busca de animales comestibles para él, y realiza el proceso de encuentro y depredación sobre ellos (véanse las funciones *searchAnimal* y *huntAnimal* de la clase `Animal` en la subsección 5.2.1).
- ***exposeToPotentialPredators()*** – Es una función que toma como entrada un puntero a una instancia de `Animal`. Se encarga de exponer este animal ante los depredadores potenciales de la celda. Estos depredadores realizan sobre el animal el proceso de encuentro y depredación (véanse las funciones *searchAnimal* y *huntAnimal* de la clase `Animal` en la subsección 5.2.1). Este fenómeno ocurre siempre que un animal visita una nueva celda.
- ***moveAnimals()*** – Esta función llama a *move()* de cada animal de la celda, para activar la función de movimiento y depredación (véase la clase `Animal` en la subsección 5.2.1). Además, después de establecer cuál será el movimiento de cada animal, se actualiza su estado dentro de la celda o se recoloca en la celda correspondiente si se ha producido un desplazamiento.
- ***feedFungivores()*** – Es una función que llama a *feed()* de cada presa existente en la celda, haciendo que estos animales se alimenten del hongo disponible (véase la clase `Animal` en la subsección 5.2.1).

- ***subtractFungus()*** – Esta función se utiliza para restar de una fuente Fungus la biomasa de hongo consumida por una presa. Se especifica como entrada la especie FungusSpecies a la que pertenece ese hongo, y la cantidad a restar. Si la biomasa de hongo cae por debajo de un mínimo establecido, ese hongo se extingue de la celda actual.
- ***assimPredators()*** – Esta función llama a *assimilate()* de cada depredador de la celda (véase la clase Animal en la subsección 5.2.1). Cada individuo asimila la masa obtenida debido a la ingesta de presas. Este proceso lo realizan también las presas, pero ya está incluido dentro de su propia función *feed()*.
- ***metabolizeAnimals()*** – Esta función llama a *metabolizeWithGrowth()* de cada animal de la celda. Cada individuo incurre en una pérdida de energía debida a su metabolismo y también debida al número de pasos que haya recorrido el día actual, y se activa su crecimiento dependiendo de la energía almacenada (véase la clase Animal en la subsección 5.2.1).
- ***breedAnimals()*** – Es una función que recorre los animales en estado de reproducción y los empareja para generar nueva descendencia llamando a la función *breed()* (ver la clase Animal en la subsección 5.2.1). Los nuevos descendientes calculan sus rasgos mediante la función *interpolateTraits()* en base a la genética obtenida de los padres. Para una explicación más detallada, véase la sección 4.4.
- ***purgeDeadAnimals()*** – Esta función se encarga de eliminar de la simulación todos los animales que hayan muerto el día actual, sea cual sea la causa de la muerte. Esto es necesario para que la memoria no se colapse con copias de animales que murieron en los días anteriores. Sin embargo, la información relativa a estos animales queda ya escrita en los ficheros de salida, de forma que se puedan analizar los patrones de la población.

5.2.10. World

Esta clase contiene la especificación para un mundo, sobre el cual se sitúan las celdas del terreno que contendrán toda la información sobre los animales y los hongos. Su función es la de servir como contenedor principal y almacenar las variables asociadas a las características del mundo y las especies, tanto de animales como de hongos:

- ***terrain*** – Es un vector en tres dimensiones que almacena punteros a instancias de TerrainCell. Esta estructura define el terreno sobre el cual se ejecuta la simulación. Cada TerrainCell contiene sus propias coordenadas, tal y como se especifica en la clase definida en la subsección 5.2.9.
- ***width*** – Es un entero sin signo que indica la anchura del mundo. Este valor, junto a *length* y *depth*, determina el número de celdas o instancias de la clase TerrainCell que se crearán.
- ***length*** – Es un entero sin signo que indica la longitud del mundo. Este valor, junto a *width* y *depth*, determina el número de celdas o instancias de la clase TerrainCell que se crearán.
- ***depth*** – Es un entero sin signo que indica la profundidad del mundo. Este valor, junto a *width* y *length*, determina el número de celdas o instancias de la clase TerrainCell que se crearán.
- ***existingFungiSpecies*** – Es un vector de punteros a instancias de la clase FungusSpecies que almacena todas las especies de hongo que existen actualmente en la simulación (véase la clase FungusSpecies en la subsección 5.2.6).
- ***existingAnimalSpecies*** – Es un vector de punteros a instancias de la clase AnimalSpecies que almacena todas las especies de animales que existen actualmente en la simulación (véase la clase AnimalSpecies en la subsección 5.2.2).

La clase World además implementa las siguientes funciones, relacionadas con la creación del terreno y la inserción de las especies dentro del mismo:

- ***World()*** – Es el constructor de la clase, que toma como entrada el archivo de configuración del nuevo mundo. Este método crea la instancia del mundo e inicializa las variables del mismo a partir del fichero de configuración especificado. Aquí se crean las celdas del terreno llamando a *initializeTerrain()* y se almacenan en el vector *terrain*. Después se instancian las especies, tanto de hongos como de animales, llamando respectivamente a las funciones *readFungusSpeciesFromJSON()* y *readAnimalSpeciesFromJSON()*, y almacenando dichas especies en los vectores *existingFungiSpecies* y *existingAnimalSpecies*. Tras haber leído la información relativa

a las especies y haber creado el terreno, se inicializan los animales y se establece su localización en las celdas de forma aleatoria, llamando a la función *initializeAnimals()*.

- ***initializeTerrain()*** – Es una función que se llama desde el constructor, y que se encarga de crear el terreno almacenando las *TerrainCell* instanciadas dentro del vector *terrain*. Al crear cada celda, se le asignan sus coordenadas y se inicializa su temperatura y su capacidad de hongo.
- ***readFungusSpeciesFromJSON()*** – Es una función que se llama desde el constructor, y que se encarga de realizar la lectura de las especies de hongo desde el archivo de configuración. Cada especie de hongo *FungusSpecies* es almacenada en el vector *existingFungiSpecies*.
- ***readAnimalSpeciesFromJSON()*** – Es una función que se llama desde el constructor, y que se encarga de realizar la lectura de las especies de hongo desde el archivo de configuración. Cada especie de hongo *FungusSpecies* es almacenada en el vector *existingFungiSpecies*.
- ***initializeAnimals()*** – Es una función que se llama desde el constructor, y que se encarga de realizar la lectura de las especies de animal desde el archivo de configuración. Cada especie de animal *AnimalSpecies* es almacenada en el vector *existingAnimalSpecies*. Tras haber leído todas las especies y haber calculado la densidad de población de cada una, esta función realiza el cálculo del número de especies de cada animal en cada instar, utilizando la densidad total de población y llamando a la función *scaleInitialPopulation()* (véase la clase *AnimalSpecies* en la subsección [5.2.2](#)).
- ***evolveWorld()*** – Esta función podría ser considerada la función principal del programa. Previamente inicializado el mundo, esta función se encarga de hacerlo evolucionar. Así, se ocupa de activar todas las funciones en todas las celdas *TerrainCell* del espacio, de tal forma que los animales ejecutan su movimiento, alimentación, metabolismo, crecimiento y reproducción, y las fuentes de hongo son actualizadas con la cantidad de biomasa correspondiente en cada paso (véase la clase *TerrainCell* en la subsección [5.2.9](#)).

6 | Pruebas y resultados

6.1. Primeras pruebas de estabilidad

Se ha llevado a cabo la ejecución de varias simulaciones con el objetivo de medir la eficiencia del algoritmo secuencial para distintos parámetros. El experimento se ha realizado sobre un procesador *Intel(R) Core(TM) i7 860* con 2.80GHz y 8GB de memoria RAM. Las simulaciones se han preparado siguiendo un patrón de escala para el cual el sistema mantiene en gran medida su estabilidad, es decir, las distintas especies siguen sobreviviendo el mayor tiempo posible.

Esta estabilidad de la que hablábamos es un tema muy importante, ya que permite analizar conjuntos de población a lo largo de varias generaciones. Una mayor longevidad en las simulaciones permite que se puedan obtener datos acerca de la evolución de la genética en función de las dinámicas ecológicas que haya sufrido la comunidad, como pueden ser el número de encuentros, la cantidad de individuos de las distintas especies o su posicionamiento dentro del espacio, entre otros factores.

En la tabla 6.1 se puede ver el tiempo de ejecución obtenido para las diferentes configuraciones de los parámetros de entrada, donde P indica el número de presas iniciales, D es el número de depredadores y C es el número de celdas del mundo. Estas pruebas se han realizado utilizando una sola dimensión del espacio, una sola especie de depredador y una sola especie de presa. Los tiempos de ejecución se representan en segundos y aparecen en las columnas T_{100} , T_{200} y T_{300} que corresponden a la simulación de 100, 200 y 300 días, respectivamente.

TABLA 6.1: Resultados de las pruebas de estabilidad. Tiempos de ejecución según los parámetros indicados.

P	D	C	T_{100}	T_{200}	T_{300}
500	50	100	9.013	32.966	61.904
1000	100	200	19.965	64.372	142.745
2000	200	400	37.134	121.607	228.322
4000	400	800	70.756	260.857	515.189

Los resultados obtenidos indican que la complejidad del sistema crece linealmente al incrementar el número de individuos y el espacio de la simulación. Este es un gran inconveniente para obtener resultados interesantes desde el punto de vista ecológico, ya que para ello se necesitará aumentar los valores de los parámetros iniciales. Además, el elevado número de parámetros configurables hace que el número de simulaciones a realizar pueda ser muy elevado, con el objetivo de encontrar los valores más adecuados para la estabilidad de la comunidad. Por estos motivos se propone la paralelización del sistema como trabajo futuro (véase la sección 7.4).

6.2. Ejemplo sencillo de simulación

En esta sección se muestra el proceso seguido para la ejecución de un ejemplo sencillo de simulación, y la obtención de las salidas de datos necesarias para analizar lo sucedido durante la misma. Todos los archivos utilizados se encuentran en un disco entregado junto con este documento, para su reproducción si se desea (usar para ello una máquina con al menos 32GB de memoria RAM y 5GB de disco libres, para asegurar que la simulación finaliza correctamente).

En el apéndice A se puede consultar el manual de usuario de Weaver, donde se explican todos los pasos necesarios para llevar a cabo una simulación utilizando este programa. Allí se especifica el formato que debe tener el archivo de configuración `run_params` y qué parámetros debe incorporar. Esta sección se limita entonces a explicar cómo se obtienen y evalúan las salidas de un ejemplo sencillo de simulación.

En el apéndice B se puede consultar el ejemplo completo de archivo de configuración `run_params` que se ha utilizado para obtener las salidas que se van a mostrar a continuación. Este archivo de configuración es uno de los que han sido utilizados en la publicación *Eco-Evolutionary Spatial Dynamics: Rapid Evolution and Isolation Explain Food Web Persistence*[19]. Por lo tanto, los valores utilizados en todos los parámetros se rigen según los valores eco-evolutivos ya explicados.

El archivo contiene la definición de 20 especies animales repartidas a lo largo del mundo simulado. Se incorpora una única especie de hongo, que se sitúa en cuatro islas separadas y colocadas en un terreno en 3 dimensiones de $20 \times 20 \times 10$ celdas. Siguiendo los pasos descritos en la sección A.1 del manual de Weaver, se procede a realizar la simulación hasta que su ejecución finaliza y obtenemos los archivos de salida descritos en la sección A.3.

En la figura 6.1 se muestra el sumario extendido que se ha obtenido de esta simulación. Como se puede observar en la figura, el sumario contiene cierta información en forma de tabla acerca del número de animales que existen en cada estado, separándolos por cada especie animal. También muestra la cantidad de biomasa total existente para cada especie de hongo. Esta tabla puede ser tratada utilizando otras herramientas, como por ejemplo el lenguaje de programación R, con el objetivo de generar datos estadísticos que ayuden a discernir o exponer los comportamientos ecológicos de la población.

Otra de las salidas importantes a analizar es aquella que proporciona información sobre los individuos existentes durante cada día de la simulación. Tal y como se explica en la sección A.3 del manual de Weaver, el directorio `animals_each_day_end` contiene una lista de archivos cuya nomenclatura sigue la pauta `animals_day_X.txt` donde **X** indica un día de simulación concreto. Cada archivo contiene toda la información acerca del estado actual de cada animal presente en la simulación **al finalizar el día** concreto. La figura 6.2 muestra uno de estos archivos, que ha sido partido en tres partes para poder mostrar todos los campos que sobrepasan el margen horizontal. En la figura se puede observar cierta información como es el identificador del individuo (**id**), la especie a la que pertenece (**species**), las coordenadas en las que se encuentra (**x**, **y**, **z**), el estado de vida en que se encuentra el individuo (**state**), el instar que ha alcanzado hasta la fecha (**instar**), el valor inicial que obtuvo para su fenología (**pheno_ini**), la fecha en la que apareció en la simulación (**date_egg**), la fecha de su primera reproducción (**age_first_rep**), el número de

day	Fungus X biomass	acal_0	acal_1	acal_2	acal_3	acal_4	acal_5	acal_6	aca2_0	aca2_1	aca2_2	aca2_3	aca2_4
0	3064.5	954	214	0	684	24	0	0	703	214	0	510	12
1	3046.17	1009	97	0	119	22	0	0	871	114	0	105	7
2	3058.6	1227	64	2	49	5	0	0	1025	78	5	36	3
3	3061.66	1218	37	8	25	8	0	0	1018	52	7	22	7
4	3061.01	1193	56	8	47	7	0	0	982	60	8	16	11
5	3056.83	922	229	3	95	7	0	0	854	175	3	42	9
6	3043.77	331	558	9	252	8	0	0	437	460	6	164	11
7	3042.68	191	493	30	174	9	0	0	412	518	35	121	7
8	3040.81	40	438	92	114	9	0	0	316	472	59	116	7
9	3043.62	18	219	109	126	15	0	0	502	293	128	73	11
10	3038.56	37	147	44	44	17	0	0	594	168	73	49	18
11	3037.14	55	98	21	28	17	0	0	606	135	37	15	34
12	3035.35	72	69	13	14	19	0	0	970	149	16	28	24
13	3027.59	126	49	12	12	15	0	0	1004	223	7	42	32
14	3023.53	126	24	5	14	21	0	0	1381	242	13	90	25
15	3016.23	123	24	1	1	22	0	0	1615	271	27	85	26
16	3009.45	90	45	4	6	24	0	0	1782	302	17	48	40
17	3008.52	153	75	4	1	13	0	0	2107	370	29	76	26
18	2994.86	130	95	4	6	16	0	0	2043	507	31	134	41
19	2988.35	108	93	10	21	19	0	0	2427	511	28	148	54
20	2975.65	107	61	22	9	21	0	0	2439	616	41	232	71
21	2971.71	157	47	15	6	15	0	0	2850	628	38	363	76
22	2980.05	141	53	8	4	20	0	0	3450	606	25	323	108
23	2972.48	73	104	7	7	23	0	0	5069	702	33	296	59
24	2953.25	67	94	1	11	27	0	0	6314	663	31	374	65
25	2954.48	67	71	11	22	28	0	0	8014	742	22	375	75
26	2946.6	101	51	20	17	23	0	0	9976	729	10	489	71
27	2940.32	64	40	10	36	25	0	0	11436	925	21	601	77

FIGURA 6.1: Ejemplo de sumario extendido que resume los datos sobre población durante la simulación

veces que se ha reproducido (**rep_count**), su fecundidad o número de descendientes hasta la fecha (**fecundity**), la fecha de su muerte (**date_death**), el número de generación que le corresponde según cada uno de sus padres (**g_numbrprt1** y **g_numbrprt2**), el identificador de cada uno de sus padres (**ID_prt1** e **ID_prt2**), el número de encuentros que ha tenido con depredadores en el día actual (**encounters_pred**), el número total de encuentros que ha tenido con depredadores a lo largo de su vida (**global_pred_encs**), el número de días que le quedan para finalizar la digestión (**days_digest**) y todas las características actuales del animal que vienen asociadas desde los rasgos genéticos y han sido modificadas de alguna forma hasta el día actual (**energy**, **growth**, **pheno**, **body**, **assim**, **vor**, **speed search**, **met**, **vorQ10**, **spdQ10**, **srchQ10**, **e_met**). Un valor -1 en cualquier campo significa que el valor no existe para ese animal. Por ejemplo, los animales que aún no han muerto tienen su *date_death* a -1. A partir de esta información diaria sobre los animales, se pueden crear scripts en cualquier lenguaje de tratamiento de datos que accedan a esta información y la resuman en forma de tablas o gráficas. En la sección 6.3 se pueden observar algunas de estas gráficas, obtenidas utilizando el lenguaje de programación R. Es importante notar que, al tener acceso a absolutamente toda la

información sobre todo lo que ha ocurrido cada día, existen muchas posibilidades para el tratamiento de esta información. De hecho, esto puede suponer un contratiempo, y es que tratar cantidades tan grandes de información puede suponer un verdadero reto. En la sección [7.5](#) se propone la optimización de estas salidas de datos como trabajo futuro.

id	species	x	y	z	state	instar	pheno_ini	date_egg	age_first_rep	rep_count	fecundity	date_death	g_numbprt1	g_numbpr
4654	aca4	6	0	0	1	1	8.54129	0	-1	0	0	-1	0	
23337	aca6	13	0	0	1	1	6.49589	3	-1	0	0	-1	1	
7092	aca6	17	0	0	2	1	8.73034	0	-1	0	0	10	0	
344	aca1	4	1	0	2	1	8.1472	0	-1	0	0	10	0	
20332	aca6	8	1	0	1	4	7.45553	1	-1	0	0	-1	1	
630	aca1	2	2	0	1	4	7.41542	0	-1	0	0	-1	0	
811	aca1	2	2	0	2	1	7.4139	0	-1	0	0	10	0	
225	aca1	6	2	0	2	1	7.72068	0	-1	0	0	10	0	
7911	aca6	10	2	0	1	4	6.82554	-1	8	1	6	-1	0	
20977	aca5	12	2	0	1	1	8.67438	1	-1	0	0	-1	1	
23274	aca6	12	2	0	1	1	5.91197	3	-1	0	0	-1	1	
7518	aca6	4	3	0	3	2	7.19084	0	-1	0	0	10	0	
22416	aca2	5	3	0	1	1	8.16014	2	-1	0	0	-1	1	
1962	aca2	15	3	0	2	1	8.82402	0	-1	0	0	10	0	
346	aca1	15	3	0	2	1	9.44133	0	-1	0	0	10	0	
23250	aca6	17	3	0	1	1	7.08052	3	-1	0	0	-1	1	
23141	aca6	17	3	0	1	1	7.46131	3	-1	0	0	-1	1	
23180	aca6	19	3	0	1	2	5.90575	3	-1	0	0	-1	1	
6090	aca5	19	3	0	3	1	7.56566	0	-1	0	0	10	0	
3715	aca3	2	4	0	1	1	7.04798	0	-1	0	0	-1	0	
348	aca1	8	4	0	1	4	7.53396	0	-1	0	0	-1	0	
g_numbprt1	g_numbprt2	IDprt1	IDprt2	encounters	pred	global_pred	encs	days_digest	energy	growth	pheno	body		
0	0	-1	-1	0	0	1	0	0.00688505	1.40629	7	0.00534			
1	1	8420	8465	0	0	1	0	0.000351025	1.39575	8	0.00213			
0	0	-1	-1	0	0	0	0	0.000147338	1.40192	7	0.0024			
0	0	-1	-1	0	0	0	0	0.000136442	1.41727	6	0.00194			
1	1	8376	8427	0	0	1	0	0.0124751	1.42304	7	0.0139			
0	0	-1	-1	0	0	1	0	0.00121036	1.4043	6	0.00423			
0	0	-1	-1	0	0	0	0	8.77052e-05	1.40336	6	0.00206			
0	0	-1	-1	0	0	2	0	0.811734e-05	1.39976	6	0.00167			
0	0	-1	-1	0	0	2	0	0.0022015	1.4043	-1	0.0127			
1	1	6864	6893	0	0	1	0	0.00060315	1.42092	8	0.00319			
1	1	8396	8399	0	0	0	0	0.000295177	1.41466	8	0.00244			
0	0	-1	-1	0	0	1	0	0.00297912	1.39865	6	0.0059			
1	1	3301	3308	1	1	1	0	0.000618159	1.36941	8	0.00303			
0	0	-1	-1	0	0	4	0	0.000110226	1.40308	7	0.00281			
0	0	-1	-1	0	0	0	0	0.000182734	1.3906	7	0.00183			
1	1	8402	8375	0	0	0	0	0.000436058	1.38775	9	0.0022			
1	1	8394	8367	0	0	0	0	0.000439862	1.40537	9	0.00252			
1	1	8388	8373	0	0	0	0	0.0034864	1.38093	8	0.00566			
0	0	-1	-1	0	0	7	0	0.000588358	1.38476	6	0.00326			
0	0	-1	-1	0	0	0	0	0.009575	1.402	6	0.00439			
0	0	-1	-1	0	0	0	0	0.00133494	1.40142	6	0.00456			
growth	pheno	body	assim	vor	speed	search	met	vorQ10	spdQ10	srchQ10	e_met			
1.40629	7	0.00534149	0.735753	0.0178294	1.37358	1	0.727999	3.65354	2.04207	2.20421	0.426027			
1.39575	8	0.00213154	0.805935	0.00952414	1.80528	3	0.73556	3.45039	1.76954	2.28873	0.409387			
1.40192	7	0.0024377	0.795327	0.0101341	1.86325	3	0.704253	3.42764	1.72158	2.17873	0.425921			
1.41727	6	0.00194366	0.792377	0.00713453	1.83779	3	0.712161	3.32926	1.88959	2.21964	0.346513			
1.42304	7	0.0139703	0.714112	0.0304638	2.82162	4	0.690798	3.59582	1.85798	2.0914	0.415322			
1.4043	6	0.00423949	0.791063	0.0136171	1.95772	3	0.668834	3.57395	2.00321	2.26084	0.389633			
1.40336	6	0.00206504	0.783065	0.00824198	2.00319	3	0.69399	3.34409	2.11865	2.32909	0.366359			
1.39976	6	0.00167098	0.795858	0.00834762	1.87509	3	0.753452	3.61003	2.01383	2.17518	0.359985			
1.4043	-1	0.0127225	0.824624	0.0282919	2.48322	3	0.645079	3.45948	1.90753	2.31409	0.404265			
1.42092	8	0.00319517	0.775718	0.00971866	2.17775	3	0.723246	3.40266	2.20907	2.31355	0.376431			
1.41466	8	0.00244921	0.787321	0.00839323	1.91901	3	0.674313	3.35049	1.88179	2.32568	0.410242			
1.39865	6	0.0059267	0.762947	0.0189561	2.20665	3	0.669465	3.49164	2.10086	2.1658	0.407751			
1.36941	8	0.00303305	0.817324	0.011909	1.99437	3	0.715719	3.74396	1.9737	2.21461	0.412079			
1.40308	7	0.00281376	0.72349	0.00916955	2.09291	3	0.701279	3.44189	2.02315	2.26144	0.406833			
1.3906	7	0.00183524	0.826412	0.00672928	1.98749	3	0.732227	3.24878	2.25039	2.25185	0.373955			
1.38775	9	0.0022815	0.782299	0.00905856	1.79966	3	0.680384	3.64808	1.76584	2.25149	0.404115			
1.40537	9	0.00252166	0.827523	0.00913002	1.79765	3	0.725856	3.54502	1.66706	2.25907	0.424294			
1.38093	8	0.00568007	0.820596	0.00941745	1.87466	3	0.667809	3.64545	1.70628	2.33601	0.385862			
1.38476	6	0.00328738	0.849858	0.0115849	2.12275	3	0.728238	3.4274	2.00897	2.14718	0.365492			
1.402	6	0.00439045	0.767932	0.014081	0.251761	0	0.701698	3.38055	2.19059	2.36433	0.377911			
1.40142	6	0.00456942	0.79384	0.0143074	2.15695	3	0.693009	3.46662	1.87635	2.26045	0.356628			

FIGURA 6.2: Ejemplo de salida diaria con información de los animales

Un archivo que proporciona información también muy importante es el llamado *animal_constitutive_traits.txt*. Este archivo contiene información similar los 13 últimos campos que se han mostrado en el archivo con información diaria de los animales, pero solo almacena los valores que tenía el animal para sus 13 rasgos justo en el día de su nacimiento. Esta salida es muy útil para realizar investigación acerca de la deriva genética de las poblaciones.

