Handbook of Chemoinformatics Algorithms

Faulon Jean-Loup, Bender Andreas

Three-Dimensional (3D) Molecular Representations

Publication details

L. Willighagen Egon

Published online on: 21 Apr 2010

How to cite: L. Willighagen Egon. 21 Apr 2010, Three-Dimensional (3D) Molecular Representations from: Handbook of Chemoinformatics Algorithms CRC Press

Accessed on: 16 Oct 2019

PLEASE SCROLL DOWN FOR DOCUMENT

Full terms and conditions of use: https://www.routledgehandbooks.com/legal-notices/terms

This Document PDF may be used for research, teaching and private study purposes. Any substantial or systematic reproductions, re-distribution, re-selling, loan or sub-licensing, systematic supply or distribution in any form to anyone is expressly forbidden.

The publisher does not give any warranty express or implied or make any representation that the contents will be complete or accurate or up to date. The publisher shall not be liable for an loss, actions, claims, proceedings, demand or costs or damages whatsoever or howsoever caused arising directly or indirectly in connection with or arising out of the use of this material.
3 Three-Dimensional (3D) Molecular Representations

Egon L. Willighagen

CONTENTS

3.1 Introduction ......................................................................65
3.2 Coordinate Systems .............................................................67
  3.2.1 Cartesian Coordinates ..................................................67
  3.2.2 Internal Coordinates ....................................................68
  3.2.3 Fractional Coordinates ..................................................69
  3.2.4 Two-Dimensional Chemical Diagrams .................................70
3.3 Interconverting Coordinate Systems ...........................................71
  3.3.1 Internal Coordinates into Cartesian Coordinates ......................71
  3.3.2 Fractional Coordinates into Cartesian Coordinates ...................72
3.4 Comparing Geometries .........................................................73
3.5 Fixed-Length Representations ..................................................74
  3.5.1 Molecular Descriptors ..................................................75
    3.5.1.1 The Length-over-Breadth Descriptor ............................75
    3.5.1.2 Charged Partial Surface Area (CPSA) Descriptors ..........75
  3.5.2 Comparative Molecular Field Analysis ..................................76
  3.5.3 Radial Distribution Functions ...........................................77
3.6 Application: Clustering of Crystal Packings ..................................79
3.7 Open-Source Implementations .................................................85
References .............................................................................85

3.1 INTRODUCTION

Three-dimensional (3D) molecular representation is at the heart of modern chemistry. The past decades have taught us that pure graph-oriented representations are typically not enough to understand the interactions of molecules with their environments. The 3D molecular geometry has a strong effect on molecular binding, as clearly seen in ligand–protein interactions and packing in crystal structures.

Understanding molecular properties requires us to understand the geometrical features of the molecule. For example, the molecular geometries of a molecule and its surroundings determine the proximity of functional groups and, therefore, why certain
molecules show strong binding affinity, due to, for example, salt and hydrogen bridges and hydrophobic interactions. Only a brief reminder is needed here that the geometry is not static and that binding affinity often involves induced fit. Visual exploration of geometries is well established in various fields of chemoinformatics, and free tools are abundant. Jmol [1] and PyMol [2] are the best-known open-source applications in this area.

Dealing with 3D geometries in computation, however, is more complex (Figure 3.1). A program does not have the visual interpretation of depth or orientation. In order to have an analysis tool to understand these patterns, the patterns need to be expressed numerically. Depth can be represented as a Euclidean distance, which, depending on the application, might be a relative distance or distance ratio. Orientation is even more complex and it involves a coordination reference to which the orientation can be measured, something that can be easily done visually. This brings us to the topic of this chapter: how to represent 3D molecular geometries such that they are useful for analysis and computation.

The molecular structure is, ultimately, governed by the quantum mechanics of the electrons that are organized in atomic and molecular orbitals. This quantum molecular structure defines all molecular properties, including the geometry and chemical reactivity. However, quantum mechanics is for many supramolecular systems too computation intensive, and simpler representations are needed to deal with the fast molecular space we nowadays work with. This simpler 3D representation typically involves an atom-and-bond representation and combines a chemical graph with the geometry information. Instead of the many electrons involved in the molecule, it focuses only on the nuclei and their coordinates. Electronic effects on the geometry are implicitly captured by the coordinates, but can be complemented with atom-type information, which typically includes hybridization information. This is the basic model behind the force field approaches.
Other applications, however, need a different representation. The above-sketched representation still increases in size with the number of atoms and bonds. However, numerical analyses in quantitative structure–activity relationship (QSAR) studies, which correlate geometrical features with binding affinity, often require a fixed-length representation that is independent of the number of atoms and bonds.

This chapter discusses the representation of molecular geometry in various coordinate systems, how to interchange those representations, and how fixed-length, numerical representations may be derived from them.

