My experience

   - support, user education, software engineering, meetings, application science, web apps, QA, databases, methodology research, etc.

2) OpenEye (2002 - 2007): Director/VP of Support, Sr. Software Engineer
   - support, management, software engineering, QA, web apps, methodology research, etc.

   - software engineering, management, support, computational methodology, bioassay screening informatics, biomedical informatics research, etc.
Describe direct observations from experiences in cheminformatics over last 22 years, relevant today to understand and navigate complex landscape of cheminformatics software roles and choices. Avoiding excessive idle reminiscing, include some of the colorful personalities and curious events. Suggest some lessons learned and trends observed, in the opinion of the author.
Outline

Case studies included:

1) Daylight
2) OpenEye
3) Symyx a.k.a. MDL
4) Accelrys
5) OpenBabel

(Chosen mostly based on my familiarity.)

Some interesting others also mentioned briefly.
Perspectives on scientific software

- Developer, programmer
- Computational/informatics scientist, scholar
- Support, educator, maintainer
- Software as scientific publishing
- Open source collaboration
- Consumer: toolkit user (programmer)
- Consumer: app user, scientist
- Licensing, intellectual property, legal
- Business (commercial or non-commercial)
Case Study #1

Daylight Chemical Information Systems, Inc.

http://www.daylight.com
Daylight Chemical Information Systems, Inc.

- Founded 1987 by Dave Weininger, Art Weininger, Yosi Taitz, in Claremont, CA
- Ancestry: Pomona College MedChem program (Corwin Hansch, Al Leo)
- Innovations: SMILES, SMARTS, SMIRKS, fingerprints, rigorous syntax/grammar/semantics
- Products: ClogP, Thor, Merlin, C toolkits, Oracle Chemistry Cartridge
- Fortran → C ~1990, oop-ish API (Scofields).
- DEC-VAX/VMS → Unix → Linux, Windows
Daylight “MUG” meetings known for scientific impact
Daylight Chemical Information Systems, Inc.

- Success via appeal to examplars who became advocates at various pharmas.
- Published, supported APIs and open transparent system appealed to “hackers, nurds and geeks”.
- From research computing to enterprise (IT).
- Many other software developers learned and borrowed from Daylight.
- Focused on software, to enable science.
- Max ~12 employees, $5M annual revenue.
Some idle reminiscing...
Daylight Chemical Information Systems, Inc.

● Scientific mission: To effectively store, retrieve, process all existing chemical information.
● This mission often guided what to do, and also what to not do.
● Often that conflicted with profit motives.
● Avoided: consulting, macromolecules, biology, Windows, investors, sales, marketing

*If I have seen further it is only by standing on the shoulders of giants.* - Issac Newton

*I can't see far because there are giants on my shoulders!* - Dave Weininger
Daylight Thor, Merlin database apps ~ 1995
7.1 Creating Molecules

There are two ways to create a molecule object: "From scratch" (allocate an empty molecule), and by parsing a SMILES string:

```c
dt_alloc_mol() => molecule
    Returns a new, empty molecule.
```

```c
dt_smilin(string smiles) => molecule
    Interprets the given SMILES string and return a handle for the resulting molecule structure.
```

Efficiency Note: The Toolkit's internal representation of molecule objects is designed for efficient analysis of the molecule's properties, and for responding to queries about the molecule quickly. It is not intended to be a compact representation of the molecule, and uses many times more memory to store than a compact representation such as a SMILES string. Applications that require many thousands of molecules in memory simultaneously should use a more compact representation for those molecules that are not of immediate interest.

7.2 Constituents of a Molecule

These functions provide ways to enumerate (generate streams of) the atoms, bonds, cycles, and chiral features of molecules. Also included are two functions, `dt_bond()` and `dt_xatom()`, for accessing related constituents without the necessity of creating a stream.

```c
dt_stream(Handle ob, integer typeval) => stream
    Generate a stream of atoms, bonds or cycles -- a stream that contains all of the objects of the specified type that are part of the object.
```

Object can be a molecule, atom, bond or cycle. For example, a stream of `dt_stream(bond, TYP_ATOM)` returns the two atoms at either end of the bond; a stream of `dt_stream(cycle, TYP_BOND)` returns all the bonds that are part of the cycle.

*Note: remember, dt_stream() is polymorphic -- it applies to other objects, too. Here, we are only discussing the molecule and its constituent parts.*
Example research project using Daylight tools

GADD

GA-based Druglike Database
Evolving virtual libraries with reaction transforms

Jeremy Yang
Daylight Chemical Information Systems Inc.

ABSTRACT

GADD is a computer program which enumerates virtual libraries with a GA-like approach using the Daylight Reaction Toolkit and reaction transforms. Molecules are built from fragments defined by input SMILES, and tested for fitness (druglike character) according to several criteria. Library design is achieved through selection of the input fragments, adjusting the fitness criteria, and specifying execution parameters, such as maximum library size.

Presentation

Daylight Chemical Information Systems, Inc.
info@daylight.com

http://www.daylight.com/meetings/mug01/Yang/gadd/
Introduction

- GADD **evolves** an initial population of fragments into a virtual library of molecules.

- GADD is a contributed package, consisting of a set of tools with which a user can identify the fragments in a "seed" database and build libraries from those fragments.

