# 多分量快速互相关（FastXC）使用说明

2024年10月18日

## 程序简介

本程序旨在通过一个高性能的 CPU-GPU 异构计算框架，高效地计算背景噪声数据中的单/九分量噪声互相关函数（NCFs）。它集成了数据预处理、互相关加速计算以及多种叠加技术（Linear, PWS, tf-PWS），特别是利用 CUDA 技术优化了计算过程，显著提升了处理速度和数据的信噪比，特别适用于处理大规模噪声数据集。

该程序的开发源于传统计算单/九分量 NCFs 方法在处理大规模数据时的低效率和高成本。通过本程序，我们能够显著提高计算效率，优化数据存储和传输，为背景噪声成像提供更准确、快速的计算工具。

本程序使用的编程语言包括python 和 CUDA-C，已通过不同规模数据集和计算设备的测试。

## 安装指南

### 2.1 准备环境

操作系统：Linux （不支持 Windows 系统）， WSL （Window 的 Linux 子系统）

CUDA Toolkit：CUDA 10.2。CUDA Toolkit 与Nvidia驱动版本有明确的兼容性要求，每个版本的CUDA Toolkit 都需要特定版本或更高版本的GPU 驱动。可以在英伟达官网查看具体细节 [CUDA Toolkit - Free Tools and Training | NVIDIA Developer](https://developer.nvidia.com/cuda-toolkit)。安装细节见第8部分。

Python版本：Python 3.10 或更高版本

必要的第三方Python模块：obspy, pandas, scipy, matplotlib, tqdm, numpy。由于使用的是这些库相对基础的功能，安装最新版本的库即可。可以使用下面的命令安装：

pip install obspy pandas scipy matplotlib tqdm numpy

### 2.2 编译

一般情况下，在FastXC文件夹下输入（#后是注释）：

make veryclean # 清空原先编译的结果（非常必要）

make

如果编译不成功，需要进入到src/ 目录。在这个目录里，有 sac2spec, xc\_dual, xc\_multi 三个目录，每个目录下有一个Makefile，找到：

NVCC=/usr/local/cuda/bin/nvcc

修改成用户的NVCC的编译器的地址。如果已经将地址写入环境变量（建议这么做），可以写成下面的形式：

NVCC=nvcc

## 快速开始

进入FastXC文件夹

cd FastXC

编译程序

make

本级目录下有几个文件夹和文件，分别是：

bin # 编译后的可执行文件

config # 运行程序的配置文件，并不是必需

fastxc # Python 代码

log # 日志文件文件夹，用于检查输出进行debug，不是必需

src # CUDA-C 代码

stalist # 存放台站列表，不是必需

test\_data # 测试数据

run.py # 主程序，跑程序的脚本

编辑测试算例的配置文件~/FastXC/config/test.ini 文件，找到其中的第4行：

sac\_dir\_1 = /mnt/c/Users/admin/Desktop/FastXC/test\_data

修改成~/FastXC/test\_data在系统中的绝对路径。

找到文件第25行:

output\_dir = /mnt/c/Users/admin/Desktop/FastXC/test\_output

修改为一个合法的输出位置。

找到文件的第82-86行：

sac2spec = /mnt/c/Users/admin/Desktop/FastXC/bin/sac2spec

xc\_multi = /mnt/c/Users/admin/Desktop/FastXC/bin/xc\_multi\_channel

xc\_dual = /mnt/c/Users/admin/Desktop/FastXC/bin/xc\_dual\_channel

stack = /mnt/c/Users/admin/Desktop/FastXC/bin/ncfstack

rotate = /mnt/c/Users/admin/Desktop/FastXC/bin/RotateNCF

修改为~/FastXC/bin 中相应可执行文件的绝对路径。

找到文件第88-96行：

[gpu\_info]

; GPU configuration for processing, specify IDs and task allocation.

gpu\_list = 0

gpu\_task\_num = 1

; GPU memory information used for task allocation and maximum spectrum calculations.

; Utilize nvidia-smi tool to obtain memory details.

gpu\_mem\_info = 24

redundant\_ratio = 0.8

根据系统信息修改这些参数。对于初次使用者，修改gpu\_mem\_info 为GPU显卡的显存。可以使用如下命令查看显卡信息：

nvidia -smi

完成以上修改后回到~/FastXC目录下，检查run.py 中是否使用了正确的config 文件，如果正确使用，在命令行运行：

python run.py

## 输出文件

输出文件的位置由用户在配置文件中设置。在测试例子中，输出目录是~/FastXC/test\_output。输出目录下会有若干个文件夹，其中你只需要关注ncf和stack，这两个文件夹。其中ncf代表每个时段的互相关结果，stack里面是叠加不同时段之后的互相关的结果。

## 功能描述和配置说明 （待补充）

通常情况下，源代码不需要进行修改，如果遇到涉及源代码的报错，欢迎随时和我联系！

## 故障排查（待补充）

## Nvidia-smi（驱动）, Nvidia Toolkit（编译器、库和开发者工具）安装

1. 在英伟达的GPU支持页面中[CUDA GPUs - Compute Capability | NVIDIA Developer](https://developer.nvidia.com/cuda-gpus)查看GPU是否支持CUDA。
2. 在英伟达驱动下载页面[Download The Official NVIDIA Drivers | NVIDIA](https://www.nvidia.com/en-us/drivers/)， 根据系统信息和GPU信息选择驱动并安装。（安装时注意关闭图形界面）
3. 访问 [CUDA Toolkit 12.6 Update 2 Downloads | NVIDIA Developer](https://developer.nvidia.com/cuda-downloads)，根据官方提示下载和执行安装脚本。（安装时注意关闭图形界面）
4. 配置环境变量

export PATH=/usr/local/cuda-X.Y/bin:$PATH

export LD\_LIBRARY\_PATH=/usr/local/cuda-X.Y/lib64:$LD\_LIBRARY\_PATH

1. 验证安装

nvidia-smi # 显示驱动信息，可以在这里看GPU占用情况

nvcc -V # 显示编译器版本 （编译器在toolkit里）

## 开发人员信息

王景熙

邮箱：[wkh16@mail.ustc.edu.cn](mailto:wkh16@mail.ustc.edu.cn) / 1531051129@qq.com

微信：15155909221