Manual de usuario de PyToxo

PyToxo 1.0

Versión en castellano

Borja González Seoane

C.e.: borja.gseoane@udc.es

Facultade de Informática

Universidade da Coruña

23 de junio de 2021

Índice general

Índice de códigos

Índice de figuras

1 Requisitos

Para poder instalar PyToxo es necesario disponer previamente de Python 3.8 o superior. Se recomienda seguir las instrucciones oficiales de la Python Software Foundation [5] para instalar Python 3.8.10, que es la última versión de Python 3.8. La instalación de Python del usuario debe incluir también la herramienta PIP [3].

En este manual se asumirá que el usuario ha completado el proceso de instalación de Python y configurado correctamente la variable PATH de su entorno para que tanto el propio Python, como PIP y los paquetes instalados a raíz del mismo, sean alcanzables desde un terminal de comandos.

PyToxo funciona en cualquiera de sus formas —librería programática, CLI y GUI— en sistemas Linux, Mac OS y Windows. Concretamente ha sido verificado sobre Linux Lubuntu 20, Apple Mac OS Big Sur y Microsoft Windows 10. Sin embargo, sus escasas dependencias hacen de PyToxo una aplicación muy portable que es improbable que presente problemas de compatibilidad incluso en plataformas más antiguas.

Además de lo anterior, si se desea utilizar la interfaz gráfica de PyToxo, es necesario disponer de Tk [6]. Tk ha sido adaptada para funcionar en la mayoría de las variantes de Linux, Mac OS y Windows, y está incluida en la mayoría de las distribuciones modernas de Python para las plataformas anteriores. No ocurre así en algunas distribuciones de Linux, en las que hay que instalar un paquete adicional. Sin embargo, el propio PyToxo detecta esta carencia cuando se intenta lanzar la interfaz, y le sugiere al usuario el comando a emplear para instalar el módulo, que en sistemas operativos basados en Debian es:

```
sudo apt install python3-tk
```

2 Instalación

PyToxo está disponible en PyPI [4], el repositorio oficial de Python, por lo que, una vez completados los pasos de la sección anterior, basta con ejecutar el siguiente comando para instalarlo:

```
pip install pytoxo
```

Hecho esto, habrán quedado disponibles en la máquina del usuario la librería `pytoxo` y los ejecutables `pytoxo_cli` y `pytoxo_gui`.

3 Uso como librería de Python

A la hora de utilizar PyToxo, y como es bastante idiosincrásico en la comunidad de Python, la mejor documentación la conforman las propias cabeceras del código fuente, que en PyToxo han sido escritas con gran detalle atendiendo a las convecciones recomendadas oficialmente.

Con propósito ilustrativo, a continuación vamos a abordar el uso de PyToxo como librería de Python por medio de una serie de ejemplos. Esta demostración está disponible en forma de Jupyter Notebook [2] en el repositorio de PyToxo.

Lo primero que tenemos que hacer para utilizar PyToxo es importar la librería:

```
import pytoxo
```

Luego, a partir de un modelo en archivo CSV, podemos generar un objeto `Model` de PyToxo con:

```
modelo = pytoxo.Model(filename="../modelos/additive_3.csv")
```

E inmediatamente después ya podemos generar una tabla de penetrancia utilizando el método apropiado del objeto `modelo`. Este método será `find_max_prevalence_table` o `find_max_heritability_table`, según queramos maximizar la prevalencia o la heredabilidad, respectivamente.

```
# Definimos los parámetros del experimento
mafs = [0.4, 0.4, 0.4]
heredabilidad = 0.85

tabla = modelo.find_max_prevalence_table(mafs=mafs, h=heredabilidad)
```

El objeto `tabla`, de la clase `PTable`, contiene nuestra tabla de penetrancia. Ahora ya podríamos imprimirla por pantalla o guardarla como un archivo, por ejemplo con:

```
tabla.print_table(format="gametes")
```

Utilizando PyToxo como librería también tenemos la posibilidad de ingresar los datos del modelo epistático original directamente, sin recurrir a un archivo CSV existente. En el Código 1 presentamos un ejemplo de uso completo que parte de dos listas con los datos del modelo.