3.2 COORDINATE SYSTEMS

Three atomic coordinate systems are commonly used: Cartesian coordinates, internal coordinates, and notional coordinates. The last is specific for crystallography data and describes both the molecular geometry as well as the crystal lattice. Internal coordinates rely on and use the chemical graph and therefore aim at single, connected molecules. Cartesian coordinates are the most versatile and are typically used for disconnected 3D structure.

3.2.1 CARTESIAN COORDINATES

Cartesian coordinates describe the atomic coordinates relative to the origin. The X, Y, and Z axes are orthogonal and Euclidean distances can be used to measure distances between atoms. Orientation and placement with respect to the origin is arbitrary.

The Cartesian coordinates for ethanol shown in Figure 3.2 are as follows:

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>O</td>
<td>1.94459</td>
<td>1.33711</td>
</tr>
<tr>
<td>C</td>
<td>1.52300</td>
<td>0.00000</td>
</tr>
<tr>
<td>C</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>H</td>
<td>1.93156</td>
<td>−0.49598</td>
</tr>
<tr>
<td>H</td>
<td>1.93156</td>
<td>−0.49598</td>
</tr>
<tr>
<td>H</td>
<td>−0.35196</td>
<td>1.05588</td>
</tr>
<tr>
<td>H</td>
<td>−0.35196</td>
<td>−0.52794</td>
</tr>
<tr>
<td>H</td>
<td>−40.35196</td>
<td>−0.52794</td>
</tr>
<tr>
<td>H</td>
<td>1.18187</td>
<td>1.88994</td>
</tr>
</tbody>
</table>

Distances, angles, and torsions are easily calculated from Cartesian coordinates, as well as many other derived properties, such as molecular volume, total polar surface area, and so on (see Section 3.5.1). For example, the molecular center-of-mass may be placed on the origin, so that molecules are located in the same location. Algorithm 3.1 describes the algorithm to calculate a molecule’s center-of-mass. Centering the molecule around the origin is then done by subtracting the coordinates of the center-of-mass from the atomic coordinates.

ALGORITHM 3.1 ALGORITHM TO CALCULATE THE MOLECULAR CENTER-OF-MASS

```
sum.x = 0
sum.y = 0
sum.z = 0
```
FIGURE 3.2 3D model of ethanol labeled by position in the input file.

```
total.weight=0
iterate over all atoms {
    sum.x = sum.x + atom.x * atom.weight
    sum.y = sum.y + atom.y * atom.weight
    sum.z = sum.z + atom.z * atom.weight
    total.weight = total.weight + atom.weight
}
com.x = sum.x / total.weight
com.y = sum.y / total.weight
com.z = sum.z / total.weight
```

However, although Cartesian coordinates are universal, they are not always the best choice with respect to computation times or algorithm simplicity. For geometry optimization calculations, internal coordinates are more suitable, requiring less computation to reach the same results. Tomczak reports an approximately fourfold speed using internal coordinates over Cartesian coordinates [3].

### 3.2.2 Internal Coordinates

Internal coordinates describe the atomic coordinates in an internal frame, that is, without an external reference. They describe the molecular geometry in terms of distances between atoms and angles and torsions between bonds. This closely overlaps with force field approaches where the molecular energy is expressed in terms of bond length, angles, and torsions, defining a well-structured search space for geometrical optimization. Many molecular dynamics and quantum mechanics algorithms take advantage of this representation.

The internal coordinates for ethanol shown in Figure 3.2 are given below. The atomic numbering is the same as for the list of Cartesian coordinates and is shown in Figure 3.2 too.

These coordinates are interpreted as follows. The first distance given (1.4020 Å) is between atom 2 (carbon) and atom 1 (oxygen), while the second distance (1.5230 Å) is
FIGURE 3.3 The torsion angle for atoms 4, 2, 1, and 3 in ethanol (see Figure 3.2) is defined as the angle between a vector through atoms 4 and 2 and the vector through atoms 1 and 3, as measured in a plane perpendicular to the vector through atoms 2 and 1. (Note that atom 1 is depicted behind atom 2.) The lines between atoms are not bonds; in fact, atom 3 is bonded to atom 2 and not to atom 1. However, the vector between atom 4 and atom 2 does coincide with an actual bond.

the bond length of the carbon–carbon bond. The first angle given (107.50°) is between the bonds between the third and second atoms and the second and first atoms. The first torsion angle (239.34°) is the angle between the two lines, one between atom 4 and atom 2 and the other between atom 1 and atom 3, as measured in a plane perpendicular to the bond between atom 1 and atom 2 (Figure 3.3). (Note that atom 1 is located behind atom 2 in this figure.) These lines do not necessarily have to coincide with bonds.

3.2.3 FRACTIONAL COORDINATES

Fractional coordinates describe the positions of the atoms as fractions of the axes of the crystal’s unit cell, which is described by its crystallographic axes A, B, and C. There are two common ways to describe these three axes themselves: as a vector in Cartesian space with nine values, or with six values listing the axes’ lengths and the angles between the axes, sometimes referred to as the notional axes. Figure 3.4 shows the unit cell of the cubic unit cell of sodium chloride. The unit cell axes can be described as in notional axes 5.6, 5.6, 5.6 Å and 90°, 90°, 90°, describing the axis lengths and the angles between them, respectively.

