
Tiko'n Documentation

Versión 2.0.3

Julien Malard

14 de julio de 2019

1. ¿Qué es Tiko'n?	3
2. ¿Qué es un modelo de redes tróficas?	5
3. ¿Por qué utilizar Tiko'n?	7
4. Contenido	9
4.1. Introducción	9
4.2. Instalación	9
4.3. Estructura general	9
4.4. Publicaciones	10
4.5. Agradecimientos	10
4.6. Construir una red	11
4.7. Insectos	11
4.8. Conexión con cultivos	13
4.9. Manejo	13
4.10. A priori	13
4.11. Ecuaciones	14
4.12. Con clima	14
4.13. Calibraciones y simulaciones	14
4.14. Modificar la Red	15
4.15. Nuevos cultivos	15
4.16. Traducciones	15
4.17. Red	15
4.18. Insectos	15
4.19. Clima	16
4.20. Cultivo	16
4.21. Manejo	16
4.22. General	16
4.23. Resultados	16
Índice de Módulos Python	17
Índice	19



CAPÍTULO 1

¿Qué es Tiko'n?

Tiko'n es un programa para desarrollar modelos de redes tróficas agroecológicas de manera sencilla, reproducible y divertida.

¿Qué es un modelo de redes tróficas?

Un modelo de redes tróficas es un modelo que representa las interacciones entre organismos vivos en un campo agrícola (en general insectos y otros artrópodos). Predice el nivel de control biológico natural, las presiones futuras de plagas dado factores como el clima y la presencia de insectos benéficos, y hacia permite predecir cuáles serían los impacto de varias estrategias de manejo de plagas.

¿Por qué utilizar Tiko'n?

Redes tróficas con cosas muy difíciles a modelizar. Aun las más sencillas (3 insectos) presentan problemas como

1. Multitud de parámetros (15+ por insecto)
2. Inestabilidad matemática (dificultando predicciones fiables)
3. Multitud de posibilidades de ecuaciones en la literatura para cada fase de vida (reproducción, edad, transiciones, muertes, depredación)
4. Muy alta incertidumbre en los valores verdaderos de los parámetros
5. Pocos datos para calibración y validación
6. Requerimiento de conocimientos informáticos avanzados para conectar con modelos de cultivos externos o con predicciones de cambios climáticos

Así que si quieres hacer modelos tróficos pero te desaniman estos retos, ¡Tiko'n es para ti! Tiko'n manejará todos los puntos arriba para ti, siempre dándote el nivel de control que quieres sobre los detalles de tu modelo, nada menos, nada más.

1. Colección interna de la mayoría de ecuaciones para redes tróficas disponibles en la literatura, con selección automática de ecuaciones para principiantes
2. Posibilidad de agregar tus propias ecuaciones si quieres
3. Conexión automática con bases de datos y con predicciones y observaciones climáticas
4. Manejo automático de parámetros y de cálculos de poblaciones y de depredación
5. Funcionalidades integradas para calibración y validación de modelos
6. Formato estandarizado para guardar y compartir modelos calibrados
7. Y, por supuesto, gráficos bonitos

4.1 Introducción

4.2 Instalación

Es muy fácil instalar Tiko'n. Primero necesitarás [Python 3.6+](#). Después, puedes instalar Tiko'n en la terminal con:

```
pip install tikon
```

Si quieres la versión más recién (en desarrollo), puedes obtenerla de GitHub directamente con:

```
pip install git+git://github.com/julienmalard/tikon.git@master
```

Nota: Si tienes Windows, es posible que tengas que instalar el C++ redistributable de [aquí](#). Toma la versión terminando en `...x86.exe` si tienes Python de 32 bits y en `...x64.exe` si tienes **Python** (no Windows) de 64 bits. Después, instálalo. Por razones oscuras, `SciPy`, un paquete requerido por Tiko'n, no funciona en Windows sin éste.

4.3 Estructura general

La modelización agroecológica es un tema un poco más complicado que otros tipos de modelos por causa de su complejidad y comportamiento dinámico. La estructura de Tiko'n toma en cuenta estos desafíos por su inclusión de repeticiones estocásticas, incertidumbre paramétrica, y filosofía modular y flexible.