4 Uso desde la interfaz en línea de comando

Para invocar la CLI basta con emplear el comando:

```
pytoxo_cli
```

Podemos resumir el uso de la interfaz en la especificación POSIX [1] siguiente, que desglosamos a continuación de la misma, comenzando por los argumentos opcionales:

```
pytoxo [-h] [--gametes] (--max_prev | --max_her) <model> <prev_or_her> <maf>
    [<maf> ...]
```

- `-h`: usando este argumento opcional desplegamos la ayuda en línea de comando.

```python
import pytoxo
import numpy

genotipos = ["AABB", "AABb", "AAbb", "AaBB", "AaBb", "Aabb", "aaBB", "aaBb",
    "aabb"]
probabilidades = numpy.array(
    [
        "x",
        "x",
        "x",
        "x",
        "x*(1+y)",
        "x*(1+y)",
        "x",
        "x*(1+y)",
        "x*(1+y)",
    ]
)  # Podemos utilizar tanto listas normales como arrays de NumPy
modelo = pytoxo.Model(
    definitions=genotipos,
    probabilities=probabilidades,
    model_name="otro_modelo",
)
tabla = modelo.find_max_heritability_table(mafs=[0.1] * modelo.order, p=0.96)
tabla.print_table()
```

Código 1: Ejemplo de uso de PyToxo como librería introduciendo el modelo manualmente

- `--gametes`: argumento opcional que implica que la tabla de salida será compuesta en el formato de GAMETES [7]. Si no se hace uso de este argumento, la tabla será conformada como CSV, por defecto.
- `--max_prev` o `--max_her`: empleando el primero se maximizará la prevalencia y empleando el segundo, la heredabilidad. Estas dos opciones son mutuamente excluyentes, es necesario especificar una de las dos, e implica que el último argumento ingresado antes de las MAF será tratado como heredabilidad, si se maximiza la prevalencia; o como prevalencia, si se maximiza la heredabilidad.

Los argumentos opcionales anteriores los podemos distribuir en cualquier posición del comando sin que esto afecte al mismo. Sin embargo, en los que sí que resulta relevante la situación es en los argumentos posicionales, que son todos los que no se correspondan sintácticamente con alguno de los anteriores, y que serán interpretados, en orden, como sigue:

- El primer argumento posicional será la ruta hasta el archivo CSV que contenga el modelo epistático.
- El segundo argumento posicional se corresponderá con la heredabilidad o con la prevalencia a fijar, dependiendo de si se ha definido `--max_prev` o `--max_her`.
- Los argumentos posicionales que vayan a continuación de los anteriores serán interpretados todos ellos como cada una de las MAF, respectivamente. Deben ser especificadas tantas MAF como orden tenga el modelo. Las MAF deben ir una detrás de otra en el comando.

```
pytoxo_cli --gametes --max_prev modelos/additive_4.csv 0.6 0.2 0.3 0.3 0.4

pytoxo_cli modelos/additive_4.csv --max_prev --gametes 0.6 0.2 0.3 0.3 0.4

pytoxo_cli modelos/additive_4.csv 0.6 0.2 0.3 0.3 0.4 --max_prev --gametes

pytoxo_cli modelos/additive_4.csv 0.6 --max_prev 0.2 0.3 0.3 0.4 --gametes

pytoxo_cli modelos/threshold_8.csv --max_her 0.88 0.01 0.01 0.01 0.01 0.01 0.01
    0.01 0.01

pytoxo_cli modelos/additive_2.csv --max_her 0.4 0.45 0.5
```

Código 2: Ejemplos de comandos bien formados que hacen uso de la CLI de PyToxo

A modo ilustrativo, presentamos en el Código 2 algunos ejemplos de comandos bien formados que hacen uso de la CLI de PyToxo. Los cuatro primeros casos anteriores son distintas formas de escribir exactamente el mismo experimento.