Alternatively, the axes can be described as vectors in Euclidean space. This leaves a choice of how to rotate the unit cell in Euclidean space. If we fix the A axis on the x axis and the B axis in the XY plane, then rotation in the Euclidean space is fixed. Using this convention, the unit cell axis vectors for the sodium chloride example are A = 5.6, 0, 0, B = 0, 5.6, 0, and C = 0, 0, 5.6. If angles deviate with 90°, then only the A axis will be parallel to an Euclidean axis.

The coordinates of atoms in the unit cell are expressed as fractions of the axes A, B, and C. The fractional coordinates of the four sodium atoms in the shown unit cell are 0, 0, 0.5, 0.5, 0, 0, 0.5, 0.5, and 0.5, 0, 0.5. The chloride ions are located at 0.5, 0, 0, 0.5, 0.5, 0, 0, 0.5, and 0.5, 0.5, 0.5.
FIGURE 3.4 Unit cell of sodium chloride with the three unit cell axes starting from the origin in the lower left corner of the cube. The notional coordinates of this unit cell are defined by the \( A, B, \) and \( C \) axis lengths (all 5.6 \( \text{Å} \)) and the three angles \( \alpha, \beta, \) and \( \gamma \) (all 90) between \( B \) and \( C \), \( A \) and \( C \), and \( A \) and \( B \), respectively.

3.2.4 TWO-DIMENSIONAL CHEMICAL DIAGRAMS

The fourth coordinate system is very common in chemistry: two-dimensional (2D) chemical diagrams. These diagrams are aimed at graphical visualization of the connection table and typically focus on depiction of atom and bond properties, such as isotope and charge details for atoms, and bond properties like bond order, delocalization, and stereochemistry. This 2D coordinate space is outside the scope of this chapter. It is mentioned here, however, because 2D diagrams are often the input in algorithms that create 3D molecular structures.

These algorithms create 3D Cartesian coordinates from the information presented in 2D molecular representations. Primarily, this information includes the connection table, and atom- and bond-type information. However, to properly reflect stereochemistry features presented in the 2D diagrams, the algorithm has to resolve such information often from wedge bond representations, and 2D coordinates for cis/trans isomorphism. Additionally, coordination generation for ring systems can use a template library that may or may not contain information on the layout of the attachment points to assemble the geometries of ring and nonring systems. The general concept is given in Algorithm 3.2.

ALGORITHM 3.2 ALGORITHM TO CREATE 3D GEOMETRIES FROM 2D DIAGRAMS

extract connection table
derive atom parities from wedge bond and 2D coordinates information
derive cis/trans isomorphism from 2D coordinates
isolate ring systems, and look up 3D coordinates from a template library
apply common geometries for non-ring substructures taking into account stereochemistry

3.3 INTERCONVERTING COORDINATE SYSTEMS

Interconversion between the three coordinate systems is important, because algorithms can perform differently depending on the chosen system, as was discussed earlier. Algorithms to interconvert coordinate systems are abundant, but may differ in detail between implementations. This section discusses two algorithms: conversion of internal coordinates into Cartesian coordinates and conversion of fractional coordinates into Cartesian coordinates.

3.3.1 INTERNAL COORDINATES INTO CARTESIAN COORDINATES

Converting internal coordinates into Cartesian coordinates is fairly straightforward: each next atom is placed into Euclidean space to conform the internal coordinates converted so far. The algorithm has two degrees of freedom: (1) in which Cartesian coordinate the first atom is placed and (2) in which plane the first two bonds are located. The algorithm description given in Algorithm 3.3 puts the first atom at the origin of the coordinate system, the first bond along the x axis, and the second bond in the xy plane.

ALGORITHM 3.3 ALGORITHM TO CONVERT INTERNAL COORDINATES INTO CARTESIAN COORDINATES. ATOM NUMBERING FOLLOWS THOSE FROM TABLE 3.1

let the first line define:
the first atom
then:
put the first atom at \{0,0,0\}
let the second line define:
a new first atom, and a second atom
a distance to a second atom
then:
\[ d = \text{distance (first atom, second atom)} \]
put the second atom on the X axis at \{d,0,0\}
let the third line define:
new first and second atoms, and a third atom
a distance to a second atom,
an angle between the first, second and third atom on this line
then:
\[ d = \text{distance (first atom, second atom)} \]
\[ \alpha = \text{angle (first atom, second atom, third atom)} \]
put the third atom in the XY plane, such that:
the distance to the second atom is \(d\), and
### Table 3.1

