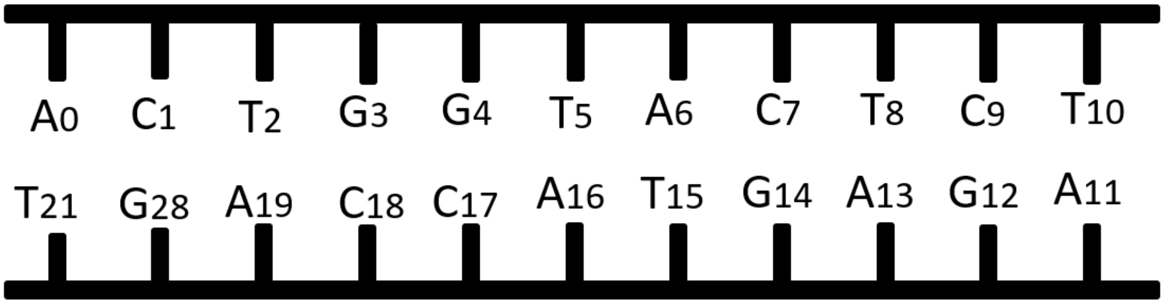

Figure 1: example of DNA with Bases labelled

RBB-NA input instructions

Aderik Voorspoels

December 16, 2022

Abstract

RBB-NA (Rigid Base Biassing for Nucleic Acids)[1] is a piece of software to enable biasing and ”on-the-fly” analysis of the so-called rigid base parameters[2] in strands of dsDNA and dsRNA. For the calculation it relies on a reimplementation of the Curves algorithm[3] and accordingly on the mclaughlin fitting procedure[4]. It operates as a collective variable in the framework of Plumed[5] and has been tested with gromacs[6]. A preprint of the journal article describing the RBB-NA algorithm can be found at `https://doi.org/10.48550/arXiv.2208.10286`. Please cite this and the other applicable references from the bibliography.

1 Instalation

To use RBB-NA one must add the RBB-NA source code to Plumed before installing Plumed it self. This is dine by copying the source code, the ”rbbNA.cpp” or ”rbbNAnoder.cpp” files, available from `https://github.com/AderikVoorspoels/RBB-NA.git` into the PLUMED/src/colvar directory. Where PLUMED is the path to your main PLUMED directory. After this is done one can compile Plumed as normal and patch it into you md engine of choice as described in the plumed documentation `https://www.plumed.org/doc-v2.8/user-doc/html/index.html`.

2 Declaring the Rigid base parameters

RBB-NA provides the capability to calculate the rigid base parameters of a double stranded piece of DNA or RNA. At the same time it will calculate the appropriate derivatives, allowing the parameters to be used as collective variables for biasing by plumed. There are three ways to input a piece of DNA, Sequence mode, Pair mode and Step mode.

2.1 Sequence mode

The most user friendly is the Sequence mode. Here the User provides two things, The sequence of DNA where they which to know and/or control the rigid base parameters and the four end points of this sequence. The sequence is provided in the direction that the atoms are numbered in your underlying

md engine (Using A, T, C, G, and U).
The endpoints are denoted by the number of the C'1 atom of the bases at the end. Note that these should be the numbers Plumed will receive from your md engine. In the case of gromacs these can be found in your .gro file. They are ordered going first with the sequence and then against the sequence on the opposing strand.
If for example one wanted to calculate rigid base parameters on the snippet shown in 1 between the base T_2 and C_7 (and the bases oposing those), The Plumed input line for this could be:

```
seq: RBBNA ENDS=167,324,546,868 SEQUENCE=TGGTAC FITTED
```

Here "var" is a label used in the rest of the plumed input that refers to this set of Rigid base parameters and "RBBNA" is the action used to define the collective variable of the parameters. The "ENDS" argument specifies the C'1 atoms of $T_2 = 167$, $C_7 = 324$, $G_{14} = 546$ and $A_{19} = 868$ and the "SEQUENCE" argument informs RBB-NA about the sequence of bases. With this information RBB-NA can find the C'1 atoms of the bases between T_2 and C_7 as well as those on the opposite strand. The "FITTED" key word here is a flag that will be discussed in section 2.4.
Notably this mode will provide all $12N - 6$ rigid base pair parameters as collective variables on which forces can be applied. If one only wants to read out the parameters (or for testing purposes) they can forego the calculation of derivatives by replacing "RBBNA" by "RBBNANODER". Be aware that this will also make RBBNA unable to enact forces or enforce biasing potentials.

