
单/多分量地震连续波形记录快速互相关 (FastXC) 使用指南和程序说明

王景熙

1531051129@qq.com

2023 年 12 月 29 日

目录

1	程序简介	2
2	环境准备	3
2.1	运行环境	3
2.2	获取工具和验证环境	3
2.3	程序编译	3
2.4	安装SeisHandler	4
2.5	获得fastxc文件夹	4
3	config.ini 配置文件参数说明	4
3.1	SeisArray_1 和 SeisArray_2 配置	4
3.1.1	SeisArray_1	4
3.1.2	SeisArray_2	5
3.2	Parameters 配置	5
3.3	Command 配置	6
3.4	mem.info 配置	6
3.5	executing 配置	6

1 程序简介	2
4 互相关的计算流程	7
4.1 基本原理	7
4.2 本程序的计算流程	7

1 程序简介

本程序主要包括两部分，以FastXC.py为核心的python脚本集合，以及包括若干个CUDA-C/C程序的源文件。程序的核心设计理念是利用灵活的脚本语言（python）的调用编译好的高性能可执行文件进行计算，实现用户友好、计算快捷的目标。

python脚本集合存放在名为fastxc的文件夹下（在解压之后的文件夹中应当是 /work/-fastxc），里面含有一系列脚本，这些脚本提供的功能包括：解析参数配置文件、生成滤波器、生成每一步CUDA-C/C 可执行程序所需的配置文件（通常是文件列表或目录）、生成可以在终端运行的命令以及将这些任务分配到不同的GPU或CPU上进行运行。

CUDA-C/C程序的源文件放在名为xc-x.x.x的文件夹中。文件夹下包括bin/，src/和一个Makefile，在终端输入make之后对src/中的源文件进行编译，之后会在bin/文件中生成6个可执行文件，分别实现下面六个功能：单分量连续波形频谱转化、多分量连续波形频谱转化、单台阵频域互相关、双台阵频域互相关、互相关叠加以及旋转。

除了以上两个最重要的组件之外，在运行程序之前还需要安装一个python模块SeisHandler，其主要功能是组织不同存储格式（路径、文件名）的地震连续波形数据，使得这个程序能够有效处理各种存储形式的地震数据。

王伟涛老师2015-2019年编写并修改在CPU上运行的单分量和九分量（多分量）互相关程序。这一套程序包括三步：首先将连续波形数据转化为频谱，第二步将频谱相乘然后再反变换为时间域互相关信号（相关证明见4章 [互相关的计算流程](#)），之后再将不同时间段计算得到的相同台站对的互相关函数进行叠加，如果是三分量数据的情形，在此之后还会有一步旋转的操作（将互相关函数从ENZ分量旋转到RTZ分量上）。

由于互相关函数在两两台站之间进行计算，随着台站数目的增加，计算量呈平方倍增长。2021年，科大网络信息与计算中心的吴超老师等人将单分量互相关程序的C代码重构为CUDA-C，将频谱转换和频谱相乘使用GPU加速，并使用OpenMP对叠加计算进行加速。2021年十月之后，由笔者先后修改了连续波形的预处理方法以及其他程序中的细节，并使用python语言编写脚本调用CUDA-C代码，之后将单分量的加速代码扩展到九分量上s。

2 环境准备

2.1 运行环境

本程序在Linux环境下运行，通常为了成功运行基于CUDA的程序需要准备一个特定的开发环境。以下是必需的组件和步骤：

1. **NVIDIA CUDA Toolkit** - CUDA Toolkit 是运行和开发基于CUDA的应用程序的核心。请确保您安装的是 10.2 或更高版本。您可以从 [NVIDIA官方网站](#) 下载最新版本。
2. **支持CUDA的NVIDIA GPU** - 您的系统需要配备NVIDIA GPU，且该GPU必须支持CUDA。您可以在 [NVIDIA官方网站](#) 查看兼容的GPU列表。
3. **合适的编译器** - 根据您的操作系统，您可能需要安装GCC或其他编译器。CUDA Toolkit 10.2 兼容的GCC版本详情可以在其官方文档中找到。
4. **Linux操作系统** - CUDA通常在Linux环境下运行得更好。确保您的Linux版本与CUDA Toolkit兼容。您可以在其[官方文档](#)中查看支持的Linux发行版。
5. **环境配置** - 在安装CUDA Toolkit后，您需要配置环境变量以确保系统可以找到CUDA的编译器和库。
6. **验证安装** - 您可以运行 `nvcc --version` 来检查CUDA编译器的版本，运行 `nvidia-smi` 来检查您的NVIDIA驱动和GPU状态。