Por último, cabe destacar una de las salidas más relevantes: aquella que proporciona la información sobre las depredaciones ocurridas entre las diferentes especies animales. El archivo *predationOnSpecies* que se encuentra en el directorio *Matrices* almacena una matriz $N \times N$ donde N es el número de especies animales presentes en la simulación. Los valores de esta matriz representan el número de depredaciones que una especie animal (eje de nombres vertical) ha efectuado sobre otra especie animal (eje de nombres horizontal). En la figura 6.3 se muestra el ejemplo de matriz obtenido tras realizar la simulación. Esta información es muy importante para evaluar cómo se comportan las poblaciones de individuos de las distintas especies, dependiendo del ambiente establecido para cada simulación (véase la sección 6.3).

	aca1	aca2	aca3	aca4	aca5	aca6	ara1	ara2	ara3	ara4	ara5	ara6	col1	col2	enc1	geo1	geo2	lit1	lit2	ori1
aca1	29458	19802	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
aca2	0	159189	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	83860
aca3	0	0	0	249	0	0	0	0	0	0	11	0	93548	29	0	0	0	0	0	0
aca4	293	231	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
aca5	0	0	0	0	0	0	0	0	0	0	19	0	247	25	0	0	0	0	0	0
aca6	0	0	0	0	446	0	59	0	0	0	0	0	0	0	0	0	0	0	2	0
ara1	249	234	0	103	0	0	0	0	0	0	14	0	138	43	0	0	0	0	0	264
ara2	0	0	0	152	0	0	0	0	0	0	26	0	215	41	648	0	0	0	0	0
ara3	0	0	0	79	0	0	0	0	40	143	40	0	152	57	307	0	0	0	0	0
ara4	307	0	0	132	0	0	0	0	0	0	33	0	208	70	469	0	0	0	0	0
ara5	216	167	0	74	0	0	0	0	0	0	0	0	0	27	0	0	0	0	0	188
ara6	0	0	825	0	166	277	15	204	5	42	0	717	608	0	256	1	0	2	3	0
col1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
col2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
enc1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
geo1	0	0	299	0	262	0	87	235	64	81	19	443	483	0	536	37	0	63	66	0
geo2	820	0	1511	185	344	651	76	342	37	75	17	664	3397	136	1073	87	423	103	76	0
lit1	0	0	208	0	248	444	57	185	33	73	21	375	293	0	411	38	0	42	68	0
lit2	0	0	268	0	231	412	64	198	49	72	0	442	0	0	401	31	129	44	72	0
ori1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FIGURA 6.3: Ejemplo de matriz de depredación entre especies

6.3. Resumen de resultados eco-evolutivos obtenidos

En esta sección se resumen una serie de resultados eco-evolutivos obtenidos utilizando el programa Weaver que se ha desarrollado en este proyecto fin de carrera. Es importante notar que todos los resultados siguientes son solamente un breve resumen de lo que se ha desarrollado en la publicación *Eco-Evolutionary Spatial Dynamics: Rapid Evolution and Isolation Explain Food Web Persistence*[19]. En las siguientes subsecciones, primero se van a plantear varios ejemplos de hipótesis cuya verificación puede ser completada mediante Weaver, y después se listarán algunos de los resultados que se han obtenido adjuntando sus correspondientes gráficas.

6.3.1. Ejemplos de hipótesis

Weaver es un programa que pretende salvar la frontera entre: las observaciones empíricas de redes tróficas basadas en individuos en espacios heterogéneos; y las predicciones teóricas provenientes de las dinámicas eco-evolutivas de metacomunidades multi-tróficas. El propósito general de las simulaciones que se han realizado en [19] es el de ilustrar la utilidad de este *framework* por medio de varios ejemplos, apuntando algunas de las cuestiones por resolver en el campo de las dinámicas eco-evolutivas y las redes ecológicas en el espacio. Se proponen seis cuestiones:

- **Cuestión 1** – ¿Afecta la conectancia a la persistencia de una red trófica? (subsección 6.3.2).
- **Cuestión 2** – La variación genética intraespecífica, ¿altera la persistencia de una red trófica mediante la activación de una evolución rápida? (subsección 6.3.3).
- **Cuestión 3** – En un contexto de metacomunidad multi-trófica, ¿de qué forma altera la estructura espacial (es decir, el grado de aislamiento en el espacio) a la persistencia y a la evolución rápida en las redes tróficas? (subsección 6.3.4).
- **Cuestión 4** – En todos los escenarios anteriores, ¿inhiben los depredadores a las poblaciones de presas en sistemas de evolución rápida depredador-presa? y si esto es así, ¿contribuyen los depredadores a mantener la diversidad de presas? (subsección 6.3.5).

- **Cuestión 5** – ¿En qué medida la variación genética altera el ratio evolutivo y la persistencia de sistemas depredador-presa? (subsecciones 6.3.6 y 6.3.7).
- **Cuestión 6** – ¿Evolucionan los rasgos de forma diferente en presas bajo ambientes puramente competitivos (sin depredadores) y en presas bajo ambientes donde la depredación y la competitividad entran en juego? (subsección 6.3.8).

Las seis cuestiones citadas anteriormente se extienden brevemente en las siguientes subsecciones. Está claro que los resultados obtenidos mediante esta herramienta son meramente aproximaciones a la naturaleza. Para aprovechar totalmente estas simulaciones, los resultados deben ser probados en sistemas reales y retroalimentar ambos sistemas iterativamente. Sin embargo, todavía se pueden proveer pruebas e hipótesis importantes utilizando únicamente esta plataforma digital, los cuales podrían ser verificados en futuros experimentos de campo.

6.3.2. Conectancia y persistencia de la red trófica

Una de las características más importantes de Weaver es que puede restringir qué especies son capaces de alimentarse de otras especies, simplemente indicándolo en un vector de especies comestibles para cada especie animal. Esto permite inicializar simulaciones con casi cualquier estructura de red trófica. Para manipular la conectancia (es decir, como de estrecha es la relación de depredación entre especies) y para poder simular redes tróficas realistas, se ha usado para la publicación [19] el programa Network3D para construir dos redes de 20 especies restringidas al modelo de nicho [20]: una con conectancia baja (0.1) y otra con conectancia relativamente alta (0.3). Además, para poder realizar las pruebas más eficientemente, se ha añadido otra red trófica hipergeneralista, en la cual todos los depredadores son capaces de depredar a todas las especies, incluidos a ellos mismos, resultando en una conectividad de la red muy alta (0.55). Las redes tróficas de conectividad 0.1 y 0.55 tienen exactamente el mismo número de especies: 11 especies de depredadores y 9 especies de fungívoros. Para ajustar la red de conectividad 0.3 al modelo de nicho, ésta se ha desarrollado con un número diferente de especies: 16 especies de depredadores y 4 especies de fungívoros. Por lo tanto, la red de conectividad 0.55 contiene las mismas especies que la de conectividad 0.1, pero estas han sido forzadas a depredar a todas las especies (hipergeneralistas) para aumentar la conectividad. Por

simplicidad, solamente se incluye una especie de hongo a partir de la cual se alimentan todos los fungívoros.

Para incorporar el componente espacial, se ha simulado un escenario de cuatro microislas imitando bolsas de agua en el suelo del bosque, en las cuales el hongo está capacitado para crecer. El espacio rodeando a estas islas es seco, y todas las islas tienen la misma productividad basal.

En la figura 6.4 se muestran las diferentes redes tróficas según su conectividad, y las estructuras espaciales de metacomunidad que se han incluido en las simulaciones.

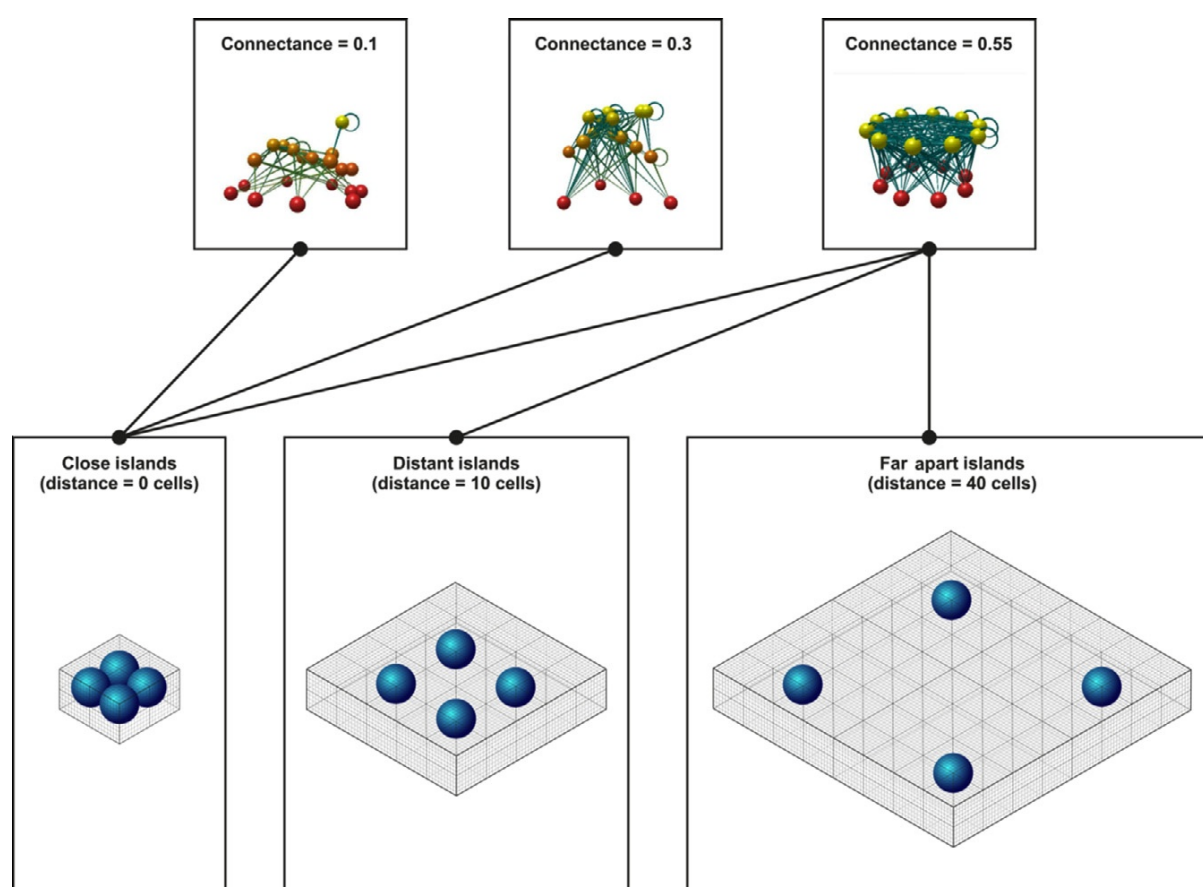


FIGURA 6.4: Redes tróficas y estructuras espaciales de metacomunidad incluidas en las simulaciones. Las líneas que unen los gráficos indican qué red trófica se ha probado en qué estructura espacial de metacomunidad. Las esferas azules del panel inferior corresponden a microislas de humedad (bolsas de humedad) en el suelo del bosque.

Un ejemplo de esta dinámica se puede observar en la figura 6.5, en la cual se muestran las dinámicas de población de cada especie (un mínimo de 20 especies) durante 200 días de simulación.

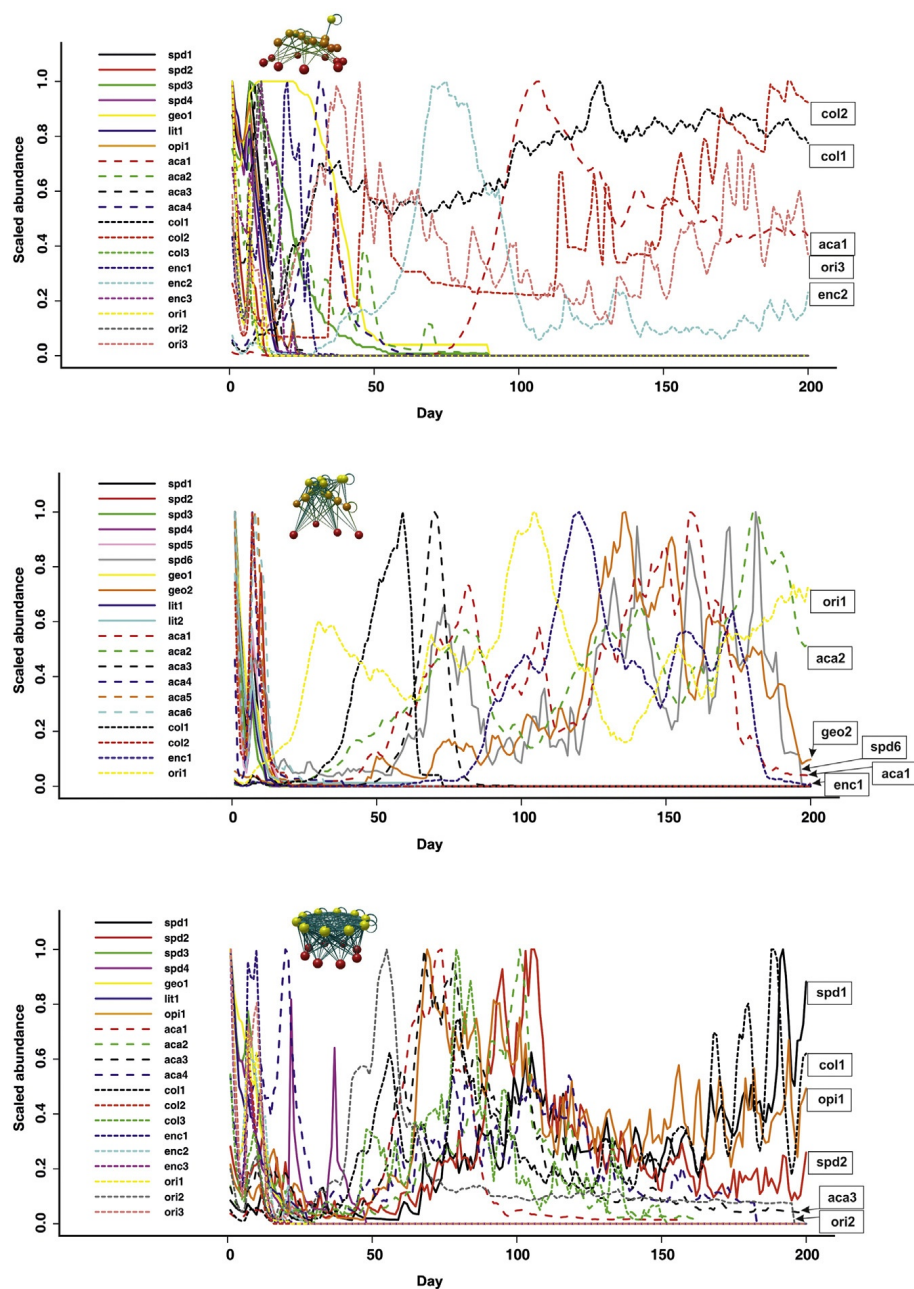


FIGURA 6.5: Dinámicas de población de cada especie usando distintas conectancias. El gráfico superior corresponde a la red trófica con conectancia baja (0.1), el central a la red con conectancia relativamente alta (0.3), y el inferior a la red trófica con conectancia muy alta (0.55). Cada gráfico solamente representa una réplica de las cinco que se han efectuado de cada simulación. La abundancia de cada especie (eje Y) se muestra en una escala 0-1. Las líneas punteadas corresponden a fungívoros (ácaros oribátidos, colémbolos y gusanos), las líneas discontinuas a depredadores pequeños (ácaros depredadores) y las líneas sólidas a depredadores grandes (arañas, opiliones y ciempiés). Los códigos de la derecha facilitan la identificación de cada especie en impresos en blanco y negro. La figura ha sido extraída del artículo publicado [19].

La conectancia ha incrementado la proporción de especies que sobreviven al final de la simulación en 1.7 veces. Este efecto es sobre todo visible en los depredadores. Sin embargo, una conectancia mayor también incrementa el ratio de extinción de las presas [19].

6.3.3. Variación genética y persistencia de la red trófica

Un ejemplo de estas dinámicas de puede observar en la figura 6.6, en la cual se muestran las dinámicas de población de cada especie durante 200 días cuando la conectancia de la red trófica es muy alta (0.55), al utilizar distintas variaciones genéticas.

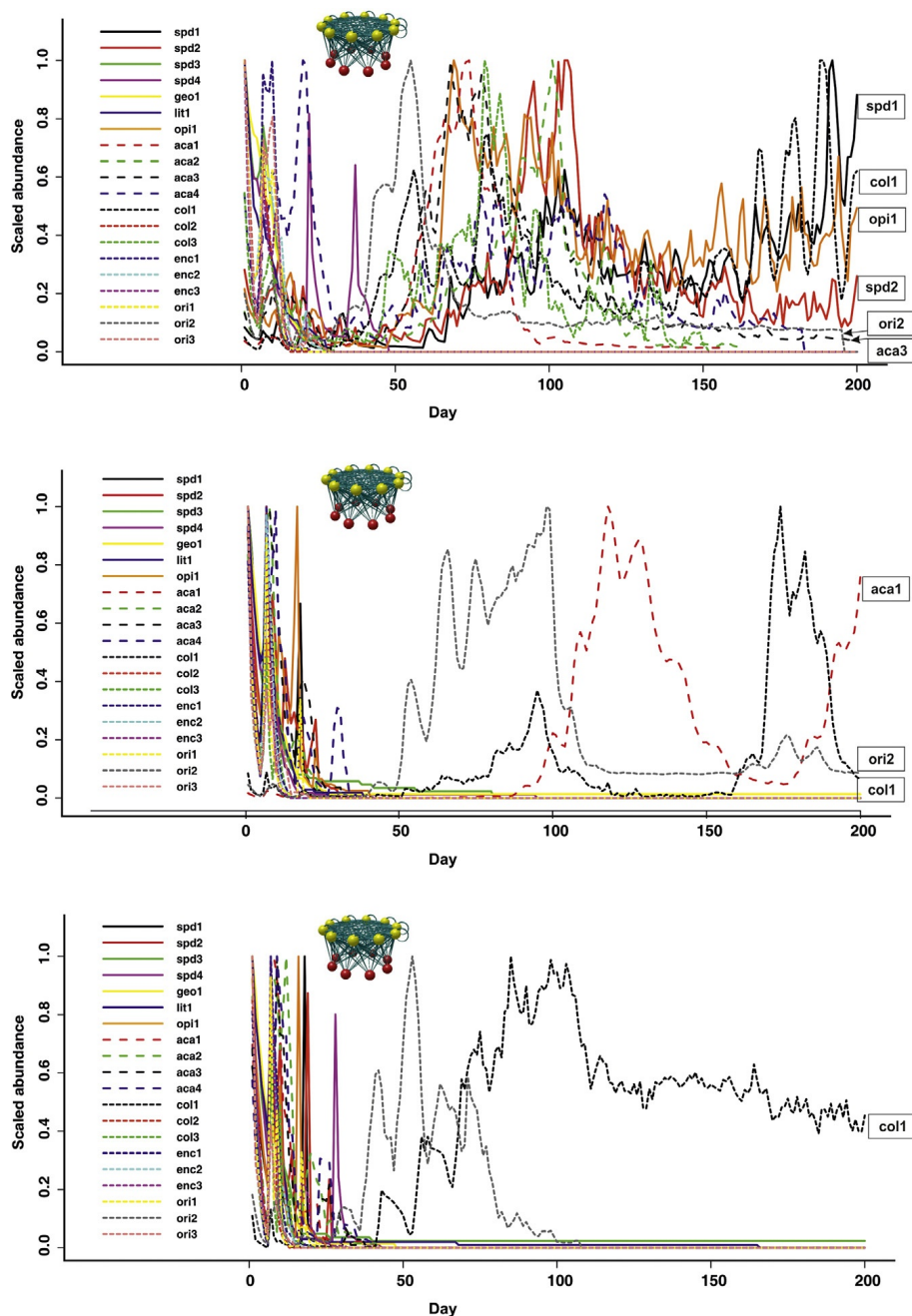


FIGURA 6.6: Dinámicas de población de cada especie al utilizar distintas variaciones genéticas. La figura refleja las dinámicas de población de cada especie durante 200 días cuando la conectancia de la red trófica es muy alta (0.55), y la variación genética era alta (panel superior), media (panel central) o casi nula (panel inferior). Cada gráfico solamente representa una réplica de las cinco que se han efectuado de cada simulación. La abundancia de cada especie (eje Y) se muestra en una escala 0-1. Las líneas punteadas corresponden a fungívoros (ácaros oribátidos, colémbolos y gusanos), las líneas discontinuas a depredadores pequeños (ácaros depredadores) y las líneas sólidas a depredadores grandes (arañas, opiliones y ciempiés). Los códigos de la derecha facilitan la identificación de cada especie en impresos en blanco y negro. La figura ha sido extraída del artículo publicado [19].

Estos gráficos revelan que una variación genética alta ha permitido la persistencia de 3.6 veces más especies que, o bien una variación genética nula, o a nivel intermedio [19].

6.3.4. Distancia entre islas y persistencia de la red trófica

Un ejemplo de estas dinámicas se puede observar en la figura 6.7, en la cual se muestran las dinámicas de población de cada especie durante 200 días cuando la conectancia de la red trófica es muy alta (0.55), con distintas distancias entre islas.

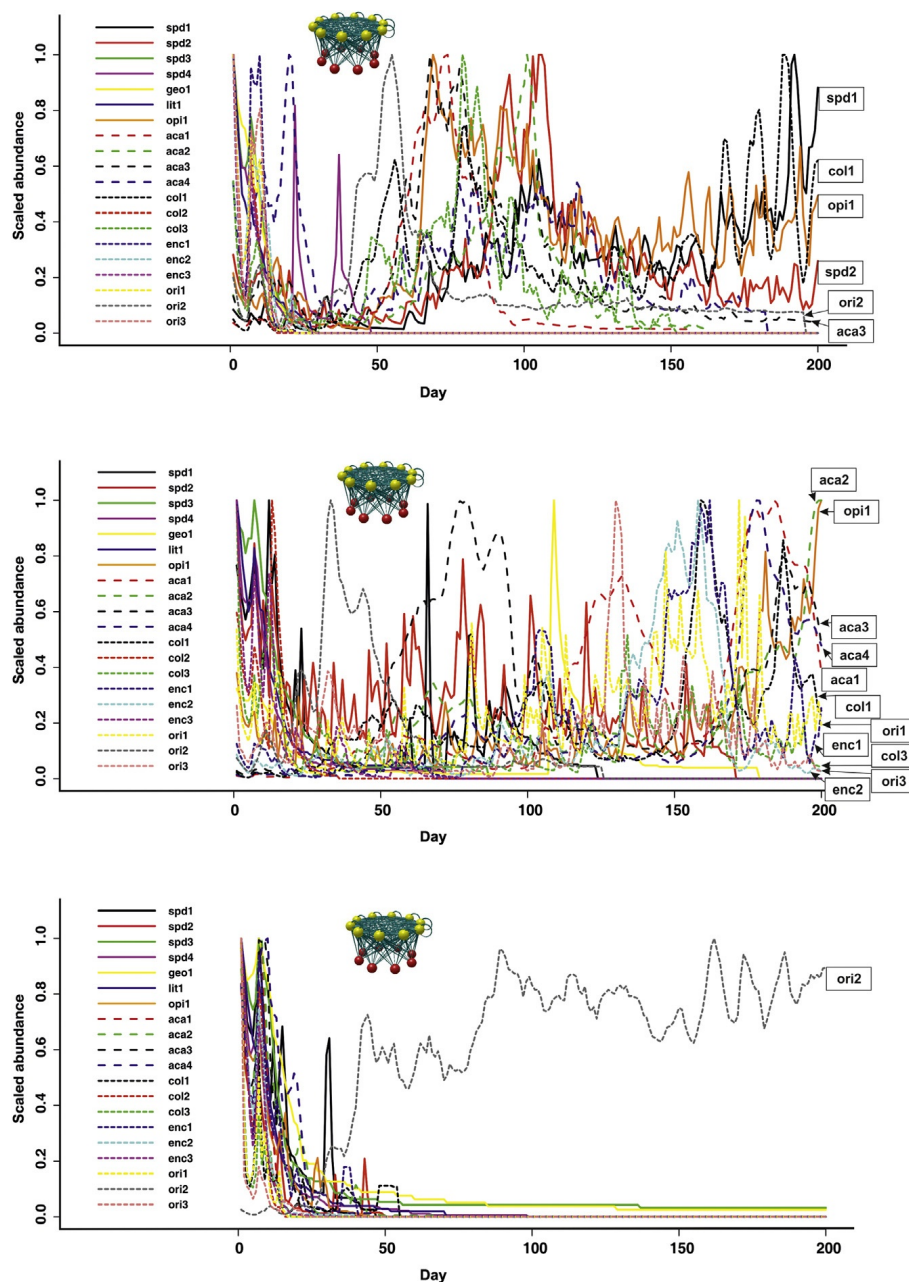


FIGURA 6.7: Dinámicas de población de cada especie con distintas distancias entre islas. La figura representa las dinámicas de población de cada especie durante 200 días cuando la conectancia de la red trófica es muy alta (0.55), pero las islas están separadas a diferentes distancias: 0 celdas (panel superior), 10 celdas (panel central) o 40 celdas (panel inferior). Cada gráfico solamente representa una réplica de las cinco que se han efectuado de cada simulación. La abundancia de cada especie (eje Y) se muestra en una escala 0-1. Las líneas punteadas corresponden a fungívoros (ácaros oribátidos, colémbolos y gusanos), las líneas discontinuas a depredadores pequeños (ácaros depredadores) y las líneas sólidas a depredadores grandes (arañas, opiliones y ciempiés). Los códigos de la derecha facilitan la identificación de cada especie en impresos en blanco y negro. La figura ha sido extraída del artículo publicado [19].

