BRIQ 2

User Manual

(Last modified: Jul 18, 2024)

Contents

1 Introduction **2**

2 License **3**

3 Installation **4**
3.1 Prerequisites 4
3.2 Installation steps 4

4 Usage **5**
4.1 Commands 5
4.2 Input file formats 5
4.2.1 Required arguments 6
4.2.2 Optional arguments 6
4.3 Output formats 7

5 Citing BRIQ2 **8**

1 Introduction

BRIQ2 is a novel algorithm for RNA tertiary structure modeling. Unlike any previous methods, such as homologies modeling, template searching, and deep learning algorithms, BRIQ2 is the first fully De Novo RNA tertiary structure modeling algorithm which discretizes the base orientation space, rotamer space and phosphate space to sample varieties of conformations and then employ BRIQ2 energy score to find the most stable conformation. The BRIQ2 program was developed by Peng Xiong at University of Science and Technology of China in collaboration with Ke Chen, Heqin Zhu. The source code is publicly available for downloading from

https://github.com/xionglab2023/BRIQ2

under the **GPLv3 license** (See more details in Section 2).

If you have any comments or suggestions, welcome to contact Peng Xiong (*xiongxp@ustc.edu.cn*).

2 License

3 Installation

Currently, BRIQ2 is only available for Unix-based systems.

3.1 Prerequisites

- *nix operating system
- gcc 12.3

3.2 Installation steps

1. Clone the GitHub repo of BRIQ2 and change to the directory.

   ```
   $ git clone git@github.com:xionglab2023/BRIQ2.git
   $ cd BRIQ2
   ```

2. Download data from and decompress it in BRIQ2 directory.

   ```
   $ tar -xzf briq_data.tar.gz
   ```

3. Modify variable of *BRIQX_DATAPATH* in setup.sh.

4. Add BRIQ2 binary to system path, temporary change:

   ```
   $ export PATH=$PATH:$HOME/apps/BRIQX/bin
   ```

   Permanent change: Modify $HOME/.bashrc

5. Run setup program.

   ```
   $ bash ./setup.sh
   ```

4 Usage

BRIQ2 takes an ”input_file” as input and outputs predicted structure in **PDB** format. The ”input_file”, in **txt** format, specifies necessary arguments and flexible constraints for an input sequence target.

4.1 Commands

BRIQ2 can handle three tasks: coarse-grained modeling, prediction and refinement. These tasks are executed in a single executable named **briqx_run**. Task types are specified in ”input_file” (described in Section 4.2).

- Coarse-grained modeling

```
$ briqx_run -in $INPUT -out $OUTPUT -seed $SEED -n $N_KEY -mp $N_THREAD
```

- Prediction or Refinement. (Specify task argument in input_file)

```
$ briqx_run -in $INPUT -out $OUTPUT -seed $RANDOM_SEED
```

Examples:

- Coarse-grained modeling

```
$ briqx_run -in input -out seq.key -seed 1234 -n 2000 -mp 1
```

- Prediction or Refinement. (Specify task argument in input_file)

```
$ briqx_run -in input -out seq.pdb -seed 1234
```

4.2 Input file formats

The following content is an example of the input file of BRIQ2.