<table>
<thead>
<tr>
<th>Number</th>
<th>Element</th>
<th>Distance</th>
<th>Angle</th>
<th>Torsion</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>O</td>
<td>1.4020</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>C</td>
<td>1.5230</td>
<td>107.50</td>
<td>239.34</td>
</tr>
<tr>
<td>3</td>
<td>C</td>
<td>1.1130</td>
<td>108.34</td>
<td>120.66</td>
</tr>
<tr>
<td>4</td>
<td>H</td>
<td>1.1130</td>
<td>108.43</td>
<td>240.00</td>
</tr>
<tr>
<td>5</td>
<td>H</td>
<td>1.1130</td>
<td>108.43</td>
<td>120.00</td>
</tr>
<tr>
<td>6</td>
<td>H</td>
<td>1.1130</td>
<td>108.43</td>
<td>0.00</td>
</tr>
<tr>
<td>7</td>
<td>H</td>
<td>1.1130</td>
<td>108.44</td>
<td>3 0.00</td>
</tr>
<tr>
<td>8</td>
<td>H</td>
<td>0.9420</td>
<td>108.44</td>
<td>3 0.00</td>
</tr>
</tbody>
</table>

The angle between first, second and third atom is $\alpha$.

Let the fourth and all later lines define:
- New first, second, and third atoms, and a fourth atom,
- A distance between the first and second atom,
- The angle between the first, second and third atom, and
- The torsion between the first, second, third, and fourth atom.

Then:
- $d =$ distance (first atom, second atom)
- $\alpha =$ angle (first atom, second atom, third atom)
- $t =$ torsion (first atom, second atom, third atom, fourth atom)

Put the first in euclidean space, such that:
- The distance to the second atom is $d$,
- The angle between first, second and third atom is $\alpha$,
- The torsion is defined by $t$.

### 3.3.2 Fractional Coordinates into Cartesian Coordinates

Converting fractional coordinates into Cartesian coordinates can in the simplest way be performed as a matrix operation:

$$
\begin{pmatrix}
    x' \\
    y' \\
    z'
\end{pmatrix} =
\begin{pmatrix}
    a & b \cdot \cos \gamma & c \cdot \cos \beta \\
    0 & b \cdot \sin \gamma & \frac{c \cdot \cos \alpha - c \cdot \cos \beta \cdot \cos \gamma}{\sin \gamma} \\
    0 & 0 & \frac{V}{a \cdot b \cdot \sin \gamma}
\end{pmatrix}
\begin{pmatrix}
    x \\
    y \\
    z
\end{pmatrix},
$$

(3.1)
with $a$, $b$, and $c$ being the length of the crystallographic axes $A$, $B$, and $C$, and $\alpha$, $\beta$, and $\gamma$ the angles between $B$ and $C$, $A$ and $C$, and $A$ and $B$, respectively, and $V$ the volume of the unit cell, defined as

$$V = abc\sqrt{1 - \cos^2 \alpha - \cos^2 \beta - \cos^2 \gamma + 2\cos \alpha \cos \beta \cos \gamma}.$$  (3.2)

### 3.4 COMPARING GEOMETRIES

There are many applications where comparing the 3D structure of molecules is interesting; molecular docking is likely the most common one. In such studies the molecule is oriented in the active site of an enzyme or receptor (Figure 3.5). CoMFA studies binding affinities and assumes that molecules are overlayed and oriented in a similar chemical direction, reflecting similar binding modes with protein (see Section 3.5.2).

Comparing two or more molecular 3D geometries is generally not directly possible: the geometries do not share a common reference origin, and they may not be oriented in the same direction. The center-of-masses of the molecules may be far apart, and the structures can be differently aligned. The first can be addressed by putting the center-of-mass of each molecule in the origin of the coordinate system.

It does not, however, orient the molecule in any particular way. While the center-of-mass is in the origin, the molecular conformer can still be oriented in any direction. To address this molecule, one may apply principal component analysis (PCA) and orient the molecule such that the first three latent variables are oriented along the $X$, $Y$, and $Z$ axes as described in Algorithm 3.4.

**ALGORITHM 3.4 ALGORITHM TO ALIGN CHEMICAL STRUCTURES BASED ON ANISOMORPHISM**

for each molecule:
- calculate the three PCs from the 3D coordinates
for each atom:
  - the new x coordinate is the score on PC1
the new y coordinate is the score on PC2
the new z coordinate is the score on PC3
overlay the molecules in the new coordinate space

The above-sketched algorithm does not take into account structural similarity between molecules, but only looks at the anisomorphism of the structures. That is, the variance in atomic coordinates is used to create new coordinates. Practically, this means that each molecule is reoriented such that the direction in which the molecule is longest, and thus has the highest variance, is aligned with the first principal component (PC1).

Instead, it is often desirable to orient the molecules based on the maximal common substructure (MCSS). For example, alignment of a series of steroids is expected to overlay the sterane skeletons. The alignment must not be disturbed by large side chains that change the overall anisomorphism of the geometry: the variance of the 3D coordinates would change and the alignment too. Using the MCSS, the alignment of the two molecules becomes more, in agreement with what one would expect. Most chemoinformatics toolkits have the means to either find the maximum common substructure or to identify a user-defined substructure using a query language like molecular query language (MQL) [4] or SMARTS.