La proporción total de especies está entre 2.6 veces y 3.5 veces más persistentes en islas que están colocadas de forma contigua, y a una distancia relativamente corta (distancia mínima de 10 celdas), respectivamente, al contrario que cuando están más lejos (a 40 celdas). Además, la proporción de especies del tipo presa es 2-2.5 veces más persistente a distancias intermedias. Un total del 50 % de especies a sobrevivido al final de las simulaciones a distancias intermedias [19].

6.3.5. Control *top-down* de los depredadores sobre la diversidad de presas

Un ejemplo de estas dinámicas se puede observar en la figura 6.8, en la cual se muestran las dinámicas de población de cada especie durante 200 días cuando la conectancia de la red trófica es muy alta (0.55), dependiendo de la presencia o ausencia de depredadores.

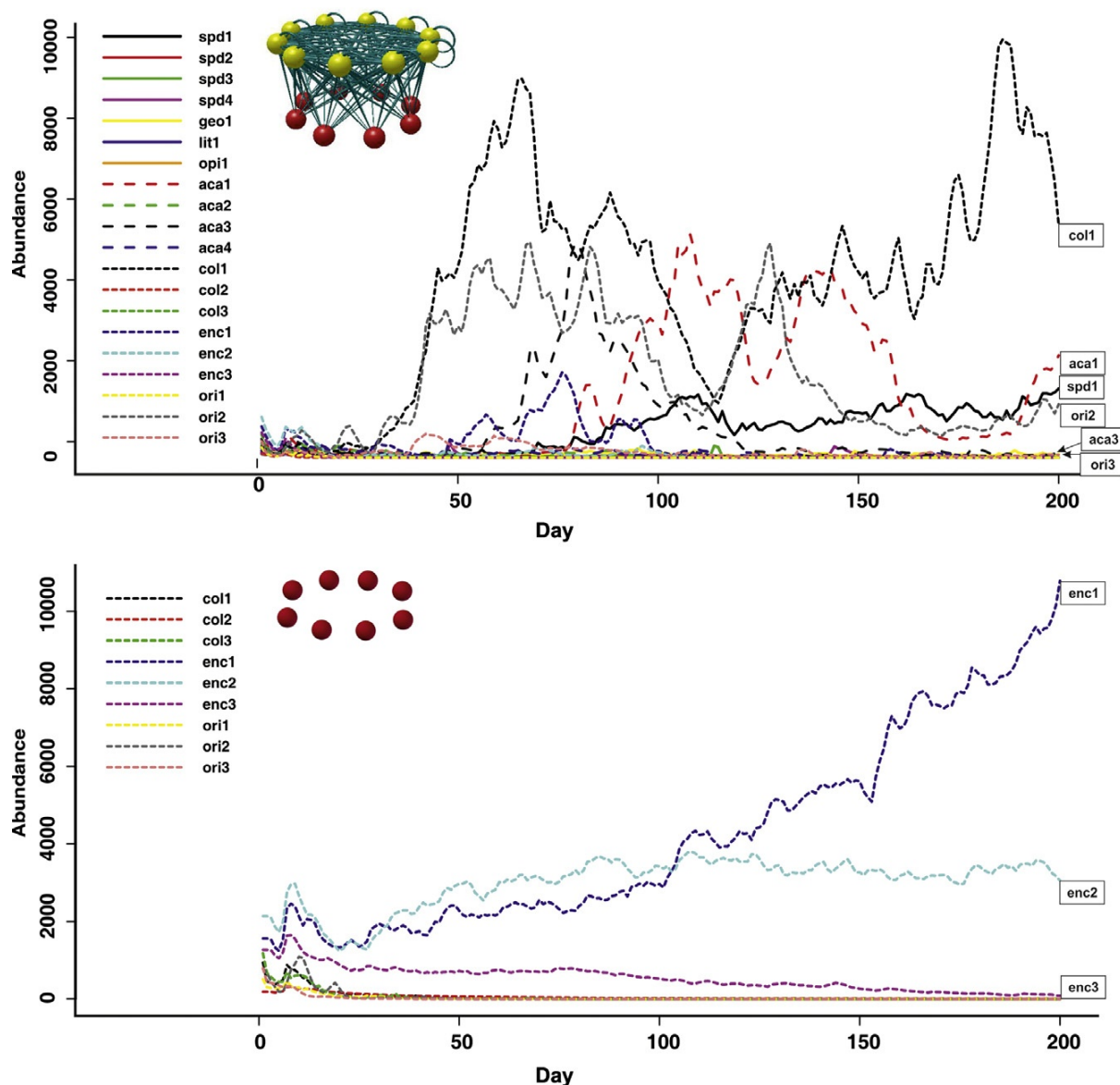


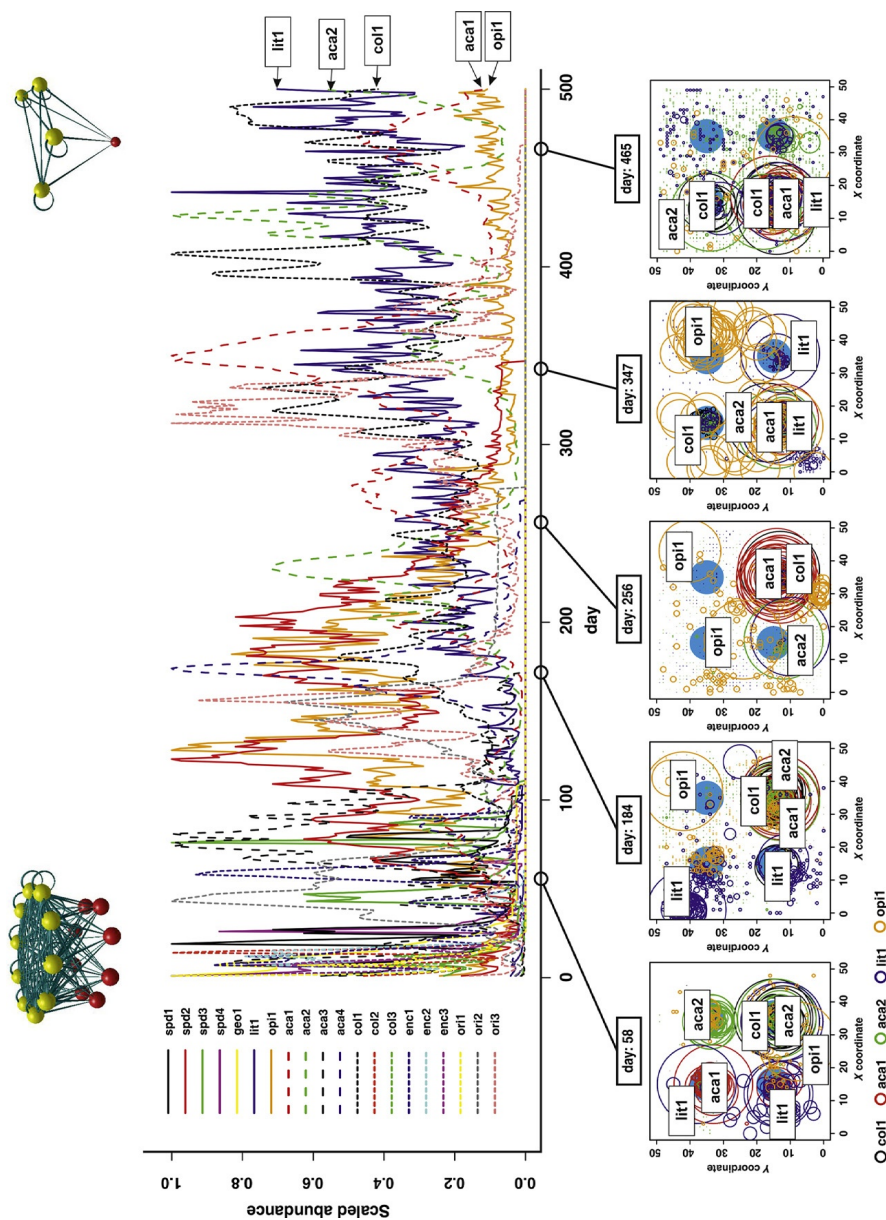
FIGURA 6.8: Dinámicas de población de cada especie dependiendo de la presencia o ausencia de depredadores. La figura representa las dinámicas de población de cada especie durante 200 días cuando la conectancia de la red trófica es muy alta (0.55), en la presencia (panel superior) o ausencia (panel inferior) de depredadores. Cada gráfico solamente representa una réplica de las cinco que se han efectuado de cada simulación. Las líneas punteadas corresponden a fungívoros (ácaros oribátidos, colémbolos y gusanos), las líneas discontinuas a depredadores pequeños (ácaros depredadores) y las líneas sólidas a depredadores grandes (arañas, opiliones y ciempiés). Los códigos de la derecha facilitan la identificación de cada especie en impresos en blanco y negro. La figura ha sido extraída del artículo publicado [19].

En general, la presencia de depredadores resulta en altos ratios de extinción de las presas. La figura 6.8 revela que la presencia de depredadores resulta en la extinción de dos especies de gusanos (panel superior) que llegan hasta el final de la simulación en un entorno competitivo sin depredadores (panel inferior). En particular, los depredadores muestran un fuerte efecto estabilizador cuando las islas se sitúan a distancias intermedias, en las cuales las especies de presas llegan hasta el final de la simulación. Y lo más importante es que los depredadores afectan directamente a la identidad de las especies que sobreviven [19].

6.3.6. Dinámicas multi-tróficas espacio-temporales durante una simulación de 500 días

Un ejemplo de estas dinámicas se puede observar en la figura 6.9, en la cual se muestran las dinámicas del escenario más estable para 500 días.

FIGURA 6.9: Dinámicas del escenario más estable para 500 días. La figura representa las dinámicas del escenario más estable: red trófica altamente conectada, con alta variabilidad genética y con islas a distancia intermedia (círculos azules -grises en la versión impresa- de los paneles inferiores) durante 500 días, cuando cinco especies (cuatro de ellas depredadoras y una presa) persisten hasta el final de la simulación (panel superior). El panel inferior muestra la fotografía de las dinámicas espaciales que están ocurriendo para esas cinco especies durante los 500 días. El tamaño de los círculos de igual color corresponde a la abundancia de una especie en esa zona en particular, relativa a la abundancia de esa misma especie en las otras zonas. En total, el experimento realizado en esta simulación ha incluido 8.891.887 individuos. Los códigos de la derecha y de las fotografías facilitan la identificación de cada especie en impresos en blanco y negro. La figura ha sido extraída del artículo publicado [19].



Como ya se ha indicado, la simulación ejecutada en el escenario más estable (alta conectividad de la red, alta variabilidad genética y distancia intermedia entre islas) durante un periodo de tiempo más largo (500 días) revela que, de 20 especies, 5 persisten (4 especies de depredadores y 1 de presas). Además de esto, se observa un incremento moderado de la abundancia del depredador más alto (el ciempiés lit1). Las dinámicas espaciales para estas cinco especies son más complejas y muestran algunos patrones emergentes:

- Primero, los depredadores grandes (ciempiés y opiliones) son altamente móviles en relación a los fungívoros y los depredadores pequeños.
- Segundo, la primera fotografía del terreno (día 58) muestra una segregación espacial emergente de dos especies de ácaros depredadores, que ocupan diferentes islas, y también un patrón emergente para la presa compartida (colémbolo), que solamente está presente en su mayoría en una isla.
- Tercero, en el día 256 las poblaciones de presas se extinguen en tres de las cuatro islas, y casi todos los individuos (depredadores y presas) se concentran en la isla en la cual el número de presas compartidas es todavía muy grande.
- En los días subsecuentes, el pico de las poblaciones de presas se observa en diferentes islas y entonces se extinguen en la originariamente más poblada de la que hablamos antes. Por lo tanto, las dinámicas espaciales tienen una gran influencia en los patrones de extinción y persistencia.

Durante la simulación anterior, también se han grabado todas las interacciones de alimentación y se ha construido una subred (figura 6.10) con la fuerza de esas interacciones.

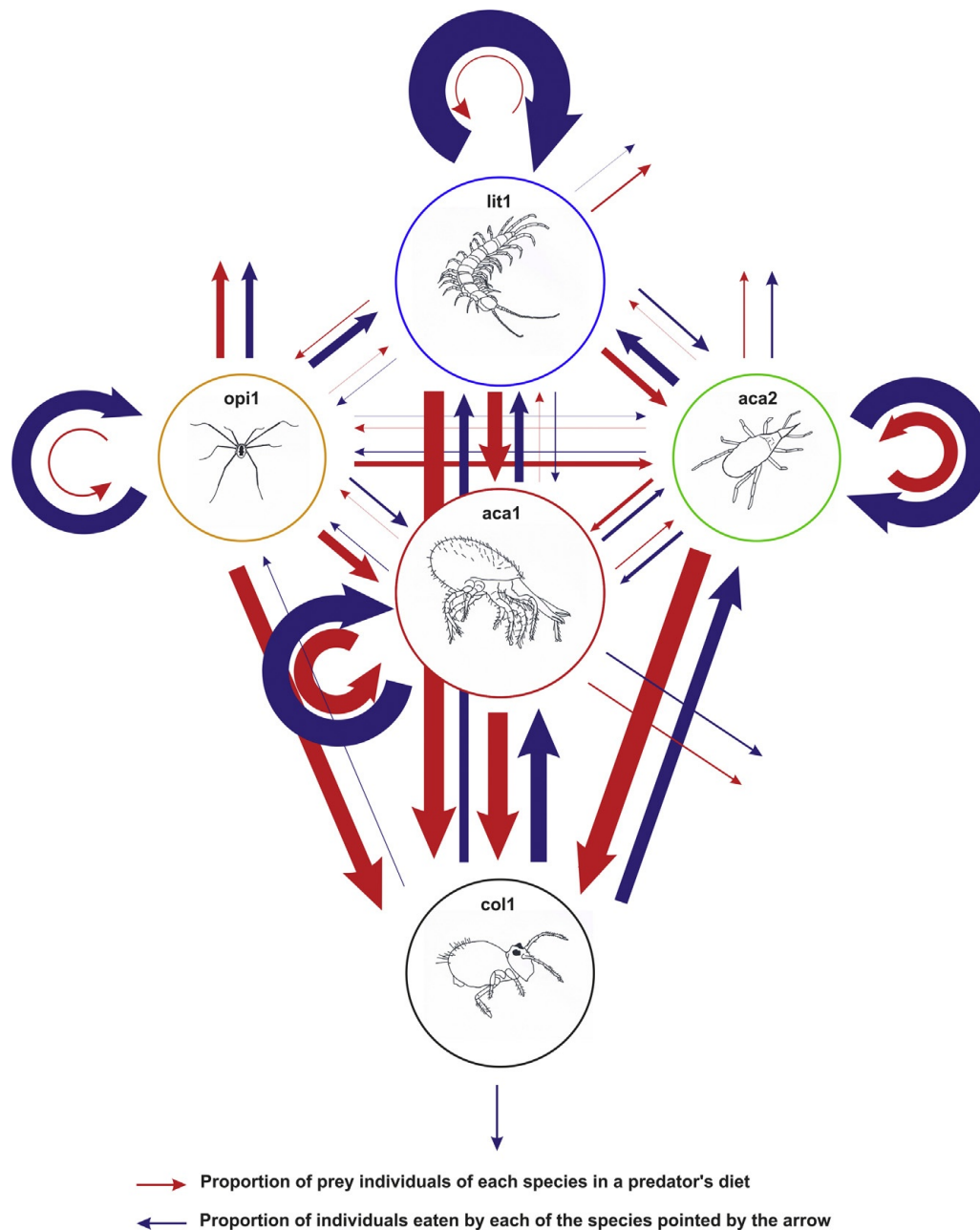


FIGURA 6.10: Fuerza de interacciones en una red persistente tras 500 días de simulación. La subred está definida en los dos sentidos: (a) la proporción de individuos de cada especie que ingiere cada especie depredadora (perspectiva del depredador, flechas rojas) y (b) la proporción de individuos de cada especie que es ingerida por cada especie depredadora (perspectiva de la presa, flechas azules). La figura ha sido extraída del artículo publicado [19].

En esta subred, el ciempiés es el depredador que más arriba se encuentra en la cadena, ya que interactúa con más fuerza al alimentarse de todas las otras especies. La mayoría de los depredadores basan su alimentación sobre todo en la presa compartida. El canibalismo entre las poblaciones de depredadores es en general elevado y las especies de depredador más pequeñas (los ácaros) interactúan con mucha menor fuerza con las demás especies que entre ellos mismos: en general, la depredación intragremio es más débil que el canibalismo [19].

6.3.7. Micro-evolución en 500 días de una red persistente

Un ejemplo de este efecto se puede observar en la figura 6.11, en la cual se muestra la evolución de 14 rasgos de las 5 especies que sobreviven en la red trófica tras 500 días.

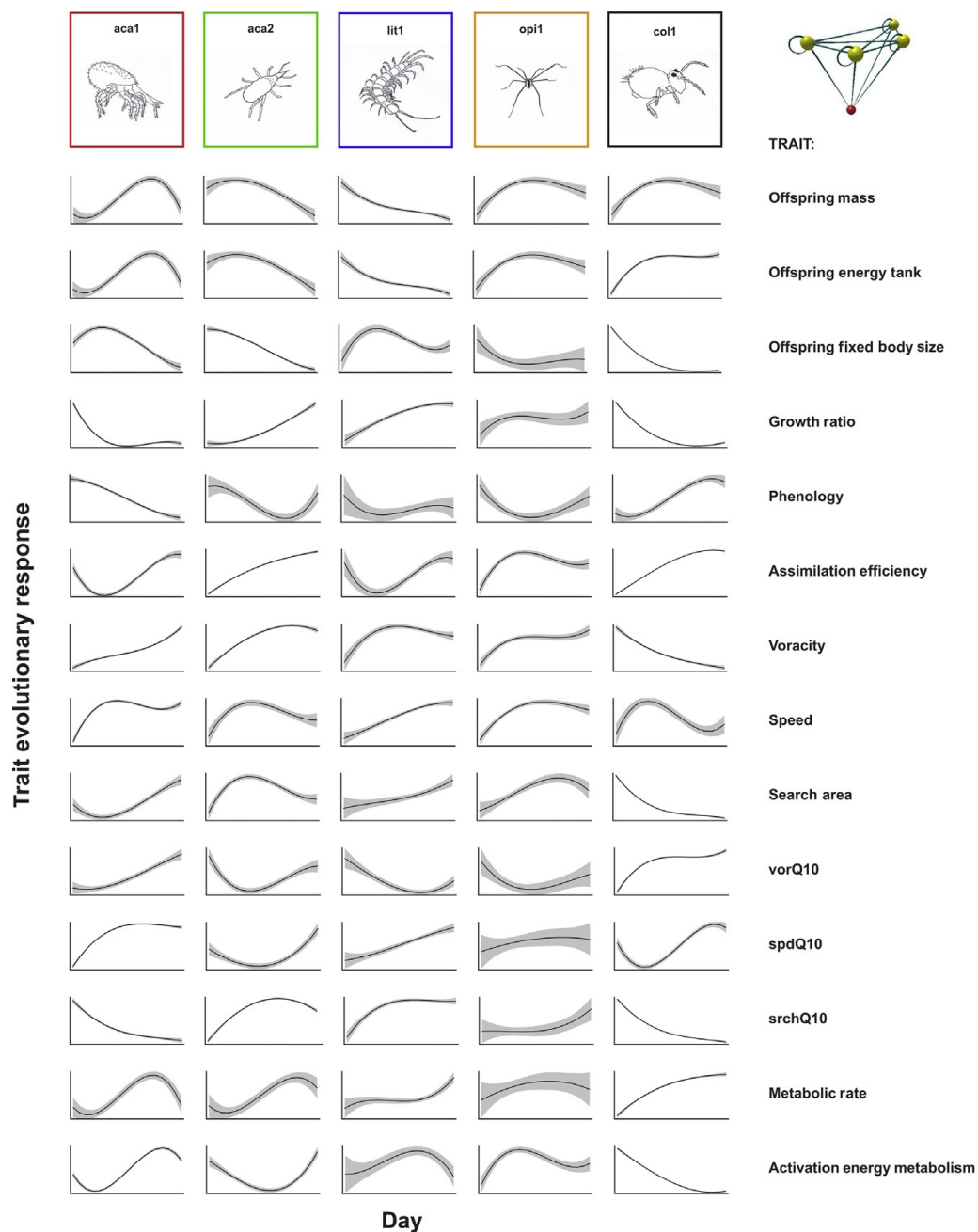


FIGURA 6.11: Evolución de 14 rasgos de las 5 especies persistentes tras 500 días. La figura representa la evolución de 14 rasgos de las 5 especies que sobreviven en la red trófica tras 500 días de simulación. De derecha a izquierda: dos ácaros (aca1, aca2), un ciempiés (lit1), un opilión (opi1) y un colémbolo (col1). El diagrama de la esquina superior derecha representa la estructura de la red trófica (ver de todas formas la figura 6.10). En todos los paneles, el eje X representa el tiempo, y el eje Y representa los valores del rasgo. Las áreas grises corresponden a bandas de confianza del 95 %, calculadas entre los individuos. Las definiciones de los rasgos pueden ser consultadas en la sección 4.2. El máximo número de generaciones alcanzadas por cada población es de: aca1=58, aca2=39, col1=52, lit1=12 y opi1=14. La figura ha sido extraída del artículo publicado [19].

La evolución se ha distribuido ampliamente entre las especies y los rasgos, con algunos de éstos mostrando claras oscilaciones hacia el curso de las dinámicas ecológicas. Se van a comentar los paneles de arriba a abajo. La primera impresión aparente es que la evolución de la masa de los descendientes está conducida por la evolución de la masa almacenada para el mantenimiento, crecimiento y reproducción y no por el tamaño estructural del cuerpo del descendiente (comparar los tres paneles superiores de la figura 6.11).

Los ratios de crecimiento de los dos ácaros depredadores pequeños muestran patrones opuestos, con uno (aca1) incrementando y el otro (aca2) disminuyendo. Los colémbolos (el único fungívoro) disminuyen en ratio de crecimiento. Entre los depredadores más grandes, uno no evoluciona en ratio de crecimiento, mientras que el otro lo incrementa (lit1). La fenología, o tiempo de desarrollo del huevo, muestra patrones opuestos en el ácaro mesostigmata (aca1) y en la presa compartida (col1), y no oscilan ni evolucionan en ninguno de las otras tres especies de depredadores. Como se esperaba, la eficiencia de asimilación aumenta generalmente a lo largo del tiempo.

Algunos rasgos incorporan un componente que los hace cambiar a partir de la temperatura ambiente (véase los rasgos definidos del tipo Q_{10} en la sección 4.2). Por ello, estos rasgos se comentan de forma conjunta. La voracidad, que determinan la actividad alimenticia del animal dentro de una celda, se incrementa consistentemente solamente en el ácaro mesostigmata (aca1). La velocidad evoluciona a valores más altos en todos los animales. Sin embargo, la respuesta evolutiva para la presa compartida ha sido la menor de todas. El área de búsqueda, que determina la actividad alimenticia del animal a lo largo de las celdas, evoluciona consistentemente solo en la presa compartida (disminuye) y para el depredador más alto (se incrementa). Finalmente, aunque hay una significativa evolución y oscilaciones en los ratios metabólicos, la única respuesta fuerte es la de la presa compartida en la cual los dos componentes del ratio metabólico, el coeficiente de escala y la energía de activación, evolucionan en direcciones opuestas [19].

6.3.8. Dinámicas evolutivas de gusanos en presencia y ausencia de depredadores

Un ejemplo de este efecto se puede observar en la figura 6.12, en la cual se muestra la evolución de los rasgos de un gusano fungívoro (*enc2*) bajo dos escenarios ecológicos distintos.