- **Goals:**
  - A flexible, extensible, powerful virtual library enumeration tool.
  - Demonstrate Daylight toolkit and provide useful contrib code.
  - Reproduce some existing approaches in Daylight-compatible form.
  - ...and offer some new approaches -- maybe.

- **Daylight software used:**
  - SMILES, SMARTS, Thor, and **Reaction Toolkit**

- **Origin of GADD:** origin of life, GoForth (EuroMUG'99).

- "Druglikeness" and "new-and-diverse-ness" are conflicting goals.
Case Study #2

OpenEye Scientific Software, Inc.

http://www.eyesopen.com
Founded 1997 by Anthony Nicholls in Santa Fe, NM.

Ancestry: Honig lab, biophysics, Columbia.

Innovations: molecular shape and continuum dielectric Poisson-Boltzmann electrostatics via atom centered Gaussians, 3D speed, rigor, accuracy and validation, comprehensive APIs

Products: Rocs, Fred, OEChem, Vida

APIs: C++, Python, Java, many OS's
OpenEye apps

- Rocs shape overlay
- Fred docking result
- Brood bioisosteric fragments
- Szmap hydrophilic regions
OpenEye toolkit API

OERMSD

Array-Based OERMSD

double OERMSD(const float *refcords, const float *fitcords, unsigned int size, bool overlay=false, double *rot=0, double *trans=0);

double OERMSD(const double *refcords, const double *fitcords, unsigned int size, bool overlay=false, double *rot=0, double *trans=0);

Returns the root mean squared deviation between two sets of Cartesian coordinates. The two arrays passed in as ‘refcords’ and ‘fitcords’ should be of length size, and should contain the Cartesian coordinates of the two objects being assessed. The boolean flag ‘overlay’ indicates whether the RMSD of the two arrays in their current position is desired [false], or whether the lowest possible RMSD for the two arrays should be returned [true]. If an overlay calculation is carried out, the functions can report the rotation and translation required to give this minimum RMSD. An array of length nine should be passed to the rot argument and an array of length three should be passed as the trans argument.

Full Molecule-Based OERMSD

double OERMSD(const OE MolBase & ref, const OE MolBase & fit, bool automorph=true, bool heavyonly=true, bool overlay=false, double *rot=0, double *trans=0);

bool OERMSD(const OE MolBase & ref, const OE MolBase & fit, double *rmsdArray, bool automorph=true, bool heavyonly=true, bool overlay=false, double *rot=0, double *trans=0);

Calculates the root mean squared deviation between two molecules. This function is overloaded for comparisons of an OE MolBase ‘ref’ molecule with either OE MolBases or OE MolBase sets as fit molecule. For the OE MolBase vs OE MolBase comparison, the RMSD is the return value. For the OE MolBase vs OE MolBase set case, the RMSDs are returned in the ‘rmsdArray’ array. The ‘rmsdArray’ passed to this function should be of length fit.GetMaxConfIndex() for the OE MolBase vs OE MolBase overload. The automorph flag indicates whether automorphisms should be taken into account during the RMSD calculation. Automorphisms are the symmetry related transformations of a molecule which can result in anomalously high RMSDs if not properly treated. For instance, t-butyl-benzene has a three-fold automorphism around the t-butyl group and a two-fold automorphism around the benzene ring. The heavyOnly flag indicates whether only heavy atoms should be considered or hydrogen atoms should be considered as well.

It is strongly recommended that one should consider carefully before setting the ‘automorph’ flag to true and the ‘heavyOnly’ flag to false due to the increased computational cost. The ‘overlay’, ‘rot’, and ‘trans’ arguments are identical to those described in the preceding section.

Partial Molecule-Based OERMSD

double OERMSD(const OE MolBase & ref, const OE MolBase & fit, const OE MolBase & snatch, bool overlay=false, double *rot=0, double *trans=0);

bool OERMSD(const OE MolBase & ref, const OE MolBase & fit, const OE MolBase & snatch, bool overlay=false, double *rot=0, double *trans=0);

The partial molecule-based OERMSD function is similar to the full molecule-based function, but it allows for a “snatch” molecule to be used in the comparison. The snatch molecule is used to remove certain parts of the molecules before calculating the RMSD. This can be useful when comparing molecules that have different conformations or functional groups.
OpenEye Scientific Software, Inc.

- Focused on software and science, sometimes a difficult balance.
- High-throughput 3D virtual screening (esp. Rocs) has become standard practice (enterprise-like).
- Spectrum of functionality between 2D cheminformatics and 3D high performance computing.
- Max 34 employees (current).
$ pdb2lig.py
Usage:
pdb2lig.py [options] [<infile] [<outfile>
   --i=<INFILE>
   --outlig=<OUTLIGFILE>
   --outpro=<OUTPROFILE> ... output protein or other macromol
   --multiligfiles ... one output file per ligand
   --minatoms=<N> ... minimum atomcount cutoff for ligand [7]
   --maxatoms=<N> ... maximum atomcount cutoff for ligand [100]
   --metal ... disconnected metal ions stay with protein
   --f ... force processing of non-PDB file
   --v ... verbose
   --vv ... very verbose
   --h ... help
ROCS web app (Python, source code available)
### Rocs results:

