2 Installation and Getting Started

2.1 Prerequisites

Smooth Emulator programs should run on UNIX, Mac OS or Linux, but is not supported for Windows OS. The software is largely written in C++, though there are some python scripts included for graphing results. In addition to a C++ compiler (CMake files assume you can accommodate the C++ 20 standard), the user needs the following software installed.

- git
- CMake
- Eigen3 (Linear Algebra Package)
- Python/Matplotlib (only for generating plots)

CMake is an open-source, cross-platform build system that helps automate the process of compiling and linking for software projects. Hopefully, CMake will perform the needed gymnastics to find the Eigen3 installation. To install CMake, either visit the CMake website (https://cmake.org/), or use the system's package manager for the specific system. For example, on Mac OS, if one uses *homebrew* as a package manager, the command is

```
% brew install cmake
```

Eigen is a C++ template library for vector and matrix math, i.e. linear algebra. The user can visit the Eigen website (https://eigen.tuxfamily.org/dox/), or use their system's package manager. For example on Mac OS with *homebrew*,

```
% brew install eigen
```

2.2 Downloading the Repository and Copying Files to Working Directories

The software requires downloading the BAND framework software repository into some directory. Should that be in the User's home directory, the User might enter

```
/Users/CarlosSmith% git clone https://github.com/bandframework/bandframework.git
```

Within the repository, there will be a directory `/Users/CarlosSmith/bandframework/software/SmoothEmulator/`. All the relevant functionality of *Smooth Emulator* is contained within this directory. Throughout the manual the phrase `${GITHOME_BAND_SMOOTH}` will refer to this directory.

The User should create a personal directory from which the User would perform most projects. This is easiest accomplished by copying the template directory from the *Smooth* distribution. For example

```
% cp -r ${GITHOME_BAND_SMOOTH}/templates /Users/CarlosSmith/smoothy
```

The User may choose any other name besides `smoothy`. The directory `../smoothy/..` is the root directory for the User's executables and analyses data, and should be outside the `bandframework/` path so that the User's work does not interfere with the repository. Within this directory one will find a template analysis directory, which is named `/Users/CarlosSmith/smoothy/analysis_template`. One can copy that directory to a new directory, e.g. `../smoothy/my_analysis`. Hence forth, `${MY_ANALYIS}` is a name chosen by the User, and will refer to this directory, including the path, from which the User will perform most of the analysis. Thus, the User may wish to have several such directories, located according to the User's preference. Any time a new analysis is performed with new parameters, and if the User wishes to save the previous analysis, a new `${MY_ANALYSIS}` directory should be created.

Within the `smoothy/` directory, there is also a directory `../smoothy/mylocal/`, which after the copying above will become: `/Users/CarlosSmith/smoothy/mylocal`. Henceforth, `${MY_LOCAL}` will refer to this directory, including its path. `${MY_LOCAL}/software/` contains the source code and CMake files for compiling the main programs. Executables will be stored in `${MY_LOCAL}/bin/`. Because the User might wish to edit the main programs, or to add similar programs, this provides the User a space to make such edits, all while leaving an original copy of the directory in the `templates/` directory. The User may find it convenient to add `${MY_LOCAL}/bin/` to their path.

The philosophy behind this structure is that it allows the User to devise and build simple programs making calls to *Smooth Emulator* libraries without altering the source code and libraries which are kept in the `${GITHOME_BAND_SMOOTH}/` directory structure. If the User wished to make modifications to this code, it would make sense to copy the `${GITHOME_BAND_SMOOTH}/` directory to a new location.

The directory structure is Fig. 2.1. As stated above, for the remainder of this manual, `${GITHOME_BAND_SMOOTH}` `${MY_LOCAL}/` and `${MY_ANALYSIS}/` will be used to denote the location of these directories.

2.3 Compiling the Software

First, change into software directories, then create the makefiles with cmake, then compile them.

```
% cd ${GITHOME_BAND_SMOOTH}/software
% ${GITHOME_BAND_SMOOTH}/software% cmake .
% ${GITHOME_BAND_SMOOTH}/software% make
```

There seems to be a common problem that `cmake` misreports the path of the `Eigen` installation. If the User should get an error stating that the Eigen header files cannot be found, the User can set the environmental variable,

```
% export EIGEN3_INCLUDE_DIR=/usr/local/include/eigen3
```

The final arguments may need to be changed depending on the User's location of the packages. If the User wishes to choose a specific C++ compiler, the `cmake` command should be replaced with `cmake -D CMAKE_CXX_COMPILER=g++-11`, or whichever compiler is preferred.