En las simulaciones con alta conectancia (0.55), el gusano *enc2* es la única especie que persiste consistentemente hasta el día 200, en ambos escenarios con y sin depredadores. Además, en un ambiente puramente competitivo, esta especie se convierte en la especie dominante junto a otro gusano. Se ha aprovechado esta persistencia en ambos ambientes para realizar pruebas sobre las diferencias entre las respuestas a la selección en *enc2* entre los dos escenarios ecológicos. En un ambiente puramente competitivo, la selección favorece la inversión en descendientes con mayor tanque de energía y menor tamaño, mientras que estos rasgos no evolucionan significativamente en presencia de depredadores. Sorprendentemente, en ausencia de depredadores la voracidad evoluciona a un valor menor y el ratio metabólico a un valor mayor. También, la plasticidad dependiente de la temperatura para el rasgo de la velocidad, que es solamente funcional bajo la amenaza de depredación, evoluciona a un valor mayor. Finalmente, la energía de activación para el ratio metabólico evoluciona a un valor menor. Por otro lado, solamente los rasgos directamente relacionados con la evasión de la depredación (es decir, velocidad y plasticidad para la velocidad) evolucionan claramente a un valor mayor en presencia de depredadores. El resto de rasgos no muestran ninguna respuesta significativa (por ejemplo, la energía de activación para el ratio metabólico) u oscilación alguna, mostrando valores que no difieren de los iniciales (por ejemplo, voracidad y área de búsqueda) [19].

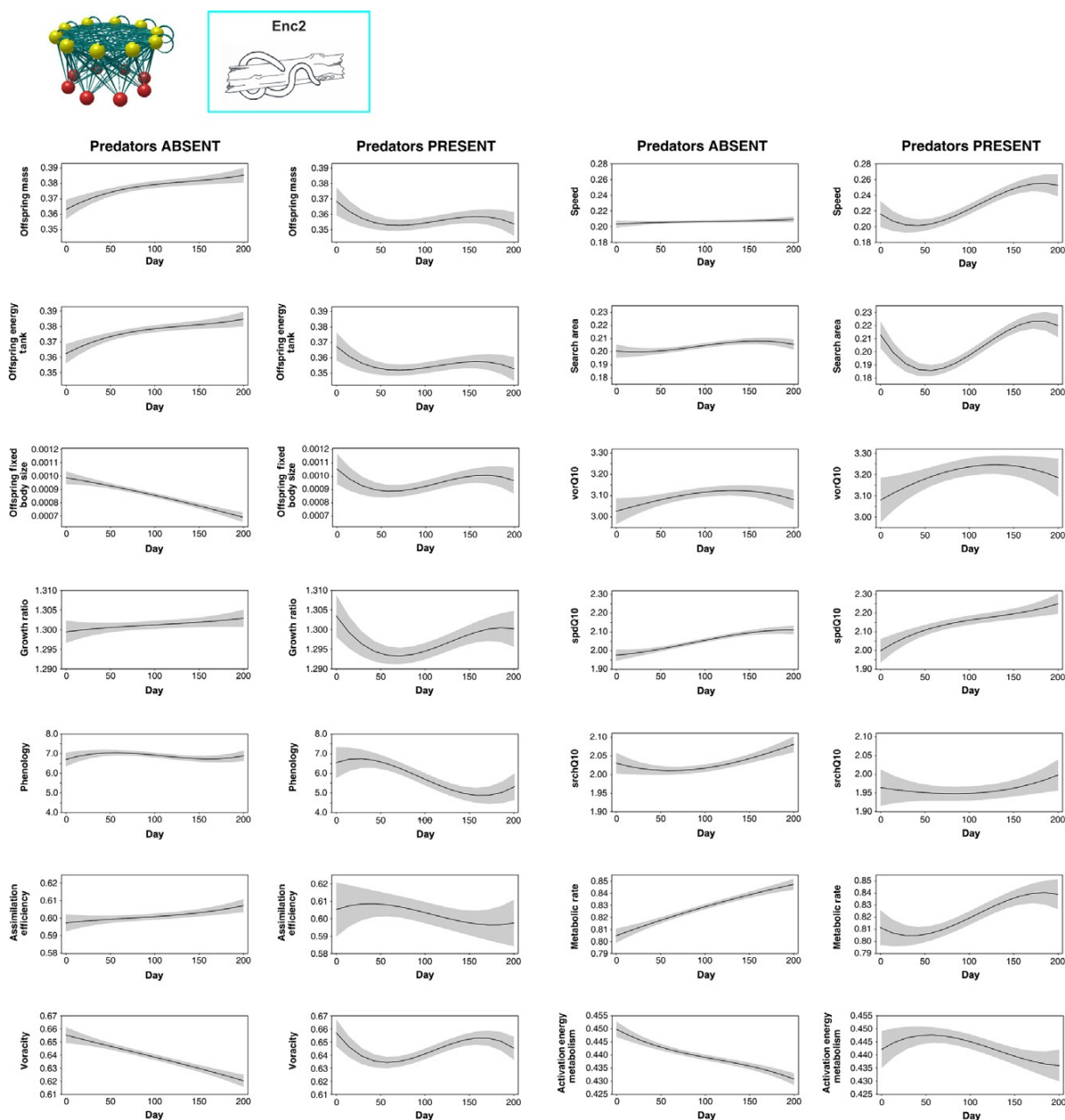


FIGURA 6.12: Evolución de los rasgos de un gusano fungívoro (*enc2*) en presencia y ausencia de depredadores. La figura representa la evolución de los rasgos de un gusano fungívoro (*enc2*) bajo dos escenarios ecológicos: depredadores ausentes, en el cual solamente están presentes las interacciones competitivas; y depredadores presentes, en el cual hay interacciones tanto de depredación como de competición. En todos los paneles, el eje X representa el tiempo, y el eje Y representa los valores del rasgo. Las áreas grises corresponden a bandas de confianza del 95 %, calculadas entre los individuos. La figura ha sido extraída del artículo publicado [19].

7 | Conclusiones y trabajo futuro

7.1. Conclusiones

Como se ha visto en la sección 6.3 de resultados, las simulaciones ejecutadas utilizando el programa Weaver han conseguido mostrar patrones relevantes que se espera puedan ayudar a entender las dinámicas eco-evolutivas en sistemas complejos. La implementación ha sido capaz de arrojar una cantidad de datos útil a la hora de realizar las investigaciones deseadas. Aunque el prototipo está bastante avanzado e incluye bastantes características, aún quedan muchas opciones posibles para mejorarlo e incluso para permitir la obtención de patrones eco-evolutivos más interesantes.

En las siguientes secciones se proponen una serie de ideas para continuar con el proyecto, ya que se espera que tenga un potencial y aceptación bastante altos.

7.2. Optimización para la estabilidad

Tras haber incluido un número superior de especies, va a ser realmente complicado encontrar los parámetros necesarios para conseguir estabilidad en un tiempo considerable. Para ello se propone el uso de algoritmos genéticos o de optimización que ayuden a definir los valores correctos de estos parámetros. Estos valores pueden ser aproximados inicialmente mediante la observación en ecosistemas reales, y viceversa, pueden ser contrastados los resultados simulados en el entorno real.

7.3. Mejora del suelo

El espacio en el que se desarrollan todos los procesos está limitado actualmente y no permite que los animales superen las paredes externas tanto hacia el exterior como hacia el interior del terreno. Como objetivo futuro se plantea la extensión del espacio de forma que puedan aparecer individuos espontáneos en las zonas cercanas a las paredes, provenientes del exterior. También se consideraría el proceso inverso, y es que desaparezcan individuos de las zonas cercanas a las paredes, ya que habrían saltado al exterior. Además, se propone la inclusión de nuevas especies de recursos basales (u hongos) e incluso modelar mejor la humedad del ambiente, o incluir la existencia de nutrientes que interactúan con el crecimiento de las especies de hongo.

7.4. Paralelización y modelo de islas

Como ya habíamos indicado en la sección 6.1 de pruebas de estabilidad, la paralelización del código está más que motivada ya que con ello se conseguirá reducir el tiempo empleado. Se propone trabajar en un modelo en el cual existan islas de población diferenciadas y ejecutándose simultáneamente y comunicándose mediante MPI. Los individuos tendrán la capacidad de moverse entre esas islas dadas unas circunstancias específicas, para lo cual habrá que establecer puntos de sincronización. Además, dentro de cada una de estas islas de población, se pueden separar los individuos identificando aquellos cuya actividad no se interfiere mutuamente, con lo cual se puede realizar la ejecución de las funciones asociadas a estos individuos simultáneamente mediante el uso de diferentes hilos de ejecución.

Esto es interesante también desde el punto de vista evolutivo: se podrían observar nuevos patrones al realizar simultáneamente las funciones de los individuos de las poblaciones, ya que además este comportamiento es intuitivamente más realista que el proceso secuencial.

7.5. Optimización de las salidas de datos

Las salidas de datos actuales han sido resumidas en la sección A.3 del manual de usuario de Weaver. Estas salidas son muy completas y aportan toda la información posible sobre

las simulaciones. Esto es un punto positivo para el programa, pero aún se puede ser más ambicioso y proponer que no solamente se escriba toda la información en los archivos de salida, sino que además esta información sea tratada directamente por Weaver, y el propio programa sea capaz de arrojar resúmenes más detallados o incluso gráficos o fotografías (snapshots) de la situación del terreno y animales en la simulación. Esta tarea puede suponer un trabajo muy complejo pero es también muy viable pues ayudaría en gran medida a los investigadores biólogos que quieran usar Weaver, pues les facilitaría la interacción con el mismo y les ahorraría el exhaustivo tratamiento de datos que, de otra forma, deben realizar utilizando otras herramientas estadísticas (como por ejemplo el lenguaje de programación R).

A | Manual de usuario de Weaver

En este documento se detalla toda la información necesaria relativa al uso de Weaver. Este manual de usuario se divide en varias secciones que van a tratar aspectos diferentes del programa: pasos para realizar una simulación, estructura del archivo de configuración, lista de archivos de salida generados.

A.1. Pasos para realizar una simulación

Los pasos a seguir por el usuario a la hora de realizar la ejecución de una simulación son los siguientes:

- **Paso 1** – Se debe colocar el archivo ejecutable *Weaver3D* en cualquier directorio junto con el archivo de configuración *run_params*. Es estrictamente necesario que ambos archivos se localicen en el mismo directorio, pues al comienzo del programa *Weaver3D* busca el archivo llamado *run_params* en el mismo directorio desde donde se lanzó la ejecución. Además, ese mismo directorio debe contener la carpeta *lib* con los cuatro archivos que se proporciona adjunta a este manual de usuario. La localización de los archivos debe ser entonces similar a la que se indica en la figura [A.1](#).
- **Paso 2** – Tras colocar los dos archivos y la carpeta *lib* en el mismo directorio, se lanza la ejecución de *Weaver3D* de la misma forma que se lanza cualquier ejecutable. En este caso bastaría con escribir el comando `./Weaver3D`, porque el ejecutable no precisa de ningún parámetro añadido por la entrada estándar. Todos los parámetros necesarios están recogidos en el archivo *run_params*, el cual se comenta con detalle en la sección [A.2](#).

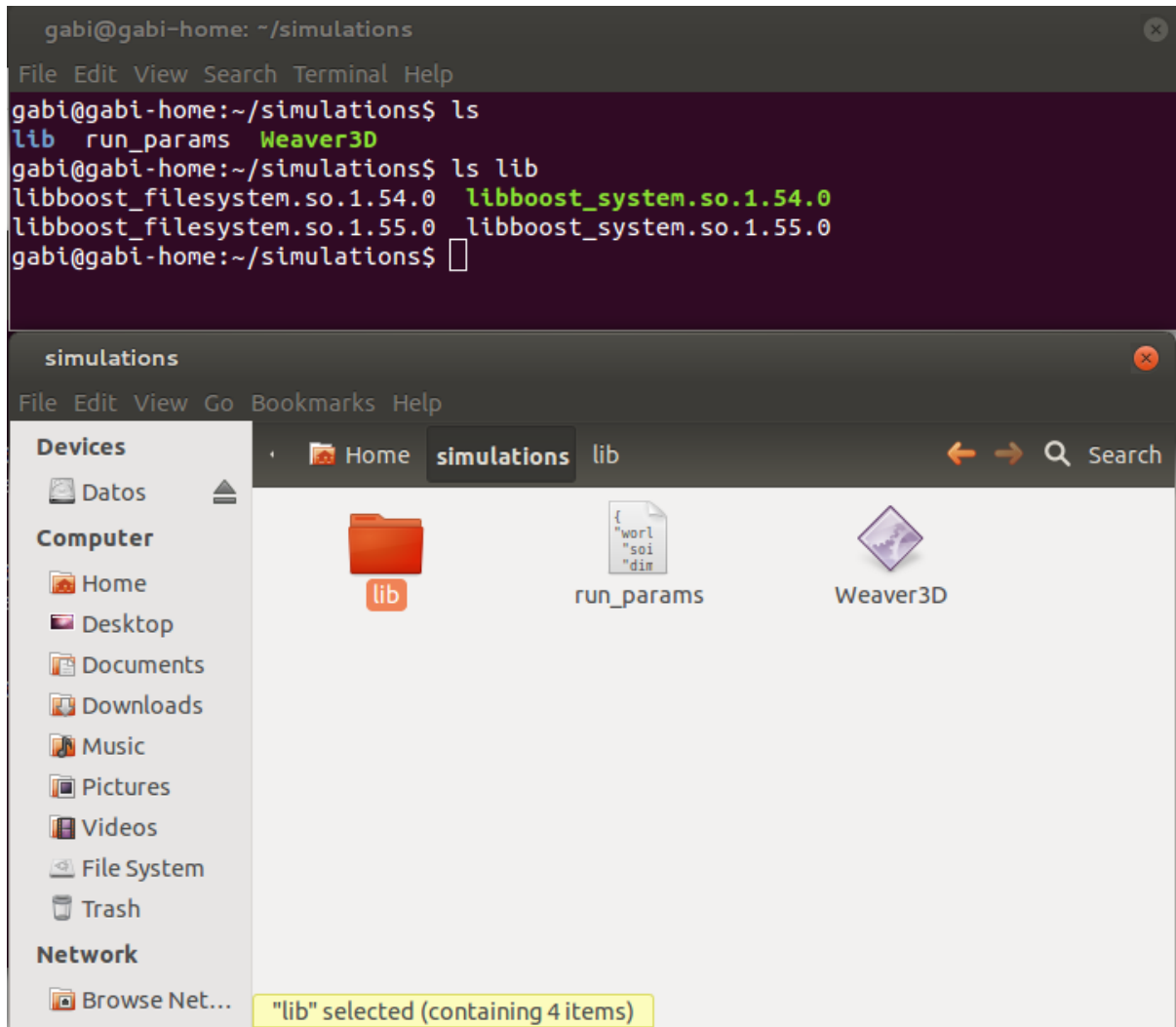
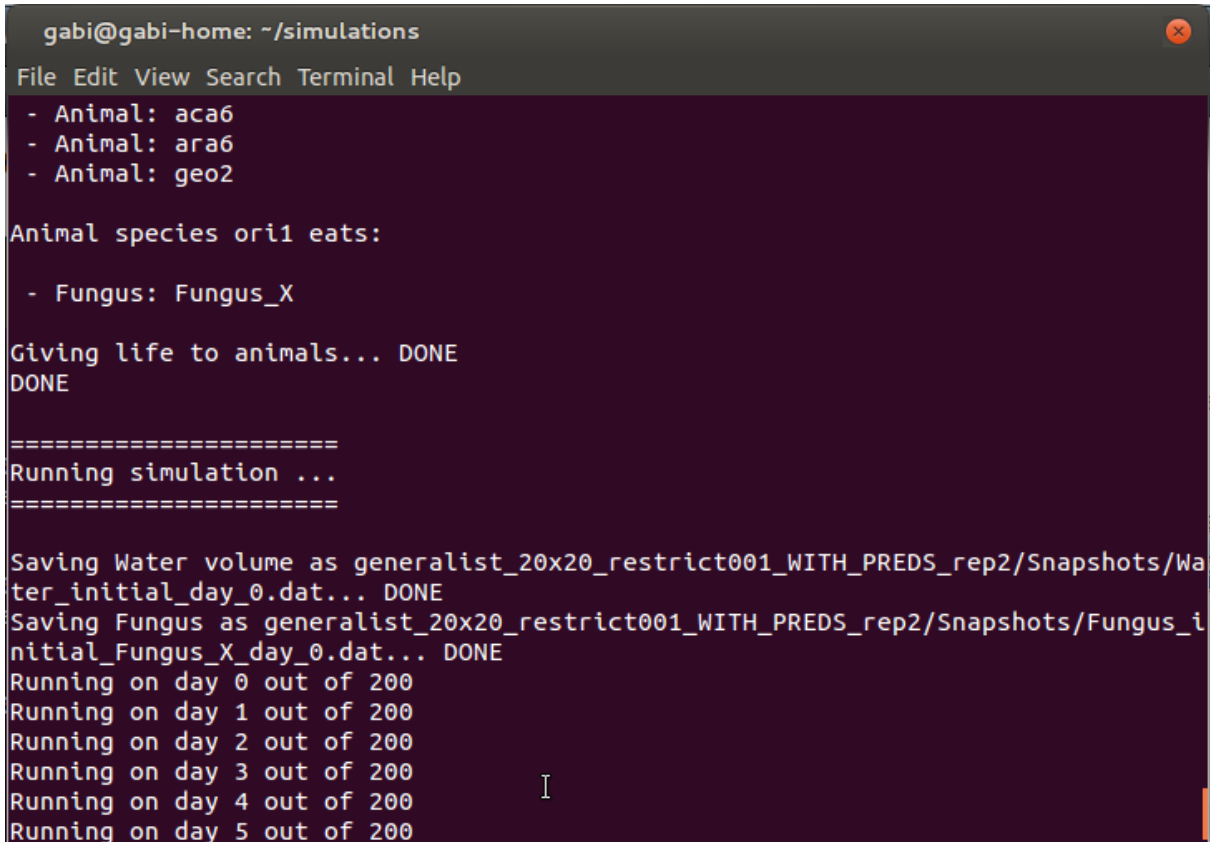


FIGURA A.1: Localización de los archivos previa ejecución de una simulación

- **Paso 3** – La simulación comienza su ejecución y muestra por consola varios datos para actualizar al usuario con el progreso de la misma, tal y como se puede observar en la figura A.2. Durante la simulación se generan progresivamente los archivos de salida en un directorio nuevo, el cual es creado dentro del mismo directorio de ejecución. Este directorio tiene como nombre el que se haya indicado en el archivo de configuración *run_params*, en el campo llamado *outputDirectory* (ver estructura del archivo de configuración en la sección A.2). Bajo el nuevo directorio se crean una serie de archivos de salida y directorios, los cuales vienen listados en su sección correspondiente en la sección A.3.

A screenshot of a terminal window titled 'gabi@gabi-home: ~/simulations'. The window has a menu bar with 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. The terminal output shows the following text:

```
gabi@gabi-home: ~/simulations
File Edit View Search Terminal Help
- Animal: aca6
- Animal: ara6
- Animal: geo2

Animal species ori1 eats:
- Fungus: Fungus_X

Giving life to animals... DONE
DONE

=====
Running simulation ...
=====

Saving Water volume as generalist_20x20_restrict001_WITH_PREDS_rep2/Snapshots/Wa
ter_initial_day_0.dat... DONE
Saving Fungus as generalist_20x20_restrict001_WITH_PREDS_rep2/Snapshots/Fungus_i
nitial_Fungus_X_day_0.dat... DONE
Running on day 0 out of 200
Running on day 1 out of 200
Running on day 2 out of 200
Running on day 3 out of 200
Running on day 4 out of 200
Running on day 5 out of 200
```

FIGURA A.2: Estado de la consola mientras se realiza una simulación

A.2. Estructura del archivo de configuración

En esta sección se explica con detalle qué significa cada uno de los parámetros que aparecen en el archivo de configuración *run_params*, que debe ser utilizado en cada una de las ejecuciones. Este fichero está escrito en un formato ligero de intercambio de datos llamado **JSON** (JavaScript Object Notation - Notación de Objetos de JavaScript). Leer y escribir datos en este formato es muy simple e intuitivo para humanos, y para las máquinas es simple interpretarlo, tratarlo y generarlo. JSON está constituido por dos estructuras principales:

- Una colección de pares de nombre/valor.
- Una lista ordenada de valores.

Estas estructuras pueden aparecer en un archivo JSON creando las siguientes formas:

- **objeto** – Un objeto es un conjunto desordenado de pares nombre/valor. Un objeto comienza con { (llave de apertura) y termina con } (llave de cierre). Cada nombre es seguido por : (dos puntos) y los pares nombre/valor están separados por , (coma).
- **array** – Un *array* es una colección de valores que comienza con *corchete izquierdo* y termina con *corchete derecho*. Los valores se separan por , (coma).
- **valor** – Un valor puede ser una cadena de caracteres con comillas dobles, o un número, o *true* o *false* o *null*, o un objeto o un array. Estas estructuras **pueden anidarse**.
- **cadena de caracteres** – Una cadena de caracteres es una colección de cero o más caracteres Unicode, encerrados entre comillas dobles, usando barras divisorias invertidas como escape.
- **número** – Un número es similar a un número C o Java, excepto que no se usan los formatos octales y hexadecimales.

