Finding Nearest Neighbors in Crystallography with NearTree

L Andrews[1], H Bernstein[2]
[1]retired, Kirkland, WA, [2]Ronin Institute for Independent Scholarship, c/o NSLS-II, Brookhaven National Lab, Bldg 745, Bellport, NY
larry6640995@gmail.com

Crystallography involves analysis of distances among points in several spaces: 3-dimensional real and reciprocal spaces, 1-, 2-, 3-, 4-, 5-, and 6- dimensional spaces of lattice parameters, and higher dimensional spaces in which those spaces may be embedded. In any or all of these spaces we may need to find one or more nearest neighbors, one or more farthest neighbors, centroids, or clusters. All of these tasks need efficient data structures with appropriate functions to allow us to implement efficient searches.

NearTree is an implementation of a solution to the nearest neighbor problem. It is based on the algorithm described by Kalantari and McDonald in 1983. The mechanics of NearTree has replaced their complex operation with recursion. Also known as the post office problem, nearest neighbor searches arise in many contexts, including many that are familiar to crystallographers.

NearTree is based on a binary tree structure. It is simple to use, and in practice it is found to operate with good speed. For the average dataset, insertion of a new data point has complexity O(n ln(n)). Retrieval of the nearest data point has complexity O(n). That means that NearTree's operation is optimal! In general, optimal algorithms will perform better than any others, at least for sufficiently large datasets; sufficiently large is often fairly small.

NearTree is designed to use an external distance measure. It can work without prior knowledge of the dimensionality of the underlying data space, and can be used as a tool to help discover local Hausdorff dimensions. All post office problems need to deal with Bellman's "Curse of Dimensionality," which makes searches in higher dimensions more challenging than searches in lower dimensions. In our experience, we commonly use NearTree for three and six dimensions, but we have used NearTree for higher dimensions, up to 36 in the case of 6x6 matrices.

One of the design objects of NearTree is ease of use. Only a few function calls, typically 2 or 3, are necessary to create a NearTree. A single function call will return the nearest neighbor. Besides the nearest neighbor, NearTree provides several other search functionalities. Farthest neighbor, those within a sphere, and those within an annulus are some of the additional search modes.

Among the current practical crystallographic uses for NearTree are analysis of atomic bonding in RasMol, searching with SAUC for close cell parameters matches in the PDB or COD for molecular replacement, and ad hoc contact searches. Our current work includes analysis of cell clusters in serial crystallography.

Andrews, Larry. "A template for the nearest neighbor problem." C/C++ Users Journal 19, no. 11 (2001): 40 -- 49.

Andrews, Lawrence C., and Herbert J. Bernstein. "NearTree, a data structure and a software toolkit for the nearest-neighbor problem." J. Appl. Cryst. 49, no. 3 (2016): 756-761.

Bellman, Robert. "Curse of dimensionality." Adaptive control processes: a guided tour. Princeton, NJ 3 (1961): 2.

Hausdorff, Felix. "Dimension und äußeres Maß." Mathematische Annalen 79, no. 1-2 (1918): 157-179.

Kalantari, Iraj, and Gerard McDonald. "A data structure and an algorithm for the nearest point problem." IEEE Transactions on Software Engineering 5 (1983): 631 -- 634.

McGill, Keith J., Mojgan Asadi, Maria T. Karakasheva, Lawrence C. Andrews, and Herbert J. Bernstein. "The geometry of Niggli reduction: SAUC – search of alternative unit cells." J. Appl. Cryst. 47, no. 1 (2014): 360 -- 364.