2.2 获取工具和验证环境

- **CUDA Toolkit 和 驱动程序** 可以从 [NVIDIA官方网站](#) 下载。
- 对于 **Linux操作系统**，您可以选择像Ubuntu、Fedora等流行的发行版。大多数发行版官方网站都提供了安装和配置的详细指南。
- 使用Linux终端的 `gcc --version` 命令来检查GCC版本，确保其与CUDA Toolkit兼容。

确保按照上述步骤准备好开发环境，您就可以顺利运行本程序了。

2.3 程序编译

打开文件夹 `/xc-x.x.x/`，在命令行中输入：

Listing 1: make

```
1 make
```

之后会在xc-x.x.x/bin下出现六个可执行文件，在当前目录下直接运行这些程序，可以看到这些程序的使用方法，例如：

Listing 2: execute

```
1 ./sac2spec_c1
2 ./sac2spec_c9
3 ./spec_xc_mono
4 ./spec_xc_dual
5 ./ncfstack
6 ./RoTateNCF
```

2.4 安装SeisHandler

在运行程序之前还需要安装SeisHandler包。SeisHandler 是一个 Python 包，专门用于组织地震数据。首先，需要从其 GitHub 项目页面下载，链接为：<https://github.com/wangkingh/SeisHandler>。在安装 SeisHandler 之前，必须先安装 pandas 包。下载 SeisHandler 后，使用命令 `pip install SeisHandler.tar.gz` 进行安装。该包通过文件名匹配和组织地震文件，支持多线程文件匹配和过滤，以及按标签分组和组织文件。

2.5 获得fastxc文件夹

work文件夹包含一个fastxc/文件夹，一个run.py和config.ini示例。修改config.ini中的参数，接下来运行run.py即可。

3 config.ini 配置文件参数说明

work文件中包含一个config.ini文件，用户可以通过修改这个文件修改参数。如果文件丢失，也可以在fastxc/template_config.ini中找到。

3.1 SeisArray_1 和 SeisArray_2 配置

3.1.1 SeisArray_1

- **sac_dir**: 指定存储第一个台阵的连续波形文件（SAC格式）的目录，输入需要是绝对路径。

- **pattern:** 定义SAC文件的命名模式如例如hefei_project/FD01/2022.195.U.sac 可以被写作home/station/YYYY.JJJ.component.suffix，其中home关键字代表sac_dir，目前支持以下关键字。文件的命名模式中必须包含足够的时间信息，至少能精确到天，例如YYYY和JJJ的组合。
 - YYYY 四位数年份
 - YY 两位数年份
 - MM 月份
 - DD 日期
 - HH 小时
 - MI 分钟
 - JJJ 年积日，一年中的第几天
 - station 台站名
 - network 台阵名
 - component 分量名
 - suffix 后缀名
- **start** 和 **end:** 定义待处理台阵数据的起始时间和终止时间。
- **sta_list:** 台站列表文件的路径。如果不需要这样的文件，则设置为 **NONE**。
- **component_list:** 列出要计算的记录分量，三分量情形下顺序为东（E）、北（N）和垂直（Z）。

3.1.2 SeisArray_2

- 配置与 **SeisArray_1** 类似，但用于第二个台阵。通过将 **sac_dir** 设置为 **NONE** 禁用。
- **sta_list:** 此台阵台站列表文件的路径。

3.2 Parameters 配置

- **output_dir:** 存储互相关输出的目录。
- **win_len:** 连续波性分段的窗口（每一段）长度（秒）。
- **whiten:** 频域谱白化选项（OFF、BEFORE、AFTER、BOTH）。

