

# cgFRAMES

Lennart de Bruin

August 16, 2017

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Usage</b>	<b>1</b>
2.1	Input files . . . . .	1
2.2	Output files . . . . .	2
<b>3</b>	<b>External libraries</b>	<b>2</b>
<b>4</b>	<b>Compilation</b>	<b>2</b>
<b>5</b>	<b>Ideal bases file format</b>	<b>2</b>
<b>6</b>	<b>Folder hierarchy</b>	<b>2</b>
<b>7</b>	<b>Program structure</b>	<b>2</b>
<b>8</b>	<b>Things left to do</b>	<b>3</b>

## 1 Introduction

cgFRAMES is a program that fits user-specified atom groups to atom groups in Molecular Dynamics trajectories, and PDB files (not supported yet).

It is set up as follows: the user specifies an ideal atom group in a specified format in a file. This atom group will contain the atom names, and coordinates. On top of that, a frame is specified w.r.t. the origin of the coordinate system the atom coordinates are in.

The program will then find these atoms in the MD topology file, and the corresponding coordinates in the MD trajectory file, fit the ideal atom group to this group in a later specified method, and output the frames for all specified atom groups for each MD snapshot in the .fra format.

## 2 Usage

### 2.1 Input files

**.top** The topology file. The standard AMBER format.

**.nc** The trajectory file. The standard nc format.

**ideal bases** The file containing the ideal bases. The format is as follows:

Each entry has:

<group name> <# atoms in group> <9 values of the frame orientation matrix> <3 values of the  
<x> <y> <z> <atom name as in the topology file> (for each atom)

Comments begin with an #. Between each entry there needs to be an empty line, or a comment.

## 2.2 Output files

## 3 External libraries

The program uses netCDF to read in the MD trajectories from Amber MD simulations.

## 4 Compilation

## 5 Ideal bases file format

## 6 Folder hierarchy

The hierarchy of the folders is as follows:

### **bin**

contains the binary produced by compiling the program.

### **doc**

contains this document, and other documentation.

### **include**

contains all header files

### **input**

contains all files necessary for using cgFRAMES. Think of ideal bases.

### **output**

contains all files outputted by cgFRAMES.

### **src**

contains the implementation files of the program.

### **test\_files**

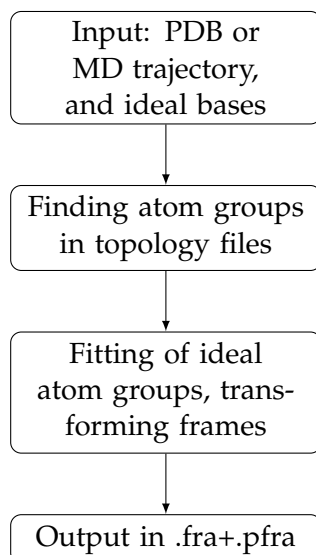
contains main files for testing the implementation. This was originally for unit testing, but the author was too lazy to bother.

### **utilities**

contains several scripts to convert the simulations output to different formats, e.g. for viewing the configuration with cgDNAjs.

## 7 Program structure

The structure of the program is laid out in the flow-chart below.



## 8 Things left to do

There are a lot of things left to do:

1. Finish the PDB format reader, so that frames can also be fitted to PDB files.
2. At the moment only DNA bases are supported. This needs to be extended to perhaps basepairs, and general atom groups, so that also frames can be fitted to proteins. For this, the topology file needs to be stored in a better way. (At the moment only the atoms are stored, with positions, not the connections between atoms. We can only find DNA bases because the program assumes that the atoms of the bases are next to each other in the MD topology file).

Where to start:

1. One could download a library to parse the PDB files, since these files tend to vary in which record different information is stored. Libraries that come to mind are one from stanford: <http://graphics.stanford.edu/~drussel/pdb>, or the official PDBx/mmCIF reader: <http://mmcif.wwpdb.org/docs/sw-examples/cpp/html/index.html>. Then put this in the general protein data structures used by cgFRAMES.
2. Not only the atoms are specified in the topology files, but also the connections between them. These can be somehow used to see the atoms' relation to the specified groups. This is in general not an easy problem to solve, and impossible if you have disjoint subgroups belonging to the same specified group that are not unique in the set of atoms you're searching in.