

guide

May 21, 2025

1 Manual de Usuario: sunitinib_analysis

Este documento describe paso a paso cómo preparar el entorno, instalar dependencias y ejecutar las notebooks del repositorio **sunitinib_analysis**, dedicado al análisis farmacocinético mono- y bi-compartmental de Sunitinib en ratones.

1.1 Requisitos Previos

- **Sistema operativo:** Windows 10/11, macOS 10.14+ o Linux (Ubuntu 18.04+).
- **Git** (para clonar el repositorio).
- **Python 3.8 o superior** (se recomienda 3.9–3.11).
- **pip** (gestor de paquetes de Python).
- Opcional pero recomendado: **conda** o **pyenv** para gestionar versiones de Python.

1.2 Instalación de Python

1.2.1 Windows

1. Descarga el instalador de Python 3.x desde <https://www.python.org/downloads/windows/>
2. Ejecuta el **.exe** y asegúrate de:
 - Marcar “**Add Python 3.x to PATH**”.
 - Seleccionar la instalación para **todos los usuarios**.
3. Tras la instalación, abre **PowerShell** o **Símbolo del sistema** y verifica:

```
python --version
pip --version
```

1.2.2 macOS

- **Homebrew** (si no lo tienes):

```
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```
- Instala Python:

```
brew update
brew install python@3.10
```

- Verifica:

```
python3 --version
pip3 --version
```

1.2.3 Linux (Ubuntu/Debian)

```
sudo apt update
sudo apt install -y python3 python3-venv python3-pip git
python3 --version
pip3 --version
```

1.3 Creación y Activación del Entorno Virtual

Es buena práctica aislar dependencias en un entorno virtual.

1. Desde la carpeta donde vas a trabajar, crea el entorno:

```
python3 -m venv venv
```

2. Actívalo:

- Linux/macOS:

```
source venv/bin/activate
```

- Windows (PowerShell):

```
.\venv\Scripts\Activate.ps1
```

- Windows (cmd.exe):

```
.\venv\Scripts\activate.bat
```

1.4 Clonación del Repositorio

```
git clone https://github.com/rorro6787/sunitinib_analysis.git
cd sunitinib_analysis
```

1.5 Instalación de Dependencias

1.5.1 Modo editable

Desde la raíz del proyecto:

```
pip install --upgrade pip
pip install -e .
```

Esto instalará el paquete principal y todas sus dependencias declaradas en `pyproject.toml`.

1.5.2 Dependencias adicionales (si existiera `requirements.txt`)

```
pip install -r requirements.txt
```

1.6 Estructura del Proyecto

```
sunitinib_analysis/
├── docs/                ← Documentación y notebooks de ejemplo
├── sunitinib_analysis/  ← Código fuente del paquete
│   ├── analysis.ipynb   ← Jupyter con el estudio
│   ├── data.csv         ← Datos de prueba
│   └── test.csv         ← Datos de test
├── tests/              ← Conjuntos de pruebas unitarias
├── .gitignore
├── LICENSE.txt
├── README.md
└── pyproject.toml      ← Configuración de empaquetado y deps
```

1.7 Uso de Jupyter Notebook

1.7.1 Inicio del servidor

Desde la carpeta raíz del proyecto:

```
jupyter notebook
```

o

```
jupyter lab
```

Se abrirá tu navegador por defecto con la lista de notebooks en `docs/` (o en la raíz, según el proyecto).

1.7.2 Ejecución de notebooks

1. Navega a la carpeta `docs/`.
2. Haz clic en el notebook que quieras abrir (`.ipynb`).
3. En el menú, selecciona **Kernel** → **Restart & Run All** para ejecutar todas las celdas en orden.

1.7.3 Exportar/Compilar notebooks

Para generar una versión estática (HTML, PDF, etc.):

```
# A HTML
```

```
jupyter nbconvert --to html sunitinib_analysis/analysis.ipynb
```

```
# A PDF (requiere TeX instalado)
```

```
jupyter nbconvert --to pdf sunitinib_analysis/analysis.ipynb
```

Los archivos generados quedarán en la misma carpeta que el notebook de origen.

1.8 Ejecución de los Tests

Si quieres validar que todo funciona correctamente:

```
pip install pytest
```

```
pytest --maxfail=1 --disable-warnings -q
```

1.9 Buenas Prácticas y Consejos

- Mantén el entorno virtual activo siempre que trabajes en el proyecto.
- Para actualizar dependencias, edita `pyproject.toml` y vuelve a ejecutar:

```
pip install -e .
```
- Usa `git pull` regularmente para incorporar mejoras o correcciones.
- Si desarrollas nuevas funcionalidades, añade tests en `tests/` y verifica con `pytest`.

1.10 Recursos Adicionales

- Documentación de NumPy: <https://numpy.org/doc>
- SciPy Reference: <https://docs.scipy.org/doc/scipy>
- Matplotlib Guide: <https://matplotlib.org/stable/contents.html>