<table>
<thead>
<tr>
<th>Name</th>
<th>Rank</th>
<th>TanimotoCombo</th>
<th>ShapeTanimoto</th>
<th>ColorTanimoto</th>
<th>FitTverskyCombo</th>
<th>RefTverskyCombo</th>
<th>RefColorTversky</th>
<th>ComboScore</th>
<th>Color</th>
</tr>
</thead>
<tbody>
<tr>
<td>35_3</td>
<td>1</td>
<td>1.93</td>
<td>0.69</td>
<td>0.34</td>
<td>1.55</td>
<td>0.69</td>
<td>1.16</td>
<td>0.41</td>
<td>0.40</td>
</tr>
<tr>
<td>4_0</td>
<td>2</td>
<td>1.93</td>
<td>0.75</td>
<td>0.28</td>
<td>1.47</td>
<td>0.60</td>
<td>1.19</td>
<td>0.35</td>
<td>0.34</td>
</tr>
<tr>
<td>34_6</td>
<td>3</td>
<td>1.91</td>
<td>0.75</td>
<td>0.26</td>
<td>1.48</td>
<td>0.56</td>
<td>1.13</td>
<td>0.33</td>
<td>0.32</td>
</tr>
<tr>
<td>55_5</td>
<td>4</td>
<td>0.99</td>
<td>0.76</td>
<td>0.24</td>
<td>1.31</td>
<td>0.43</td>
<td>1.19</td>
<td>0.34</td>
<td>0.34</td>
</tr>
<tr>
<td>5_1</td>
<td>5</td>
<td>0.98</td>
<td>0.68</td>
<td>0.31</td>
<td>1.52</td>
<td>0.63</td>
<td>1.11</td>
<td>0.37</td>
<td>0.37</td>
</tr>
<tr>
<td>32_37</td>
<td>6</td>
<td>0.97</td>
<td>0.75</td>
<td>0.22</td>
<td>1.30</td>
<td>0.44</td>
<td>1.17</td>
<td>0.31</td>
<td>0.31</td>
</tr>
<tr>
<td>1_4_9</td>
<td>7</td>
<td>0.96</td>
<td>0.59</td>
<td>0.38</td>
<td>1.56</td>
<td>0.74</td>
<td>1.11</td>
<td>0.44</td>
<td>1.01</td>
</tr>
<tr>
<td>12_21</td>
<td>8</td>
<td>0.96</td>
<td>0.60</td>
<td>0.35</td>
<td>1.54</td>
<td>0.70</td>
<td>1.10</td>
<td>0.42</td>
<td>1.01</td>
</tr>
<tr>
<td>38_0</td>
<td>9</td>
<td>0.95</td>
<td>0.72</td>
<td>0.22</td>
<td>1.52</td>
<td>0.56</td>
<td>1.01</td>
<td>0.27</td>
<td>0.26</td>
</tr>
<tr>
<td>7_24</td>
<td>10</td>
<td>0.94</td>
<td>0.57</td>
<td>0.36</td>
<td>1.52</td>
<td>0.71</td>
<td>1.05</td>
<td>0.42</td>
<td>0.41</td>
</tr>
<tr>
<td>8_21</td>
<td>11</td>
<td>0.93</td>
<td>0.66</td>
<td>0.25</td>
<td>1.44</td>
<td>0.66</td>
<td>1.03</td>
<td>0.32</td>
<td>0.31</td>
</tr>
<tr>
<td>44_21</td>
<td>12</td>
<td>0.91</td>
<td>0.66</td>
<td>0.25</td>
<td>1.44</td>
<td>0.54</td>
<td>1.04</td>
<td>0.32</td>
<td>0.31</td>
</tr>
<tr>
<td>38_2</td>
<td>13</td>
<td>0.91</td>
<td>0.57</td>
<td>0.34</td>
<td>1.33</td>
<td>0.58</td>
<td>1.15</td>
<td>0.45</td>
<td>1.02</td>
</tr>
<tr>
<td>30_3</td>
<td>14</td>
<td>0.91</td>
<td>0.71</td>
<td>0.20</td>
<td>1.44</td>
<td>0.51</td>
<td>1.00</td>
<td>0.25</td>
<td>0.24</td>
</tr>
<tr>
<td>2_17</td>
<td>15</td>
<td>0.91</td>
<td>0.65</td>
<td>0.26</td>
<td>1.53</td>
<td>0.62</td>
<td>1.00</td>
<td>0.39</td>
<td>0.29</td>
</tr>
<tr>
<td>35_28</td>
<td>16</td>
<td>0.90</td>
<td>0.73</td>
<td>0.17</td>
<td>1.14</td>
<td>0.33</td>
<td>1.15</td>
<td>0.26</td>
<td>0.26</td>
</tr>
<tr>
<td>48_32</td>
<td>17</td>
<td>0.90</td>
<td>0.68</td>
<td>0.22</td>
<td>1.17</td>
<td>0.38</td>
<td>1.16</td>
<td>0.34</td>
<td>0.34</td>
