



RESURRECTING A ROGER RELIC: IMPLEMENTING THE DOUBLE CUBIC LATTICE VOLUME (DCLV) FOR RDKIT

Rachael Pirie, John Mayfield, Roger Sayle



A BRIEF HISTORY LESSON

The Double Cubic Lattice Method: Efficient Approaches to Numerical Integration of Surface Area and Volume and to Dot Surface Contouring of Molecular Assemblies

FRANK EISENHABER*

*Biochemisches Institut der Charité der Humboldt-Universität zu Berlin, Hessische Str. 3–4,
D-10115 Berlin-Mitte, Germany*

**PHILIP LIJNZAAD, PATRICK ARGOS, CHRIS SANDER, and
MICHAEL SCHARF**

*European Molecular Biology Laboratory, Meyerhofstr. 1, Postfach 10.2209,
D-69012 Heidelberg, Germany*

Received 24 March 1994; accepted 17 August 1994

```
/* pdbaccess.c
 * PDB Surface Accessibility Calculation
 * Roger Sayle, GSK
 * Version 1.2, July 1998
 */
```

Eisenhaber, F. et al, J. Comp. Chem. 16, 3, 273-284 (1995)



HOW DID THIS END UP ON MY DESK (AND IN RDKIT)?

RS

Roger Sayle

Some ancient C code [that probably no longer compiles].

To: Rachael Pirie, Cc: John Mayfield

Hi Rachael,
I thought the attached C source code might be useful to study.
It probably needs a rewrite to compile with modern C compilers,
or a port to python, but it's perhaps something we should contribute
to RDKit or our external git. It's designed to calculate the surface
area (and solvent volume) buried by a ligand upon docking.

Interestingly, it was (GSK's/EBI's) Andrew Leach that originally
asked me to write this code, nearly 30 years ago.

A numerical approximation to van de Waal's surface area might also
be useful to compare to the analytical equations used in your PhD work.

I hope this is useful/interesting.

Cheers,
Roger

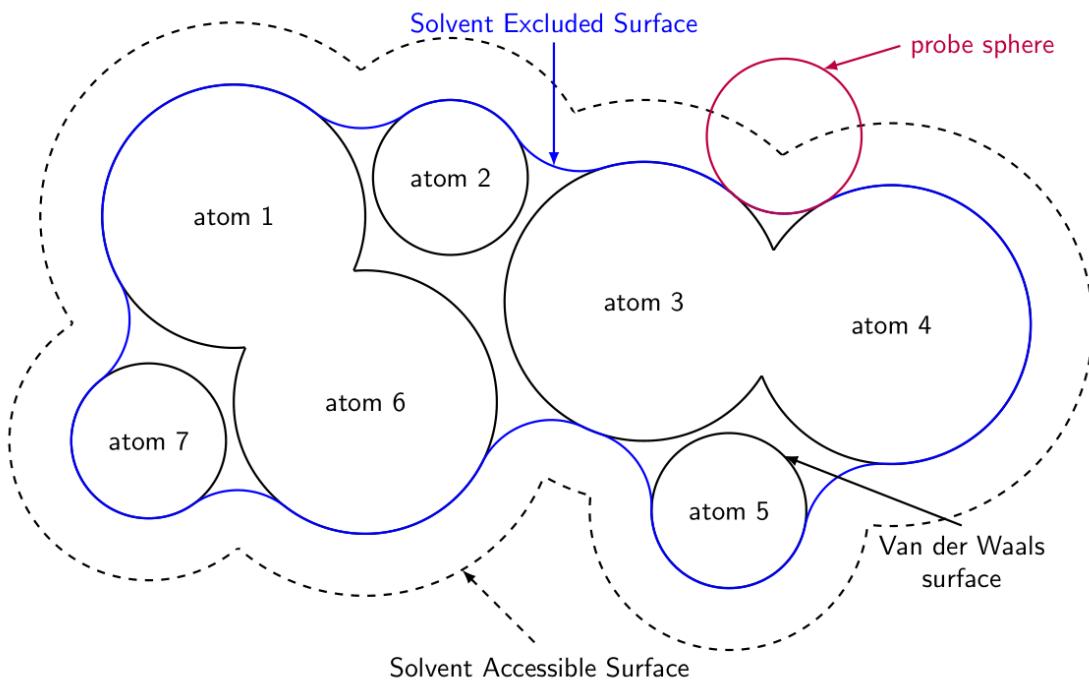
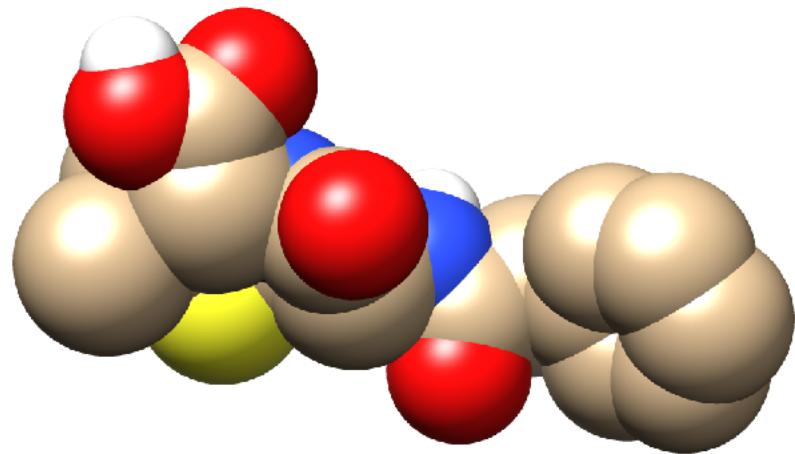
--



