manual_psa

April 6, 2018

1 Hausdorff distance for cpptraj

We use the *Hausdorff metric* (see [1]) with the RMSD between protein structures as a measure of distance between trajectories P and Q:

$$H(P,Q) := \max[h(P,Q), h(Q,P)] \tag{1}$$
$$h(P,Q) := \max_{p\in P} \min_{q\in Q} \mathrm{RMSD}(p,q) \tag{2}$$

The brute-force calculation can be formulated in terms of a 2D RMSD distance matrix $\mathtt{d} = (d_{ij})$, $1 \le i \le N_P$, $1 \le j \le N_Q$ (frame i in P vs frame j in Q):

$$H(\mathtt{d}) = \max[h_{PQ}(\mathtt{d}), h_{QP}(\mathtt{d})] \tag{3}$$
$$h_{PQ}(\mathtt{d}) = \max_{1\le i\le N_P} \min_{1\le j\le N_Q} d_{ij} \tag{4}$$
$$h_{QP}(\mathtt{d}) = \max_{1\le j\le N_Q} \min_{1\le i\le N_P} d_{ij} \tag{5}$$

There is a generally faster early-break algorithm for h [2], implemented as scipy.spatial.distance.directed_hausdorff(), which avoids calculating the whole distance matrix. However, there is no parallel version of the Taha algorithm that we are aware of. On the other hand, the distance matrix calculation is pleasingly parallel and can be scaled for large distance matrices.

1. D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge. Comparing images using the Hausdorff distance. IEEE Transactions on Pattern Analysis and Machine Intelligence, 15(9):850–863, 1993.
2. A. A. Taha and A. Hanbury. An efficient algorithm for calculating the exact Hausdorff distance. IEEE Trans Pattern Anal Mach Intell, 37(11):2153–63, Nov 2015.

1.1 Method implementations

```
In [1]: import numpy as np
```

```
        def load_2drmsd(filename):
            d = np.loadtxt(filename)
            d = d[:, 1:]  # strip first column (frame number)
```

```
        return d

    def Hausdorff_simple(d):
        hPQ = np.max(np.min(d, axis=1))
        hQP = np.max(np.min(d, axis=0))
        return np.max([hPQ, hQP])
```

1.2 Results

Data files:

```
In [2]: %ls *.dat

rmsd2.1-1.dat  rmsd2.1-2.dat
```

Compute the Hausdorff distance for the sample files:

```
In [3]: d = load_2drmsd("rmsd2.1-1.dat")
        Hausdorff_simple(d)

Out[3]: 0.0
```

The data file contains the 2D RMSD for the same trajectory because the Hausdorff distance is zero.

The second data file shows that the two trajectories are different:

```
In [4]: d = load_2drmsd("rmsd2.1-2.dat")
        Hausdorff_simple(d)

Out[4]: 1.872
```