Aquí sigue una breve introducción a los términos específicos a Tiko'n.

4.3.1 Simulador

Un *Simulador* es la unidad fundamental de un modelo en Tiko'n. Contiene varios *Simulador*.

4.3.2 Módulos

Cada *Módulo* en Tiko'n representa una parte del agroecosistema. Por ejemplo, existen módulos para la red agroecológica, para el clima, para el cultivo y para el manejo humano. Los módulos pueden intercambiar valores de variables en el transcurso de una simulación.

4.3.3 Experimentos

Un *Exper* representa un experimento, o sea, una combinación de decisiones de observaciones (reales o hipotéticas) para una simulación. Aun simulaciones sin datos observados implementan un experimento vacío automáticamente.

4.3.4 Simulaciones

Se efectúan simulaciones por llamar `simular()` con especificaciones de escala temporal, repeticiones paramétricas y estocásticas, y experimento. Adentro de cada simulación, el modelo se va a `iniciar()`, `correr()`, y finalmente `cerrar()`.

4.3.5 Resultados

Los resultados de simulación tienen su propia clase (*ResultadosSimul*), la cual incluye los resultados (*ResultadosMódulo*) de cada módulo del simulador, los cuales a su vez contienen los resultados (*Resultado*) de cada variable del módulo. Resultados se pueden `validar()` y también `graficar()`.

4.3.6 Parámetros

Por supuesto, todo modelo necesita parámetros. En Tiko'n, los parámetros se implementan por *Parám*, y cada parámetro puede tener varias calibraciones conteniendo distintas distribuciones de valores (*Dist*).

4.3.7 Ecuaciones

El módulo *RedAE* implementa ecuaciones (*Ecuación*) para representar cada fase del ciclo de vida de los insectos en la red. Las ecuaciones se pueden por supuesto modificar, agregar, o desactivar según sus necesidades.

4.4 Publicaciones

En cuanto se publiquen artículos sobre Tiko'n, se incluirán aquí.

4.5 Agradecimientos

Tiko'n es un proyecto de fuente abierta. Las personas e instituciones siguientes hicieron posible su desarrollo.

4.5.1 Autores del código

- Julien Malard ([Sitio internet](#))
- Marcela Rojas ([Sitio internet](#))

4.5.2 Traductores

- Julien Malard ([Sitio internet](#))

4.5.3 Instituciones afiliadas

Tiko'n se desarrolla en colaboración con las instituciones siguientes:



4.6 Construir una red

4.6.1 Especificar insectos

4.6.2 A prioris

4.6.3 Validación

4.6.4 Calibración

4.6.5 Guardar y cargar

4.7 Insectos

Tiko'n implementa una gran variedad de ciclos de vida de insectos a través de subclases especiales de `Insecto`.

Nota: Tiko'n toma una vista *ecológica* de lo que es un insecto. Es decir, si come como un insecto y se come como un insecto, entonces *es* un insecto por lo que le importa a Tiko'n.

Así que no te sorprendes al ver arañas y gusanos en las listas de insecto. Sé que entomológicamente es una herecía, pero programáticamente es la mejor solución.

Cada insecto viene con sus etapas (huevo, larva, etc.) y las ecuaciones correspondientes ya especificadas.

4.7.1 Sencillo

El tipo de insecto más sencillo posible. Por lo tanto es también generalmente inútil, pruebas teóricas a parte. Solamente lleva una forma adulta, y se implementa con la clase *Sencillo*.

4.7.2 Metamorfosis completa

Tiko'n lleva la clase *MetamCompleta* para representar a insectos con ciclos de vida completos (de huevo a adulto, pasando por una pupa).

4.7.3 Metamorfosis incompleta

Insectos con ciclos de vida incompletos (sin pupa) se pueden representar con la clase *MetamIncompleta*.

4.7.4 Parasitoides

Parasitoides, aunque técnicamente por su mayor parte insectos con metamorfosis completa, se representan por su propia clase (*Parasitoide*) porque se debe tomar en cuenta el hecho de que su fase juvenil se desarrolla adentro de su huésped.