- Potential thesis correction
- Useful RDKit contribution
- Opportunity to learn some C++



RECAP



<https://tex.stackexchange.com/questions/463219/solvent-accessible-and-excluded-surface>

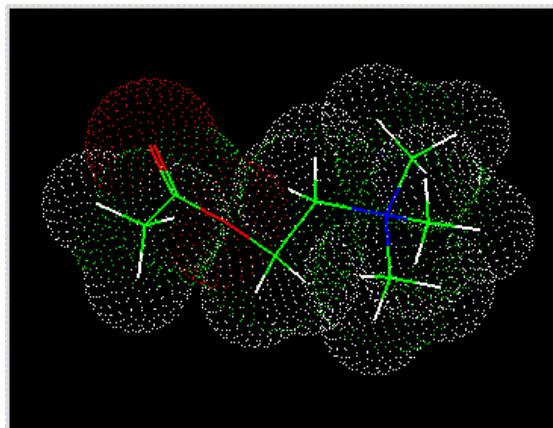


RECAP

$$A = \sum \left(\frac{R}{\sqrt{R^2 - Z_i^2}} \right) \cdot D \cdot Li$$

$$D = \frac{\Delta Z}{2 + \Delta' Z}$$

Lee and Richards



<https://images.app.goo.gl/axdKg5bqtZUznH226>

Shrake and Rupley

Lee, B. and Richards, F. M, J Mol. Biol. 55, 379-400 (1971)

Shrake, A. and Rupley, J. A, J Mol. Biol. 79, 351-371 (1973)



METHOD - OVERVIEW

```
class rdkit.Chem.rdMolDescriptors.DoubleCubicLatticeVolume( (Mol)mol, (bool)isProtein=False,  
 (bool)includeLigand=True, (float)probeRadius=1.2, (int)depth=4, (int)dotDensity=0)
```

ARGUMENTS:

- mol: molecule or protein under consideration
- isProtein: flag to indicate if the input is a protein
- includeLigand: flag to include or exclude a bound ligand when input is a protein
- probeRadius: radius of the solvent probe
- depth: control of number of dots per atom
- dotDensity: control of accuracy

GetCompactness((DoubleCubicLatticeVolume)arg1) → float:
Get the Compactness of the Protein

GetPackingDensity((DoubleCubicLatticeVolume)arg1) → float:
Get the PackingDensity of the Protein

GetSurfaceArea((DoubleCubicLatticeVolume)arg1) → float:
Get the Surface Area of the Molecule or Protein

GetVDWVolume((DoubleCubicLatticeVolume)arg1) → float:
Get the van der Waals Volume of the Molecule or Protein

GetVolume((DoubleCubicLatticeVolume)arg1) → float:
Get the Total Volume of the Molecule or Protein



Added in **2024.03** release



Proteins, Ligands (default) and
Protein-Ligand complexes.