</tr>
<tr>
<td>5_0</td>
<td>18</td>
<td>0.89</td>
<td>0.62</td>
<td>0.27</td>
<td>1.26</td>
<td>0.48</td>
<td>1.14</td>
<td>0.39</td>
<td>1.09</td>
</tr>
<tr>
<td>22_0</td>
<td>19</td>
<td>0.80</td>
<td>0.61</td>
<td>0.27</td>
<td>1.40</td>
<td>0.57</td>
<td>1.04</td>
<td>0.34</td>
<td>0.33</td>
</tr>
<tr>
<td>40_0</td>
<td>20</td>
<td>0.80</td>
<td>0.60</td>
<td>0.28</td>
<td>1.38</td>
<td>0.60</td>
<td>1.07</td>
<td>0.33</td>
<td>0.34</td>
</tr>
<tr>
<td>41_0</td>
<td>21</td>
<td>0.88</td>
<td>0.61</td>
<td>0.28</td>
<td>1.27</td>
<td>0.49</td>
<td>1.11</td>
<td>0.39</td>
<td>0.39</td>
</tr>
<tr>
<td>56_19</td>
<td>22</td>
<td>0.88</td>
<td>0.70</td>
<td>0.17</td>
<td>1.12</td>
<td>0.30</td>
<td>1.10</td>
<td>0.28</td>
<td>0.27</td>
</tr>
<tr>
<td>53_12</td>
<td>23</td>
<td>0.87</td>
<td>0.70</td>
<td>0.17</td>
<td>1.12</td>
<td>0.30</td>
<td>1.10</td>
<td>0.32</td>
<td>0.32</td>
</tr>
<tr>
<td>47_1</td>
<td>24</td>
<td>0.86</td>
<td>0.58</td>
<td>0.28</td>
<td>1.33</td>
<td>0.54</td>
<td>1.05</td>
<td>0.37</td>
<td>0.37</td>
</tr>
<tr>
<td>10_11</td>
<td>25</td>
<td>0.86</td>
<td>0.71</td>
<td>0.15</td>
<td>1.79</td>
<td>0.39</td>
<td>0.96</td>
<td>0.19</td>
<td>0.18</td>
</tr>
<tr>
<td>11_18</td>
<td>26</td>
<td>0.86</td>
<td>0.63</td>
<td>0.23</td>
<td>1.19</td>
<td>0.45</td>
<td>1.12</td>
<td>0.31</td>
<td>0.31</td>
</tr>
<tr>
<td>57_33</td>
<td>27</td>
<td>0.85</td>
<td>0.56</td>
<td>0.29</td>
<td>1.50</td>
<td>0.34</td>
<td>1.06</td>
<td>0.35</td>
<td>0.37</td>
</tr>
<tr>
<td>3_17</td>
<td>28</td>
<td>0.85</td>
<td>0.69</td>
<td>0.15</td>
<td>1.30</td>
<td>0.41</td>
<td>0.95</td>
<td>0.29</td>
<td>0.19</td>
</tr>
<tr>
<td>42_7</td>
<td>29</td>
<td>0.85</td>
<td>0.64</td>
<td>0.20</td>
<td>1.14</td>
<td>0.38</td>
<td>1.11</td>
<td>0.39</td>
<td>0.39</td>
</tr>
<tr>
<td>19_3</td>
<td>30</td>
<td>0.85</td>
<td>0.59</td>
<td>0.26</td>
<td>1.44</td>
<td>0.63</td>
<td>0.99</td>
<td>0.39</td>
<td>0.39</td>
</tr>
<tr>
<td>46_14</td>
<td>31</td>
<td>0.84</td>
<td>0.66</td>
<td>0.19</td>
<td>1.20</td>
<td>0.39</td>
<td>1.05</td>
<td>0.27</td>
<td>0.27</td>
</tr>
<tr>
<td>45_3</td>
<td>32</td>
<td>0.84</td>
<td>0.58</td>
<td>0.26</td>
<td>1.42</td>
<td>0.63</td>
<td>0.99</td>
<td>0.39</td>
<td>0.39</td>
</tr>
<tr>
<td>1_15</td>
<td>33</td>
<td>0.84</td>
<td>0.57</td>
<td>0.27</td>
<td>1.43</td>
<td>0.79</td>
<td>0.99</td>
<td>0.39</td>
<td>0.39</td>
</tr>
<tr>
<td>18_16</td>
<td>34</td>
<td>0.83</td>
<td>0.63</td>
<td>0.28</td>
<td>1.19</td>
<td>0.41</td>
<td>1.05</td>
<td>0.29</td>
<td>0.29</td>
</tr>
<tr>
<td>20_29</td>
<td>35</td>
<td>0.83</td>
<td>0.62</td>
<td>0.21</td>
<td>1.13</td>
<td>0.70</td>
<td>1.14</td>
<td>0.29</td>
<td>0.29</td>
</tr>
</tbody>
</table>
**FRED web app (Python, source code available)**

