

Reconstruction of Protein Backbones using **BB**;
installation and usage instructions.

[INCOMPLETE DRAFT]

Stewart A. Adcock

Dept. Chemistry and Biochemistry,

University of California, San Diego,

4234 Urey Hall,

9500 Gilman Drive,

La Jolla, CA 92093-0365.

email: adcock@mccammon.ucsd.edu

Last updated: January 12, 2004

<i>CONTENTS</i>	2
-----------------	---

Contents

1 Preface	3
2 Introduction	4
3 Copyright and Licensing	7
4 Installation	8
4.1 Binary Packages	8
4.1.1 RedHat Linux	8
4.1.2 Debian Linux	8
4.1.3 Binary tarballs	9
4.2 Source Packages	9
4.2.1 Source tarball	9
4.2.2 Source RPM	9
5 Command-line usage	10
5.1 Environment variables	11
5.2 An example calculation	11
6 C-API usage	13
7 Acknowledgments	14

1 Preface

Reconstruction of complete protein backbones from C_{α} coordinates only is a common step in protein modelling procedures. A novel protocol was recently developed and proven to produce highly accurate backbone structures. This protocol is implemented efficiently in computer program named **BB**. This document contains installation and usage instructions for **BB**, and it corresponds to version 0.9-0.

Software and data files are freely available for download at <http://mccammon.ucsd.edu/~adcock/bb.html>.

Like most of my documentation, this document is incomplete!

2 Introduction

The process of reconstructing an all-atom peptide, or protein, model from a subset of the atomic coordinates is the subject of a considerable body of literature. Numerous applications for this process exist, including enhancement of low-resolution models from crystallography into usable models or conversion of the coarse structures typical of *ab initio* folding computations or comparative protein modelling techniques into all-atom models.

It is common to split the process of reconstructing all-atom structures into two separate tasks. The first task is to predict the full backbone, and the second is to build side chains onto this. This partitioning of the problem is considered reasonable because it is observed that the backbone geometry of well defined secondary structure elements is mostly invariant to the identity of its amino acid residues.

A variety of approaches are described in the literature for the determination of complete backbone models from C_α coordinates. These methods either generate *de novo* backbone conformations [1–8] or utilize fragment libraries [9–13] derived from known structures to locate possible conformations that do not violate the provided C_α trace. These conformations are selected using energetic, homology-based or geometric criteria.

Payne described a dynamic programming method which located optimal rotations of peptide units with respect to a potential of mean force between adjacent residues [3]. Rey and Skolnick applied frequency tables derived from the PDB to locate C_β coordinates and then generated suitable coordinates for the remaining atoms to fit with those [1]. Bassolino-Klimas *et al.* minimised an

empirical potential by applying a directed conformational search [2]. The very rapid method of Milik *et al.* used information extracted from known structures to generate statistical positions for reconstructed atoms [5]. Feldman and Hogue built the backbone sequentially, selecting conformations according to database derived trajectory distributions [6]. Liwo *et al.* maximised the peptide dipole alignments to predict the backbone conformation [4]. Iwata *et al.* devised an analytical approach to selecting coordinates compatible with favoured regions on the Ramachandran map [7]. Several groups have applied molecular dynamics [7, 14–16] or Monte Carlo procedures [8, 17, 18] to construct, or refine, backbone conformations, usually through the evaluation of standard molecular mechanics forcefields.

BB applies a dead-end elimination procedure, a knowledge-based scoring function and a library of 3-residue fragments to construct all backbone atoms. In a previous study, **BB** was found to produce structures with an average all-backbone atom RMSD of 0.41Å from the C_α coordinates alone [19].

If you use **BB** in your work, please cite:

- Adcock, S. A. "Accurate reconstruction of peptide backbone using dead-end elimination and a knowledge-based forcefield", *Journal of Computational Chemistry*, **25**:16-27 (2004).

That manuscript describes the search procedure and scoring function in some detail. The method is also validated and benchmarked. Note that numerous enhancements have been applied to **BB** since the manuscript was written, so the current version will not exactly reproduce the published results... it should, in general, produce better results!¹

¹As of version BB-0.9-0, a suboptimal forcefield is distributed for the general case. The distributed forcefield is optimised for core structures at the expense of loop structures.

Additional details about **BB** are available in:

- Adcock, S. A. "BB: Accurate reconstruction of protein backbones", *Bioinformatics*, submitted.

3 Copyright and Licensing

BB, its documentation and its data files are copyright © 2003, Regents of The University of California. All rights reserved.

BB is freely distributed in binary format with no conditions other than a request that it is suitably cited if published works relied upon it in any way. The source code could be made available under certain circumstances. Send requests to `adcock@mccammon.ucsd.edu`.

