Calculation of Q-scores from Command Line

Updated June 29, 2021
MapQ v. 1.6.5

First, download and install UCSF Chimera and the MapQ plugin:
https://github.com/gregdp/mapq

Q-scores can be calculated from the command line using a python script. The script is in the zip file downloaded when installing the MapQ plugin, it has the name mapq_cmd.py. The command has the following format:

```
python mapq_cmd.py [path to Chimera] [path to map] [path to model 1] [path to model 2] [path to model …] [option1] [option2] …
```

Example:

```
python mapq_cmd.py ~/_mol/Chimera ~/Dropbox/_data/Ferritin/k175/k175_fi.mrc ~/Dropbox/_data/Ferritin/k175/3ajo_f.pdb np=4 res=3.0 bfactor=150
```

Parameters:

- [path to Chimera]
  - **required**: the path where Chimera is installed **(not to the binary file)**
- [path to model #]
  - **required**: at least one model should be specified, but more can also be included
- np=N (e.g. np=4)
  - optional – default is 1 - it specifies number of processes to use; use up to the number of cores (hyperthreading does not seem to offer an advantage)
- res=N.N (e.g. res=3.2)
  - optional - default is 3.0 - specifies resolution of map; it is used to output per-residue statistics along with expected Q-score at this resolution
- bfactor=N (suggested bfactor=150)
  - optional - if specified, a separate pdb file will be written where bfactor=N*(1-Q-score) for each atom. This file can be used to generate a model-map that more closely matches the cryoEM map, e.g. with Phenix, as follows:
    - `phenix.fmodel high_resolution=[res] scattering_table=electron [pdb file]`
      - may also need to add `generate_fake_p1_symmetry=True`
    - `phenix.mtz2map high_resolution=[res] include_fmodel=true scattering_table=electron [pdb file] [pdb file].mtz`
- sigma=N.N (e.g. sigma=0.6)
  - optional – default is 0.6 – specifies width of reference Gaussian
    - for sigma=0.6, the higest Q-score of 1 occurs at ~1.5Å resolution
      - current statistics (e.g. expected Q-score at a given resolution) are using this value currently
    - for sigma=0.4, highest is at ~1.0Å
      - this has been used to compare newer maps up to 1.15Å resolution

The output should be something like this (will vary slightly depending on whether multi-processing was used or not, and whether more than one model was specified):

```
Q Scores:
 - map: emd_20026.map
 - mol: 3ajo.pdb, chain: _all_
 - sigma: 0.50
 - mind: -0.061, maxd: 0.109
 - atoms to do: 1714
 - path: /Users/greg/Desktop/Chimera.app/Contents/Resources/share/mapqp.py
 - on mac

[...]

 - done, time: 70.172916
 - done, time: 1 min, 10.2 sec
Saving pdb with Q-scores: /Users/greg/_data/mmetrics/emd_20026/3ajo__Q__emd_20026.pdb

Q for 413 res...
Model Q-score: 0.776

Chain   Avg.Q-score    Est.Res.(A)
 chain A (hoh) 0.506   3.45
 chain A (mg)  0.510   3.42
 chain A (protein)     0.820   1.69

Type    Avg.Q-score    Est.Res.(A)
 Protein       0.820   1.69
 MG     0.510   3.42
 HOH    0.506   3.45

Map: emd_20026.map, Model: 3ajo.pdb
 --> avg. Q score: 0.776

Saving per-chain & per-residue Q-scores:
 -> res= 1.75
 -> file: /Users/greg/_data/maps2/FerritinT/3ajo__Q__emd_20026_All.txt
```

- The score for the entire model is on the last line (0.776).
- The other few last lines summarize the scores by atom type: protein, nucleic acids, and other, e.g. Mg, Hoh (water atoms).
- Per-atom Q-scores are stored in the pdb file [model name]__Q__[map name] in the B-factor column. Note that the full path to the model must be given in the command for this file to be put in the same folder as this model.