4.7.5 Esfécidos

Esfécidos son avispas similares a parasitoides pero que paralizan e inactivan su presa al momento del parasitismo, lo cual puede ser interior o exterior. Se deben representar de manera distinta (*Esfécido*) a parasitoides convencionales, porque la presa se quita del ecosistema al momento del acto de parasitismo y no al momento de la emergencia de la avispa adulta.

4.7.6 Cambiar ecuaciones

Puedes modificar las ecuaciones empleadas para un insecto en particular.

4.7.7 Cambiar etapas

Si quieres cambiar la estructura de las etapas de un insecto, también puedes implementar una clase para un nuevo tipo de insecto (ver nuevos-insectos).

4.8 Conexión con cultivos

4.9 Manejo

4.9.1 Reglas

4.9.2 Condiciones

4.9.3 Acciones

4.10 A prioris

Distribuciones *a priori* especifican rangos probables para los parámetros de un modelo. Se pueden especificar por el usuario y después se tomarán en consideración para corridas de simulaciones y calibraciones.

Existen dos maneras de especificar una distribución *a priori*: por densidad y por forma analítica.

Advertencia: Distribuciones *a priori* solamente se toman en cuenta automáticamente para corridas de calibraciones. Ver [Calibraciones y simulaciones](#) para más detalles.

4.10.1 Densidad

Se puede especificar *a prioris* por un rango de dos valores y el nivel de confianza que el valor verdadero se encuentre efectivamente en el rango. Tiko'n generará automáticamente una distribución que corresponde a la especificación, y eso, tomando también en cuenta los límites teóricos del parámetro.

4.10.2 Analíticas

También se puede especificar un *a priori* directamente por su forma analítica.

4.10.3 Uso en modelos

Interacciones

En el caso de parámetros con interacciones, los *a prioris* especificados se aplicarán a la raíz del árbol de interacciones del parámetro. Igualmente puedes especificar un nivel de interacción al cual aplicar el *a priori*.

Nota: Ver [Ecuaciones](#) para una explicación de niveles en árboles de ecuaciones, y [Heredar interacciones](#) para una descripción de cómo desactivar la propagación de *a prioris* a través los niveles de interacción.

4.11 Ecuaciones

4.11.1 Árbol de ecuaciones

4.11.2 Especificar ecuaciones

4.11.3 Nuevas ecuaciones

4.12 Con clima

4.13 Calibraciones y simulaciones

En Tiko'n, una *calibración* se refiere a un conjunto de valores para los parámetros de un modelo, todos calibrados en conjunto. Una *simulación* se refiere a cualquier ejecución del modelo. Por ejemplo, calibrar un modelo generalmente requiere la ejecución de una cantidad importante de simulaciones.

Por lo tanto, cada *simulación* normal se corre con una especificación de cuales *calibraciones* tomar en cuenta para establecer los valores de sus parámetros.

4.13.1 Comportamiento automático

Si no se especifican calibraciones para una corrida, Tiko'n tomará una decisión razonable para ti.

- Si es una corrida de **simulación**, se tomará cada calibración disponible, y se ignorarán *a priori*s especificados
- Si es una corrida de **calibración**, se tomarán primero los *a priori*s especificados, y, para los parámetros sin *a priori*s, se generarán distribuciones a base de las calibraciones ya existentes.

4.13.2 Calibraciones específicas

Si quieres cambiar el comportamiento automático, puedes especificar cuáles calibraciones existentes quieres incluir en la corrida.

4.13.3 Opciones avanzadas

Puedes cambiar el comportamiento automático por pasar un objeto `EspecCalibsCorrida` a la función de simulación `simular()` del `Simulador` (e igualmente a las que la llaman indirectamente, como `calibrar()` y `sensib()`). `EspecCalibsCorrida` toma opciones para especificar el uso de *a priori*s, de correspondencia entre parámetros, y de herencia de interacciones.

A priori

Puedes especificar si se deberían emplear distribuciones *a priori*s para los parámetros donde han sido especificados. En este caso, se ignorarán todas otras calibraciones disponibles para los parámetros con *a priori*s disponibles.

Correspondencia

Cuando se calibra un modelo con varios parámetros, no solamente importan los valores estimados de los parámetros sino también las interacciones entre los parámetros sí mismos. Es decir, en la calibración no se busca el valor óptimo de cada parámetro individualmente sino un conjunto de valores para todos los parámetros que da buenos resultados. Por eso es importante, tanto como sea posible, guardar las correlaciones entre los valores de los parámetros calibrados y Tiko'n hará su posible para únicamente tomar valores de calibraciones que están disponibles para todos los parámetros necesarios a la simulación.