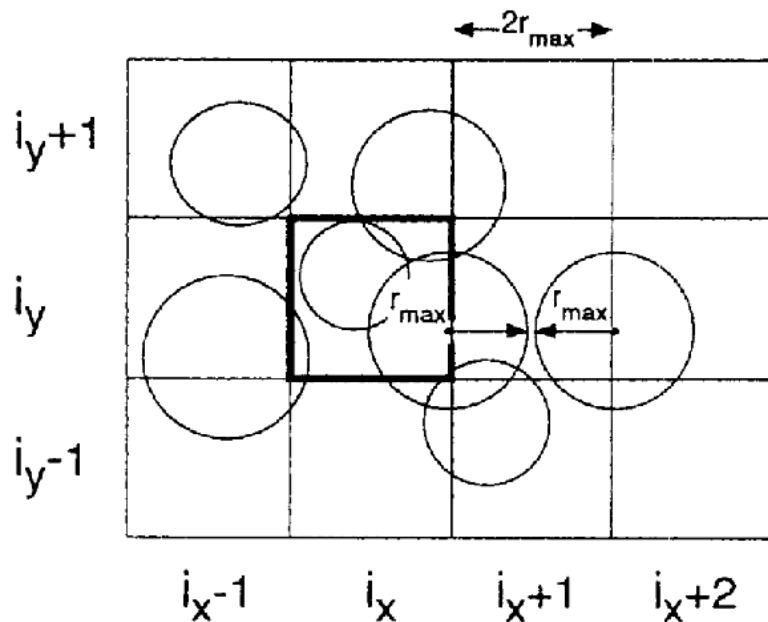
Excludes Hydrogen atoms.



