

# Phlash软件安装教程

## 总结

- 1.使用pip工具自动安装plash，遇到python3.12和dinopy不兼容问题
- 2.手动下载、解压、sed -i 修改 (int, long) 为 int。
- 3.手动编译dinopy
- 4.其他依赖管理保持自动，下载plash

## 成功安装 plash 的完整步骤总结：

1. 识别问题：
  - 使用 pip 自动安装 plash[gpu] 时遇到 Python 3.12 与 dinopy 的兼容性问题
  - 具体错误是 dinopy/processors.pyx 中的 long 类型在 Python 3 中已不存在
2. 手动修复 dinopy：
  - 下载 dinopy 源代码： `pip download dinopy --no-deps`
  - 解压源代码包
  - 使用 sed 命令修复兼容性问题： `sed -i 's/(int, long)/int/g' dinopy/processors.pyx`
3. 手动编译安装 dinopy：
  - 进入 dinopy 目录
  - 使用开发模式安装： `pip install -e .`
4. 安装 plash：
  - 保持其他依赖的自动管理
  - 成功安装 plash[gpu]
5. 验证安装：
  - 验证 plash 安装成功： `python -c "import plash; print('plash installed successfully')"`
  - 测试 JAX 设备： `python -c "import jax; print(jax.devices())"`

## 具体软件安装步骤

### 安装指南

```
# 确认安装环境
which python
/usr/bin/python
```

```
# 使用 genome 环境的路径（你可以换成别的 conda 环境路径）
```

```

CONDA_ENV_PATH="$HOME/miniconda3/envs/genome/bin"

# 临时将 conda 环境路径插入到 PATH 前面
export PATH="$CONDA_ENV_PATH:$PATH"

which python
~/miniconda3/envs/genome/bin/python

# 根据安装指南
$ python3 -mvenv /home/litianxing/100My_Jino/114.PHLASH/software # replace with
desired path
$ source /home/litianxing/100My_Jino/114.PHLASH/software/bin/activate
$ pip3 install -U pip setuptools # recent version of pip and setuptools are
required

$ pip install 'phlash[gpu]' # as long as NVIDIA

# 并不需要cd

```

pip 会去 **PyPI**（或你配置的镜像）下载对应的包，然后安装到当前激活的环境里。它并不是依赖你所在的工作目录（除非你是要从本地目录安装，比如 `pip install .` 或 `pip install ./包目录` 才需要 `cd` 到对应路径）。设置环境变量——执行 `python`、`pip` 等命令时，都用这个环境里的版本。所以 `pip install` 装的库就会进入这个环境的 `site-packages`，不需要再管当前目录。创建了一个新的虚拟环境。激活之后，它自己有 `bin/python`、`bin/pip`，所以安装的东西就只会进这个环境，不会乱到系统里。

## 解决报错

### Review

直接安装会卡在这一步

```

Collecting dinopy>=3.0.0 (from phlash)
  Using cached dinopy-3.0.0.tar.gz (744 kB)

```

这个错误是因为在 Python 3 中，`long` 类型已经被移除，所有整数都统一为 `int` 类型。但在 `dinopy/processors.pyx` 文件中仍然使用了 `long` 类型，导致编译失败。

解决方案是修改 `dinopy/processors.pyx` 文件中的第649行，将：

```

elif isinstance(seq, (int, long)):

```

改为：

```
elif isinstance(seq, int):
```

你可以通过以下步骤修复这个问题：

**1. 找到并编辑源文件：**

首先需要找到 `dinopy` 包的源文件。根据错误信息，它正在从缓存中安装（Using cached `dinopy-3.0.0.tar.gz`），所以你需要找到这个压缩包或解压后的目录。

**2. 修改 `processors.pyx` 文件：**

在解压后的目录中找到 `dinopy/processors.pyx` 文件，将第649行的 `(int, long)` 改为 `int`。

**3. 从本地安装修改后的版本：**

修改完成后，从本地目录安装修复后的版本：

```
pip install /path/to/modified/dinopy-3.0.0/
```

**4. 重新安装 `phlash[gpu]`：**

安装完修复的 `dinopy` 后，再次尝试安装 `phlash[gpu]`：

```
pip install phlash[gpu]
```

如果找不到源文件的位置，你可以尝试直接下载 `dinopy` 的源代码：

```
# 使用 sed 命令直接修改文件
sed -i 's/(int, long)/int/g'
/home/litianxing/100My_Jino/114.PHLASH/software/dinopy-3.0.0/dinopy/processors.pyx

# 然后修改 processors.pyx 文件
cd dinopy-3.0.0
pip install .

# 然后再安装一次看看
pip install 'phlash[gpu]'
```

## 报错之2，手动安装的dinopy需要解决C++编译问题

```
# 验证 phlash 是否安装成功：
python -c "import phlash; print('phlash installed successfully')"

# 测试 GPU 支持
python -c "import jax; print(jax.devices())"
```

错误表明虽然 dinopy 包已经安装，但其 Cython 模块没有被正确编译或导入。问题在于 `dinopy.output_opener` 模块没有被找到，这通常是因为 Cython 扩展没有正确编译。

## 确保 Cython 扩展已正确编译

进入 dinopy 源代码目录，重新编译并安装：

```
cd /home/litianxing/100My_Jino/114.PHLASH/software/dinopy-3.0.0
python setup.py build_ext --inplace
pip install -e . # 使用开发模式安装

# 下面为命令行输出
copying build/lib.linux-x86_64-cpython-312/dinopy/auxiliary.cpython-312-x86_64-linux-gnu.so -> dinopy
.....

# 检查编译结果
# 查看 dinopy 目录中是否有 `.so` 文件（编译后的 Cython 扩展）
ls -la /home/litianxing/100My_Jino/114.PHLASH/software/dinopy-3.0.0/dinopy/*.so

# **重新安装 phlash**
# 在确保 dinopy 正确安装后，重新安装 phlash:
# 一直在虚拟环境中，使用的是虚拟环境中的Python，没有deactivate
which python
~/100My_Jino/114.PHLASH/software/bin/python

pip install --force-reinstall phlash[gpu] #事后证明这一步是不需要的
```

```
python -c "import phlash; print('plash installed successfully')"
```

ei，好了，根本不需要重新安装plash  
手动编译一下dinopy之后直接就好了

```
# 验证 phlash 是否安装成功:
python -c "import phlash; print('plash installed successfully')"
```

plash installed successfully

```
pip show phlash
```

  

```
# 测试 GPU 支持
python -c "import jax; print(jax.devices())"
```

```
[CpuDevice(id=0)]
```