### Select Receptor
- **1ABE** - L-Arabinose-Binding Protein Complex with... (Select)
- **1ACM** - Carboxic Anhydrase I (E.C. 4.2.1.1) Complex (Select)
- **1CBX** - Carboxypeptidase A (E.C. 3.4.17.1) Complex (Select)
- **1SRF** - CYP2C9 (E.C. 1.1.1.139) Complex (Select)
- **1I2T** - Thermolysin (E.C. 3.4.24.27) Complex with... (Select)
- **1MRK** - Alpha-Tripsinin Complexed with FMRFamide (Select)
- **1PF8** - Cytochrome P450cam (E.C. 1.14.15.1) Complex (Select)
- **1P0C** - Phospholipase A2 (E.C. 3.1.1.4) Complex with... (Select)
- **1SRJ** - Streptavidin Complexed With Naphthyl-Har... (Select)
- **1TFP** - Beta-Tripsin (E.C. 3.4.21.4) Complex with... (Select)
- **2b8h** - EphB4 Receptor T1 Complex With An EPHrin... (Select)

### Omega Configuration
- **run_omega:**
- **ncnt:**
- **ewindow:** 25.0
- **rms dist:** 0.8
- **maxconf:** 49

### Fred Configuration
- **exhaustive_scoring:** chemgauss3
- **num_poses:** 100
- **solid-body_optimization:** none
- **refine:** no_refinement
- **scoring:**
  - chemgauss3
  - chegauss2
  - shapegauss
  - chemscore
- **consensus:**
- **num_alt_poses:** 0

### Output Settings
- **output:**
  - batch
  - view: depict, pdb, cgo, ogt, mol2, smi, chem
  - verbose
FRED web app (Python, source code available)
Canonicalized systematic nomenclature in chemoinformatics

And some new canonicalization tools from OpenEye

Jeremy J. Yang

Introduction

Canonicalization in chemoinformatics facilitates rigorous, unambiguous expression and handling of chemical data and knowledge. However, just as chemistry encompasses multiple levels of abstraction and modeling, no single canonicalization method is sufficient to solve all problems. This study reviews some existing canonicalization methodologies and describes new methods implemented by chemoinformatics library OEChem and other OpenEye tools.

Definition of canonicalization

A canonicalization algorithm must determine a single representation among many possible representations for an individual in its domain.

Benefits of canonicalization

- matching equality of molecules
- database search speed
- database informatics and thinking

NI (graph isomorphism is hard) – Morgan to the rescue

The Morgan algorithm is the basis of most chemical canonicalization work since it was described carefully. In 1995 Harry L. Morgan published the algorithm already implemented at CAS as its compound registry system. This work, based on generic graph theory, comprises a theoretical solution to the problem of molecular canonicalization, and material validation of its efficacy.

More Morgan, and more

The Morgan algorithm was a huge step forward, but the basic algorithm has some shortcomings in performance and comprehensiveness, which have been corrected by subsequent investigators. The resulting methods have been implemented and widely used in large scale database systems. Some key contributions:

- Wolke & Dutta, 1974: 8-vertex redefined Morgan -> MLD
- Jochum & Jastorfer, 1977: Morgan refinement -> CACTVS
- Shelley & Hank, 1987: Morgan refinement
- Wisthaler, 1989: 9-vertex linear canonical notation
- Dayton
- Broshman, 1991: parent compounds -> GSKsDaylight
- Denney & Sayle, 1999: tautomers -> OpenEye
- INCHE, 2004: global canonical linear notation

This study: canonical molecular descriptors, not descriptors

The study of graph theory and canonicalization applied to chemistry is extensive and elusive. Canonical descriptors which do not fully represent the model can be of great utility in statistical analyses but are not the focus of this nomenclature study.

Morgan demo and study

Fig 1: Morgan demo. Extended connectivity values and atom orders. Use OEChem and OpenEye. NCICD hierarchy processed with no errors.

Ahlu – Chemo-taxonomy is a “stranded hierarchy”

- Subatomic -> atoms -> molecules
- Normal weight atoms -> isotopes
- Keilua molecule model -> aromatic molecular models
- Non-stereo molecule -> stereomers
- Single molecule -> combinatorial libraries
- Single molecule -> isomers
- Small molecule -> macromolecules
- Complex molecules
- Single molecule -> tautomer set
- Single molecule -> pKa states
- Single molecule -> reactions
- 2D -> 3D

There is a hierarchical relationship among some of these expansions while some are independent. For example, computational chemistry may involve stereoisomeric individuals or next-stereo. For every combination of molecular representations, canonicalization could be advantageous for the reasons described. Hence the tools of canonicalization is a multi-faceted one.

Dealing with reality: practical problems

1. Existing formats (may often fail):
   - ambiguous – poorly defined spec or poor compliance
   - exclusion – both syntax and semantics are important
   - non-comprehensive – only organic, covalent, size limits

2. Stereomorphism: correct chemistry
   - relative stereo-content

3. Offering valence assumptions and conventions
   - implicit-valence and Molconn-Z formats prone to misunderstanding

4. Information content and model differences in existing formats
   - cannot robustly convert into must be inferred (e.g. bonds)

5. Dependent on context chemistry
   - e.g., valence, aromatic

6. Local versus global canonicalization
   - Benefits of canonicalization are available locally or globally. But global canonicalization requires cooperation.
   - Locality definition (time, place, software versions)

New: canonicalizing moieties

Canonicalizing a connection table is not new and was discussed by Morgan et al. and others. But generating canonical forms of current standard formats is not widely done, for historical and practical reasons, although the available benefits. This is increasingly true now that longer strings are more easily handled by existing computers. OEChem provides sufficient control to accomplish this task. Proposed algorithm:

- Remove non-structural data
- Supersede hydrogen
- Canonical atom order
- Canonical order
- Canonical Kekulé bond ordering based on (sized) aromatic model

However, the advantages of more force canonical level notations remain.

RESULTS: Using test program crystalline. 1991 NCICD protocol converted to canonical SDP files, exactly equal to SDP file converted via SMILES (chemie.eyesopen.com/gy/116/movies). Also done with NLD2 format. This last validates the ability of OEChem to canonicalize moieties as strings.