Variant of Shrake and Rupley

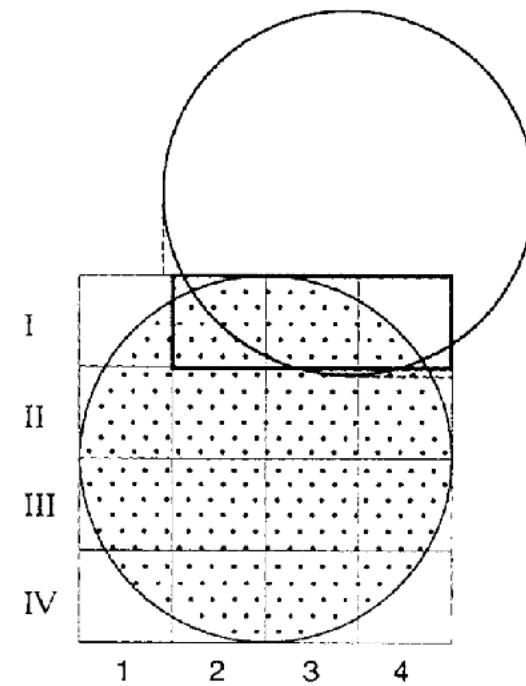


METHOD – DOUBLE CUBIC LATTICE



Lattice No. 1

Grouping atoms to identify neighbours



Lattice No. 2

Grouping dots to identify which are shared

Reduces CPU time by avoiding redundant distance checks

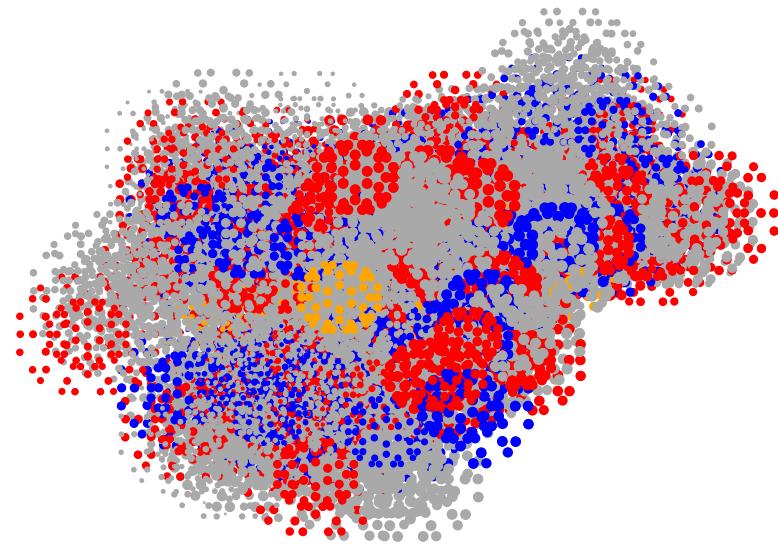
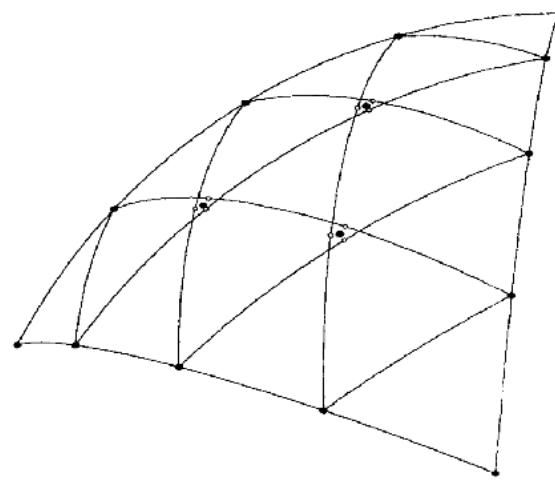


METHOD - PRODUCING THE DOT SURFACE

```
if (isProtein) {  
    generateElemPoints(&elemA, 1.87, probeRadius, dotDensity);  
    generateElemPoints(&elemC, 1.76, probeRadius, dotDensity);  
    generateElemPoints(&elemN, 1.65, probeRadius, dotDensity);  
    generateElemPoints(&elemO, 1.4, probeRadius, dotDensity);  
    generateElemPoints(&elemX, 1.8, probeRadius, dotDensity);  
} else {  
    generateElemPoints(&elemC, 1.87, probeRadius, dotDensity);  
    generateElemPoints(&elemN, 1.5, probeRadius, dotDensity);  
    generateElemPoints(&elemO, 1.4, probeRadius, dotDensity);  
    generateElemPoints(&elemS, 1.84, probeRadius, dotDensity);  
    generateElemPoints(&elemX, 1.44, probeRadius, dotDensity);  
}
```



METHOD - PRODUCING THE DOT SURFACE



Eisenhaber, F. et al, J. Comp. Chem. 16, 3, 273-284 (1995)



METHOD – SURFACE AREA

$$A = 4\pi \sum_i r_i^2 \frac{m_{acc}(i)}{m}$$

Radius

Dots buried by neighbour

Total No. Dots

Atom Index



METHOD – VOLUME

$$V = \frac{4\pi}{3m} \sum_i [r_i^2 a_i \cdot (\sum_{k(i)} p_k) + r_i^3 m_{acc}(i)]$$

Volume (V_{VDW} for $r=0$)

X, Y, Z coords Dots buried by neighbour

Total No. of Dots Radius Vector of Dots Radius



METHOD - METRICS

$$Z = A \cdot (36\pi \cdot V^2)^{-1/3}$$

↑
Surface Area ↑
 Volume

Compactness

$$p = V_{vdw}/V_{total}$$

Packing Density



BENCHMARKING – EXISTING RDKIT METHODS

Surface Area

```
rdkit.Chem.MolSurf.pyLabuteASA(mol, includeHs=1)
```

-  Calculates Labute's Approx. Surface Area (from MOE)
-  Sum of VDW area of each sphere - area two spheres intersect
-  Ignores case where portion of sphere A is contained within more than one other sphere
-  Python and C++

```
rdkit.Chem.rdFreeSASA.CalcSASA((Mol)mol, (AtomPairsParameters)radii,  
(int)confIdx=-1, (Atom)query=None, (SASAOpts)opts=<rdkit.Chem.rdFree  
SASA.SASAOpts object at 0x70bdff1786d0>) → float:
```