After having identified the MCSS of the molecular geometries, the full structures can be rotated in the coordinates space to minimize the root mean square deviation (RMSD) of the coordinates of the shared substructure (Algorithm 3.5).

**ALGORITHM 3.5 ALGORITHM TO ALIGN 3D MOLECULAR STRUCTURES BASED ON THE COMMON SUBSTRUCTURE**

find the maximal common substructure (MCSS)
find a rotation that minimizes the RMSD of the atomic coordinates of the MCSS

### 3.5 FIXED-LENGTH REPRESENTATIONS

One disadvantage of representation in any of the three discussed coordinates systems is that the size depends on the number of atoms. Many chemometrical modeling methods, however, require a numerical and fixed-length vector representation of the molecular structure [5,6]. The above representations do not fulfill this requirement, and hence derived descriptors have been and still are being developed to bridge the gap between those representations and the mathematical modeling methods. These descriptors allow statistical modeling and analysis with, for example, classical methods like PCA, partial least squares (PLS) and neural networks (NNs) and classification methods like linear discriminant analysis (LDA). Only very few methods, such as classification and regression trees (CART), do not require a numerical representation. Distance-based clustering, for example, can work directly with an MCSS-based distance matrix in which two molecules that have a large substructure in common have a smaller distance and are considered more alike.

The *Handbook of Molecular Descriptors* published in 2000 [7] gives a broad overview of known molecular descriptors. Depending on the information content,
Three-Dimensional (3D) Molecular Representations

descriptors are usually classified as 0D, 1D, 2D, and 3D descriptors. The last category, 3D, takes into account the 3D geometry of the molecule. Recently, a fifth category has been proposed: 4D descriptors for which several different but related definitions have been given. Todeschini defines the 4th dimension to describe the interaction field of the molecule [7], while others reserve this dimension to describe its conformations [8]. The latter takes into account the flexibility of molecules, where coordinate systems only treat the molecules as rigid bodies.

3.5.1 MOLECULAR DESCRIPTORS

To illustrate how molecular descriptors convert the variable-length 3D molecular geometries into a fixed-length representation, two descriptor algorithms are described in this section. It is important to realize that the representation not only needs to be of fixed length, but also needs to be orientation independent. That is, the descriptor value must not change when the molecular geometry is rotated in coordinate space. Consequently, these descriptors are suitable for comparing molecular geometries without the need for alignment. This requirement is also the reason why these molecular descriptors typically do not describe angular features of the molecule, other than collapsed onto a single value.

3.5.1.1 The Length-over-Breadth Descriptor

The length-over-breadth descriptor describes the anisomorphism of the molecule, but uses their ratio to collapse the length and breadth features into a single number (Algorithm 3.6).

ALGORITHM 3.6 ALGORITHM FOR THE LENGTH-OVER-BREADTH DESCRIPTOR

calculate the geometrical dimensions of the molecule
determine the length and breadth
calculate the ratio length over breadth

Calculation of the molecular length and breadth is quite similar to the use of PCA alignment (see Section 3.4), which rotates the molecule such that the longest molecular axis is aligned with the PC1. The length is then defined as the difference between the maximum and minimum coordinates on this axis (PC1); the breadth would be the difference on the second axis (PC2). This calculation can include the van der Waals radii of the atoms to reflect the size of the molecule as a function of its molecular surface. However, to simplify calculation, not all possible rotations are taken into account. For example, implementations may only rotate the molecular structure around a single coordinate system axis.

3.5.1.2 Charged Partial Surface Area (CPSA) Descriptors

The molecular surface area and the molecular volume are other methods to reduce the 3D geometry to a fixed-length representation. Neither of the two describe the internal geometry of the molecules, but are aimed at describing the molecular features
governing intermolecular interactions. Both surface area descriptors and the molecular formula require the calculation of the molecular surface area (Algorithm 3.7). Depending on the actual surface of interest, different atomic contributions to the total surface can be used. For example, the van der Waals surface will use a smaller sphere around each atom than the solvent accessible surface.

**ALGORITHM 3.7 ALGORITHM TO DETERMINE THE 3D MOLECULAR SURFACE**

For each atom:
- use tessellation to define a sphere of points around the 3D coordinate of the atom
- remove all sphere points which are buried inside the spheres of neighboring atoms
- the molecular surface is defined by the remaining points

The CPSA descriptor uses the atomic contributions to this surface, combined with the partial atomic charges, as the starting point to come to 25 descriptor values [9]. A full description of all values is outside the scope of this chapter and is well described in the original paper, but it is illustrative to describe the first six: partial positive surface area (PPSA), total charge weighted PPSA, atomic charge weighted PPSA, and their negative charge equivalents, namely partial negative surface area (PNSA), total charge weighted PNSA, and atomic charge weighted PNSA.