Una vez que la tabla de penetrancia es calculada desde la interfaz en línea de comando, esta se imprime directamente a la salida estándar configurada en el terminal. Para guardar la tabla en un archivo, se debe seguir cualquiera de los métodos habituales en este tipo de flujos de trabajo, como por ejemplo redireccionando la salida:

```
pytoxo_cli modelos/additive_3.csv --max_prev 0.75 0.5 0.45 0.5 > mi_tabla.csv
```

5 Uso desde la interfaz gráfica de usuario

Para iniciar la GUI basta con emplear el comando:

```
pytoxo_gui
```

En la Figura 1 hemos marcado los distintos componentes que integran la interfaz gráfica, y que comentamos a continuación:

- **a**: menú contextual de archivo, desde el que cargar un modelo, borrarlo, guardar una tabla de penetrancia como archivo o cerrar la aplicación.
- **b**: menú contextual de ayuda, desde el que se puede acceder a información de utilidad sobre la aplicación.
- **c**: parrilla principal de la pantalla, con la representación del modelo. En el caso de la Figura 1 contiene también en la tercera columna las penetrancias ya calculadas.
- **d**: textos informativos de apoyo al usuario, que se van actualizando dinámicamente.

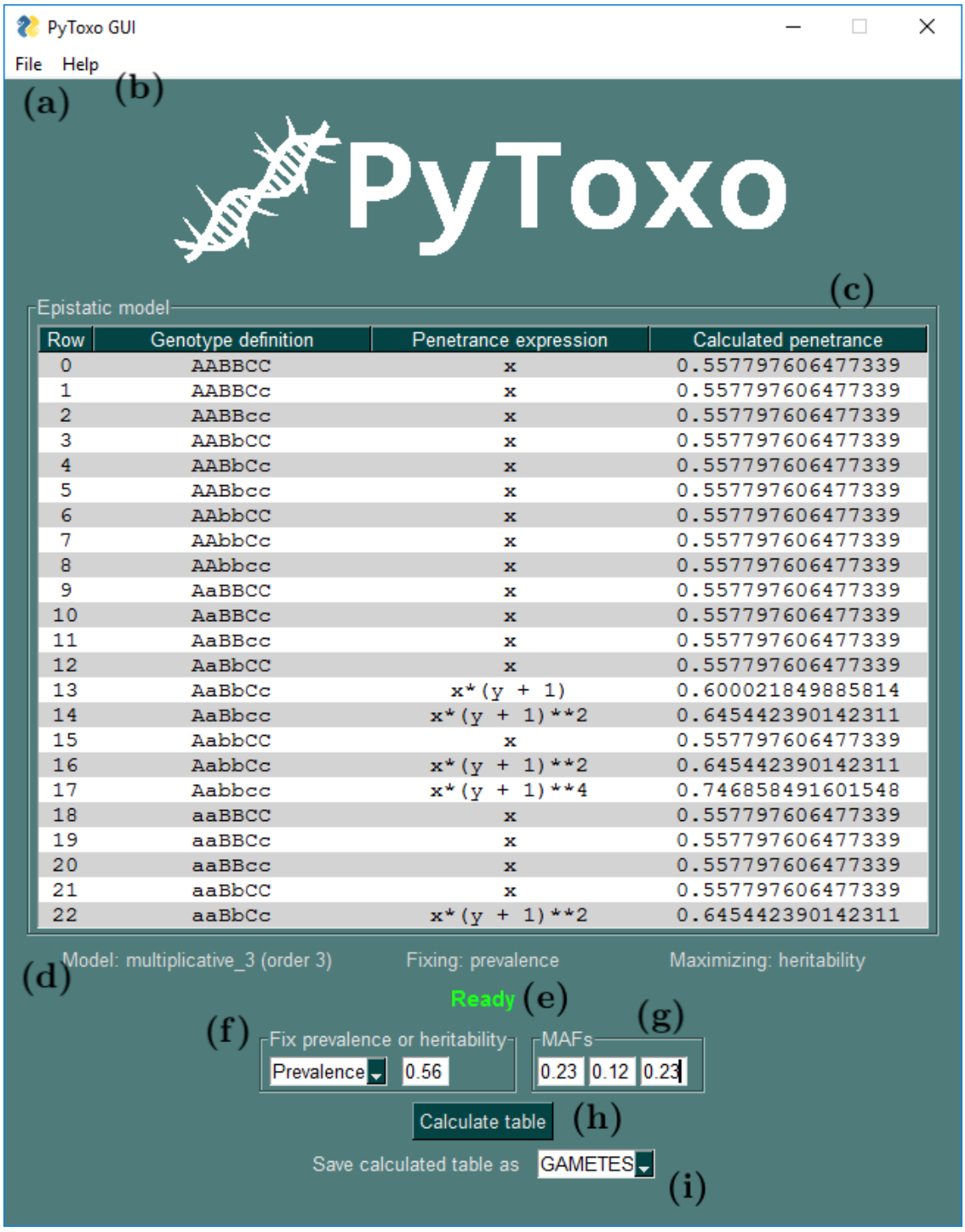

Figura 1: GUI de PyToxo desde Windows 10 con sus diferentes componentes marcados para referenciar

- **e**: indicador del estado del programa, que se alterna cuando se inicia el proceso de cálculo de la tabla.
- **f**: desplegable para seleccionar el parámetro a fijar de entre prevalencia y heredabilidad, y espacio para ingresar su valor.
- **g**: espacios para rellenar las MAF a emplear. El número de espacios se adapta dinámicamente al orden del modelo cargado.