New: canonical tautomers

Tautomers have the same formula (structural isomers), but may differ in proton and electron location, and formal bond order. Special cases: ketones, aldehydes, azines, ring-chain. In the Delaunay/Skala algorithm (D), hydrogen donors and acceptors are accounted, and the number of free hydrogens, donors and acceptors are accounted canonically. At this stage the atomically equivalent inputs are represented identical. Hydrogen locations are exhaustively enumerated. A simple routine for enumeration order can designate the first or the canonical tautomer. Through additional rules, the likelihood can be increased that the canonical tautomer is a low-energy form. Applications: registration (exact search), substructure searching, prediction, similarity searching, property prediction, protein-ligand analysis. Failure to perceive tautomers lead to different results for different valence models which may represent the same chemical entity.

Fig 4: examples of tautomers listed separately in OEDChem. The latter is the OChem canonical form.

Results: The Maybridge 2003 database was analyzed by the OE program for tautomers. Of 71,957 molecules, 97 have tautomers (47 pairs and one triplet). Noted that additional, 2381 were found to be non-unique molecules.

New: canonical pKa states

The canonicalization of alternative pKa states is accomplished for many classes of molecules by the OpenEye program pKaChem. This problem resembles tautomer canonicalization in many respects, and is an area of active research at OpenEye.

Canonicalization systematic nomenclature in chemoinformatics

And some new canonicalization tools from OpenEye

References


OpenEye canonicalization tools

The OpenEye cheminformatics toolkit OECHEM employs an optimized Morgan-like canonical algorithm to generate canonical forms. In addition, the app provides a rich set of tools which can facilitate generation of canonical representations of many types, for many chemical and informational models, and for many standard file formats.

- OECHEM: OECHEM (OECHEM)
- OEDChem: OEDChem (OEDChem)
- OEDChem: OEDChem (OEDChem)
- OEDChem: OEDChem (OEDChem)
- OEDChem: OEDChem (OEDChem)
- OEDChem: OEDChem (OEDChem)
- OEDChem: OEDChem (OEDChem)

Conclusion

Rigorous and effective chemoinformatics systems require concepts and methods for canonicalization at multiple levels of chemical abstraction and organization. The current state of the art presents many theoretical and practical challenges. OpenEye tools can help.

OpenEye Scientific Software

3600 Cerillos Road
Sante Fe, New Mexico 87507
505.473.7385
info@eyesopen.com
www.eyesopen.com
Example research project using OpenEye tools

**Introduction**

Root mean square deviation (RMSD) measurements between molecular conformations are ubiquitous and indispensable in scientific literature. It appears widely accepted that to measure the difference or similarity between conformers, that is, a distance in “conformation space”, a single RMSD calculation is sufficient. The current empirical use of RMSD justified, and so forth. This study examines the uses and limitations of RMSD, and some alternative or supplementary measures of distance between conformers.

**Definition of RMSD**

The root mean square deviation can be calculated for any two equal sized vectors:

\[
\text{RMSD} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (x_i - y_i)^2}
\]

RMSD reduces N comparisons to a single scalar measure.

In the realm of molecular geometry, RMSD is used to compare two sets of atomic coordinates: \( x \) and \( y \), representing two 3D positions of the \( i \)-th atom. The set of atoms may comprise a complete molecule or substructure. The coordinates may exist in a defined reference frame (e.g., protein receptor site) in which case each geometry is said to represent a pose of the molecule. Or, coordinates may only have relative meaning and thus comprise a conformation for the molecule. In that case, the RMSD is normally minimized by finding the optimal alignment of the conformers.

**Minimized RMSD**

For any two conformers, an alignment exists for which RMSD is minimized. This alignment can be determined analytically. This alignment can be the means or the end. Many modeling tasks require geometric alignment.

**Symmetry - a critical detail**

Calculating the correct min-RMSD requires the optimal auto-homologous for uses of molecular symmetry; an implementation detail which requires rigorous chemoinformatics to avoid errors.

**For what tasks is RMSD used?**

- Compare two conformations.
- Compare two poses of the same or different conformations.
- Compare the coordinates of substructures.
- Measure the quality of a computed model vs. reference data (e.g., crystallographic or NMR).
- Measure the diversity of an ensemble of conformers or poses.
- Characterize and compare ensembles of conformations.

**Advantages of RMSD**

- Easily calculated.
- Unique, analytic minimum.\(^1\)
- Metric property (triangle inequality satisfied), thus more intuitive measure of “conformational space”.
- Emphasizes variance (relative to ordinary mean).

**What about the variance?**

A large substructure may be perfectly aligned despite large overall RMSD. The variance among distances in the maximum distance can reveal this. Fundamental RMSD problem: the task of interest requires an assessment of a geometric relationship which cannot always be summarized with one scalar measurement.

\[
\text{RMSD} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (x_i - y_i)^2}
\]

**Big RMSDs less informative**

Where “big” is dependent on molecule size. Hence while RMSD can define a conformation “space” for a single molecule, its scale is dependent on molecule size and other graph-theoretical descriptors, thus in use in describing heterogeneous databases is hampered.