2.2 Pair mode

Alternatively just the rigid base parameters in a pair can be calculated by using the pair mode. Here one has two options, either one can rely on the base completion or you can provide the triplet of atoms needed for calculating the base frame. When relying on the base completion the input for calculating the pair parameters in the T_2-A_{19} pair in the diagram becomes:

```
pair: RBBNA Base1=167 Base2=868 TYPE1=T TYPE2=A FITTED
```

Here "var" and "RBBNA" have the same function as above. The "Base1" and "Base2 arguments again tell RBBNA where it can find the C'1 atoms of the desired bases. The "TYPE1" and "TYPE2" arguments give the type of bases dealt with. Notably here The keyword "FITTED" is important as it indicates RBB-NA should use base completion, ie find the atoms of the two bases based on the index of the C'1 atom and the type of base.
To work with non canonical bases, or to skip the fitting procedure one can instead use the "Base1" and "Base2" arguments to specify the indices of the three atoms relevant to define the local reference frame of the bases. These are C'1, N9 and C4 in purine bases and C'1, N1 and C2 in pyrimidine bases. Such an input for the same pair would look like:

```
pair: RBBNA Base1=167,169,179 Base2=868,870,883
```

Here the types of bases don't need to be specified, if done anyway RBB-NA will simply disregard them. Note that the "FITTED" key word has not been included. Fitting can only be used in conjunction with base completion, adding the "Fitted" key word when specifying more than just the C'1 atom will accordingly result in an error. It is at the moment not possible to use base completion but not fitting when the input is given in pair mode.

2.3 Step mode

The Use of Step mode Inputs is very similar to pair mode input. The only difference is that now a step must be specified, and as such only the step parameters will be computed. Using Base completion and fitting input for the parameters of step C_7-T_8 would be:

```
step: RBBNA Base1=324 Base2=546 Base3=354 Base4=514 TYPE1=C TYPE2=G TYPE3=T TYPE4=A FITTED
```

Note that Base1 opposes Base2 and Base3 opposes Base4, also note Base1 and Base 3 are on the same strand.
Similar to the Pair mode one can forego fitting and base completion by instead specifying three atoms (C'1, N9/N1 and C4/C2). For the same set of bases this would be done by:

```
step: RBBNA Base1=324,326,336 Base2=546,548,562 Base3=354,356,368 Base4=514,516,529
```

2.4 Additional flags

Aside from the above discussed input arguments RBB-NA uses two key words. The first is ”FITTED” which turns on the fitting procedure used to reduce noise from fluctuations of the base. As mentioned above it also controls whether or not base completion will be used in Pair and step modes.
The Second Key word input is ”RNA”. This key word must be used when working on RNA. It makes sure the complementary sequence in sequence mode uses Uracil to oppose Adenine instead of Thymine and changes the code to find the bases to deal with the ribose instead of desoxyribose sugar.