Una vez descrito el lenguaje, se muestra a continuación un fragmento de ejemplo de un archivo *run_params*, en el cual se puede apreciar intuitivamente como están distribuidos los datos de cada componente del sistema. Más adelante se explica qué significa cada uno de los objetos y valores.

```
{
{
"world": {
  "soil": {
    "dimensions": {
      "depth": "10",
      "length": "20",
      "width": "20",
      "cellSize": "1"
    },
    "moisture": {
      "patches": [
        {
          "type": "sphere",
          "radius": "5",
          "xPos": "5",
          "yPos": "5",
          "zPos": "5",
          "value": "87.5"
        },
        {
          "type": "sphere",
          (...) (completar datos de esta zona)
        }
      ]
    }
  }
}
```

```

    },
    (...) (añadir más zonas si se necesita)
  ],
},
"nutrients": {
  "minC": "0",
  "maxC": "23",
  "minN": "0",
  "maxN": "23",
  "minP": "0",
  "maxP": "23"
},
"temperature": "18",
"maxK": "9.6",
"minK": "0.1",
"timelapseForChemostatEffect": "1",
"thresholdForChemostatEffect": "0.05",
"increaseForChemostatEffect": "1.0"
},
"life": {
  "animals": [
    {
      "name": "aca1",
      "huntingMode": "active_hunting",
      "genetics": {
        "NumberOfLoci": "20",
        "NumberOfAlleles": "10",
        "NumberOfTraits": "13",
        "TRAITS_PER_MODULE": "3"
      },
      "edibleAnimalSpecies": ["aca1","aca2"],
      "edibleFungusSpecies": [],
      "initialPopulation": "414",
      "restrict": "0.01",
      "correlationCoefficientRH0": "0.1",
      "A": "0.053",
      "B": "2.494",
      "forDensitiesGrowth": "1.4",
      "forDensitiesEggSize": "0.002",
      "ecosystemSize": "20000",
      "minCondition": "0.01",
      "maxVoracityT": "0.100",
      "keepForSurv": "3.0",
      "maxReproductionEvents": "5",
      "assignedForReproduction": "0.9",
      "forControllingDryBodyMass": "1.0",
      "probabilityDeathFromBackground": "0.0",

      "deadlyTank": "0.1",
      "assignedForMetab": "0.9",
      "numberOfInstars": "4",
      "percentOfTimeSinceLastInstar": "0.9",
      "alphaForPredation": "1.0",
      "minRandomForEncounters": "0.337",
      "maxRandomForEncounters": "0.338",
      "minRandomForPredation": "0.607",
      "maxRandomForPredation": "0.614",

      "maxEncountersT": "10",
      "maxSearchAreaT": "10",
    }
  ]
}

```

```

"Q10phenology": "0.35",
"Q10digestion": "0.25",

"meanSizeHunted": "20.08",
"sdSizeHunted": "7.5",
"meanVorHunted": "13.96",
"sdVorHunted": "4.9",
"meanSpdHunted": "40.65",
"sdSpdHunted": "15",

"meanSizeHunter": "20.08",
"sdSizeHunter": "7.5",
"meanVorHunter": "13.96",
"sdVorHunter": "4.9",
"meanSpdHunter": "40.65",
"sdSpdHunter": "15",
"meanSearchAreaHunter": "0.0",
"sdSearchAreaHunter": "0.0",

"meanSizeXSize": "806.4",
"sdSizeXSize": "275",
"meanVorXVor": "389.6",
"sdVorXVor": "140",
"meanSpdRatio": "25.5",
"sdSpdRatio": "9",
"meanSizeRatio": "397.2",
"sdSizeRatio": "140",

"minTraitsRanges": {
  "energy_tank": "0.25",
  "growth": "1.35",
  "pheno": "3",
  "body_size": "0.001",
  "assim": "0.7",
  "voracity": "0.55",
  "speed": "0.1",
  "search_area": "0.1",
  "met_rate": "0.6",
  "vorQ10": "3",
  "spdQ10": "1.5",
  "srchQ10": "2.0",
  "actE_met": "0.33"
},
"maxTraitsRanges": {
  "energy_tank": "0.5",
  "growth": "1.45",
  "pheno": "11",
  "body_size": "0.003",
  "assim": "0.9",
  "voracity": "0.75",
  "speed": "0.3",
  "search_area": "0.4",
  "met_rate": "0.8",
  "vorQ10": "4",
  "spdQ10": "2.5",
  "srchQ10": "2.5",
  "actE_met": "0.42"
},
"minTraitLimits": {
  "energy_tank": "0.01",
  "growth": "1.01",

```

```

        "pheno": "1",
        "body_size": "0.00001",
        "assim": "0.1",
        "voracity": "0.4",
        "speed": "0.05",
        "search_area": "0.05",
        "met_rate": "0.4",
        "vorQ10": "1",
        "spdQ10": "1",
        "srchQ10": "1",
        "actE_met": "0.2"
    },
    "maxTraitLimits": {
        "energy_tank": "1",
        "growth": "2.0",
        "pheno": "100",
        "body_size": "394.7",
        "assim": "1",
        "voracity": "0.8",
        "speed": "0.35",
        "search_area": "0.5",
        "met_rate": "0.9",
        "vorQ10": "6",
        "spdQ10": "3",
        "srchQ10": "3",
        "actE_met": "0.9"
    }
},
{
    "name": "aca2",
    "huntingMode": "active_hunting",
    "genetics" : {
        (...) (completar datos de esta especie)
    },
    (...) (añadir más especies si se necesita)
],
"fungi": [
    {
        "name": "Fungus_X",
        "sporeMass": "4",
        "minimumFungus": "0",
        "ACTIVATION_ENERGY": "0.68",
        "NORMALIZATION_B": "25.98",
        "minHR": "85",
        "maxHR": "90",
        "maxRScale": "0.5",
        "zeroFungi": "8",
        "patches": [
            {
                "type": "sphere",
                "radius": "5",
                "xPos": "5",
                "yPos": "5",
                "zPos": "5",
                "value": "1.5"
            },
            {
                "type": "sphere",
                (...) (completar datos
                de esta zona)
            }
        ],
    },

```

```

        (...) (añadir más zonas
        si se necesita)
    ]
}
]
}
},
"simulation": {
    "runDays" : "200",
    "outputDirectory": "generalist_20x20_restrict001_WITH_PREDS_rep2",
    "saveIntermediateVolumes": "false",
    "saveIntermediateVolumesPeriodicity": "100000",
    "encountersMatrixFilename": "encountersMatrix",
    "predationsMatrixFilename": "predationsMatrix",
    "nodesMatrixFilename": "nodesMatrix",
    "predationEventsOnOtherSpeciesFilename": "predationOnSpecies"
}
}
}

```

A.3. Lista de archivos de salida generados

Cada vez que se ejecuta una nueva simulación se generan progresivamente los archivos de salida en un directorio nuevo, el cual es creado dentro del mismo directorio de ejecución. Este directorio tiene como nombre el que se haya indicado en el archivo de configuración *run_params*, en el campo llamado *outputDirectory* (ver estructura del archivo de configuración en la sección A.2). Bajo el nuevo directorio se crean una serie de archivos de salida y directorios, los cuales se listan a continuación. Las salidas marcadas como *DESACTIVADO* están desactivadas en la versión actual de la aplicación para no colapsar la memoria o retrasar la ejecución en la máquina en la cual se esté probando la simulación.

(ATENCIÓN: si se realiza una segunda ejecución utilizando el mismo nombre para el campo de outputDirectory que la anterior, los archivos antiguos serán eliminados del disco.)

- *animal_constitutive_traits.txt* – Este archivo almacena en formato tabla la información relativa a cada individuo animal que nace. En concreto, se utiliza para tener un recuento de los rasgos constitutivos que tiene cada animal al nacer. Dentro del archivo, los nombres de los campos para estos trece rasgos son los siguientes: **energy**, **growth**, **pheno**, **body**, **assim**, **vor**, **speed search**, **met**, **vorQ10**, **spdQ10**, **srchQ10**, **e_met**.

Además de esta información, el archivo incluye los siguientes datos para poder rastrear a qué gama pertenece cada uno de los individuos: su identificador (**id**), la especie a la que pertenece (**species**), el número de generación que le corresponde según cada uno de sus padres (**g_numb_prt1** y **g_numb_prt2**), el identificador de cada uno de sus padres (**ID_prt1** e **ID_prt2**).

- ***animal_genetics.txt*** (*DESACTIVADO*) – Este archivo almacena todos y cada uno de los alelos presentes en cada cromosoma de cada animal de la simulación.

Además de esta información, el archivo incluye los siguientes datos para poder rastrear a qué gama pertenece cada uno de los individuos: su identificador (**id**), la especie a la que pertenece (**species**), el número de generación que le corresponde según uno de sus padres (**g_numb_prt1**), el identificador de cada uno de sus padres (**ID_prt1** e **ID_prt2**).

- ***dailySummary.txt*** (*DESACTIVADO*) – Este archivo almacena un resumen diario en forma de tabla sobre el número de animales que existen en cada estado, y la cantidad de biomasa total existente. Estos animales se dividen en este archivo por dos tipos: depredadores o presas. Así, los campos contenidos en esta salida son los siguientes: **DAY**, **BIOMASS**, **PREY_UNBORN**, **PREY_ACTIVE**, **PREY_STARVED**, **PREY_PREDATED**, **PREY_REPRODUCING**, **PREY_SENESCED**, **PREDATOR_UNBORN**, **PREDATOR_ACTIVE**, **PREDATOR_STARVED**, **PREDATOR_PREDATED**, **PREDATOR_REPRODUCING**, **PREDATOR_BACKGROUND**, **PREDATOR_SENESCED**.

- ***extendedDailySummary.txt*** – Este archivo almacena un resumen diario en forma de tabla sobre el número de animales que existen en cada estado, separándolos por cada especie animal. También muestra la cantidad de biomasa total existente para cada especie de hongo. La tabla muestra los siguientes campos: el día de simulación (**day**), la cantidad de biomass de cada especie de hongo, de la forma **scientificName_biomass**, y la cantidad de animales actuales en cada estado de vida para cada especie, que se escriben de la forma **scientificName_X** donde **X** simboliza el estado del animal (0-no nacido, 1-activo, 2-muerto por inanición,

3-muerto por depredación, 4-en reproducción, 5-muerto por background, 6-muerto por senescencia).

- *run_params* – Este archivos es una copia exacta del archivo *run_params* utilizado para ejecutar la simulación. Se crea esta copia para garantizar que se conservan los parámetros utilizados para posteriores comprobaciones.
- *animals_each_day_start* (*DESACTIVADO*) – Este directorio contiene una lista de archivos cuya nomenclatura sigue la pauta **animals_day_X.txt** donde **X** indica un día de simulación concreto. Cada archivo contiene toda la información acerca del estado actual de cada animal presente en la simulación **al comienzo del día** concreto. Entre dicha información se incluye el identificador del individuo (**id**), la especie a la que pertenece (**species**), las coordenadas en las que se encuentra (**x, y, z**), el estado de vida en que se encuentra el individuo (**state**), el instar que ha alcanzado hasta la fecha (**instar**), el valor inicial que obtuvo para su fenología (**pheno_ini**), la fecha en la que apareció en la simulación (**date_egg**), la fecha de su primera reproducción (**age_first_rep**), el número de veces que se ha reproducido (**rep_count**), su fecundidad o número de descendientes hasta la fecha (**fecundity**), la fecha de su muerte (**date_death**), el número de generación que le corresponde según cada uno de sus padres (**g_numbprt1** y **g_numbprt2**), el identificador de cada uno de sus padres (**IDprt1** e **IDprt2**), el número de encuentros que ha tenido con depredadores en el día actual (**encounters_pred**), el número total de encuentros que ha tenido con depredadores a lo largo de su vida (**global_pred_encs**), el número de días que le quedan para finalizar la digestión (**days_digest**) y todas las características actuales del animal que vienen asociadas desde los rasgos genéticos y han sido modificadas de alguna forma hasta hoy (**energy, growth, pheno, body, assim, vor, speed search, met, vorQ10, spdQ10, srchQ10, e_met**).
- *animals_each_day_end* – Contiene la misma información que el directorio anterior, pero referida a los animales justo **al finalizar el día** indicado en el nombre de archivo.
- *animals_each_day_encounterProbabilities* (*DESACTIVADO*) – Este directorio contiene una lista de archivos cuya nomenclatura sigue la pauta **animals_each_day_encounterProbabilities_X.txt** donde **X** indica un día

concreto de la simulación. Estos archivos de salida han sido implementados para depurar los ratios de encuentro entre las diferentes especies.

- ***animals_each_day_predationProbabilities*** (*DESACTIVADO*) – Este directorio contiene una lista de archivos cuya nomenclatura sigue la pauta ***animals_each_day_predationProbabilities_X.txt*** donde *X* indica un día concreto de la simulación. Estos archivos de salida han sido implementados para depurar los ratios de depredación entre las diferentes especies.

animals_each_day_voracities (*DESACTIVADO*) – Este directorio contiene una lista de archivos cuya nomenclatura sigue la pauta

animals_each_day_voracities_X.txt donde *X* indica un día concreto de la simulación. Estos archivos de salida han sido implementados para puro uso de depuración de forma que los parámetros biológicos no den lugar a comportamientos ir-reales. Por ejemplo, no pueden existir animales de tan gran tamaño tales que coman millones de animales de menor tamaño. Esta salida proporciona información acerca de las ingestas diarias de cada animal, de forma que se pueda comprobar que ningún valor se sale de los límites establecidos.

Matrices – Este directorio contiene un archivo llamado *predationOnSpecies* que almacena una matriz $N \times N$ donde *N* es el número de especies animales presentes en la simulación. Los valores de esta matriz representan el número de depredaciones que una especie animal (eje de nombres vertical) ha efectuado contra otra especie animal (eje de nombres horizontal). Esta información es muy importante para evaluar cómo se comportan las poblaciones de individuos de las distintas especies dependiendo del ambiente establecido para cada simulación.

Snapshots (*DESACTIVADO*) – Este directorio está desactivado actualmente pues solo se han realizado pruebas de depuración con el mismo. Se utiliza para almacenar archivos relativos a los volúmenes de agua, animales y hongo a lo largo del terreno. Por ahora esta característica del programa no está lo suficientemente avanzada como para generar salidas observables.

B | Ejemplo completo de archivo de configuración *run_params*

A continuación se muestra el archivo de configuración *run_params* completo que se ha utilizado para llevar a cabo la explicación de cómo ejecutar un ejemplo sencillo de simulación en la sección 6.2.

```
{
  "world": {
    "soil": {
      "dimensions": {
        "depth": "10",
        "length": "20",
        "width": "20",
        "cellSize": "1"
      },
      "moisture": {
        "patches": [
          {
            "type": "sphere",
            "radius": "5",
            "xPos": "5",
            "yPos": "5",
            "zPos": "5",
            "value": "87.5"
          },
          {
            "type": "sphere",
            "radius": "5",
            "xPos": "5",
            "yPos": "15",
            "zPos": "5",
            "value": "87.5"
          },
          {
            "type": "sphere",
            "radius": "5",
            "xPos": "15",
            "yPos": "5",
            "zPos": "5",
            "value": "87.5"
          },
          {
            "type": "sphere",
            "radius": "5",
            "xPos": "15",
            "yPos": "15",
            "zPos": "5",
            "value": "87.5"
          }
        ]
      },
      "nutrients": {
```

```

        "minC": "0",
        "maxC": "23",
        "minN": "0",
        "maxN": "23",
        "minP": "0",
        "maxP": "23"
    },
    "temperature": "18",
    "maxK": "9.6",
    "minK": "0.1",
    "timelapseForChemostatEffect": "1",
    "thresholdForChemostatEffect": "0.05",
    "increaseForChemostatEffect": "1.0"
},
"life": {
    "animals": [
        {
            "name": "aca1",
            "huntingMode": "active_hunting",
            "genetics": {
                "NumberOfLoci": "20",
                "NumberOfAlleles": "10",
                "NumberOfTraits": "13",
                "TRAITS_PER_MODULE": "3"
            },
            "edibleAnimalSpecies": ["aca1", "aca2"],
            "edibleFungusSpecies": [],
            "initialPopulation": "414",
            "restrict": "0.01",
            "correlationCoefficientRH0": "0.1",
            "A": "0.053",
            "B": "2.494",
            "forDensitiesGrowth": "1.4",
            "forDensitiesEggSize": "0.002",
            "ecosystemSize": "20000",
            "minCondition": "0.01",
            "maxVoracityT": "0.100",
            "keepForSurv": "3.0",
            "maxReproductionEvents": "5",
            "assignedForReproduction": "0.9",
            "forControllingDryBodyMass": "1.0",
            "probabilityDeathFromBackground": "0.0",

            "deadlyTank": "0.1",
            "assignedForMetab": "0.9",
            "numberOfInstars": "4",
            "percentOfTimeSinceLastInstar": "0.9",
            "alphaForPredation": "1.0",
            "minRandomForEncounters": "0.337",
            "maxRandomForEncounters": "0.338",
            "minRandomForPredation": "0.607",
            "maxRandomForPredation": "0.614",

            "maxEncountersT": "10",
            "maxSearchAreaT": "10",

            "Q10phenology": "0.35",
            "Q10digestion": "0.25",

            "meanSizeHunted": "20.08",
            "sdSizeHunted": "7.5",
            "meanVorHunted": "13.96",
            "sdVorHunted": "4.9",
            "meanSpdHunted": "40.65",
            "sdSpdHunted": "15",

            "meanSizeHunter": "20.08",
            "sdSizeHunter": "7.5",
            "meanVorHunter": "13.96",
            "sdVorHunter": "4.9",
            "meanSpdHunter": "40.65",
            "sdSpdHunter": "15",
            "meanSearchAreaHunter": "0.0",
            "sdSearchAreaHunter": "0.0",

            "meanSizeXSize": "806.4",
            "sdSizeXSize": "275",
            "meanVorXVor": "389.6",
            "sdVorXVor": "140",
            "meanSpdRatio": "25.5",
            "sdSpdRatio": "9",

```

```

"meanSizeRatio": "397.2",
"sdSizeRatio": "140",
"minTraitsRanges": {
  "energy_tank": "0.25",
  "growth": "1.35",
  "pheno": "3",
  "body_size": "0.001",
  "assim": "0.7",
  "voracity": "0.55",
  "speed": "0.1",
  "search_area": "0.1",
  "met_rate": "0.6",
  "vorQ10": "3",
  "spdQ10": "1.5",
  "srchQ10": "2.0",
  "actE_met": "0.33"
},
"maxTraitsRanges": {
  "energy_tank": "0.5",
  "growth": "1.45",
  "pheno": "11",
  "body_size": "0.003",
  "assim": "0.9",
  "voracity": "0.75",
  "speed": "0.3",
  "search_area": "0.4",
  "met_rate": "0.8",
  "vorQ10": "4",
  "spdQ10": "2.5",
  "srchQ10": "2.5",
  "actE_met": "0.42"
},
"minTraitLimits": {
  "energy_tank": "0.01",
  "growth": "1.01",
  "pheno": "1",
  "body_size": "0.00001",
  "assim": "0.1",
  "voracity": "0.4",
  "speed": "0.05",
  "search_area": "0.05",
  "met_rate": "0.4",
  "vorQ10": "1",
  "spdQ10": "1",
  "srchQ10": "1",
  "actE_met": "0.2"
},
"maxTraitLimits": {
  "energy_tank": "1",
  "growth": "2.0",
  "pheno": "100",
  "body_size": "394.7",
  "assim": "1",
  "voracity": "0.8",
  "speed": "0.35",
  "search_area": "0.5",
  "met_rate": "0.9",
  "vorQ10": "6",
  "spdQ10": "3",
  "srchQ10": "3",
  "actE_met": "0.9"
}
},
{
  "name": "aca2",
  "huntingMode": "active_hunting",
  "genetics": {
    "NumberOfLoci": "20",
    "NumberOfAlleles": "10",
    "NumberOfTraits": "13",
    "TRAITS_PER_MODULE": "3"
  },
  "edibleAnimalSpecies": ["aca2", "ori1"],
  "edibleFungusSpecies": [],
  "initialPopulation": "414",
  "restrict": "0.01",
  "correlationCoefficientRHO": "0.1",
  "A": "0.053",
  "B": "2.494",
  "forDensitiesGrowth": "1.4",

```

```

"forDensitiesEggSize": "0.003",
"ecosystemSize": "20000",
"minCondition": "0.01",
"maxVoracityT": "0.181",
"keepForSurv": "3.0",
"maxReproductionEvents": "5",
"assignedForReproduction": "0.9",
"forControllingDryBodyMass": "1.0",
"probabilityDeathFromBackground": "0.0",

"deadlyTank": "0.1",
"assignedForMetab": "0.9",
"numberOfInstars": "5",
"percentOfTimeSinceLastInstar": "0.9",
"alphaForPredation": "1.0",
"minRandomForEncounters": "0.337",
"maxRandomForEncounters": "0.338",
"minRandomForPredation": "0.607",
"maxRandomForPredation": "0.614",

"maxEncountersT": "10",
"maxSearchAreaT": "14",

"Q10phenology": "0.35",
"Q10digestion": "0.25",

"meanSizeHunted": "20.08",
"sdSizeHunted": "7.5",
"meanVorHunted": "13.96",
"sdVorHunted": "4.9",
"meanSpdHunted": "40.65",
"sdSpdHunted": "15",

"meanSizeHunter": "20.08",
"sdSizeHunter": "7.5",
"meanVorHunter": "13.96",
"sdVorHunter": "4.9",
"meanSpdHunter": "40.65",
"sdSpdHunter": "15",
"meanSearchAreaHunter": "0.0",
"sdSearchAreaHunter": "0.0",

"meanSizeXSize": "806.4",
"sdSizeXSize": "275",
"meanVorXVor": "389.6",
"sdVorXVor": "140",
"meanSpdRatio": "25.5",
"sdSpdRatio": "9",
"meanSizeRatio": "397.2",
"sdSizeRatio": "140",

"minTraitsRanges": {
  "energy_tank": "0.25",
  "growth": "1.35",
  "pheno": "3",
  "body_size": "0.002",
  "assim": "0.7",
  "voracity": "0.55",
  "speed": "0.1",
  "search_area": "0.1",
  "met_rate": "0.6",
  "vorQ10": "3",
  "spdQ10": "1.5",
  "srchQ10": "2.0",
  "actE_met": "0.36"
},
"maxTraitsRanges": {
  "energy_tank": "0.5",
  "growth": "1.45",
  "pheno": "11",
  "body_size": "0.004",
  "assim": "0.9",
  "voracity": "0.75",
  "speed": "0.3",
  "search_area": "0.4",
  "met_rate": "0.8",
  "vorQ10": "4",
  "spdQ10": "2.5",
  "srchQ10": "2.5",
  "actE_met": "0.46"
},

```

```

    "minTraitLimits": {
      "energy_tank": "0.01",
      "growth": "1.01",
      "pheno": "1",
      "body_size": "0.00001",
      "assim": "0.1",
      "voracity": "0.4",
      "speed": "0.05",
      "search_area": "0.05",
      "met_rate": "0.4",
      "vorQ10": "1",
      "spdQ10": "1",
      "srchQ10": "1",
      "actE_met": "0.2"
    },
    "maxTraitLimits": {
      "energy_tank": "1",
      "growth": "2.0",
      "pheno": "100",
      "body_size": "394.7",
      "assim": "1",
      "voracity": "0.8",
      "speed": "0.35",
      "search_area": "0.5",
      "met_rate": "0.9",
      "vorQ10": "6",
      "spdQ10": "3",
      "srchQ10": "3",
      "actE_met": "0.9"
    }
  },
  {
    "name": "aca3",
    "huntingMode": "active_hunting",
    "genetics": {
      "NumberOfLoci": "20",
      "NumberOfAlleles": "10",
      "NumberOfTraits": "13",
      "TRAITS_PER_MODULE": "3"
    },
    "edibleAnimalSpecies": ["aca4", "ara5", "col1", "col2"],
    "edibleFungusSpecies": [],
    "initialPopulation": "414",
    "restrict": "0.01",
    "correlationCoefficientRRHO": "0.1",
    "A": "0.053",
    "B": "2.494",
    "forDensitiesGrowth": "1.4",
    "forDensitiesEggSize": "0.004",
    "ecosystemSize": "20000",
    "minCondition": "0.01",
    "maxVoracityT": "0.308",
    "keepForSurv": "3.0",
    "maxReproductionEvents": "5",
    "assignedForReproduction": "0.9",
    "forControllingDryBodyMass": "1.0",
    "probabilityDeathFromBackground": "0.0",

    "deadlyTank": "0.1",
    "assignedForMetab": "0.9",
    "numberOfInstars": "5",
    "percentOfTimeSinceLastInstar": "0.9",
    "alphaForPredation": "1.0",
    "minRandomForEncounters": "0.337",
    "maxRandomForEncounters": "0.338",
    "minRandomForPredation": "0.607",
    "maxRandomForPredation": "0.614",

    "maxEncountersT": "10",
    "maxSearchAreaT": "19",

    "Q10phenology": "0.35",
    "Q10digestion": "0.25",

    "meanSizeHunted": "20.08",
    "sdSizeHunted": "7.5",
    "meanVorHunted": "13.96",
    "sdVorHunted": "4.9",
    "meanSpdHunted": "40.65",
    "sdSpdHunted": "15",

```

```

"meanSizeHunter": "20.08",
"sdSizeHunter": "7.5",
"meanVorHunter": "13.96",
"sdVorHunter": "4.9",
"meanSpdHunter": "40.65",
"sdSpdHunter": "15",
"meanSearchAreaHunter": "0.0",
"sdSearchAreaHunter": "0.0",

"meanSizeXSize": "806.4",
"sdSizeXSize": "275",
"meanVorXVor": "389.6",
"sdVorXVor": "140",
"meanSpdRatio": "25.5",
"sdSpdRatio": "9",
"meanSizeRatio": "397.2",
"sdSizeRatio": "140",

"minTraitsRanges": {
  "energy_tank": "0.25",
  "growth": "1.35",
  "pheno": "3",
  "body_size": "0.003",
  "assim": "0.7",
  "voracity": "0.55",
  "speed": "0.1",
  "search_area": "0.1",
  "met_rate": "0.6",
  "vorQ10": "3",
  "spdQ10": "1.5",
  "srchQ10": "2.0",
  "actE_met": "0.33"
},
"maxTraitsRanges": {
  "energy_tank": "0.5",
  "growth": "1.45",
  "pheno": "11",
  "body_size": "0.005",
  "assim": "0.9",
  "voracity": "0.75",
  "speed": "0.3",
  "search_area": "0.4",
  "met_rate": "0.8",
  "vorQ10": "4",
  "spdQ10": "2.5",
  "srchQ10": "2.5",
  "actE_met": "0.42"
},
"minTraitLimits": {
  "energy_tank": "0.01",
  "growth": "1.01",
  "pheno": "1",
  "body_size": "0.00001",
  "assim": "0.1",
  "voracity": "0.4",
  "speed": "0.05",
  "search_area": "0.05",
  "met_rate": "0.4",
  "vorQ10": "1",
  "spdQ10": "1",
  "srchQ10": "1",
  "actE_met": "0.2"
},
"maxTraitLimits": {
  "energy_tank": "1",
  "growth": "2.0",
  "pheno": "100",
  "body_size": "394.7",
  "assim": "1",
  "voracity": "0.8",
  "speed": "0.35",
  "search_area": "0.5",
  "met_rate": "0.9",
  "vorQ10": "6",
  "spdQ10": "3",
  "srchQ10": "3",
  "actE_met": "0.9"
}
},
{
"name": "aca4",