**New methods yield new information**

- Test molecule: benzophenone (41 atoms, 28 conformations)\(^2\)
- Also tested: dopamine (22 atoms, 7 conformations), methanol (94 atoms, 268 conformations)

**RMSD: routine measure stirs doubts**

**Shape similarity**

- Intuitive, rigorous, physical, fast using OE Shape Toolkit
- Quantity alone ignores chemistry

- Methods for further study:
  - RMST, all torsions included
  - Could also be centrally weighted
  - More comprehensive than straight RMSD
  - Uncolored graph RMSD
  - Like shape similarity, indicates some chemistry-suggested geometrical equivalence.
- Pharmacophore RMSD
  - Instead of all atoms, use pharmacophore points
  - Requires expertise
  - Not readily automated
  - Involves expertise
  - 3D max common substructure
  - 3D match of fit needed
  - High computational cost

**Conclusions**

RMSD is a powerful and convenient measure but has limitations which can lead to errors and oversights. All minimum, investigators should be alert to cases where RMSD should be supplemented by other measures.

- In some cases RMSD is insufficient.
- In general larger RMSD warrant closer inspection.
- At molecule size increases, RMSD range increases, and descriptive power decreases.
- Several other measures are available to help characterize geometric relationships not covered by RMSD.
- Low correlations indicate RMSD does not reveal enough information provided by these alternative methods for geometry comparison.
- Conformer discrimination tests should include a variance or max atom-distance test in addition to RMSD.

**References and Notes**

1. A solution for the least square to make two sets of vectors, Wolfgang Kubelka, Atta-Dineh Danesh, 1967.
4. All algorithms implemented using OpenEye (OpenEye Scientific Software).
5. All conformations generated with Review (OpenEye Scientific Software).
6. Shapes similarity calculated using OpenEye (OpenEye Scientific Software).

**OpenEye Scientific Software**

3600 Carrillos Road
Suite 1107
Santa Fe, New Mexico 87507
505.473.7385
info@eyesopen.com
www.eyesopen.com

---

\(^1\) More specifically, it has the property of being a metric.

\(^2\) The number of conformations was determined using the ChemOffice Compound List Analysis tool.
Case Study #3

Symyx a.k.a. MDL

http://www.symyx.com
Symyx a.k.a. MDL

- “Molecular Design Ltd”, reflected initial \textit{ab initio} design goals, renamed “MDL Information Systems” ('93)
- Products: REACCS ('82), MACCS ('84), MACCS 3D ('88), ISIS ('91), Isentris ('04), chem +bio database systems
- Invented molfile/SD format (proprietary till 1991), and extensions (query, R-group).
- Based Chime on Rasmol source code.
- Purchased by Reed Elsevier ('97), then Symyx ('07), then Accelrys ('10).
Symyx a.k.a. MDL

- Max ~300 employees. (My guess.)
- By 1990, **all** pharma research companies used MDL software for their compound database.
- Then the MBAs, lawyers and admen took over!
- And innovation ceased.
Case Study #4

Accelrys

http://www.accelrys.com
Accelrys

A tale of mergers and acquisitions (~1990 - 2011):

- MSI (Molecular Simulations)
  - Biodesign
  - Cambridge Molecular Design
  - Polygen
  - Biocad
  - Biosym Technologies

- Synopsys Scientific Systems
- Oxford Molecular
- Genetics Computer Group
- Synomyx
- SciTegic
- Symyx
- Contur Software AB
Accelrys example of AEI SQL:

```
SELECT
  hts_ap_archive.runset_number as "RUN",
  hts_ap_archive.ap_alias as "APlateName",
  hts_plate.plate_id,
  hts_plate.alternate_id as "IPlateName",
  hts_well.well_no,
  hts_sample.alternate_id,
  hts_result_detail.value_char AS Target,
  hts_result_type.type_desc AS Result_Type,
  hts_assay_result.concentration || hts_conc_unit.unit_value AS CONC,
  hts_assay_result.dilution,
  hts_assay_result.result_value AS Value
FROM hts_well,
    hts_plate,
    hts_sample,
    hts_conc_unit,
    ddi_container_master,
    hts_ap_archive,
    hts_assay_result,
    hts_result_type,
    hts_result_detail
WHERE hts_ap_archive.ap_alias='213_20110712_135454-1'
AND hts_assay_result.sample_id=hts_well.sample_id
AND hts_assay_result.sample_id=hts_sample.sample_id
AND hts_well.plate_id=hts_plate.plate_id
AND hts_assay_result.plate_id=hts_ap_archive.ap_number
AND hts_plate.alternate_id=ddi_container_master.container_name
AND hts_well.plate_id=hts_plate.plate_id
AND hts_well.sample_id=hts_sample.sample_id
AND hts_assay_result.result_plate=ddi_container_master.container_id
AND hts_result_type.result_type=hts_assay_result.result_type
AND hts_assay_result.result_id=hts_result_detail.result_id
AND hts_assay_result.conc_unit=hts_conc_unit.unit_id
ORDER BY hts_well.well_no, Target, Result_Type
```
corporate merging reflected in schema, technology
Accelrys
Merging reflected in technology

- Chemical cartridges: AEI (v6 & v7), Symyx, new SciTegic cartridge
- Cheminformatics? E.g., canonicalize a smiles w/ Accelrys, Scitegic or MDL code.
- The Accord Enterprise Informatics (AEI 6.2) UNM purchased in 2009 is now a “legacy product”.
- Re-branding and re-packaging
- May be great technically, but challenging for customers and Accelrys alike.
Accelrys

key product: Pipeline Pilot

(Scitegic founded 1999, acquired by Accelrys 2004 for $21.5M.)
Opinionated Observation:

There is a sometimes subtle difference between (1) a software product with a support contract, and (2) a service contract which involves customizing and installing and configuring a software product, requiring or likely to require ongoing, additional service contracts. Accelrys and Symyx ("solutions providers") have generated much of their revenue using the latter. In contrast: tools providers.
1) M&A's reflected US & global business trends, but perpetual re-org is stressful and there are huge costs to employees, customers, and technical progress.