This program is distributed in the hope that it will be useful, but **WITHOUT ANY WARRANTY**; without even the implied warranty of **MERCHANTABILITY** or **FITNESS FOR A PARTICULAR PURPOSE**.

4 Installation

Installation will depend upon the form of the distribution (i.e. source or binary) and the target platform. Brief details for the major cases are given in this section. The user is expected to use their initiative, however.

4.1 Binary Packages

The precompiled binary packages contain the executable programs, the necessary data files and the documentation.

4.1.1 RedHat Linux

Download the appropriate RPM package from <http://mccammon.ucsd.edu/~adcock/bb.html>. Each RPM package is specific to a particular hardware platform, and you will need the correct one. *bb-0.9-0.i386.rpm* is for x86-based computer systems and *bb-0.9-0.alpha.rpm* is for Alpha-based computer systems.

To install an RPM package, you must log in as the root user. To install or upgrade a package, type `rpm -Uvh filename`, where `filename` is the package, or packages, you want. To uninstall a package, type `rpm -e package_name`, in this case `package_name` will be `bb`. To view a list of all packages installed on your machine, type `rpm -qa`.

4.1.2 Debian Linux

Sorry, no `.deb` packages are available at this time.

4.1.3 Binary tarballs

Unpack the tarball by typing `"tar xvzf bb-0.9-0.arch.tgz"`, obviously where `bb-0.9-0.arch.tgz` is the name of the file you downloaded.

4.2 Source Packages

The source packages contain the raw source code for **BB** and its helper utilities. Before use they will require compilation. Source packages are intended for developers only. Full installation instructions may be found in the `INSTALL` file if you downloaded a tarball. Source RPMs may be installed as above. In both cases, compilation is performed via a two step process. Firstly, configuration must be performed by typing `"./configure [options]"` where the options are listed if you type `"./configure --help"`. Secondly, the source must be compiled by typing `"make"`, and optionally installed with `"make install"`.

4.2.1 Source tarball

The file `bb-0.9-0.tgz` contains compressed source code, along with data files. Extract files with `"tar -xvzf bb-0.9-0.tgz"`. This will create a directory called `"bb-x.x-0"`. Full installation instructions may be found in the text file named `"INSTALL"`. Basically, you will want to type `"./configure ; make"` to compile the source code and subsequently `"make install"` to install the files.

4.2.2 Source RPM

For convenience, the source code is also available in a RPM package. Install with `"rpm -Uvh bb-0.9-0.src.rpm"`.

5 Command-line usage

Typing "bb -usage" will produce output looking something like this:

Available command line options include:

```
--help, --usage, -h          This stuff.
--version, -v                Display version information then quit.
--log_level, -V INTEGER      Set logging level 0(none)-4(lots). [Default = 2]
--log_file, -L FILENAME      Set log file, none=stdout. [Default = stdout]
--quiet, -q                  Do not produce verbose output.
--diagnostic, -D             Output diagnostics and quit.
--test, -T                   Run DEE and superimpose tests and quit.
--seed, -s INTEGER           Set pseudo-random number seed value.
--aaiff, -a FILENAME         Select file containing SPLIFF-AAi forcefield data.
--aaxff, -A FILENAME         Select file containing SPLIFF-AAx forcefield data.
--bbff, -B FILENAME          Select file containing SPLIFF-BB forcefield data.
--topology, -t FILENAME      Select file containing SPLIFF topology data.
--lib, -l FILENAME           Select file containing BB fragment data.
--pdb, -p FILENAME           Select file containing C-alpha coordinates.
--out, -o FILENAME           Select file for output structure in PDB format.
--pout FILENAME              Select file for randomized structure in PDB format.
--minprob REAL               Set probability for acceptable fragments.
--energycutoff REAL          Set energy cutoff for DEE search. (Energy above minimum)
--random REAL                Amount of random perturbation to apply to input. (Default = 0.0)
--segments [[INT INT]...]    Specify segments to build. (Default = 1 N, where N is last residue)
```

They are your options. Some additional options are available, but not supported in any way.

The forcefield and fragment data files are essential, and suitable files are included in the data directory. It **BB** has been fully installed the data directory will, by default, be located at /usr/share/bb-x.x/data/. Otherwise, in the source tree, the data directory is in bb-x.x/data. In addition to the prerequisite data files, non-standard protein topology may be handled by specifying a topology file in SPLIFF

format. An example topology file is provided in the data directory as "spliff.top". The SPLIFF format is covered in documentation at <http://mccammon.ucsd.edu/~adcock/restricted/spliff.html>².