These six descriptors provide a numerical vector representation of the geometrical features of the molecule, but at the same time introduce electronic features that affect intermolecular interactions. The PPSA and PNSA use the aforementioned algorithm to determine the atomic contributions to the molecular surface area. While the PPSA only takes into account atomic contributions of atoms with a positive partial charge ($\sum (SA^+_i)$), the PNSA only takes into account contributions from the negatively charged atoms ($\sum (SA^-_i)$). This introduces a nice area where implementations of the general algorithm will differ in results, depending on which algorithm has been used to calculate the partial charges. For example, the original paper used an empirical method, whereas the CDK implementation of this descriptor uses Gasteiger charges. The other four descriptors are also derived from the atomic contributions, but are weighted sum of positive ($Q^+_i$) or negative ($Q^-_i$) partial charges. An overview of the six descriptor values of the CPSA descriptor is given in Table 3.2.

### 3.5.2 COMPARATIVE MOLECULAR FIELD ANALYSIS

That an insight into the 3D interaction of a ligand with protein cavities is important in the modeling of biochemical endpoints, such as binding affinity, became apparent and computationally feasible in the last decade. Comparative molecular field analysis (CoMFA) is the primary example of this concept [10]. The CoMFA method studies molecule–environment interaction by putting the molecules in an equidistant grid of points in 3D space. At each point, the interaction energy is calculated using a hypothetical probe, for example, using the Lennard–Jones potential function and the Coulomb potential energy function. It is important to note that because the molecules
### TABLE 3.2
First Six of the 25 CPSA Descriptors, with the Formulas to Calculate them

<table>
<thead>
<tr>
<th>Descriptor</th>
<th>Label</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Partial positive surface area</td>
<td>PPSA</td>
<td>( \sum (S_{A_i}^+ + Q_{T_i}^+) )</td>
</tr>
<tr>
<td>Partial negative surface area</td>
<td>PNSA</td>
<td>( \sum (S_{A_i}^- - Q_{T_i}^-) )</td>
</tr>
<tr>
<td>Total charge weighted PPSA</td>
<td>PPSA-2</td>
<td>( \frac{PPSA}{Q^+_T} )</td>
</tr>
<tr>
<td>Total charge weighted PNSA</td>
<td>PNSA-2</td>
<td>( \frac{PNSA}{Q^-_T} )</td>
</tr>
<tr>
<td>Atomic charge weighted PPSA</td>
<td>PPSA-3</td>
<td>( \sum (S_{A_i}^+ \cot Q_{T_i}^+) )</td>
</tr>
<tr>
<td>Atomic charge weighted PNSA</td>
<td>PNSA-3</td>
<td>( \sum (S_{A_i}^- \cot Q_{T_i}^-) )</td>
</tr>
</tbody>
</table>

**Note:** \( S_{A_i}^+ \) and \( S_{A_i}^- \) are the atomic contributions to the surface area for the atoms with positive and negative partial charge, respectively. \( Q_{T_i}^+ \) and \( Q_{T_i}^- \) are the sum of positive partial charges \( Q_i^+ \) and the sum of negative partial charges \( Q_i^- \), respectively.

are aligned, the interaction similarities of the ligands can be compared by calculating the difference in the interaction energies of the matching grid points for all molecules. Afterwards, PLS is used to correlate the matrix expansion of the grid with the activity, such as ligand–target binding affinities [11,12].

CoMFA requires, however, geometrical alignment of the molecules, as discussed earlier, and only considers one conformation for each molecule, which is only a simplification of reality. Therefore, the focus has moved on to descriptors that are independent of the orientation of the molecules in its reference frame, and possibly even include information of multiple conformations. This was already acknowledged in 1997 by Hopfinger, who made a scheme which incorporated some ideas from CoMFA but which was alignment independent and took into account multiple conformations [13].

### 3.5.3 Radial Distribution Functions

Another common approach to remove alignment effects is to use the radial distribution function (RDF). This kind of function, as the name says, describes the distribution of certain features as a function of the distance to the central point. RDFs are particularly interesting when distance-related interactions need to be captured. The basic RDF describes the occurrence of a chemical feature at a certain distance, for example, the presence of an atom. For example, Aires-de-Sousa et al. have used five RDFs to describe the environment of protons to predict proton NMR shifts [14] and for the simulation of infrared spectra [15,16].

Figure 3.6 shows a basic spike-like RDF and the effect of smoothing with Gaussian function. This smoothing is particularly useful when a (dis)similarity between two RDFs is calculated: small displacements of the atom positions captured in the RDF will lead to large changes in the similarity between the two functions. However, when a Gaussian smoothing is used, changes in the similarity are less abrupt. Other
approaches can be used too, and one such is used in the application described in the next section.