- **h**: botón para calcular la tabla de penetrancia con la configuración cumplimentada.
- **i**: desplegable para seleccionar el formato de la tabla para guardarla como archivo, de entre CSV y formato de GAMETES [7].

La GUI resulta muy sencilla de utilizar debido a que se adapta dinámicamente al estado en el que se halla el flujo de trabajo de la aplicación. Los botones se van habilitando o deshabilitando en función de si se tiene cargado un modelo, rellenados los campos pertinentes, etc. Por ejemplo, no podemos rellenar las MAF hasta haber cargado un modelo y no podemos usar el botón de calcular hasta que todos los parámetros hayan sido cumplimentados.

Bibliografía

[1] IEEE Computer Society y The Open Group, «IEEE Standard for Information Technology–Portable Operating System Interface (POSIX(TM)) Base Specifications, Issue 7», *IEEE Std 1003.1-2017 (Revision of IEEE Std 1003.1-2008)*, 2018. DOI: 10.1109/IEEESTD.2018.8277153.

[2] Project Jupyter, «Jupyter», https://jupyter.org, consultado el 16 de junio de 2021.

[3] Python Packaging Authority, «PIP Documentation v21.1.2», https://pip.pypa.io/en/stable/, consultado el 3 de junio de 2021.

[4] Python Software Foundation, «PyPI», https://pypi.org, consultado el 28 de mayo de 2021.

[5] ——, «Python 3.8.10», https://www.python.org/downloads/release/python-3810/, consultado el 31 de mayo de 2021.

[6] Tcl Core Team, «Tcl Developer Xchange», http://www.tcl.tk, consultado el 9 de junio de 2021.

[7] R. J. Urbanowicz, J. Kiralis, N. A. Sinnott-Armstrong, T. Heberling, J. M. Fisher, y J. H. Moore, «GAMETES: a fast, direct algorithm for generating pure, strict, epistatic models with random architectures», *BioData Mining*, vol. 5, n.º 1, p. 16, 2012. DOI: 10.1186/1756-0381-5-16.