```

```

"huntingMode": "active_hunting",
"genetics" : {
  "NumberOfLoci": "20",
  "NumberOfAlleles": "10",
  "NumberOfTraits": "13",
  "TRAITS_PER_MODULE": "3"
},
"edibleAnimalSpecies" : ["aca1","aca2"],
"edibleFungusSpecies" : [],
"initialPopulation": "414",
"restrict": "0.01",
"correlationCoefficientRHO": "0.1",
"A": "0.053",
"B": "2.494",
"forDensitiesGrowth": "1.4",
"forDensitiesEggSize": "0.005",
"ecosystemSize": "20000",
"minCondition": "0.01",
"maxVoracityT": "0.376",
"keepForSurv": "3.0",
"maxReproductionEvents": "5",
"assignedForReproduction": "0.9",
"forControllingDryBodyMass": "1.0",
"probabilityDeathFromBackground": "0.0",

"deadlyTank": "0.1",
"assignedForMetab": "0.9",
"numberOfInstars": "5",
"percentOfTimeSinceLastInstar": "0.9",
"alphaForPredation": "1.0",
"minRandomForEncounters": "0.337",
"maxRandomForEncounters": "0.338",
"minRandomForPredation": "0.607",
"maxRandomForPredation": "0.614",

"maxEncountersT": "10",
"maxSearchAreaT": "22",

"Q10phenology": "0.35",
"Q10digestion": "0.25",

"meanSizeHunted": "20.08",
"sdSizeHunted": "7.5",
"meanVorHunted": "13.96",
"sdVorHunted": "4.9",
"meanSpdHunted": "40.65",
"sdSpdHunted": "15",

"meanSizeHunter": "20.08",
"sdSizeHunter": "7.5",
"meanVorHunter": "13.96",
"sdVorHunter": "4.9",
"meanSpdHunter": "40.65",
"sdSpdHunter": "15",
"meanSearchAreaHunter": "0.0",
"sdSearchAreaHunter": "0.0",

"meanSizeXSize": "806.4",
"sdSizeXSize": "275",
"meanVorXVor": "389.6",
"sdVorXVor": "140",
"meanSpdRatio": "25.5",
"sdSpdRatio": "9",
"meanSizeRatio": "397.2",
"sdSizeRatio": "140",

"minTraitsRanges": {
  "energy_tank": "0.25",
  "growth": "1.35",
  "pheno": "3",
  "body_size": "0.004",
  "assim": "0.7",
  "voracity": "0.55",
  "speed": "0.1",
  "search_area": "0.1",
  "met_rate": "0.6",
  "vorQ10": "3",
  "spdQ10": "1.5",
  "srchQ10": "2.0",
  "actE_met": "0.36"
},

```

```

    "maxTraitsRanges": {
      "energy_tank": "0.5",
      "growth": "1.45",
      "pheno": "11",
      "body_size": "0.006",
      "assim": "0.9",
      "voracity": "0.75",
      "speed": "0.3",
      "search_area": "0.4",
      "met_rate": "0.8",
      "vorQ10": "4",
      "spdQ10": "2.5",
      "srchQ10": "2.5",
      "actE_met": "0.46"
    },
    "minTraitLimits": {
      "energy_tank": "0.01",
      "growth": "1.01",
      "pheno": "1",
      "body_size": "0.00001",
      "assim": "0.1",
      "voracity": "0.4",
      "speed": "0.05",
      "search_area": "0.05",
      "met_rate": "0.4",
      "vorQ10": "1",
      "spdQ10": "1",
      "srchQ10": "1",
      "actE_met": "0.2"
    },
    "maxTraitLimits": {
      "energy_tank": "1",
      "growth": "2.0",
      "pheno": "100",
      "body_size": "394.7",
      "assim": "1",
      "voracity": "0.8",
      "speed": "0.35",
      "search_area": "0.5",
      "met_rate": "0.9",
      "vorQ10": "6",
      "spdQ10": "3",
      "srchQ10": "3",
      "actE_met": "0.9"
    }
  },
  {
    "name": "aca5",
    "huntingMode": "active_hunting",
    "genetics" : {
      "NumberOfLoci": "20",
      "NumberOfAlleles": "10",
      "NumberOfTraits": "13",
      "TRAITS_PER_MODULE": "3"
    },
    "edibleAnimalSpecies" : ["ara5", "col1", "col2"],
    "edibleFungusSpecies" : [],
    "initialPopulation": "414",
    "restrict": "0.01",
    "correlationCoefficientRHO": "0.1",
    "A": "0.053",
    "B": "2.494",
    "forDensitiesGrowth": "1.4",
    "forDensitiesEggSize": "0.003",
    "ecosystemSize": "20000",
    "minCondition": "0.01",
    "maxVoracityT": "0.232",
    "keepForSurv": "3.0",
    "maxReproductionEvents": "5",
    "assignedForReproduction": "0.9",
    "forControllingDryBodyMass": "1.0",
    "probabilityDeathFromBackground": "0.0",

    "deadlyTank": "0.1",
    "assignedForMetab": "0.9",
    "numberOfInstars": "5",
    "percentOfTimeSinceLastInstar": "0.9",
    "alphaForPredation": "1.0",
    "minRandomForEncounters": "0.337",
    "maxRandomForEncounters": "0.338",
    "minRandomForPredation": "0.607",

```



```

"maxRandomForPredation": "0.614",
"maxEncountersT": "10",
"maxSearchAreaT": "16",
"Q10phenology": "0.35",
"Q10digestion": "0.25",
"meanSizeHunted": "20.08",
"sdSizeHunted": "7.5",
"meanVorHunted": "13.96",
"sdVorHunted": "4.9",
"meanSpdHunted": "40.65",
"sdSpdHunted": "15",
"meanSizeHunter": "20.08",
"sdSizeHunter": "7.5",
"meanVorHunter": "13.96",
"sdVorHunter": "4.9",
"meanSpdHunter": "40.65",
"sdSpdHunter": "15",
"meanSearchAreaHunter": "0.0",
"sdSearchAreaHunter": "0.0",
"meanSizeXSize": "806.4",
"sdSizeXSize": "275",
"meanVorXVor": "389.6",
"sdVorXVor": "140",
"meanSpdRatio": "25.5",
"sdSpdRatio": "9",
"meanSizeRatio": "397.2",
"sdSizeRatio": "140",
"minTraitsRanges": {
  "energy_tank": "0.25",
  "growth": "1.35",
  "pheno": "3",
  "body_size": "0.0028",
  "assim": "0.7",
  "voracity": "0.55",
  "speed": "0.1",
  "search_area": "0.1",
  "met_rate": "0.6",
  "vorQ10": "3",
  "spdQ10": "1.5",
  "srchQ10": "2.0",
  "actE_met": "0.33"
},
"maxTraitsRanges": {
  "energy_tank": "0.5",
  "growth": "1.45",
  "pheno": "11",
  "body_size": "0.0035",
  "assim": "0.9",
  "voracity": "0.75",
  "speed": "0.3",
  "search_area": "0.4",
  "met_rate": "0.8",
  "vorQ10": "4",
  "spdQ10": "2.5",
  "srchQ10": "2.5",
  "actE_met": "0.42"
},
"minTraitLimits": {
  "energy_tank": "0.01",
  "growth": "1.01",
  "pheno": "1",
  "body_size": "0.00001",
  "assim": "0.1",
  "voracity": "0.4",
  "speed": "0.05",
  "search_area": "0.05",
  "met_rate": "0.4",
  "vorQ10": "1",
  "spdQ10": "1",
  "srchQ10": "1",
  "actE_met": "0.2"
},
"maxTraitLimits": {
  "energy_tank": "1",
  "growth": "2.0",

```

```

        "pheno": "100",
        "body_size": "394.7",
        "assim": "1",
        "voracity": "0.8",
        "speed": "0.35",
        "search_area": "0.5",
        "met_rate": "0.9",
        "vorQ10": "6",
        "spdQ10": "3",
        "srchQ10": "3",
        "actE_met": "0.9"
    }
},
{
    "name": "aca6",
    "huntingMode": "active_hunting",
    "genetics" : {
        "NumberOfLoci": "20",
        "NumberOfAlleles": "10",
        "NumberOfTraits": "13",
        "TRAITS_PER_MODULE": "3"
    },
    "edibleAnimalSpecies" : ["lit1","ara1","aca5"],
    "edibleFungusSpecies" : [],
    "initialPopulation": "414",
    "restrict": "0.01",
    "correlationCoefficientRH0": "0.1",
    "A": "0.053",
    "B": "2.494",
    "forDensitiesGrowth": "1.4",
    "forDensitiesEggSize": "0.0025",
    "ecosystemSize": "20000",
    "minCondition": "0.01",
    "maxVoracityT": "0.143",
    "keepForSurv": "3.0",
    "maxReproductionEvents": "5",
    "assignedForReproduction": "0.9",
    "forControllingDryBodyMass": "1.0",
    "probabilityDeathFromBackground": "0.0",

    "deadlyTank": "0.1",
    "assignedForMetab": "0.9",
    "numberOfInstars": "4",
    "percentOfTimeSinceLastInstar": "0.9",
    "alphaForPredation": "1.0",
    "minRandomForEncounters": "0.337",
    "maxRandomForEncounters": "0.338",
    "minRandomForPredation": "0.607",
    "maxRandomForPredation": "0.614",

    "maxEncountersT": "10",
    "maxSearchAreaT": "12",

    "Q10phenology": "0.35",
    "Q10digestion": "0.25",

    "meanSizeHunted": "20.08",
    "sdSizeHunted": "7.5",
    "meanVorHunted": "13.96",
    "sdVorHunted": "4.9",
    "meanSpdHunted": "40.65",
    "sdSpdHunted": "15",

    "meanSizeHunter": "20.08",
    "sdSizeHunter": "7.5",
    "meanVorHunter": "13.96",
    "sdVorHunter": "4.9",
    "meanSpdHunter": "40.65",
    "sdSpdHunter": "15",
    "meanSearchAreaHunter": "0.0",
    "sdSearchAreaHunter": "0.0",

    "meanSizeXSize": "806.4",
    "sdSizeXSize": "275",
    "meanVorXVor": "389.6",
    "sdVorXVor": "140",
    "meanSpdRatio": "25.5",
    "sdSpdRatio": "9",
    "meanSizeRatio": "397.2",
    "sdSizeRatio": "140",

```

```

    "minTraitsRanges": {
      "energy_tank": "0.25",
      "growth": "1.35",
      "pheno": "3",
      "body_size": "0.002",
      "assim": "0.7",
      "voracity": "0.55",
      "speed": "0.1",
      "search_area": "0.1",
      "met_rate": "0.6",
      "vorQ10": "3",
      "spdQ10": "1.5",
      "srchQ10": "2.0",
      "actE_met": "0.36"
    },
    "maxTraitsRanges": {
      "energy_tank": "0.5",
      "growth": "1.45",
      "pheno": "11",
      "body_size": "0.003",
      "assim": "0.9",
      "voracity": "0.75",
      "speed": "0.3",
      "search_area": "0.4",
      "met_rate": "0.8",
      "vorQ10": "4",
      "spdQ10": "2.5",
      "srchQ10": "2.5",
      "actE_met": "0.46"
    },
    "minTraitLimits": {
      "energy_tank": "0.01",
      "growth": "1.01",
      "pheno": "1",
      "body_size": "0.00001",
      "assim": "0.1",
      "voracity": "0.4",
      "speed": "0.05",
      "search_area": "0.05",
      "met_rate": "0.4",
      "vorQ10": "1",
      "spdQ10": "1",
      "srchQ10": "1",
      "actE_met": "0.2"
    },
    "maxTraitLimits": {
      "energy_tank": "1",
      "growth": "2.0",
      "pheno": "100",
      "body_size": "394.7",
      "assim": "1",
      "voracity": "0.8",
      "speed": "0.35",
      "search_area": "0.5",
      "met_rate": "0.9",
      "vorQ10": "6",
      "spdQ10": "3",
      "srchQ10": "3",
      "actE_met": "0.9"
    }
  },
  {
    "name": "ara1",
    "huntingMode": "active_hunting",
    "genetics": {
      "NumberOfLoci": "20",
      "NumberOfAlleles": "10",
      "NumberOfTraits": "13",
      "TRAITS_PER_MODULE": "3"
    },
    "edibleAnimalSpecies": ["aca4", "ara5", "col1", "col2",
      "aca1", "aca2", "ori1"],
    "edibleFungusSpecies": [],
    "initialPopulation": "414",
    "restrict": "0.01",
    "correlationCoefficientRRH0": "0.1",
    "A": "0.011",
    "B": "3.184",
    "forDensitiesGrowth": "1.25",
    "forDensitiesEggSize": "0.023",
    "ecosystemSize": "20000",

```

```

"minCondition": "0.01",
"maxVoracityT": "3.218",
"keepForSurv": "3.0",
"maxReproductionEvents": "5",
"assignedForReproduction": "0.9",
"forControllingDryBodyMass": "1.0",
"probabilityDeathFromBackground": "0.0",

"deadlyTank": "0.1",
"assignedForMetab": "0.9",
"numberOfInstars": "8",
"percentOfTimeSinceLastInstar": "0.8",
"alphaForPredation": "1.0",
"minRandomForEncounters": "0.337",
"maxRandomForEncounters": "0.3375",
"minRandomForPredation": "0.607",
"maxRandomForPredation": "0.612",

"maxEncountersT": "10",
"maxSearchAreaT": "74",

"Q10phenology": "0.35",
"Q10digestion": "0.25",

"meanSizeHunted": "20.08",
"sdSizeHunted": "7.5",
"meanVorHunted": "13.96",
"sdVorHunted": "4.9",
"meanSpdHunted": "40.65",
"sdSpdHunted": "15",

"meanSizeHunter": "20.08",
"sdSizeHunter": "7.5",
"meanVorHunter": "13.96",
"sdVorHunter": "4.9",
"meanSpdHunter": "40.65",
"sdSpdHunter": "15",
"meanSearchAreaHunter": "0.0",
"sdSearchAreaHunter": "0.0",

"meanSizeXSize": "806.4",
"sdSizeXSize": "275",
"meanVorXVor": "389.6",
"sdVorXVor": "140",
"meanSpdRatio": "25.5",
"sdSpdRatio": "9",
"meanSizeRatio": "397.2",
"sdSizeRatio": "140",

"minTraitsRanges": {
  "energy_tank": "0.25",
  "growth": "1.2",
  "pheno": "3",
  "body_size": "0.013",
  "assim": "0.7",
  "voracity": "0.55",
  "speed": "0.1",
  "search_area": "0.1",
  "met_rate": "0.5",
  "vorQ10": "3",
  "spdQ10": "1.5",
  "srchQ10": "2.0",
  "actE_met": "0.45"
},
"maxTraitsRanges": {
  "energy_tank": "0.5",
  "growth": "1.3",
  "pheno": "11",
  "body_size": "0.033",
  "assim": "0.9",
  "voracity": "0.75",
  "speed": "0.3",
  "search_area": "0.4",
  "met_rate": "0.7",
  "vorQ10": "4",
  "spdQ10": "2.5",
  "srchQ10": "2.5",
  "actE_met": "0.55"
},
"minTraitLimits": {

```

```

        "energy_tank": "0.01",
        "growth": "1.01",
        "pheno": "1",
        "body_size": "0.00001",
        "assim": "0.1",
        "voracity": "0.4",
        "speed": "0.05",
        "search_area": "0.05",
        "met_rate": "0.4",
        "vorQ10": "1",
        "spdQ10": "1",
        "srchQ10": "1",
        "actE_met": "0.2"
    },
    "maxTraitLimits": {
        "energy_tank": "1",
        "growth": "2.0",
        "pheno": "100",
        "body_size": "394.7",
        "assim": "1",
        "voracity": "0.8",
        "speed": "0.35",
        "search_area": "0.5",
        "met_rate": "0.9",
        "vorQ10": "6",
        "spdQ10": "3",
        "srchQ10": "3",
        "actE_met": "0.9"
    }
},
{
    "name": "ara2",
    "huntingMode": "active_hunting",
    "genetics" : {
        "NumberOfLoci": "20",
        "NumberOfAlleles": "10",
        "NumberOfTraits": "13",
        "TRAITS_PER_MODULE": "3"
    },
    "edibleAnimalSpecies" : ["aca4","ara5","col1","col2",
    "enc1"],
    "edibleFungusSpecies" : [],
    "initialPopulation": "414",
    "restrict": "0.01",
    "correlationCoefficientRRHO": "0.1",
    "A": "0.063",
    "B": "2.530",
    "forDensitiesGrowth": "1.3",
    "forDensitiesEggSize": "0.009",
    "ecosystemSize": "20000",
    "minCondition": "0.01",
    "maxVoracityT": "0.852",
    "keepForSurv": "3.0",
    "maxReproductionEvents": "5",
    "assignedForReproduction": "0.9",
    "forControllingDryBodyMass": "1.0",
    "probabilityDeathFromBackground": "0.0",

    "deadlyTank": "0.1",
    "assignedForMetab": "0.9",
    "numberOfInstars": "7",
    "percentOfTimeSinceLastInstar": "0.8",
    "alphaForPredation": "1.0",
    "minRandomForEncounters": "0.337",
    "maxRandomForEncounters": "0.3375",
    "minRandomForPredation": "0.607",
    "maxRandomForPredation": "0.612",

    "maxEncountersT": "10",
    "maxSearchAreaT": "35",

    "Q10phenology": "0.35",
    "Q10digestion": "0.25",

    "meanSizeHunted": "20.08",
    "sdSizeHunted": "7.5",
    "meanVorHunted": "13.96",
    "sdVorHunted": "4.9",
    "meanSpdHunted": "40.65",
    "sdSpdHunted": "15",

```

```

"meanSizeHunter": "20.08",
"sdSizeHunter": "7.5",
"meanVorHunter": "13.96",
"sdVorHunter": "4.9",
"meanSpdHunter": "40.65",
"sdSpdHunter": "15",
"meanSearchAreaHunter": "0.0",
"sdSearchAreaHunter": "0.0",

"meanSizeXSize": "806.4",
"sdSizeXSize": "275",
"meanVorXVor": "389.6",
"sdVorXVor": "140",
"meanSpdRatio": "25.5",
"sdSpdRatio": "9",
"meanSizeRatio": "397.2",
"sdSizeRatio": "140",

"minTraitsRanges": {
  "energy_tank": "0.25",
  "growth": "1.25",
  "pheno": "3",
  "body_size": "0.008",
  "assim": "0.7",
  "voracity": "0.55",
  "speed": "0.1",
  "search_area": "0.1",
  "met_rate": "0.5",
  "vorQ10": "3",
  "spdQ10": "1.5",
  "srchQ10": "2.0",
  "actE_met": "0.45"
},
"maxTraitsRanges": {
  "energy_tank": "0.5",
  "growth": "1.35",
  "pheno": "11",
  "body_size": "0.010",
  "assim": "0.9",
  "voracity": "0.75",
  "speed": "0.3",
  "search_area": "0.4",
  "met_rate": "0.7",
  "vorQ10": "4",
  "spdQ10": "2.5",
  "srchQ10": "2.5",
  "actE_met": "0.55"
},
"minTraitLimits": {
  "energy_tank": "0.01",
  "growth": "1.01",
  "pheno": "1",
  "body_size": "0.00001",
  "assim": "0.1",
  "voracity": "0.4",
  "speed": "0.05",
  "search_area": "0.05",
  "met_rate": "0.4",
  "vorQ10": "1",
  "spdQ10": "1",
  "srchQ10": "1",
  "actE_met": "0.2"
},
"maxTraitLimits": {
  "energy_tank": "1",
  "growth": "2.0",
  "pheno": "100",
  "body_size": "394.7",
  "assim": "1",
  "voracity": "0.8",
  "speed": "0.35",
  "search_area": "0.5",
  "met_rate": "0.9",
  "vorQ10": "6",
  "spdQ10": "3",
  "srchQ10": "3",
  "actE_met": "0.9"
}
},
{
  "name": "ara3",

```

```

"huntingMode": "active_hunting",
"genetics" : {
  "NumberOfLoci": "20",
  "NumberOfAlleles": "10",
  "NumberOfTraits": "13",
  "TRAITS_PER_MODULE": "3"
},
"edibleAnimalSpecies" : ["aca4", "ara4", "ara5", "col1",
"col2", "ara3", "enc1"],
"edibleFungusSpecies" : [],
"initialPopulation": "414",
"restrict": "0.01",
"correlationCoefficientRH0": "0.1",
"A": "0.057",
"B": "2.829",
"forDensitiesGrowth": "1.2",
"forDensitiesEggSize": "0.059",
"ecosystemSize": "20000",
"minCondition": "0.01",
"maxVoracityT": "12.063",
"keepForSurv": "3.0",
"maxReproductionEvents": "5",
"assignedForReproduction": "0.9",
"forControllingDryBodyMass": "1.0",
"probabilityDeathFromBackground": "0.0",

"deadlyTank": "0.1",
"assignedForMetab": "0.9",
"numberOfInstars": "12",
"percentOfTimeSinceLastInstar": "0.8",
"alphaForPredation": "1.0",
"minRandomForEncounters": "0.337",
"maxRandomForEncounters": "0.3375",
"minRandomForPredation": "0.607",
"maxRandomForPredation": "0.612",

"maxEncountersT": "10",
"maxSearchAreaT": "157",

"Q10phenology": "0.35",
"Q10digestion": "0.25",

"meanSizeHunted": "20.08",
"sdSizeHunted": "7.5",
"meanVorHunted": "13.96",
"sdVorHunted": "4.9",
"meanSpdHunted": "40.65",
"sdSpdHunted": "15",

"meanSizeHunter": "20.08",
"sdSizeHunter": "7.5",
"meanVorHunter": "13.96",
"sdVorHunter": "4.9",
"meanSpdHunter": "40.65",
"sdSpdHunter": "15",
"meanSearchAreaHunter": "0.0",
"sdSearchAreaHunter": "0.0",

"meanSizeXSize": "806.4",
"sdSizeXSize": "275",
"meanVorXVor": "389.6",
"sdVorXVor": "140",
"meanSpdRatio": "25.5",
"sdSpdRatio": "9",
"meanSizeRatio": "397.2",
"sdSizeRatio": "140",

"minTraitsRanges": {
  "energy_tank": "0.25",
  "growth": "1.15",
  "pheno": "3",
  "body_size": "0.049",
  "assim": "0.7",
  "voracity": "0.55",
  "speed": "0.1",
  "search_area": "0.1",
  "met_rate": "0.5",
  "vorQ10": "3",
  "spdQ10": "1.5",
  "srchQ10": "2.0",
  "actE_met": "0.45"
}

```

```

    },
    "maxTraitsRanges": {
      "energy_tank": "0.5",
      "growth": "1.25",
      "pheno": "11",
      "body_size": "0.069",
      "assim": "0.9",
      "voracity": "0.75",
      "speed": "0.3",
      "search_area": "0.4",
      "met_rate": "0.7",
      "vorQ10": "4",
      "spdQ10": "2.5",
      "srchQ10": "2.5",
      "actE_met": "0.55"
    },
    "minTraitLimits": {
      "energy_tank": "0.01",
      "growth": "1.01",
      "pheno": "1",
      "body_size": "0.00001",
      "assim": "0.1",
      "voracity": "0.4",
      "speed": "0.05",
      "search_area": "0.05",
      "met_rate": "0.4",
      "vorQ10": "1",
      "spdQ10": "1",
      "srchQ10": "1",
      "actE_met": "0.2"
    },
    "maxTraitLimits": {
      "energy_tank": "1",
      "growth": "2.0",
      "pheno": "100",
      "body_size": "394.7",
      "assim": "1",
      "voracity": "0.8",
      "speed": "0.35",
      "search_area": "0.5",
      "met_rate": "0.9",
      "vorQ10": "6",
      "spdQ10": "3",
      "srchQ10": "3",
      "actE_met": "0.9"
    }
  },
  {
    "name": "ara4",
    "huntingMode": "active_hunting",
    "genetics" : {
      "NumberOfLoci": "20",
      "NumberOfAlleles": "10",
      "NumberOfTraits": "13",
      "TRAITS_PER_MODULE": "3"
    },
    "edibleAnimalSpecies" : ["aca4", "ara5", "col1", "col2",
    "aca1", "enc1"],
    "edibleFungusSpecies" : [],
    "initialPopulation": "414",
    "restrict": "0.01",
    "correlationCoefficientRH0": "0.1",
    "A": "0.032",
    "B": "3.229",
    "forDensitiesGrowth": "1.2",
    "forDensitiesEggSize": "0.019",
    "ecosystemSize": "20000",
    "minCondition": "0.01",
    "maxVoracityT": "2.406",
    "keepForSurv": "3.0",
    "maxReproductionEvents": "5",
    "assignedForReproduction": "0.9",
    "forControllingDryBodyMass": "1.0",
    "probabilityDeathFromBackground": "0.0",

    "deadlyTank": "0.1",
    "assignedForMetab": "0.9",
    "numberOfInstars": "12",
    "percentOfTimeSinceLastInstar": "0.8",
    "alphaForPredation": "1.0",
    "minRandomForEncounters": "0.337",

