CLAIMS

1. A computer-implemented method of efficiently determining which elements of a sequence are nearest to uniformly spaced cells in a grid, wherein the elements have element coordinates, and ~~the~~ cells have dimension-wise cell indices and cell coordinates, the computer-implemented method comprising:

generating an element-to-cells mapping that maps, to each of the elements, a subset of the cells, wherein the subset of the cells mapped to a particular element in the sequence includes a nearest cell in the grid and one or more neighborhood cells in the grid, wherein the nearest cell is selected based on matching element coordinates of the particular element to the cell coordinates, and wherein the one or more neighborhood cells are contiguously adjacent to the nearest cell and selected based on being within a distance proximity range from the particular element;

generating a cell-to-elements mapping that maps, to each of the cells, a subset of the elements, wherein the subset of the elements mapped to a particular cell in the grid includes those elements in the sequence that are mapped to the particular cell by the element-to-cells mapping; and

using the cell-to-elements mapping to determine, for each of the cells, a nearest element in the sequence, wherein the nearest element to the particular cell is determined based on distances between the particular cell and the elements in the subset of the elements.

2. The computer-implemented method of claim 1, wherein ~~the~~ matching the element coordinates of the particular element to the cell coordinates further includes truncating a decimal portion of the element coordinates to generate truncated element coordinates.

3. The computer-implemented method of claim 1 or 2, wherein ~~the~~ matching the element coordinates of the particular element to the cell coordinates further includes:

for a first dimension, matching a first truncated element coordinate in the truncated element coordinates to a first cell coordinate of a first cell in the grid, and selecting a first dimension index of the first cell;

for a second dimension, matching a second truncated element coordinate in the truncated element coordinates to a second cell coordinate of a second cell in the grid, and selecting a second dimension index of the second cell;

for a third dimension, matching a third truncated element coordinate in the truncated element coordinates to a third cell coordinate of a third cell in the grid, and selecting a third dimension index of the third cell;

using the selected first dimension index, the selected second dimension index, and the selected third dimension ~~indices~~ index to generate an accumulated sum based on position-wise weighting the selected first dimension index, the selected second dimension index, and the selected third dimension ~~indices~~ index by powers of a radix; and

using the accumulated sum as a cell index for selection of the nearest cell.

4. The computer-implemented method of any of claims 1-3, wherein the distances are calculated between cell coordinates of the particular cell and the element coordinates of the elements in the subset of the elements.

5. The computer-implemented method of any of claims 1-4, wherein the sequence is a protein sequence of amino acids.

6. The computer-implemented method of claim 5, wherein the elements are atoms of the amino acids from the protein sequence of amino acids.

7. The computer-implemented method of claim ~~5 or~~ 6, wherein ~~the steps of~~ generating the element-to-cells mapping, generating the cell-to-elements mapping, and using the cell-to-elements mapping to determine, for each of the cells, the nearest element have a runtime complexity of O(a * f + v), wherein:

a is a number of the atoms of the amino acids,

f is a number of the amino acids,

v is a number of the cells, and

* is a multiplication operation.

8. The computer-implemented method of claim 6 or 7, wherein the atoms include alpha carbon atoms.

9. The computer-implemented method of any of claims 6-8, wherein the atoms include beta carbon atoms.

10. The computer-implemented method of any of claims 6-9, wherein the atoms include non-carbon atoms.

11. The computer-implemented method of any of claims 1-10, wherein the cells are three-dimensional voxels.

12. The computer-implemented method of any of claims 1-11, wherein the cell coordinates are three-dimensional coordinates.

13. The computer-implemented method of any of claims 1-12, wherein the element coordinates are three-dimensional coordinates.

14. The computer-implemented method of any of claim 1-13, wherein the one or more neighborhood cells are selected based on being within an index adjacency range from the nearest cell.

15. The computer-implemented method of any of claims 1-14, wherein the one or more neighborhood cells are selected based on being within a cell neighborhood in the grid that includes the nearest cell.

16. The computer-implemented method of any of claims 1-15, wherein the sequence includes M elements, wherein the subset of the elements includes N elements, and wherein M > > N.

17. A computer-implemented method of efficiently determining which atoms in a protein are nearest to voxels in a grid, wherein the atoms have three-dimensional (3D) atom coordinates, and the voxels have 3D voxel coordinates, including:

generating an atom-to-voxels mapping that maps, to each of the atoms, a containing voxel selected based on matching 3D atom coordinates of a particular atom of the protein to the 3D voxel coordinates in the grid;

generating a voxel-to-atoms mapping that maps, to each of the voxels, a subset of the atoms, wherein the subset of the atoms mapped to a particular voxel in the grid includes those atoms in the protein that are mapped to the particular voxel by the atom-to-voxels mapping; and

using the voxel-to-atoms mapping to determine, for each of the voxels, a nearest atom in the protein.

18. The computer-implemented method of claim 17, wherein generating the atom-to-voxels mapping, generating the atom-to-voxels mapping, and using the voxel-to-atoms mapping to determine, for each of the voxels, the nearest atom~~the steps of claim 17~~ have a runtime complexity of O(number of atoms).

19. A system for efficiently determining which atoms in a protein are nearest to voxels in a grid, wherein the atoms have three-dimensional (3D) atom coordinates, and the voxels have 3D voxel coordinates, the system including:

at least a processor; and

a non-transitory computer readable storage medium storing instructions that, when executed by at least the processor, cause the system to:

generate an atom-to-voxels mapping that maps, to each of the atoms, a containing voxel selected based on matching 3D atom coordinates of a particular atom of the protein to the 3D voxel coordinates in the grid;

generate a voxel-to-atoms mapping that maps, to each of the voxels, a subset of the atoms, wherein the subset of the atoms mapped to a particular voxel in the grid includes those atoms in the protein that are mapped to the particular voxel by the atom-to-voxels mapping; and

use the voxel-to-atoms mapping to determine, for each of the voxels, a nearest atom in the protein.

20. The system of claim 19, further storing instructions that, when executed by at least the processor, cause the system to:

generate the atom-to-voxels mapping using a runtime complexity determined by a number of the atoms;

generate the voxel-to-atoms mapping using the runtime complexity determined by the number of the atoms; and

use the voxel-to-atoms mapping to determine, for each of the voxels, the nearest atom in the protein based on distances between each voxel of the voxels and each atom in the subset of the atoms.