The algorithm for calculating an RDF for an atom in a molecule is fairly simple and is described in Algorithm 3.8. While the RDF itself is an analogous function, particularly when Gaussian smoothing is used, the function is typically digitized, for example, using binning. Given a central atom, the RDF of atoms around that atom is calculated by iterating over all atoms in the molecule, and determine where it contributes to the RDF. The amount it contributes is defined by a weighing scheme. In its simplest form, the contribution is 1 for each atom present (in black in Figure 3.6). If a Gaussian smoothing is used, then the neighboring bins are increased too, effectively convoluting the spike with a Gaussian function of selectable width (in light gray and dark gray in Figure 3.6).

An interesting feature of RDFs is that they can be tuned to particular applications. The aforementioned application in NMR shift prediction uses five such customized RDFs. The contribution an atom gives to the RDF can be weighted in various ways.
Three-Dimensional (3D) Molecular Representations

FIGURE 3.7 The coulombic interaction weighted (solid line) and non-weighted RDF (dashed line) for the oxygen atom in ethanol shown in Figure 3.2, showing the effect of the weighing scheme.

Commonly, the contribution is weighted by the distance to the central atom: the farther away from the center, the smaller the contribution. This compensates for the fact that at larger distances, each bin describes an increasing amount of spherical space.

Additionally, the contribution can be weighted by the properties of the atom that affect the contribution. For example, the coulombic interaction can be used, which represents the electronic interaction between the point charges of the atoms (Figure 3.7) and which originates from the desire to describe electronic features of the molecule. The application described in the next section of this chapter applies this approach too, where it uses RDFs to describe complete organic crystal structures.

Importantly, it should be clear that the algorithm allows for any weighting function, offering interesting flexibility in describing molecular geometries.

3.6 APPLICATION: CLUSTERING OF CRYSTAL PACKINGS

Comparing crystal structures is important in both classification and clustering problems. Classification is important for the understanding of the relation between physical properties and the underlying structure of materials. The specific packing of molecules in a crystal directly influences the physical properties of compounds. As an example, in crystal engineering, crystal packings are classified according to intermolecular interactions [17–21]. A second application of the similarity measure is in the clustering stage of ab initio crystal structure prediction [22,23]. In this process, hundreds or thousands of different hypothetical crystal packings for the same molecule, called polymorphs, are generated. They need to be clustered to arrive at representative subsets for which analysis and geometry optimization are feasible.

Two things are needed for clustering and classification of crystal structures: a properly defined descriptor and a similarity function applied to this descriptor. A few requirements for both the descriptor of crystal structures and the similarity function are described in the literature [24–26]: the most obvious requirement for a descriptor–similarity combination is that more dissimilar crystal structures result in larger dissimilarity values. Although this seems trivial, several well-known descriptors
do not generally satisfy this requirement [24–27]. Many descriptors require a choice of origin, or some other setting. Among such descriptors is the combination of unit cell parameters and fractional coordinates discussed earlier in this chapter. Caused by this choice of origin, a descriptor based on reduced unit cell parameters can vary significantly with only minor lattice distortions [28,29]. Although it is in some cases possible to adapt the similarity function to deal with such instabilities, this issue can better be addressed by using RDFs [30]. Using this descriptor a dissimilarity measure that expresses the differences between two crystal structures can be defined. The resulting dissimilarity value can then be used to cluster or classify the crystal structures by grouping together structures that have a low dissimilarity between them.

Crystal structures can be uniquely represented by an RDF describing the distribution of neighboring atoms around a central atom. Each neighboring atom gives rise to a peak in the function. RDFs are independent of cell choice and can be physically interpreted. In the application presented here, the RDF is adapted to include more specific information about the atoms. To do so, the RDF is weighted by the electrostatic interactions. To indicate the inclusion of electrostatic information in the descriptor, we will refer to this as the electronic RDF, or R_eDF. The reason for including electrostatics is the assumption that these play a major role in crystal packing [18,31,32]. By including partial atomic charges, the R_eDF focuses on atom groups with large partial charges, in particular functional groups, and differentiates between attractive interactions between oppositely charged atoms and repulsive interactions.

An atomic R_eDF describes the distribution of coulombic interactions of one atom with surrounding atoms; the R_eDF for the crystal structure is obtained by summing all atomic R_eDFs of all N atoms in the asymmetric unit:

$$\text{R}_e\text{DF}(r) = \sum_{i=1}^{N} \sum_{j=1}^{M} \frac{q_i q_j}{N \cdot r_{i,j}^2} \delta(r - r_{i,j}),$$

where M is the number of neighboring atoms within a radius r, q_i and q_j are partial atomic charges of the atoms i and j, and δ places the electrostatic interaction at the right distance by its definition δ(x) = 1 if x = 0 and δ(x) = 0 if x ≠ 0. Alternatively, the δ(x) can reflect Gaussian smoothing. The function is scaled for the number of atoms in the asymmetric unit, N.