-  Free == Free to use
-  Shrike and Rupley + Lee and Richards
-  Python and C++



BENCHMARKING - EXISTING RDKIT METHODS

Volume

```
rdkit.Chem.AllChem.ComputeMolVolume(mol, confId=-1,  
gridSpacing=0.2, boxMargin=2.0)
```

- 👨‍💻 Approximate volume from occupied voxels in grid containing VDW spheres
- 👨‍💻 Documentation ????
- 👨‍💻 Python Only

All methods run from Python for fair comparison on M2 MacBook Pro

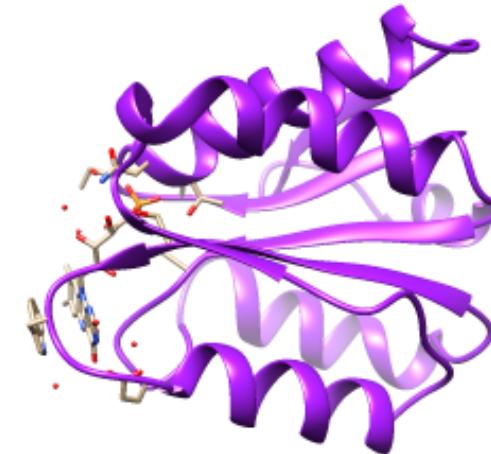


BENCHMARKING

3FXN (now 5NLL), 138 residues

	Analytical	Eisenhaber et al.	DCLV
--	------------	-------------------	------

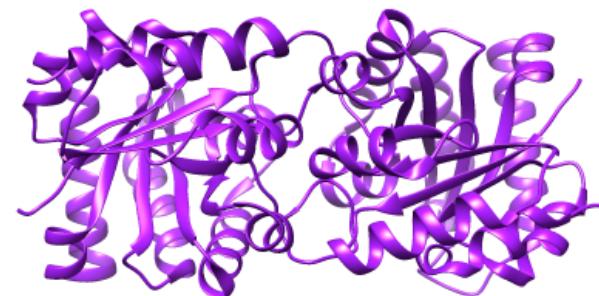
Surface Area (\AA^2)	6943.8	6933.4	6850.02
Volume (\AA^3)	26350.0	26281.9	28863.7
Compactness	1.62	1.62	1.51



1TIM, 492 residues

	Analytical	Eisenhaber et al.	DCLV
--	------------	-------------------	------

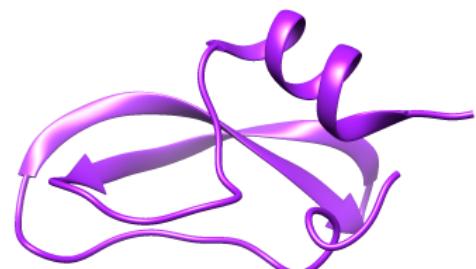
Surface Area (\AA^2)	20002.8	19997.1	20538.8
Volume (\AA^3)	89100.7	88972.7	89905.8
Compactness	2.07	2.08	2.12



4PTI, 58 Residues

	Analytical	Eisenhaber et al.	DCLV
--	------------	-------------------	------

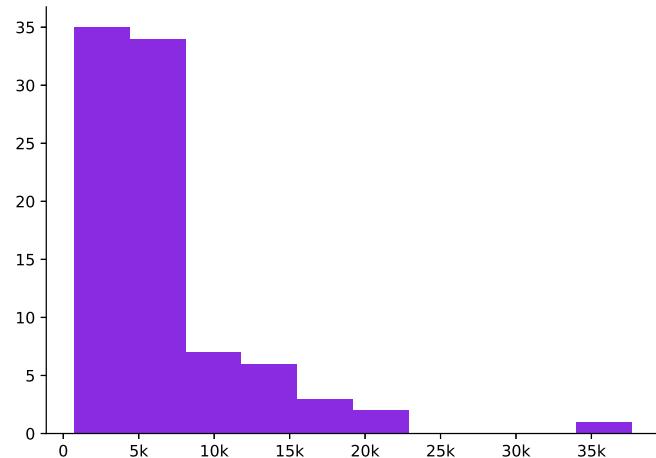
Surface Area (\AA^2)	3973.8	3971.8	4052.16
Volume (\AA^3)	11915.3	11923.7	12019.5
Compactness	1.58	1.58	1.6