3 Outputting

After calling the ”RBBNA” or ”RBBNANODER” action in a Plumed input file one can use the labels given to the collective variables to access the individual rigid base parameters. If input was given in pair mode and labelled ”pair” one can use ”pair.buckle”, ”pair.propeller”, ”pair.opening”, ”pair.shear”, ”pair.stretch” and ”pair.stagger” to access the corresponding parameters.
If step input was used with the label ”step” one can use ”step.tilt”, ”step.roll”, ”step.twist”, ”step.shift”, ”step.slide” and ”step.rise” to access the rigid pair parameters.
Finally if sequence mode input is employed on a sequence of length N base pairs, the resulting collective variable will have 12N-6 components. These are accessible in the same manner as with pair and step mode by appending the index in the sequence of the preceding base pair. when the input from the section on sequence mode2.1 was used one can then get the rise in the T_2-G_3 step by using ”seq.rise0” and the buckle in the T_5-A_{16} pair by using ”seq.buckle3”.
These components can be used in the Plumed actions that take collective variables, notably a statement like:

```
PRINT ARG=seq.tilt3,seq.roll3,seq.twist3 stride=100 FILE=out.dat
```

will every 100 timesteps print a line containing the tilt roll and twist of the T_5-A_6 step to a file named ”out.dat”. Importantly if The ”RBBNA” action was used they can also serve as input for plumed actions which bias colective variable such as ”RESTRAINT”.

4 Bias in Brief

Finally, a short note on how to add bias to rigid base parameters. Because RBB-NA implements the rigid base parameters as collective variables, all functionality of Plumed for such variables can be used. For a full view of the possibilities the reader is referred to the documentation of Plumed: https://www.plumed.org/doc-v2.8/user-doc/html/index.html.
If the goal is to add a simple quadratic potential to one of the parameters adding a line similar to:

```
bias: RESTRAINT ARG=seq.tilt0 KAPPA=1000 AT=0.1
```

to the plumed input will work. Here ”bias” is a label like before and ”RESTRAINT” is a plumed action that can be used to add a quadratic potential or constant force to a collective variable. The ”ARG” argument is used to denote which collective variable should be biased (in this case the tilt in the T_2-G_3 step, following the input in section2.1). The ”KAPPA” argument gives the strength of the quadratic potential (here $1000kJ/mol$) and the ”AT” argument gives the center of the potential in radians (when applied to rotational parameters) or in nm (when applied to translational parameters). in this case the ”bias” would be a restraint adding this potential:

$$V_\alpha = \frac{1000kJ/mol}{2}\left(\Omega_1(0) - 0.1rad\right)^2 \tag{1}$$

Multiple such restraints can be combined by either adding arguments separated by a comma like:

```
bias: RESTRAINT ARG=seq.tilt0,seq.roll3 KAPPA=1000,500 AT=0.1,-0.3
```

or by adding different lines like:

```
bias1: RESTRAINT ARG=seq.tilt0 KAPPA=1000 AT=0.1
bias2: RESTRAINT ARG=seq.roll3 KAPPA=500 AT=-0.3
```

References

[1] Voorspoels, A., Vreede, J., and Carlon, E. (2022) Rigid Base Biasing in Molecular Dynamics enables enhanced sampling of DNA conformations.

[2] Olson, W. K., Gorin, A. A., Lu, X.-J., Hock, L. M., and Zhurkin, V. B. (1998) DNA sequence-dependent deformability deduced from protein–DNA crystal complexes. *Proc. Natl. Acad. Sci. USA,* **95**, 11163–11168.

[3] Lavery, R., Moakher, M., Maddocks, J., Petkeviciute, D., and Zakrzewska, D. (2009) Conformational analysis of nucleic acids revisited: Curves+. *Nucl. Acids Res.,* **37**(37), 5917–5929.

[4] McLachlan, A. D. (1979) Gene Duplications in the Structural Evolution of Chymotrypsin. *J. Mol. Biol.,* **128**, 49–79.

[5] The PLUMED consortium (Aug, 2019) Promoting transparency and reproducibility in enhanced molecular simulations. *Nature Methods,* **16**(8), 670–673.

[6] Abrahams, M., Murtola, T., Schulz, R., Páll, S., Smith, J., Hess, B., and Lindahl, E. (2015) GROMACS: high performance molecular simulations through multi-level parallelism from laptops to supercomputers. *SoftwareX,* **1-2**, 19–25.