```



```

"maxRandomForEncounters": "0.3375",
"minRandomForPredation": "0.607",
"maxRandomForPredation": "0.612",

"maxEncountersT": "10",
"maxSearchAreaT": "63",

"Q1Ophenology": "0.35",
"Q1Odigestion": "0.25",

"meanSizeHunted": "20.08",
"sdSizeHunted": "7.5",
"meanVorHunted": "13.96",
"sdVorHunted": "4.9",
"meanSpdHunted": "40.65",
"sdSpdHunted": "15",

"meanSizeHunter": "20.08",
"sdSizeHunter": "7.5",
"meanVorHunter": "13.96",
"sdVorHunter": "4.9",
"meanSpdHunter": "40.65",
"sdSpdHunter": "15",
"meanSearchAreaHunter": "0.0",
"sdSearchAreaHunter": "0.0",

"meanSizeXSize": "806.4",
"sdSizeXSize": "275",
"meanVorXVor": "389.6",
"sdVorXVor": "140",
"meanSpdRatio": "25.5",
"sdSpdRatio": "9",
"meanSizeRatio": "397.2",
"sdSizeRatio": "140",

"minTraitsRanges": {
  "energy_tank": "0.25",
  "growth": "1.15",
  "pheno": "3",
  "body_size": "0.009",
  "assim": "0.7",
  "voracity": "0.55",
  "speed": "0.1",
  "search_area": "0.1",
  "met_rate": "0.5",
  "vorQ10": "3",
  "spdQ10": "1.5",
  "srchQ10": "2.0",
  "actE_met": "0.45"
},
"maxTraitsRanges": {
  "energy_tank": "0.5",
  "growth": "1.25",
  "pheno": "11",
  "body_size": "0.029",
  "assim": "0.9",
  "voracity": "0.75",
  "speed": "0.3",
  "search_area": "0.4",
  "met_rate": "0.7",
  "vorQ10": "4",
  "spdQ10": "2.5",
  "srchQ10": "2.5",
  "actE_met": "0.55"
},
"minTraitLimits": {
  "energy_tank": "0.01",
  "growth": "1.01",
  "pheno": "1",
  "body_size": "0.00001",
  "assim": "0.1",
  "voracity": "0.4",
  "speed": "0.05",
  "search_area": "0.05",
  "met_rate": "0.4",
  "vorQ10": "1",
  "spdQ10": "1",
  "srchQ10": "1",
  "actE_met": "0.2"
},
"maxTraitLimits": {

```

```

        "energy_tank": "1",
        "growth": "2.0",
        "pheno": "100",
        "body_size": "394.7",
        "assim": "1",
        "voracity": "0.8",
        "speed": "0.35",
        "search_area": "0.5",
        "met_rate": "0.9",
        "vorQ10": "6",
        "spdQ10": "3",
        "srchQ10": "3",
        "actE_met": "0.9"
    },
    {
        "name": "ara5",
        "huntingMode": "active_hunting",
        "genetics" : {
            "NumberOfLoci": "20",
            "NumberOfAlleles": "10",
            "NumberOfTraits": "13",
            "TRAITS_PER_MODULE": "3"
        },
        "edibleAnimalSpecies" : ["aca4", "col2", "aca1", "aca2",
            "ori1"],
        "edibleFungusSpecies" : [],
        "initialPopulation": "414",
        "restrict": "0.01",
        "correlationCoefficientRH0": "0.1",
        "A": "0.057",
        "B": "2.829",
        "forDensitiesGrowth": "1.2",
        "forDensitiesEggSize": "0.07",
        "ecosystemSize": "20000",
        "minCondition": "0.01",
        "maxVoracityT": "15.434",
        "keepForSurv": "3.0",
        "maxReproductionEvents": "5",
        "assignedForReproduction": "0.9",
        "forControllingDryBodyMass": "1.0",
        "probabilityDeathFromBackground": "0.0",

        "deadlyTank": "0.1",
        "assignedForMetab": "0.9",
        "numberOfInstars": "13",
        "percentOfTimeSinceLastInstar": "0.8",
        "alphaForPredation": "1.0",
        "minRandomForEncounters": "0.337",
        "maxRandomForEncounters": "0.3375",
        "minRandomForPredation": "0.607",
        "maxRandomForPredation": "0.612",

        "maxEncountersT": "10",
        "maxSearchAreaT": "181",

        "Q10phenology": "0.35",
        "Q10digestion": "0.25",

        "meanSizeHunted": "20.08",
        "sdSizeHunted": "7.5",
        "meanVorHunted": "13.96",
        "sdVorHunted": "4.9",
        "meanSpdHunted": "40.65",
        "sdSpdHunted": "15",

        "meanSizeHunter": "20.08",
        "sdSizeHunter": "7.5",
        "meanVorHunter": "13.96",
        "sdVorHunter": "4.9",
        "meanSpdHunter": "40.65",
        "sdSpdHunter": "15",
        "meanSearchAreaHunter": "0.0",
        "sdSearchAreaHunter": "0.0",

        "meanSizeXSize": "806.4",
        "sdSizeXSize": "275",
        "meanVorXVor": "389.6",
        "sdVorXVor": "140",
        "meanSpdRatio": "25.5",
        "sdSpdRatio": "9",

```

```

"meanSizeRatio": "397.2",
"sdSizeRatio": "140",
"minTraitsRanges": {
  "energy_tank": "0.25",
  "growth": "1.15",
  "pheno": "3",
  "body_size": "0.06",
  "assim": "0.7",
  "voracity": "0.55",
  "speed": "0.1",
  "search_area": "0.1",
  "met_rate": "0.5",
  "vorQ10": "3",
  "spdQ10": "1.5",
  "srchQ10": "2.0",
  "actE_met": "0.45"
},
"maxTraitsRanges": {
  "energy_tank": "0.5",
  "growth": "1.25",
  "pheno": "11",
  "body_size": "0.08",
  "assim": "0.9",
  "voracity": "0.75",
  "speed": "0.3",
  "search_area": "0.4",
  "met_rate": "0.7",
  "vorQ10": "4",
  "spdQ10": "2.5",
  "srchQ10": "2.5",
  "actE_met": "0.55"
},
"minTraitLimits": {
  "energy_tank": "0.01",
  "growth": "1.01",
  "pheno": "1",
  "body_size": "0.00001",
  "assim": "0.1",
  "voracity": "0.4",
  "speed": "0.05",
  "search_area": "0.05",
  "met_rate": "0.4",
  "vorQ10": "1",
  "spdQ10": "1",
  "srchQ10": "1",
  "actE_met": "0.2"
},
"maxTraitLimits": {
  "energy_tank": "1",
  "growth": "2.0",
  "pheno": "100",
  "body_size": "394.7",
  "assim": "1",
  "voracity": "0.8",
  "speed": "0.35",
  "search_area": "0.5",
  "met_rate": "0.9",
  "vorQ10": "6",
  "spdQ10": "3",
  "srchQ10": "3",
  "actE_met": "0.9"
}
},
{
  "name": "ara6",
  "huntingMode": "active_hunting",
  "genetics": {
    "NumberOfLoci": "20",
    "NumberOfAlleles": "10",
    "NumberOfTraits": "13",
    "TRAITS_PER_MODULE": "3"
  },
  "edibleAnimalSpecies": ["lit1", "lit2", "ara1", "ara2",
    "aca3", "geo1", "ara4", "ara5", "col1", "ara3", "aca5", "enc1",
    "aca6", "ara6"],
  "edibleFungusSpecies": [],
  "initialPopulation": "414",
  "restrict": "0.01",
  "correlationCoefficientRRHO": "0.1",
  "A": "0.063",

```

```

"B": "2.530",
"forDensitiesGrowth": "1.2",
"forDensitiesEggSize": "0.0046",
"ecosystemSize": "20000",
"minCondition": "0.01",
"maxVoracityT": "0.344",
"keepForSurv": "3.0",
"maxReproductionEvents": "5",
"assignedForReproduction": "0.9",
"forControllingDryBodyMass": "1.0",
"probabilityDeathFromBackground": "0.0",

"deadlyTank": "0.1",
"assignedForMetab": "0.9",
"numberOfInstars": "6",
"percentOfTimeSinceLastInstar": "0.8",
"alphaForPredation": "1.0",
"minRandomForEncounters": "0.337",
"maxRandomForEncounters": "0.3375",
"minRandomForPredation": "0.607",
"maxRandomForPredation": "0.612",

"maxEncountersT": "10",
"maxSearchAreaT": "21",

"Q10phenology": "0.35",
"Q10digestion": "0.25",

"meanSizeHunted": "20.08",
"sdSizeHunted": "7.5",
"meanVorHunted": "13.96",
"sdVorHunted": "4.9",
"meanSpdHunted": "40.65",
"sdSpdHunted": "15",

"meanSizeHunter": "20.08",
"sdSizeHunter": "7.5",
"meanVorHunter": "13.96",
"sdVorHunter": "4.9",
"meanSpdHunter": "40.65",
"sdSpdHunter": "15",
"meanSearchAreaHunter": "0.0",
"sdSearchAreaHunter": "0.0",

"meanSizeXSize": "806.4",
"sdSizeXSize": "275",
"meanVorXVor": "389.6",
"sdVorXVor": "140",
"meanSpdRatio": "25.5",
"sdSpdRatio": "9",
"meanSizeRatio": "397.2",
"sdSizeRatio": "140",

"minTraitsRanges": {
  "energy_tank": "0.25",
  "growth": "1.15",
  "pheno": "3",
  "body_size": "0.0035",
  "assim": "0.7",
  "voracity": "0.55",
  "speed": "0.1",
  "search_area": "0.1",
  "met_rate": "0.5",
  "vorQ10": "3",
  "spdQ10": "1.5",
  "srchQ10": "2.0",
  "actE_met": "0.45"
},
"maxTraitsRanges": {
  "energy_tank": "0.5",
  "growth": "1.25",
  "pheno": "11",
  "body_size": "0.0045",
  "assim": "0.9",
  "voracity": "0.75",
  "speed": "0.3",
  "search_area": "0.4",
  "met_rate": "0.7",
  "vorQ10": "4",
  "spdQ10": "2.5",
  "srchQ10": "2.5",

```

```

        "actE_met": "0.55"
    },
    "minTraitLimits": {
        "energy_tank": "0.01",
        "growth": "1.01",
        "pheno": "1",
        "body_size": "0.00001",
        "assim": "0.1",
        "voracity": "0.4",
        "speed": "0.05",
        "search_area": "0.05",
        "met_rate": "0.4",
        "vorQ10": "1",
        "spdQ10": "1",
        "srchQ10": "1",
        "actE_met": "0.2"
    },
    "maxTraitLimits": {
        "energy_tank": "1",
        "growth": "2.0",
        "pheno": "100",
        "body_size": "394.7",
        "assim": "1",
        "voracity": "0.8",
        "speed": "0.35",
        "search_area": "0.5",
        "met_rate": "0.9",
        "vorQ10": "6",
        "spdQ10": "3",
        "srchQ10": "3",
        "actE_met": "0.9"
    }
},
{
    "name": "col1",
    "huntingMode": "does_not_hunt",
    "genetics" : {
        "NumberOfLoci": "20",
        "NumberOfAlleles": "10",
        "NumberOfTraits": "13",
        "TRAITS_PER_MODULE": "3"
    },
    "edibleAnimalSpecies" : [],
    "edibleFungusSpecies" : ["Fungus_X"],
    "initialPopulation": "1171",
    "restrict": "0.01",
    "correlationCoefficientRHO": "0.1",
    "A": "0.0056",
    "B": "2.809",
    "forDensitiesGrowth": "1.3",
    "forDensitiesEggSize": "0.002",
    "ecosystemSize": "20000",
    "minCondition": "0.01",
    "maxVoracityT": "0.080",
    "keepForSurv": "3.0",
    "maxReproductionEvents": "2",
    "assignedForReproduction": "0.8",
    "forControllingDryBodyMass": "0.3",
    "probabilityDeathFromBackground": "0.0",

    "deadlyTank": "0.1",
    "assignedForMetab": "0.75",
    "numberOfInstars": "4",
    "percentOfTimeSinceLastInstar": "3.0",
    "alphaForPredation": "1.0",
    "minRandomForEncounters": "0.337",
    "maxRandomForEncounters": "0.339",
    "minRandomForPredation": "0.607",
    "maxRandomForPredation": "0.617",

    "maxEncountersT": "10",
    "maxSearchAreaT": "9",

    "Q10phenology": "0.35",
    "Q10digestion": "0.25",

    "meanSizeHunted": "20.08",
    "sdSizeHunted": "7.5",
    "meanVorHunted": "13.96",
    "sdVorHunted": "4.9",

```

```

"meanSpdHunted": "40.65",
"sdSpdHunted": "15",

"meanSizeHunter": "20.08",
"sdSizeHunter": "7.5",
"meanVorHunter": "13.96",
"sdVorHunter": "4.9",
"meanSpdHunter": "40.65",
"sdSpdHunter": "15",
"meanSearchAreaHunter": "0.0",
"sdSearchAreaHunter": "0.0",

"meanSizeXSize": "806.4",
"sdSizeXSize": "275",
"meanVorXVor": "389.6",
"sdVorXVor": "140",
"meanSpdRatio": "25.5",
"sdSpdRatio": "9",
"meanSizeRatio": "397.2",
"sdSizeRatio": "140",

"minTraitsRanges": {
  "energy_tank": "0.25",
  "growth": "1.25",
  "pheno": "3",
  "body_size": "0.001",
  "assim": "0.7",
  "voracity": "0.55",
  "speed": "0.1",
  "search_area": "0.1",
  "met_rate": "0.65",
  "vorQ10": "2",
  "spdQ10": "1.5",
  "srchQ10": "1.5",
  "actE_met": "0.61"
},
"maxTraitsRanges": {
  "energy_tank": "0.5",
  "growth": "1.35",
  "pheno": "11",
  "body_size": "0.003",
  "assim": "0.9",
  "voracity": "0.75",
  "speed": "0.3",
  "search_area": "0.3",
  "met_rate": "0.85",
  "vorQ10": "4",
  "spdQ10": "2.5",
  "srchQ10": "2.5",
  "actE_met": "0.70"
},
"minTraitLimits": {
  "energy_tank": "0.01",
  "growth": "1.01",
  "pheno": "1",
  "body_size": "0.00001",
  "assim": "0",
  "voracity": "0.4",
  "speed": "0.05",
  "search_area": "0.05",
  "met_rate": "0.4",
  "vorQ10": "1",
  "spdQ10": "1",
  "srchQ10": "1",
  "actE_met": "0.2"
},
"maxTraitLimits": {
  "energy_tank": "1",
  "growth": "2.0",
  "pheno": "100",
  "body_size": "394.7",
  "assim": "1",
  "voracity": "0.8",
  "speed": "0.35",
  "search_area": "0.5",
  "met_rate": "0.9",
  "vorQ10": "6",
  "spdQ10": "3",
  "srchQ10": "3",
  "actE_met": "0.9"
}

```

```

},
{
  "name": "col2",
  "huntingMode": "does_not_hunt",
  "genetics" : {
    "NumberOfLoci": "20",
    "NumberOfAlleles": "10",
    "NumberOfTraits": "13",
    "TRAITS_PER_MODULE": "3"
  },
  "edibleAnimalSpecies" : [],
  "edibleFungusSpecies" : ["Fungus_X"],
  "initialPopulation": "1171",
  "restrict": "0.01",
  "correlationCoefficientRHO": "0.1",
  "A": "0.0056",
  "B": "2.809",
  "forDensitiesGrowth": "1.4",
  "forDensitiesEggSize": "0.014",
  "ecosystemSize": "20000",
  "minCondition": "0.01",
  "maxVoracityT": "1.566",
  "keepForSurv": "3.0",
  "maxReproductionEvents": "2",
  "assignedForReproduction": "0.8",
  "forControllingDryBodyMass": "0.3",
  "probabilityDeathFromBackground": "0.0",

  "deadlyTank": "0.1",
  "assignedForMetab": "0.75",
  "numberOfInstars": "6",
  "percentOfTimeSinceLastInstar": "3.0",
  "alphaForPredation": "1.0",
  "minRandomForEncounters": "0.337",
  "maxRandomForEncounters": "0.339",
  "minRandomForPredation": "0.607",
  "maxRandomForPredation": "0.617",

  "maxEncountersT": "10",
  "maxSearchAreaT": "49",

  "Q10phenology": "0.35",
  "Q10digestion": "0.25",

  "meanSizeHunted": "20.08",
  "sdSizeHunted": "7.5",
  "meanVorHunted": "13.96",
  "sdVorHunted": "4.9",
  "meanSpdHunted": "40.65",
  "sdSpdHunted": "15",

  "meanSizeHunter": "20.08",
  "sdSizeHunter": "7.5",
  "meanVorHunter": "13.96",
  "sdVorHunter": "4.9",
  "meanSpdHunter": "40.65",
  "sdSpdHunter": "15",
  "meanSearchAreaHunter": "0.0",
  "sdSearchAreaHunter": "0.0",

  "meanSizeXSize": "806.4",
  "sdSizeXSize": "275",
  "meanVorXVor": "389.6",
  "sdVorXVor": "140",
  "meanSpdRatio": "25.5",
  "sdSpdRatio": "9",
  "meanSizeRatio": "397.2",
  "sdSizeRatio": "140",

  "minTraitsRanges": {
    "energy_tank": "0.25",
    "growth": "1.35",
    "pheno": "3",
    "body_size": "0.004",
    "assim": "0.7",
    "voracity": "0.55",
    "speed": "0.1",
    "search_area": "0.1",
    "met_rate": "0.65",
    "vorQ10": "2",
    "spdQ10": "1.5",
  }
}

```

```

        "srchQ10": "1.5",
        "actE_met": "0.61"
    },
    "maxTraitsRanges": {
        "energy_tank": "0.5",
        "growth": "1.45",
        "pheno": "11",
        "body_size": "0.024",
        "assim": "0.9",
        "voracity": "0.75",
        "speed": "0.3",
        "search_area": "0.3",
        "met_rate": "0.85",
        "vorQ10": "4",
        "spdQ10": "2.5",
        "srchQ10": "2.5",
        "actE_met": "0.70"
    },
    "minTraitLimits": {
        "energy_tank": "0.01",
        "growth": "1.01",
        "pheno": "1",
        "body_size": "0.00001",
        "assim": "0",
        "voracity": "0.4",
        "speed": "0.05",
        "search_area": "0.05",
        "met_rate": "0.4",
        "vorQ10": "1",
        "spdQ10": "1",
        "srchQ10": "1",
        "actE_met": "0.2"
    },
    "maxTraitLimits": {
        "energy_tank": "1",
        "growth": "2.0",
        "pheno": "100",
        "body_size": "394.7",
        "assim": "1",
        "voracity": "0.8",
        "speed": "0.35",
        "search_area": "0.5",
        "met_rate": "0.9",
        "vorQ10": "6",
        "spdQ10": "3",
        "srchQ10": "3",
        "actE_met": "0.9"
    }
},
{
    "name": "enc1",
    "huntingMode": "does_not_hunt",
    "genetics": {
        "NumberOfLoci": "20",
        "NumberOfAlleles": "10",
        "NumberOfTraits": "13",
        "TRAITS_PER_MODULE": "3"
    },
    "edibleAnimalSpecies": [],
    "edibleFungusSpecies": ["Fungus_X"],
    "initialPopulation": "305",
    "restrict": "0.01",
    "correlationCoefficientRH0": "0.1",
    "A": "0.0039",
    "B": "2.529",
    "forDensitiesGrowth": "1.4",
    "forDensitiesEggSize": "0.001",
    "ecosystemSize": "20000",
    "minCondition": "0.01",
    "maxVoracityT": "0.052",
    "keepForSurv": "3.0",
    "maxReproductionEvents": "2",
    "assignedForReproduction": "0.8",
    "forControllingDryBodyMass": "1.0",
    "probabilityDeathFromBackground": "0.0",

    "deadlyTank": "0.1",
    "assignedForMetab": "0.75",
    "numberOfInstars": "6",
    "percentOfTimeSinceLastInstar": "3.0",
    "alphaForPredation": "1.0",

```



```

"minRandomForEncounters": "0.337",
"maxRandomForEncounters": "0.339",
"minRandomForPredation": "0.607",
"maxRandomForPredation": "0.617",

"maxEncountersT": "10",
"maxSearchAreaT": "7",

"Q10phenology": "0.35",
"Q10digestion": "0.25",

"meanSizeHunted": "20.08",
"sdSizeHunted": "7.5",
"meanVorHunted": "13.96",
"sdVorHunted": "4.9",
"meanSpdHunted": "40.65",
"sdSpdHunted": "15",

"meanSizeHunter": "20.08",
"sdSizeHunter": "7.5",
"meanVorHunter": "13.96",
"sdVorHunter": "4.9",
"meanSpdHunter": "40.65",
"sdSpdHunter": "15",
"meanSearchAreaHunter": "0.0",
"sdSearchAreaHunter": "0.0",

"meanSizeXSize": "806.4",
"sdSizeXSize": "275",
"meanVorXVor": "389.6",
"sdVorXVor": "140",
"meanSpdRatio": "25.5",
"sdSpdRatio": "9",
"meanSizeRatio": "397.2",
"sdSizeRatio": "140",

"minTraitsRanges": {
  "energy_tank": "0.25",
  "growth": "1.35",
  "pheno": "3",
  "body_size": "0.0001",
  "assim": "0.5",
  "voracity": "0.55",
  "speed": "0.1",
  "search_area": "0.1",
  "met_rate": "0.7",
  "vorQ10": "2",
  "spdQ10": "1.5",
  "srchQ10": "1.5",
  "actE_met": "0.39"
},
"maxTraitsRanges": {
  "energy_tank": "0.5",
  "growth": "1.45",
  "pheno": "11",
  "body_size": "0.002",
  "assim": "0.7",
  "voracity": "0.75",
  "speed": "0.3",
  "search_area": "0.3",
  "met_rate": "0.9",
  "vorQ10": "4",
  "spdQ10": "2.5",
  "srchQ10": "2.5",
  "actE_met": "0.49"
},
"minTraitLimits": {
  "energy_tank": "0.01",
  "growth": "1.01",
  "pheno": "1",
  "body_size": "0.0001",
  "assim": "0",
  "voracity": "0.4",
  "speed": "0.05",
  "search_area": "0.05",
  "met_rate": "0.4",
  "vorQ10": "1",
  "spdQ10": "1",
  "srchQ10": "1",
  "actE_met": "0.2"
},

```

```

    "maxTraitLimits": {
      "energy_tank": "1",
      "growth": "2.0",
      "pheno": "100",
      "body_size": "394.7",
      "assim": "1",
      "voracity": "0.8",
      "speed": "0.35",
      "search_area": "0.5",
      "met_rate": "0.9",
      "vorQ10": "6",
      "spdQ10": "3",
      "srchQ10": "3",
      "actE_met": "0.9"
    }
  },
  {
    "name": "geo1",
    "huntingMode": "active_hunting",
    "genetics": {
      "NumberOfLoci": "20",
      "NumberOfAlleles": "10",
      "NumberOfTraits": "13",
      "TRAITS_PER_MODULE": "3"
    },
    "edibleAnimalSpecies": ["lit1", "lit2", "ara1", "ara2",
      "aca3", "geo1", "ara4", "ara5", "col1", "ara3", "aca5", "enc1",
      "ara6"],
    "edibleFungusSpecies": [],
    "initialPopulation": "51",
    "restrict": "0.01",
    "correlationCoefficientRRH0": "0.1",
    "A": "0.0036",
    "B": "2.626",
    "forDensitiesGrowth": "1.2",
    "forDensitiesEggSize": "0.078",
    "ecosystemSize": "20000",
    "minCondition": "0.01",
    "maxVoracityT": "18.009",
    "keepForSurv": "3.0",
    "maxReproductionEvents": "5",
    "assignedForReproduction": "0.9",
    "forControllingDryBodyMass": "1.0",
    "probabilityDeathFromBackground": "0.0",

    "deadlyTank": "0.1",
    "assignedForMetab": "0.9",
    "numberOfInstars": "14",
    "percentOfTimeSinceLastInstar": "0.8",
    "alphaForPredation": "1.0",
    "minRandomForEncounters": "0.337",
    "maxRandomForEncounters": "0.3375",
    "minRandomForPredation": "0.607",
    "maxRandomForPredation": "0.612",

    "maxEncountersT": "10",
    "maxSearchAreaT": "198",

    "Q10phenology": "0.35",
    "Q10digestion": "0.25",

    "meanSizeHunted": "20.08",
    "sdSizeHunted": "7.5",
    "meanVorHunted": "13.96",
    "sdVorHunted": "4.9",
    "meanSpdHunted": "40.65",
    "sdSpdHunted": "15",

    "meanSizeHunter": "20.08",
    "sdSizeHunter": "7.5",
    "meanVorHunter": "13.96",
    "sdVorHunter": "4.9",
    "meanSpdHunter": "40.65",
    "sdSpdHunter": "15",
    "meanSearchAreaHunter": "0.0",
    "sdSearchAreaHunter": "0.0",

    "meanSizeXSize": "806.4",
    "sdSizeXSize": "275",
    "meanVorXVor": "389.6",
    "sdVorXVor": "140",