Figure 3.8 shows the R_eDF for an artificial crystal with two atoms in the unit cell, a positively and a negatively charged one (a = 7.97, b = 10.26, c = 18.77, and α = β = γ = 90°). The first negative peak is the interaction between the two atoms at exactly the bonding distance. The other negative peaks are also peaks between two oppositely charged atoms. The overall decrease in intensities is caused by the 1/r term in the R_eDF equation. The first positive peak is related to the translation along the a axis, that is, ± a, and the second peak to the translation along the b axis. The third peak is the translation in the direction a ± b. For this orthogonal structure, there are twice as many contributions to this peak as for the first two positive peaks, resulting in the higher intensity.

The R_eDFs of four experimental cephalosporin crystal structures are shown in Figures 3.9 and 3.10. They show a few distinct high-intensity peaks and many smaller
FIGURE 3.8 Example $R_e$DF of an artificial crystal structure with a positively and a negatively charged atom ($a = 8.0$, $b = 10.3$, $c = 18.8$, and $\alpha = \beta = \gamma = 90^\circ$). Positive peaks are caused by the interaction of atoms with both positive and both negative charges. Consequently, they cause positive peaks at the distances matching the translational symmetry of the crystal. This explains, for example, the positive peaks at 8.0, 10.3, and 18.8 Å.
FIGURE 3.9 Example $R_e$DFs of three cephalosporin compounds: (a) A9, (b) A10 from the same class A.

peaks. The locations of these peaks are specific for the crystal packing: Figure 3.9a and b shows the $R_e$DFs of two cephalosporin structures from the same class, while Figure 3.9c shows the $R_e$DF for a different packing. Figure 3.10a shows the function for a simulated estrone crystal structure; a similar pattern can be observed. Figure 3.10b shows the effect of cutting away peaks with intensities lower than some
threshold. It was found that the cutoff value must be around 20% of the highest peak. Cutting away the smaller peaks emphasizes the major features of the $R_e$DF and leads to better discrimination.

Because of the nature of the $R_e$DF, one can expect positive contributions at those distances that match the translational symmetry in the crystal. This causes the positive
FIGURE 3.10  Example $R_e$DF of one of the simulated estrone structures shown in (a), and the effect of cutting away of peaks below 20\% of the intensity of the highest peak in (b).

peaks at 8.0, 10.3, and 18.8 Å. However, since such contributions can be canceled out by other, negative contributions, they do not always show up in the $R_e$DF. Moreover, peaks not related to translational symmetry are particularly interesting, because they provide information additional to symmetry in the crystal.
TABLE 3.3
Open-Source Implementations of Algorithms Discussed in the Chapter

<table>
<thead>
<tr>
<th>Algorithm Number</th>
<th>Algorithm</th>
<th>Libraries</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Calculate the center-of-mass</td>
<td>CDK</td>
<td></td>
</tr>
<tr>
<td>3.2</td>
<td>Create 3D geometries from 2D diagrams</td>
<td>CDK</td>
<td></td>
</tr>
<tr>
<td>3.3</td>
<td>Convert internal to Cartesian coordinates</td>
<td>CDK OpenBabel</td>
<td></td>
</tr>
<tr>
<td>3.4</td>
<td>Align chemical structures based on anisomorphism</td>
<td>R</td>
<td></td>
</tr>
<tr>
<td>3.5</td>
<td>Align 3D molecular structures based on the common substructure</td>
<td>CDK R</td>
<td>MCSS search For algorithm 3.4</td>
</tr>
<tr>
<td>3.6</td>
<td>Calculate the length-over-breadth ratio</td>
<td>CDK</td>
<td></td>
</tr>
<tr>
<td>3.7</td>
<td>Calculate the 3D molecular surface</td>
<td>CDK</td>
<td>NumericSurface.class</td>
</tr>
<tr>
<td>3.8</td>
<td>Calculate an atomic RDF</td>
<td>CDK</td>
<td>RDFCalculator.class</td>
</tr>
</tbody>
</table>

Using this description, dissimilarities between crystal structures are represented by the difference between the two corresponding $R_e$ DFs. For this, a weighted cross correlation (WCC) is used [19], which is applied to the high-intensity peaks of the $R_e$ DF. Using this approach, both experimental and simulated crystal structures have been clustered and classified successfully [30].

3.7 OPEN-SOURCE IMPLEMENTATIONS

This chapter has presented a variety of basic algorithms involved in the representation of 3D molecular geometries. Because support for these geometries is so fundamental to chemoinformatics, it will not be difficult to find implementations in open-source software for the algorithms described in this chapter. Visualization of 3D geometries can be done in Jmol (http://www.jmol.org, [1]) and PyMOL (http://www.pymol.org). Converting different coordinate systems is also supported by various open-source toolkits, including the CDK (http://cdk.sourceforge.net, [33,34]) and OpenBabel (http://openbabel.org). Table 3.3 gives a more detailed overview.

REFERENCES