Si quieres desactivar esta funcionalidad, puedes indicarlo así:

Heredar interacciones

Parámetros en Tiko'n pueden tener interacciones con otros objetos, por ejemplo, en el caso de un parámetro de eficacia de depredación que también se ve influido por la identidad de la presa. La opción automática es que cada parámetro, si le faltan calibraciones, puede heredar las calibraciones o *a priori* del nivel de interacción subyacente.

Nota: Ver [Ecuaciones](#) para una explicación de niveles en árboles de ecuaciones.

4.14 Modificar la Red

Hay dos maneras de agregar funcionalidad a la Red en Tiko'n: se pueden agregar nuevas opciones de organismos, y también se pueden agregar nuevas ecuaciones para modelizar sus ciclos de vida.

4.15 Nuevos cultivos

4.16 Traducciones

Tiko'n es un proyecto abierto y beneficia de se lo más accesible posible. Puedes ayudar a traducir los varios componentes de Tiko'n en tu idioma aquí. Necesitarás crear una cuenta gratis en [Transifex](#) primero.

1. [Documentación](#) (lo que estás leyendo ahora)
2. [Programa](#) (mensajes de error, etc.)
3. [Interfaz](#) (para un día futuro)

Nota: Si no aparece tu idioma favorito en la lista, por favor haz una demanda de activación de idioma en los enlaces arriba.

4.17 Red

4.18 Insectos

Esta clase representa insectos con ciclos de vida sencillos (para cuales sólo se incluye la etapa adulta en el modelo).

Esta clase representa insectos que tienen una metamorfosis completa. No necesita extensión propia, visto que no tiene métodos o atributos distintos a los de Insecto.

Parasitoides son una clase muy especial de insecto, porque sus larvas crecen adentro de los cuerpos de otros organismos. Después de mucho dolor de cabeza, decidimos (decidí) implementarlos así.

Los esfécidos son una familia de avispas que ponen sus huevos en los cuerpos (vivos) de sus presas. Al contrario de parasitoides típicos, estos paralizan y quitan su presa de la planta. Por lo mismo, se debe considerar su papel ecológico de manera distinta. (Se considera como depredación con reproducción basada en el éxito de la depredación).

4.19 Clima

4.20 Cultivo

4.21 Manejo

4.22 General

4.23 Resultados

t

`tikon.estruc.coso.Coso`, [16](#)
`tikon.estruc.módulo.Módulo`, [16](#)
`tikon.estruc.simulador.Simulador`, [16](#)
`tikon.estruc.tiempo.Tiempo`, [16](#)
`tikon.rae.orgs.insectos.Esfécido`, [16](#)
`tikon.rae.orgs.insectos.MetamCompleta`,
[15](#)
`tikon.rae.orgs.insectos.MetamIncompleta`,
[16](#)
`tikon.rae.orgs.insectos.Parasitoide`, [16](#)
`tikon.rae.orgs.insectos.Sencillo`, [15](#)
`tikon.result.obs.Obs`, [16](#)
`tikon.result.res.Resultado`, [16](#)
`tikon.result.res.ResultadosSimul`, [16](#)

T

tikon.estruc.coso.Coso (*módulo*), 16
tikon.estruc.módulo.Módulo (*módulo*), 16
tikon.estruc.simulador.Simulador (*módulo*), 16
tikon.estruc.tiempo.Tiempo (*módulo*), 16
tikon.rae.orgs.insectos.Esfécido (*módulo*), 16
tikon.rae.orgs.insectos.MetamCompleta (*módulo*), 15
tikon.rae.orgs.insectos.MetamIncompleta (*módulo*), 16
tikon.rae.orgs.insectos.Parasitoide (*módulo*), 16
tikon.rae.orgs.insectos.Sencillo (*módulo*), 15
tikon.result.obs.Obs (*módulo*), 16
tikon.result.res.Resultado (*módulo*), 16
tikon.result.res.ResultadosSimul (*módulo*), 16