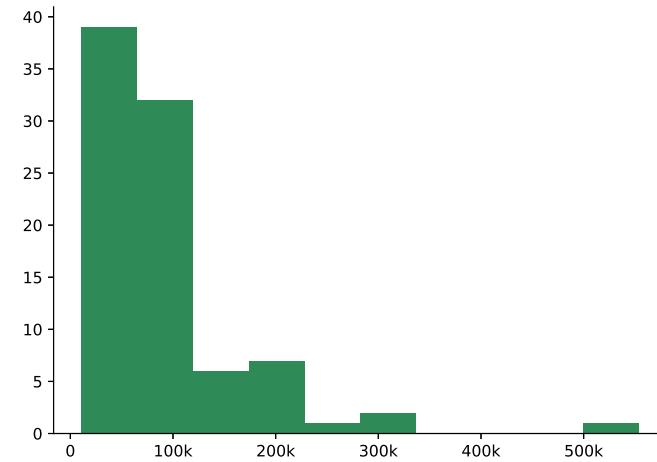
BENCHMARKING - PICSES DATA

PICSES data set as used in benchmarking FreeSASA - 88 proteins selected at random from the full database across a distribution of size. All chains included. All resolutions $\leq 1.6\text{\AA}$.

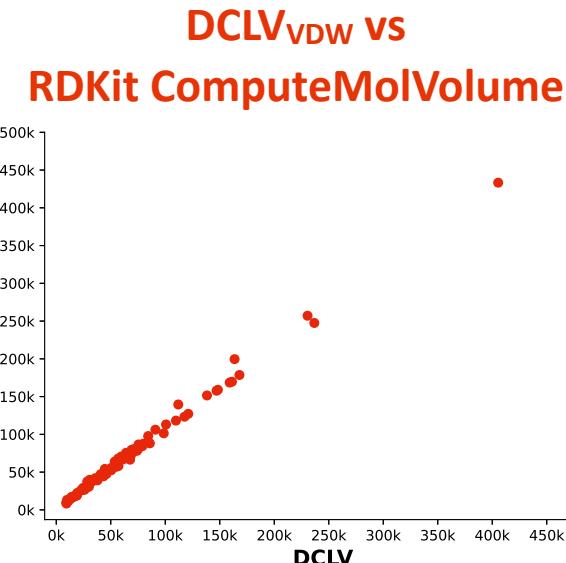
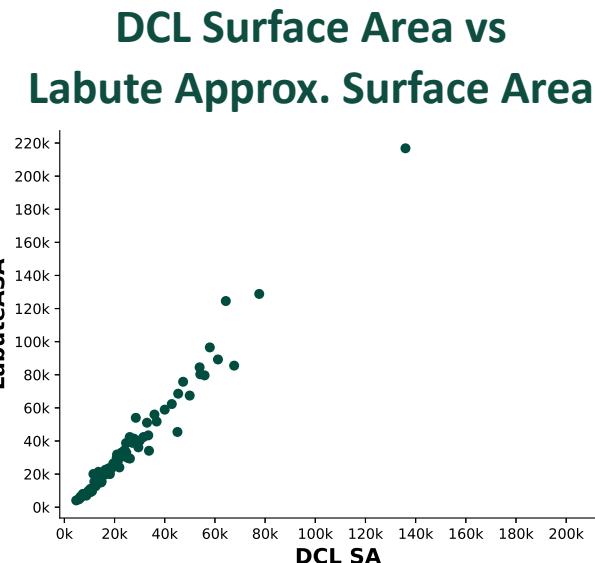
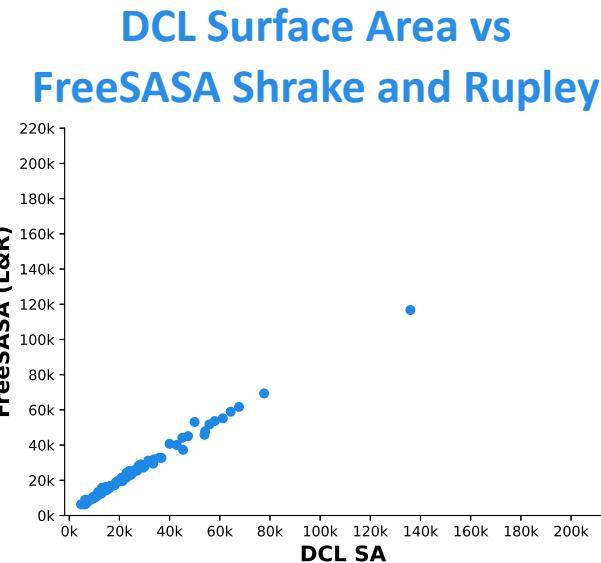