input_file

```
task predic
pdb /public/home/pengx/casp/R1209/fragA/init.pdb
seq GGCACUAUGGGCGCAGCUGACGGUACAGGCCAGAUCUGGUAUAGUGCC
sec (((((((((..(((((()))..))...)).((((((((((.)))))))
cst FFF00000A0000000000000000A0000000000000000000FFF
csn FF00000000000000000000000000000000000000000000FF
cnt ---------------|-----------------|-------------|
key !!!"!!!!!P!!!"!#!!!"!O!!!#!N!!!$!M!!!%!L!!!&!'!!!&!K!!!'!J
    !!!(!)!!!(!;!!!)!*!"!)!:!!!*!+!!!+!,!!!+!7!$
    !,!-!!!,!6!!!-!5!!!.!3!!!/!0!!!/!2!!!0!1!!!2!3!!!3!4!-!4!5!!!8!9!#!9!:!!!;!<!"!<!
    I!$!=!>!!!=!H!!!>!G!!!?!F!!!@!A!!!@!E!!!A!B!!!A!D!!!B!C!!!E!F
    !!!F!G!!!H!I!!!I!J"$!K!L!!!L!M!!!M!N!!
```

```
ct 0 10 wc-GC
ct 1 9 nat_move
ct 4 6 nat_cluster
ct 2 8 dm16 1.4201 5.1777 6.3092 5.7182 5.1753 10.5323 11.4260
    9.5116 6.2936 11.4024 10.1200 5.9646 5.7007 9.5167 5.9944
    6.5222
ct 3 7 cm -0.188591 -0.870345 0.130125 0.1077081 0.8045070
    -0.5840955 -0.1524416 0.5939300 0.7899422 0.9824258 0.0039572
    0.1866114
ct 5 20 clusters -1 0.23 0 -0.33 1 0.12
```

Each line in the input file contains an argument and correlated value. Among them, **task**, **pdb**, **seq**, **sec** are required arguments, while the others are optional arguments which are used for constraints when sampling. The detailed explanations are as follows:

4.2.1 Required arguments

- **task**. Its value should be either "predict" or "refine", which indicates the different two tasks of BRIQ2, namely prediction and refinement of RNA tertiary structure.
- **pdb**. The path of the initial conformation of RNA structure in format of PDB, which can be generated by BRIQ2.
- **seq**. The target RNA base seqeunce to be predicted for tertiary structure. The sequence should be composed of the four kinds of bases only, namely 'A', 'U', 'G' and 'C'.
- **sec**. The reference dot-bracket notations of secondary structure of input RNA sequence, which should consists of valid paired brackets.

4.2.2 Optional arguments

- **cst**. Constraints on edges. When sampling, BRIQ2 will generate a graph-like structure, named as NuGraph, and its minimum spanning tree, named as NuTree, in which each node represents a base and each edge represents an SE(3) transformation from one base to another base. The argument **cst** is a constraint for each edge. '0' indicates no constraint and other letters such as 'A', 'B', ..., 'F' indicate correlated edge is fixed, which means this edge will be initilalized with the initial conformation and won't be chaged when sampling.
- **csn**. Constraints on nodes. Similar to the above argument **cst**, this argument is the constraint of the base nodes. Note that with this constraint, the base rotamer is determinated while the robose rotamer and phosphate rotamer are undetermined.
- **cnt**. Continuation, indicates the chain breaks. '-' represents no chain breaks while '|' represents chain breaks.

- **key**. A hash key representation of the above NuGraph, which is generated by BRIQ2 in coarse-grained sampling procedure. In BRIQ2 program, the SE(3) transformation is discretized into 9 billion clusters, with each transformation equal to one 16-dimension vector, named as DM16. Each DM16 is encoded into hash code which begins with the letter '!'. Therefore, in the NuGraph, each edge contains one transformation (equals to an DM16), in which this transformation is encoded into a hash code. The argument **key** is the concatenation of all hash codes from all edges of the NuGraph.

- **ct**. More specified constraints. There are five categories of values for this arguments. Values in form of *"i j type"* represent the edge connecting the i-th and j-th bases is the specified type, namely "wc-GC" for Watson-Crick GC pair, "nat_move" for native moveSet or "nat_cluster" for native cluster of transformation. Values in form of *"i j type x1 x2 ..."* represent the extract transformation (cm, 4×3 rotation-translation matrix) or DM16, which determines the transformation of base i and base j and keeps it unchanged in sampling process.

4.3 Output formats

The results of BRIQ2 refinement or prediction of RNA tertiary structures are given in format of PDB.

5 Citing BRIQ2

If you use BRIQ2 in your work, please cite:

Xiong, P., Wu, R., Zhan, J., & Zhou, Y. (2021). Pairing a high-resolution statistical potential with a nucleobase-centric sampling algorithm for improving RNA model refinement. Nature communications, 12(1), 2777.