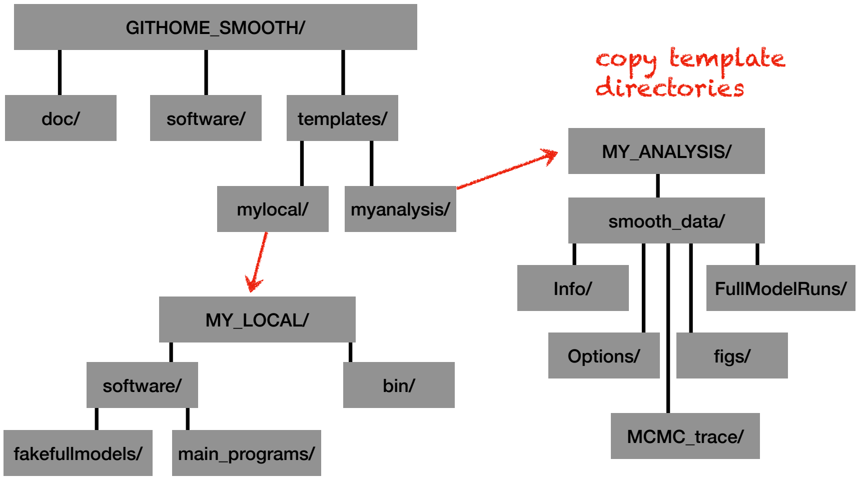

Figure 2.1: The directory structure: The User clones the repository into some location, which will be referred to as `${GITHOME_BAND_SMOOTH}`. From the `templates` directory, the User copies the `mylocal` and `myanalysis` directories to two new locations, `${MY_LOCAL}` and `${MY_ANALYSIS}`. The User will likely make multiple copies of the `myanalysis` directory for different projects or for different analyses with different options. The main source code and compiled libraries will be built and kept within the `${GITHOME_SMOOTH}/` tree. Source code for simple main programs can be found in `${MY_LOCAL}/software/` and executables are stored in `${MY_LOCAL}/bin/`. The software is designed to be run from the command line in the `${MY_ANALYSIS}` directory. Data used and created by *Smooth Emulator* software is stored in `${MY_ANALYSIS}/smooth_data/`.

At this point all the libraries are built, but this does not include the main programs. The main programs are short, and are meant to serve as examples which the User might copy and edit at will. Before compiling, one needs to set an environmental variable so that the compilation can find the libraries needed for compilation.

```
% export GITHOME_BAND_SMOOTH=/Users/CarlosSmith/bandframework/software/SmoothEmulator
```

The first part of the path needs to be replaced with the location of the `bandframework` git repository. If one prefers not to set an environmental variable, one can instead edit `${MY_LOCAL}/software/CMakeLists.txt`, and replace the line where the variable `${GITHOME_BAND_SMOOTH}` is set to the shell's environmental variable with one where it is set explicitly.

Below, this illustrates how to compile the programs used for generating training points with Simplex and for tuning the emulator with *Smooth Emulator*:

```
% cd ${MY_LOCAL}/software
% ${MY_LOCAL}/software% cmake .
% ${MY_LOCAL}/software% make
```

```
.
.
```

The cmake command will also recompile the main libraries in ${GITHOME_BAND_SMOOTH}/software/ if necessary. Several source codes for main programs can be found in ${MY_LOCAL}/software/main_programs/. If you build your own main programs (probably using these as examples), you can edit the CMakeList.txt file in ${MY_LOCAL}/software/main_programs/, using the existing entries as an example. The executables should appear in ${MY_LOCAL}/bin/.

2.4 The ${MY_ANALYSIS/smooth_data} Directory

Within ${MY_ANALYSIS}/smooth_data there are several sub-directories (assuming it was created from the template). The first is smooth_data/Info/. Information about the model parameters, and their priors is stored in smooth_data/Info/prior_info.txt, and information about the observables is stored in smooth_data/Info/observable_info.txt. The file smooth_data/Info/experimental_info.txt stores information about the measurements for each observable, and is not needed for building the emulator, but is used by the MCMC investigation of the posterior. The smooth_data/Options/ directory stores text files used to set options for the training-point optimization, the emulator and the MCMC programs. The smooth_data/FullModelRuns data has subdirectories run0/, run1/ $\cdots$. Each subdirectory stores information describing a full model run. The file smooth_data/FullModelRuns/runX/ stores the model parameters used for run "X". This file can be provided by the User, or it can be created by the training-point optimization codes. Once those files are created, it is the User's responsibility to run the full model and record the observables in smooth_data/FullModelRuns/runX/obs.txt. Given those observables, the User can then run the emulator software. If the User runs the MCMC software, the MCMC traces are stored as text files in smooth_data/MCMC_Trace/. Examples of PYTHON scripts for graphing using MATPLOTLIB can be found in smooth_data/figs/.