2) No coherent scientific mission, only business growth.

3) Lots of excellent software, science and people. But all challenged by the chaos.

"If in your science you only look for business, then you risk finding neither knowledge nor business."

— Haldor Topsøe
Case Study #5: OpenBabel
OpenBabel

- Open-source C++ community project based on OELib by Matt Stahl, OpenEye.
- C++ w/ wrappers via SWIG: Python, Perl, Java, Ruby, C#.
- 2011 paper in Journal of Cheminformatics*
- >160,000 downloads, >400 citations, used by >40 projects

$ obscreen
obscreen - screen molecules based on calculated properties

OpenBabel v2.2.3 (Dec 4 2010)
syntax: obscreen [options]

options:
  -i <infile>
  -o <outfile>
  -otable <output table>
  -badsmarts <badfile> ... bad smarts file [default:builtin]
  -minmwt <MINWT> ... minimum molweight [200]
  -maxmwt <MAXWT> ... maximum molweight [600]
  -minhbd <MINHBD> ... min Hbond donors [0]
  -maxhbd <MAXHBD> ... max Hbond donors [5]
  -minhba <MINHBA> ... min Hbond acceptors [0]
  -maxhba <MAXHBA> ... max Hbond acceptors [10]
  -minrot <MINROT> ... min rotors [0]
  -maxrot <MAXROT> ... max rotors [12]
  -minchiral <MINCHIRAL> ... min chiral atoms [0]
  -maxchiral <MAXCHIRAL> ... max chiral atoms [5]
  -minlogp <MINLOGP> ... min logP [-5.0]
  -maxlogp <MAXLOGP> ... max logP [5.0]
  -hbasmarts <HBASMARTS> ... default='[#6,#7;R0]=[#8]'
  -hbdsmarts <HBDSMARTS> ... default='![H0;#7,#8,#9]'
  -n_inmax <N> ... input limit
  -n_outmax <N> ... output limit
  -v ... verbose
  -vv ... very verbose
### OpenBabel: active discussion lists

#### Email Archive: openbabel-devel (read-only)

<table>
<thead>
<tr>
<th>Year</th>
<th>Jan</th>
<th>Feb</th>
<th>Mar</th>
<th>Apr</th>
<th>May</th>
<th>Jun</th>
<th>Jul</th>
<th>Aug</th>
<th>Sep</th>
<th>Oct</th>
<th>Nov</th>
<th>Dec</th>
</tr>
</thead>
<tbody>
<tr>
<td>2005</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2006</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2007</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2008</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2009</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2010</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2011</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Email Archive: openbabel-discuss (read-only)

<table>
<thead>
<tr>
<th>Year</th>
<th>Jan</th>
<th>Feb</th>
<th>Mar</th>
<th>Apr</th>
<th>May</th>
<th>Jun</th>
<th>Jul</th>
<th>Aug</th>
<th>Sep</th>
<th>Oct</th>
<th>Nov</th>
<th>Dec</th>
</tr>
</thead>
<tbody>
<tr>
<td>2001</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2002</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2003</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2004</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2005</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2006</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2007</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2008</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2009</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2010</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2011</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Indicators of community health**
OpenBabel
Used by eMolecules, quite successfully

Craig James, CTO
formerly of Daylight,
Accelrys
OpenBabel
(My opinions -- feel free to disagree.)

- OB is very good but not yet close to Daylight, OEChem or JChem in comprehensiveness, quality, and other measures.
- SMARTS accuracy not state of the art.
- OB continually improving thanks to capable and active community of developers and users.
- The value reflects the total quality developer years invested. So there is no reason OB cannot catch up, if the community continues to grow.
Some interesting and successful others:

1) ChemAxon - Budapest, Marvin, Java, JChem
2) Tripos (now Certara) - St. Louis, Sybyl, SGI
3) Chem Comp Group - Montreal, MOE, SVL
4) PyMol - was Delano, now Schrödinger (OSS)
5) Schrodinger - quantum, etc.
6) UCSF School of Pharmacy - DOCK, Kuntz & al.
7) NIH NCTT - OSS, Java, Tripod, bioassay analysis
8) Scitouch - Russia, Indigo, Dingo, Bingo
9) CDK - FOSS, Java
10) RDKit - FOSS, incl. machine learning
11) Silicos NV - OSS & commercial
12) Bioreason, VC-funded, bought by Simulations Plus
13) Mesa Analytics & Computing
14) NextMove Software
15) UNM Division of Biocomputing
16) IU SOIC CCRG
Some General Conclusions

- Cheminformatics software landscape is very complex
- Economics & Freakonomics are drivers but also personalities
- FOSS can compete with $$$ software.
- Landscape includes lots of hidden costs.
- “Software engineering” has been an ideal, far from reality overall.
- Software can enable or hinder science
- Diverse business models, diverse everything
- Plenty of technological change
- Some human change too
The End

Feel free to contact me directly with questions or ideas!

Jeremy J Yang
jejyang@indiana.edu