```

```

"meanSpdRatio": "25.5",
"sdSpdRatio": "9",
"meanSizeRatio": "397.2",
"sdSizeRatio": "140",

"minTraitsRanges": {
  "energy_tank": "0.25",
  "growth": "1.15",
  "pheno": "3",
  "body_size": "0.09",
  "assim": "0.7",
  "voracity": "0.55",
  "speed": "0.1",
  "search_area": "0.1",
  "met_rate": "0.46",
  "vorQ10": "3",
  "spdQ10": "1.5",
  "srchQ10": "2.0",
  "actE_met": "0.45"
},
"maxTraitsRanges": {
  "energy_tank": "0.5",
  "growth": "1.25",
  "pheno": "11",
  "body_size": "0.10",
  "assim": "0.9",
  "voracity": "0.75",
  "speed": "0.3",
  "search_area": "0.4",
  "met_rate": "0.66",
  "vorQ10": "4",
  "spdQ10": "2.5",
  "srchQ10": "2.5",
  "actE_met": "0.55"
},
"minTraitLimits": {
  "energy_tank": "0.01",
  "growth": "1.01",
  "pheno": "1",
  "body_size": "0.00001",
  "assim": "0.1",
  "voracity": "0.4",
  "speed": "0.05",
  "search_area": "0.05",
  "met_rate": "0.4",
  "vorQ10": "1",
  "spdQ10": "1",
  "srchQ10": "1",
  "actE_met": "0.2"
},
"maxTraitLimits": {
  "energy_tank": "1",
  "growth": "2.0",
  "pheno": "100",
  "body_size": "394.7",
  "assim": "1",
  "voracity": "0.8",
  "speed": "0.35",
  "search_area": "0.5",
  "met_rate": "0.9",
  "vorQ10": "6",
  "spdQ10": "3",
  "srchQ10": "3",
  "actE_met": "0.9"
}
},
{
  "name": "geo2",
  "huntingMode": "active_hunting",
  "genetics": {
    "NumberOfLoci": "20",
    "NumberOfAlleles": "10",
    "NumberOfTraits": "13",
    "TRAITS_PER_MODULE": "3"
  },
  "edibleAnimalSpecies": ["lit1", "lit2", "ara1", "ara2",
    "aca3", "geol", "aca4", "ara4", "ara5", "col1", "col2", "ara3",
    "aca5", "aca1", "enc1", "aca6", "ara6", "geo2"],
  "edibleFungusSpecies": [],
  "initialPopulation": "51",
  "restrict": "0.01",

```

```

"correlationCoefficientRH0": "0.1",
"A": "0.0036",
"B": "2.626",
"forDensitiesGrowth": "1.2",
"forDensitiesEggSize": "0.061",
"ecosystemSize": "20000",
"minCondition": "0.01",
"maxVoracityT": "12.559",
"keepForSurv": "3.0",
"maxReproductionEvents": "5",
"assignedForReproduction": "0.9",
"forControllingDryBodyMass": "1.0",
"probabilityDeathFromBackground": "0.0",

"deadlyTank": "0.1",
"assignedForMetab": "0.9",
"numberOfInstars": "13",
"percentOfTimeSinceLastInstar": "0.8",
"alphaForPredation": "1.0",
"minRandomForEncounters": "0.337",
"maxRandomForEncounters": "0.3375",
"minRandomForPredation": "0.607",
"maxRandomForPredation": "0.612",

"maxEncountersT": "10",
"maxSearchAreaT": "161",

"Q10phenology": "0.35",
"Q10digestion": "0.25",

"meanSizeHunted": "20.08",
"sdSizeHunted": "7.5",
"meanVorHunted": "13.96",
"sdVorHunted": "4.9",
"meanSpdHunted": "40.65",
"sdSpdHunted": "15",

"meanSizeHunter": "20.08",
"sdSizeHunter": "7.5",
"meanVorHunter": "13.96",
"sdVorHunter": "4.9",
"meanSpdHunter": "40.65",
"sdSpdHunter": "15",
"meanSearchAreaHunter": "0.0",
"sdSearchAreaHunter": "0.0",

"meanSizeXSize": "806.4",
"sdSizeXSize": "275",
"meanVorXVor": "389.6",
"sdVorXVor": "140",
"meanSpdRatio": "25.5",
"sdSpdRatio": "9",
"meanSizeRatio": "397.2",
"sdSizeRatio": "140",

"minTraitsRanges": {
  "energy_tank": "0.25",
  "growth": "1.15",
  "pheno": "3",
  "body_size": "0.05",
  "assim": "0.7",
  "voracity": "0.55",
  "speed": "0.1",
  "search_area": "0.1",
  "met_rate": "0.46",
  "vorQ10": "3",
  "spdQ10": "1.5",
  "srchQ10": "2.0",
  "actE_met": "0.45"
},
"maxTraitsRanges": {
  "energy_tank": "0.5",
  "growth": "1.25",
  "pheno": "11",
  "body_size": "0.07",
  "assim": "0.9",
  "voracity": "0.75",
  "speed": "0.3",
  "search_area": "0.4",
  "met_rate": "0.66",
  "vorQ10": "4",

```

```

        "spdQ10": "2.5",
        "srchQ10": "2.5",
        "actE_met": "0.55"
    },
    "minTraitLimits": {
        "energy_tank": "0.01",
        "growth": "1.01",
        "pheno": "1",
        "body_size": "0.00001",
        "assim": "0.1",
        "voracity": "0.4",
        "speed": "0.05",
        "search_area": "0.05",
        "met_rate": "0.4",
        "vorQ10": "1",
        "spdQ10": "1",
        "srchQ10": "1",
        "actE_met": "0.2"
    },
    "maxTraitLimits": {
        "energy_tank": "1",
        "growth": "2.0",
        "pheno": "100",
        "body_size": "394.7",
        "assim": "1",
        "voracity": "0.8",
        "speed": "0.35",
        "search_area": "0.5",
        "met_rate": "0.9",
        "vorQ10": "6",
        "spdQ10": "3",
        "srchQ10": "3",
        "actE_met": "0.9"
    }
},
{
    "name": "lit1",
    "huntingMode": "active_hunting",
    "genetics" : {
        "NumberOfLoci": "20",
        "NumberOfAlleles": "10",
        "NumberOfTraits": "13",
        "TRAITS_PER_MODULE": "3"
    },
    "edibleAnimalSpecies" : ["lit1","lit2","ara1","ara2",
    "aca3","geol","ara4","ara5","col1","ara3","aca5",
    "enc1","aca6","ara6"],
    "edibleFungusSpecies" : [],
    "initialPopulation": "51",
    "restrict": "0.01",
    "correlationCoefficientRH0": "0.1",
    "A": "0.0036",
    "B": "2.626",
    "forDensitiesGrowth": "1.2",
    "forDensitiesEggSize": "0.055",
    "ecosystemSize": "20000",
    "minCondition": "0.01",
    "maxVoracityT": "11.024",
    "keepForSurv": "3.0",
    "maxReproductionEvents": "5",
    "assignedForReproduction": "0.9",
    "forControllingDryBodyMass": "1.0",
    "probabilityDeathFromBackground": "0.0",

    "deadlyTank": "0.1",
    "assignedForMetab": "0.9",
    "numberOfInstars": "13",
    "percentOfTimeSinceLastInstar": "0.8",
    "alphaForPredation": "1.0",
    "minRandomForEncounters": "0.337",
    "maxRandomForEncounters": "0.3375",
    "minRandomForPredation": "0.607",
    "maxRandomForPredation": "0.612",

    "maxEncountersT": "10",
    "maxSearchAreaT": "150",

    "Q10phenology": "0.35",
    "Q10digestion": "0.25",

```

```

"meanSizeHunted": "20.08",
"sdSizeHunted": "7.5",
"meanVorHunted": "13.96",
"sdVorHunted": "4.9",
"meanSpdHunted": "40.65",
"sdSpdHunted": "15",

"meanSizeHunter": "20.08",
"sdSizeHunter": "7.5",
"meanVorHunter": "13.96",
"sdVorHunter": "4.9",
"meanSpdHunter": "40.65",
"sdSpdHunter": "15",
"meanSearchAreaHunter": "0.0",
"sdSearchAreaHunter": "0.0",

"meanSizeXSize": "806.4",
"sdSizeXSize": "275",
"meanVorXVor": "389.6",
"sdVorXVor": "140",
"meanSpdRatio": "25.5",
"sdSpdRatio": "9",
"meanSizeRatio": "397.2",
"sdSizeRatio": "140",

"minTraitsRanges": {
  "energy_tank": "0.25",
  "growth": "1.15",
  "pheno": "3",
  "body_size": "0.045",
  "assim": "0.7",
  "voracity": "0.55",
  "speed": "0.1",
  "search_area": "0.1",
  "met_rate": "0.45",
  "vorQ10": "3",
  "spdQ10": "1.5",
  "srchQ10": "2.0",
  "actE_met": "0.45"
},
"maxTraitsRanges": {
  "energy_tank": "0.5",
  "growth": "1.25",
  "pheno": "11",
  "body_size": "0.065",
  "assim": "0.9",
  "voracity": "0.75",
  "speed": "0.3",
  "search_area": "0.4",
  "met_rate": "0.65",
  "vorQ10": "4",
  "spdQ10": "2.5",
  "srchQ10": "2.5",
  "actE_met": "0.55"
},
"minTraitLimits": {
  "energy_tank": "0.01",
  "growth": "1.01",
  "pheno": "1",
  "body_size": "0.00001",
  "assim": "0.1",
  "voracity": "0.4",
  "speed": "0.05",
  "search_area": "0.05",
  "met_rate": "0.4",
  "vorQ10": "1",
  "spdQ10": "1",
  "srchQ10": "1",
  "actE_met": "0.2"
},
"maxTraitLimits": {
  "energy_tank": "1",
  "growth": "2.0",
  "pheno": "100",
  "body_size": "394.7",
  "assim": "1",
  "voracity": "0.8",
  "speed": "0.35",
  "search_area": "0.5",
  "met_rate": "0.9",
  "vorQ10": "6",

```

```

        "spdQ10": "3",
        "srchQ10": "3",
        "actE_met": "0.9"
    },
    {
        "name": "lit2",
        "huntingMode": "active_hunting",
        "genetics" : {
            "NumberOfLoci": "20",
            "NumberOfAlleles": "10",
            "NumberOfTraits": "13",
            "TRAITS_PER_MODULE": "3"
        },
        "edibleAnimalSpecies" : ["lit1","lit2","ara1","ara2",
            "aca3","geo1","ara4","ara3","aca5","enc1","aca6",
            "ara6","geo2"],
        "edibleFungusSpecies" : [],
        "initialPopulation": "51",
        "restrict": "0.01",
        "correlationCoefficientRH0": "0.1",
        "A": "0.0036",
        "B": "2.626",
        "forDensitiesGrowth": "1.2",
        "forDensitiesEggSize": "0.047",
        "ecosystemSize": "20000",
        "minCondition": "0.01",
        "maxVoracityT": "8.784",
        "keepForSurv": "3.0",
        "maxReproductionEvents": "5",
        "assignedForReproduction": "0.9",
        "forControllingDryBodyMass": "1.0",
        "probabilityDeathFromBackground": "0.0",

        "deadlyTank": "0.1",
        "assignedForMetab": "0.9",
        "numberOfInstars": "13",
        "percentOfTimeSinceLastInstar": "0.8",
        "alphaForPredation": "1.0",
        "minRandomForEncounters": "0.337",
        "maxRandomForEncounters": "0.3375",
        "minRandomForPredation": "0.607",
        "maxRandomForPredation": "0.612",

        "maxEncountersT": "10",
        "maxSearchAreaT": "131",

        "Q10phenology": "0.35",
        "Q10digestion": "0.25",

        "meanSizeHunted": "20.08",
        "sdSizeHunted": "7.5",
        "meanVorHunted": "13.96",
        "sdVorHunted": "4.9",
        "meanSpdHunted": "40.65",
        "sdSpdHunted": "15",

        "meanSizeHunter": "20.08",
        "sdSizeHunter": "7.5",
        "meanVorHunter": "13.96",
        "sdVorHunter": "4.9",
        "meanSpdHunter": "40.65",
        "sdSpdHunter": "15",
        "meanSearchAreaHunter": "0.0",
        "sdSearchAreaHunter": "0.0",

        "meanSizeXSize": "806.4",
        "sdSizeXSize": "275",
        "meanVorXVor": "389.6",
        "sdVorXVor": "140",
        "meanSpdRatio": "25.5",
        "sdSpdRatio": "9",
        "meanSizeRatio": "397.2",
        "sdSizeRatio": "140",

        "minTraitsRanges": {
            "energy_tank": "0.25",
            "growth": "1.15",
            "pheno": "3",
            "body_size": "0.040",
        }
    }
}

```

```

        "assim": "0.7",
        "voracity": "0.55",
        "speed": "0.1",
        "search_area": "0.1",
        "met_rate": "0.45",
        "vorQ10": "3",
        "spdQ10": "1.5",
        "srchQ10": "2.0",
        "actE_met": "0.45"
    },
    "maxTraitsRanges": {
        "energy_tank": "0.5",
        "growth": "1.25",
        "pheno": "11",
        "body_size": "0.05",
        "assim": "0.9",
        "voracity": "0.75",
        "speed": "0.3",
        "search_area": "0.4",
        "met_rate": "0.65",
        "vorQ10": "4",
        "spdQ10": "2.5",
        "srchQ10": "2.5",
        "actE_met": "0.55"
    },
    "minTraitLimits": {
        "energy_tank": "0.01",
        "growth": "1.01",
        "pheno": "1",
        "body_size": "0.00001",
        "assim": "0.1",
        "voracity": "0.4",
        "speed": "0.05",
        "search_area": "0.05",
        "met_rate": "0.4",
        "vorQ10": "1",
        "spdQ10": "1",
        "srchQ10": "1",
        "actE_met": "0.2"
    },
    "maxTraitLimits": {
        "energy_tank": "1",
        "growth": "2.0",
        "pheno": "100",
        "body_size": "394.7",
        "assim": "1",
        "voracity": "0.8",
        "speed": "0.35",
        "search_area": "0.5",
        "met_rate": "0.9",
        "vorQ10": "6",
        "spdQ10": "3",
        "srchQ10": "3",
        "actE_met": "0.9"
    }
},
{
    "name": "ori1",
    "huntingMode": "does_not_hunt",
    "genetics": {
        "NumberOfLoci": "20",
        "NumberOfAlleles": "10",
        "NumberOfTraits": "13",
        "TRAITS_PER_MODULE": "3"
    },
    "edibleAnimalSpecies": [],
    "edibleFungusSpecies": ["Fungus_X"],
    "initialPopulation": "458",
    "restrict": "0.01",
    "correlationCoefficientRRH0": "0.1",
    "A": "0.053",
    "B": "2.494",
    "forDensitiesGrowth": "1.4",
    "forDensitiesEggSize": "0.004",
    "ecosystemSize": "20000",
    "minCondition": "0.01",
    "maxVoracityT": "0.308",
    "keepForSurv": "3.0",
    "maxReproductionEvents": "2",
    "assignedForReproduction": "0.8",
    "forControllingDryBodyMass": "1.0",

```



```

"probabilityDeathFromBackground": "0.0",
"deadlyTank": "0.1",
"assignedForMetab": "0.75",
"numberOfInstars": "5",
"percentOfTimeSinceLastInstar": "3.0",
"alphaForPredation": "1.0",
"minRandomForEncounters": "0.337",
"maxRandomForEncounters": "0.339",
"minRandomForPredation": "0.607",
"maxRandomForPredation": "0.617",

"maxEncountersT": "10",
"maxSearchAreaT": "19",

"Q10phenology": "0.35",
"Q10digestion": "0.25",

"meanSizeHunted": "20.08",
"sdSizeHunted": "7.5",
"meanVorHunted": "13.96",
"sdVorHunted": "4.9",
"meanSpdHunted": "40.65",
"sdSpdHunted": "15",

"meanSizeHunter": "20.08",
"sdSizeHunter": "7.5",
"meanVorHunter": "13.96",
"sdVorHunter": "4.9",
"meanSpdHunter": "40.65",
"sdSpdHunter": "15",
"meanSearchAreaHunter": "0.0",
"sdSearchAreaHunter": "0.0",

"meanSizeXSize": "806.4",
"sdSizeXSize": "275",
"meanVorXVor": "389.6",
"sdVorXVor": "140",
"meanSpdRatio": "25.5",
"sdSpdRatio": "9",
"meanSizeRatio": "397.2",
"sdSizeRatio": "140",

"minTraitsRanges": {
  "energy_tank": "0.25",
  "growth": "1.35",
  "pheno": "3",
  "body_size": "0.003",
  "assim": "0.7",
  "voracity": "0.55",
  "speed": "0.1",
  "search_area": "0.1",
  "met_rate": "0.57",
  "vorQ10": "2",
  "spdQ10": "1.5",
  "srchQ10": "1.5",
  "actE_met": "0.66"
},
"maxTraitsRanges": {
  "energy_tank": "0.5",
  "growth": "1.45",
  "pheno": "11",
  "body_size": "0.005",
  "assim": "0.9",
  "voracity": "0.75",
  "speed": "0.3",
  "search_area": "0.3",
  "met_rate": "0.77",
  "vorQ10": "4",
  "spdQ10": "2.5",
  "srchQ10": "2.5",
  "actE_met": "0.76"
},
"minTraitLimits": {
  "energy_tank": "0.01",
  "growth": "1.01",
  "pheno": "1",
  "body_size": "0.00001",
  "assim": "0.1",
  "voracity": "0.4",
  "speed": "0.05",

```

```

        "search_area": "0.05",
        "met_rate": "0.4",
        "vorQ10": "1",
        "spdQ10": "1",
        "srchQ10": "1",
        "actE_met": "0.2"
    },
    "maxTraitLimits": {
        "energy_tank": "1",
        "growth": "2.0",
        "pheno": "100",
        "body_size": "394.7",
        "assim": "1",
        "voracity": "0.8",
        "speed": "0.35",
        "search_area": "0.5",
        "met_rate": "0.9",
        "vorQ10": "6",
        "spdQ10": "3",
        "srchQ10": "3",
        "actE_met": "0.9"
    }
},
"fungi": [
    {
        "name": "Fungus_X",
        "sporeMass": "4",
        "minimumFungus": "0",
        "ACTIVATION_ENERGY": "0.68",
        "NORMALIZATION_B": "25.98",
        "minHR": "85",
        "maxHR": "90",
        "maxRScale": "0.5",
        "zeroFungi": "8",
        "patches": [
            {
                "type": "sphere",
                "radius": "5",
                "xPos": "5",
                "yPos": "5",
                "zPos": "5",
                "value": "1.5"
            },
            {
                "type": "sphere",
                "radius": "5",
                "xPos": "5",
                "yPos": "15",
                "zPos": "5",
                "value": "1.5"
            },
            {
                "type": "sphere",
                "radius": "5",
                "xPos": "15",
                "yPos": "5",
                "zPos": "5",
                "value": "1.5"
            },
            {
                "type": "sphere",
                "radius": "5",
                "xPos": "15",
                "yPos": "15",
                "zPos": "5",
                "value": "1.5"
            }
        ]
    }
]
},
"simulation": {
    "runDays": "200",
    "outputDirectory": "generalist_20x20_restrict001_WITH_PREDS_rep2",
    "saveIntermediateVolumes": "false",
    "saveIntermediateVolumesPeriodicity": "100000",
    "encountersMatrixFilename": "encountersMatrix",
    "predationsMatrixFilename": "predationsMatrix",

```

```
"nodesMatrixFilename": "nodesMatrix",  
"predationEventsOnOtherSpeciesFilename": "predationOnSpecies"  
}
```

Bibliografía

- [1] Douglas J. Futuyma. *Evolutionary Biology*. Sinauer, 3d ed. Sunderland, MA, 1998 1998.
- [2] M. Begon, C.R. Townsend, and J.L. Harper. *Ecology: From Individuals to Ecosystems*. Wiley, 2009. ISBN 9781405151986. URL <http://books.google.es/books?id=Lsf1lkYKoHEC>.
- [3] F. Pelletier, D. Garant, and A. P. Hendry. Eco-evolutionary dynamics. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 364(1523):1483–1489, 2009.
- [4] Donald L. DeAngelis and Wolf M. Mooij. Individual-based modeling of ecological and evolutionary processes. *Annual Review of Ecology, Evolution, and Systematics*, 36(1):147–168, 2005. doi:10.1146/annurev.ecolsys.36.102003.152644.
- [5] J. Moya-Laraño, O. Verdeny-Vilalta, J. Rowntree, N. Melguizo-Ruiz, M. Montserrat, and P. Laiolo. Climate change and eco-evolutionary dynamics in food webs. *Advances in Ecological Research*, 47:1–80, 2012.
- [6] V. M. Savage, J. F. Gillooly, J. H. Brown, G. B. West, and E. L. Charnov. Effects of body size and temperature on population growth. *American Naturalist*, 163(3): 429–441, 2004.
- [7] Quart Dong and Gary A Polis. The dynamics of cannibalistic populations: a foraging perspective. In Crespi BJ Elgar MA, editor, *Cannibalism: ecology and evolution among diverse taxa*, pages 13–37. Oxford University Press, New York, NY, USA, 1992.
- [8] J.F. Gillooly, E.L. Charnov, G.B. West, V.M. Savage, and J.H. Brown. Effects of size and temperature on developmental time. *Nature*, 417:70–73, 2002.

-
- [9] B.C. Rall, O. Vucic-Pestic, R.B. Ehnes, M. Emmerson, and U. Brose. Temperature, predator-prey interaction strength and population stability. *Global Change Biology*, 16:2145–2157, 2010.
- [10] P. Yodzis and S. Innes. Body size and consumer-resource dynamics. *American Naturalist*, 139:1151–1175, 1992.
- [11] J. Moya-Laraño, D. Vinkovic, E. De Mas, G. Corcobado, and E. Moreno. Morphological evolution of spiders predicted by pendulum mechanics. *PLoS One*, 3:e1841, 2008.
- [12] J.H. Brown, J.F. Gillooly, A.P. Allen, V.M. Savage, and G.B. West. Toward a metabolic theory of ecology. *Ecology*, 85:1771–1789, 2004.
- [13] R.B. Ehnes, B.C. Rall, and U. Brose. Phylogenetic grouping, curvature and metabolic scaling in terrestrial invertebrates. *Ecology Letters*, 14:993–1000, 2011.
- [14] D. Hawlena and O.J. Schmitz. Herbivore physiological response to predation risk and implications for ecosystem nutrient dynamics. *Proceedings of the National Academy of Sciences of the United States of America*, 107:15503–15507, 2010.
- [15] K. Norrdahl and E. Korpimäki. Does mobility or sex of voles affect risk of predation by mammalian predators? *Ecology*, 79:226–232, 1998.
- [16] A.G. Jones, S.J. Arnold, and R. Borger. Stability of the g-matrix in a population experiencing pleiotropic mutation, stabilizing selection, and genetic drift. *Evolution*, 57:1747–1760, 2003.
- [17] D. Martín-Galvez, J.J. Soler, J.G. Martínez, A.P. Krupa, M. Richard, M. Soler, A.P. Moller, and T. Burke. A quantitative trait locus for recognition of foreign eggs in the host of a brood parasite. *J. Evol. Biol.*, 19:543–550, 2010.
- [18] J. Stapley, J. Reger, P.G.D. Feulner, C. Smadja, J. Galindo, R. Ekblom, C. Bennison, A.D. Ball, A.P. Beckerman, and J. Slate. Adaptation genomics: the next generation. *Trends Ecol. Evol.*, 25:705–712, 2010.

-
- [19] J Moya-Laraño, J.R. Bilbao-Castro, G. Barrionuevo-Rosales, D. Ruiz-Lupi3n, Leocadio G. Casado, Montserrat, Meli3n, and Magalh3es. Eco-evolutionary spatial dynamics: Rapid evolution and isolation explain food web persistence. In Jordi Moya-Laraño, Jennifer Rowntree, and Guy Woodward, editors, *Eco-Evolutionary Dynamics*, volume 50, pages 75–143. AECR, UK: Academic Press, 2014. ISBN 978-0-12-801374-8.
- [20] R.J. Williams and N.D. Martinez. Simple rules yield complex food webs. *Nature*, 404:180–183, 2000.