# August 2

From doi:10.1371/journal.pone.0026400 on the PLoS website, downloaded Figures S2 and S3.

Copied Qall from Figure S2 into a text file, "parser test.txt", to test my parser.

Checking a few values at random.

Row N, column Q should be 2.511.

>>> x['N']['Q']

'2.511'

Row E, column S should be .728.

>>> x['E']['S']

'0.728'

Row S, column E should be .728.

>>> x['S']['E']

'0.728'

Here's the working code:

**def** parse**(**mat\_file**):**

found\_resns **=** False

**for** line **in** mat\_file**:**

# Ignore comments

**if** line**[**0**]** **==** '#'**:**

**continue**

# Find which resn corresponds to which column number

**if** **not** found\_resns**:**

found\_resns **=** True

col\_names **=** line**.**split**()**

# Check to make sure they're all there

**for** resn **in** **[**'C'**,** 'N'**,** 'H'**,** 'D'**,** 'S'**,** 'Q'**,** 'K'**,** 'M'**,** 'P'**,**

'T'**,** 'F'**,** 'A'**,** 'G'**,** 'I'**,** 'L'**,** 'R'**,** 'W'**,** 'E'**,**

'Y'**,** 'V'**]:**

**assert** resn **in** col\_names**,** "missing " **+** resn

# Make the matrix that will be returned

output **=** dict**([(**resn**,** **None)** **for** resn **in** col\_names**])**

**for** key**,** value **in** output**.**items**():**

output**[**key**]** **=** dict**([(**resn**,** **None)** **for** resn **in** col\_names**])**

**continue**

row **=** line**.**split**()**

row\_name **=** line**[**0**]**

**for** rate**,** col\_name **in** zip**(**row**[**1**:],** col\_names**):**

output**[**row\_name**][**col\_name**]** **=** rate

**return** output

"x" was generated with the following commands:

**>>>** execfile**(**r'C:\cygwin\home\alex\beta barrels\bbtm derivation\bbtm.py'**)**

**>>>** f **=** open**(**'parser test.txt'**,** 'r'**)**

**>>>** x **=** parse**(**f**)**

I used Mathematica to find . Surprisingly, it was pretty much uniform, that is, all the same number. When I changed 40 to a very small nuber, it was no longer uniform. It's as if 40 is an incredibly large evolutionary time, such that a position becomes equally likely to be any residue.

However, against this interpretation, if I put in a huge number like 100 000, all the numbers get very small (and still very uniform). The matrix does not appear to be a stochastic matrix, which makes me doubt it is really a transition probability matrix.

Also, I do not see why positions would become equally likely to be any residue after a very large time. Some states I would expect to be stronger attractors than others; I'd expect a non-uniform stationary state, in other words.