There are a couple of parameters that allow some tweaking of the balance between computational expense and overall accuracy of the calculations. "–minprob" allows one to disallow unlikely fragments based on their occurrence in a selection of structure from the PDB. Increasing this value excludes more fragments and therefore accelerates calculations. However, low energy structures that contain those less likely fragments will not be detected.

Most other options are provided for special case calculations and are not really recommended for general use.

5.1 Environment variables

A number of environment variables may be used to control some specialist behaviours of **BB**. **=== ADD MORE ===** .

5.2 An example calculation

In this example, a C_{α} trace of the Ubiquitin (1UBQ) [20] is used to build the entire backbone structure. The files may be found in the example subdirectory of the **BB** distribution. Build the mainchain atoms with:

```
bb -lib data/frag_1.0.lib -aaiff data/intra_aa_potential.dat -aaxff
data/inter_aa_potential.dat -topology data/default.top -pdb 1ubq.ca.pdb -out
out_1ubq.pdb 1ubq.log
```

²Access of this file is restricted to McCammon Group members only. It is not very interesting anyway.

The output file will be named, in this case, "out_1ubq.pdb". Simple, heh?

6 C-API usage

The **BB** internal routines are accessible via a C interface as a dynamic library. This aspect is not documented because I know of nobody besides myself that would need this, and I have the source as my documentation... please tell me if I'm wrong.

7 Acknowledgments

The author acknowledges Prof. J.A. McCammon and Dr. J.E. Nielsen, who took the time to provide constructive feedback on the backbone reconstruction manuscripts. The development of the **BB** software took place in the research group of Prof. J.A. McCammon at the University of California, San Diego. This research group is supported through grants from NSF, NIH, HHMI, SDSC, the W.M. Keck Foundation, and NBCR.

References

- [1] Rey, A.; Skolnick, J. *Journal of Computational Chemistry* **1992**, *13*, 443-456.
- [2] Bassolino-Klimas, D.; Buccoleri, R. E. *Proteins: Structure, Function, and Genetics* **1992**, *14*, 465-474.
- [3] Payne, P. W. *Protein Science* **1993**, *2*, 315-324.
- [4] Liwo, A.; Pincus, M. R.; Wawak, R. J.; Rackovsky, S.; Scheraga, H. A. *Protein Science* **1993**, *2*, 1697-1714.
- [5] Milik, M.; Kolinski, A.; Skolnick, J. *Journal of Computational Chemistry* **1997**, *18*, 80-85.
- [6] Feldman, H. J.; Hogue, C. W. V. *Proteins: Structure, Function, and Genetics* **2000**, *39*, 112-131.
- [7] Iwata, Y.; Kasuya, A.; Miyamoto, S. *Journal of Molecular Graphics and Modelling* **2002**, *21*, 119-128.
- [8] Kazmierkiewicz, R.; Liwo, A.; Scheraga, H. A. *Journal of Computational Chemistry* **2002**, *23*, 715-723.
- [9] Jones, T. A.; Thirup, S. *E.M.B.O. Journal* **1986**, *5*, 819-822.
- [10] Reid, L. S.; Thorton, J. M. *Proteins: Structure, Function, and Genetics* **1989**, *5*, 170-182.
- [11] Claessens, M.; van Cutsem, E.; Lasters, I.; Wodak, S. *Protein Engineering* **1989**, *2*, 335-345.
- [12] Holm, L.; Sander, C. *Journal of Molecular Biology* **1991**, *218*, 183-194.
- [13] Levitt, M. *Journal of Molecular Biology* **1992**, *226*, 507-533.
- [14] Correa, P. E. *Proteins: Structure, Function, and Genetics* **1990**, *7*, 366-377.

- [15] van Gelder, C. W. G.; Leusen, F. J. J.; Leunissen, J. A. M.; Noordik, J. H. *Proteins: Structure, Function, and Genetics* **1994**, *18*, 174-185.
- [16] van Hooft, P. A. M.; Holtje, H.-D. *Journal of Computer-Aided Molecular Design* **2000**, *14*, 719-730.
- [17] Mathiowetz, A. M.; Goddard III, W. A. *Protein Science* **1995**, *4*, 1217-1232.
- [18] Gan, K.; Coxon, J. M.; McKinnon, A. J.; Worth, G. H. *Biopolymers* **1997**, *41*, 391-407.
- [19] Adcock, S. A. *Journal of Computational Chemistry* **2003**, in press.
- [20] Vijay-Kumar, S.; Bugg, C. E.; Cook, W. J. *Journal of Molecular Biology* **1987**, *194*, 531-544.