- **normalize**: 时域归一化选项 (RUN-ABS、ONE-BIT、OFF)。
- **bands**: 分频带滑窗时间域归一化 (RUN-ABS) 所使用的多个频带, 单位为Hz, 例如1/2 2/5 5/10。
- **max_lag**: 互相关结果保留的长度, 单位为秒。
- **do_xc**: 设置为True, 则计算所有互相关; 设置为False, 计算自相关。
- **skip_step**: 跳过某一段数据, 避免某些时段的信号, 不需要可以设为-1。
- **stack_norm**: 设置为True将在叠加的时候进行归一化处理。

3.3 Command 配置

- 定义用于命令行工具的绝对路径, 包括我上面提及的解压之后的六个可执行文件, 如 `sac2spec_c1`、`xc_mono`、`stack` 等。

3.4 mem_info 配置

- **npts**: SAC文件的点数, 在进行计算前务必保证所有SAC文件点数一致。
- **delta**: SAC文件采样间隔, 单位为秒。
- **seg_len**: 分段互相关的长度, 单位为秒, 和Parameter中的win_len是同一概念。
- **gpu_list**: 使用的GPU设备的编号, 用逗号分隔。例如1, 2, 3, 4
- **gpu_task_num**: 第二步频谱互相关中每一计算批次在每个GPU设备上分配的任务数, 长度、格式与gpu_list相同。
- **gpu_mem_info**: 所使用的GPU设备的内存大小, 长度、格式与gpu_list相同。可以使用nvidia-smi查看。
- **cpu_mem_info**: CPU的内存信息, 使用free -h查看系统剩余内存。
- **redundant_ratio**: 指定CPU内存使用的冗余比率。

3.5 executing 配置

- **do_xc**, 设置为True, 则计算所有互相关; 设置为False, 计算自相关。与Parameters中一致。
- **parallel**: 控制程序在CPU运行的部分是否采取多进程, 一般设置为True

- **debug**: 控制调试模式，一般情况下设置为False。
- **threads**: CPU并行时使用的线程数。
- **log_file_path**: 日志文件的路径。

4 互相关的计算流程

4.1 基本原理

通过计算两两台站连续波形记录的互相关函数可以得到地震波在台站之间的传播信息（格林函数），进而为地下结构的解释提供地震学依据。

从背景噪声中恢复格林函数有一个十分重要的假设：噪声源的时空分布需要是随机和均匀的。然而在真实地球情形下，情况却往往更加复杂。第一，地球环境不总是平静的，长时间的连续波形记录中会记录到一些地震事件，而地震事件的分布是不均匀的；第二，在地球上存在一些局域、长期的噪声源，这些噪声源往往有着特定的频带。除以上两点外，地球上的噪声模型也有特定的频带特征。

这三点原因导致了这样几个后果：连续波形中的强信号（地震）事件、地球上的局域源会破坏源分布的均匀性，地球的噪声模型使得某些频带的信号要远远强过其他频带。因此，在进行互相关计算之前，我们要通过滑动时窗时间域归一化消除强信号，并且使用频率域谱白化压制局域源的信号同时增强频带较弱的信号。

因此，互相关计算的流程可以描述如下：对数据进行预处理，将连续切割为若干段。对每一段进行预处理，包括去均值、去趋势、两端尖灭、谱白化、时间域归一化。值得一提的是，Bensen（2007）提出谱白化应当在时间域归一化之后进行，但是亦有学者在时间域归一化之前做谱白化。根据我自己的经验，以上两种思路都是可以的，实际上如果数据足够好，甚至不需要过多的预处理。

在这之后，就对谱白化的数据在两两台站之间进行互相关计算。计算的步骤为，时间域补零到二倍长度，随后进行傅里叶变换到频率域，在频率域做两道数据之间的共轭相乘，再将结果进行偏移，最后反变换到时间域输出。

以上步骤会计算出每一天的两两台站之间的互相关，之后就将相同台站对不同时间段的互相关进行叠加，得到最后的结果。

4.2 本程序的计算流程

本程序的操作流程基本按照上述原理进行。首先将连续波形数据转化为频谱，随后对频谱进行乘积并反变换到时域，最后进行叠加。本程序使用了多张卡，大大提高了程序的效率。