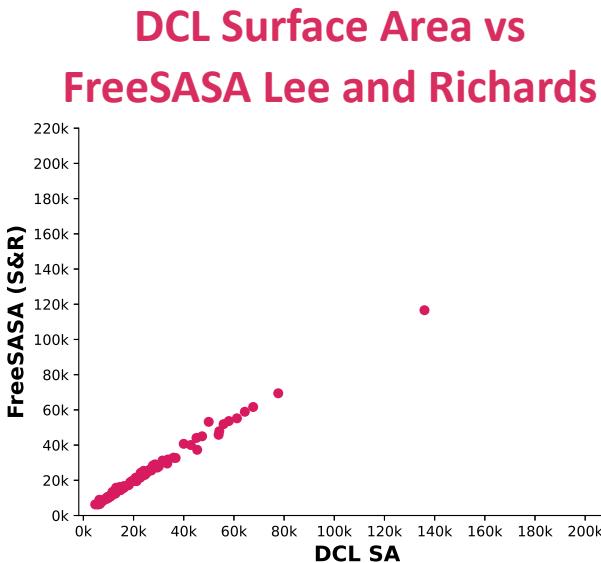
Atom Count



Molecular Weight



BENCHMARKING - PICSES RESULTS



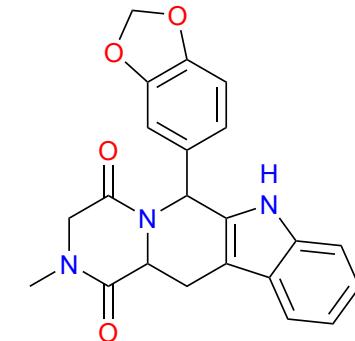
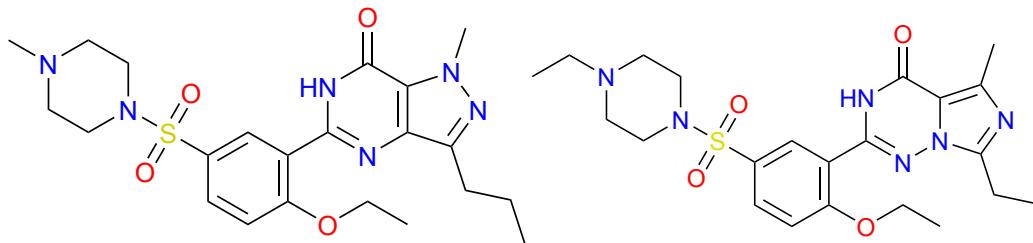
BENCHMARKING – PICSES RESULTS

	DCLSA	FreeSASA (L&R)	FreeSASA (S&R)	LabuteASA
Time (s)	41.71	15.81	4.41	1.78
Mol/s	2.11	5.57	19.96	49.52

	DCLV	RDKit ComputeMolVolume
Time (s)	38.35	754.79
Mol/s	2.29	0.15



BENCHMARKING – LIGANDS



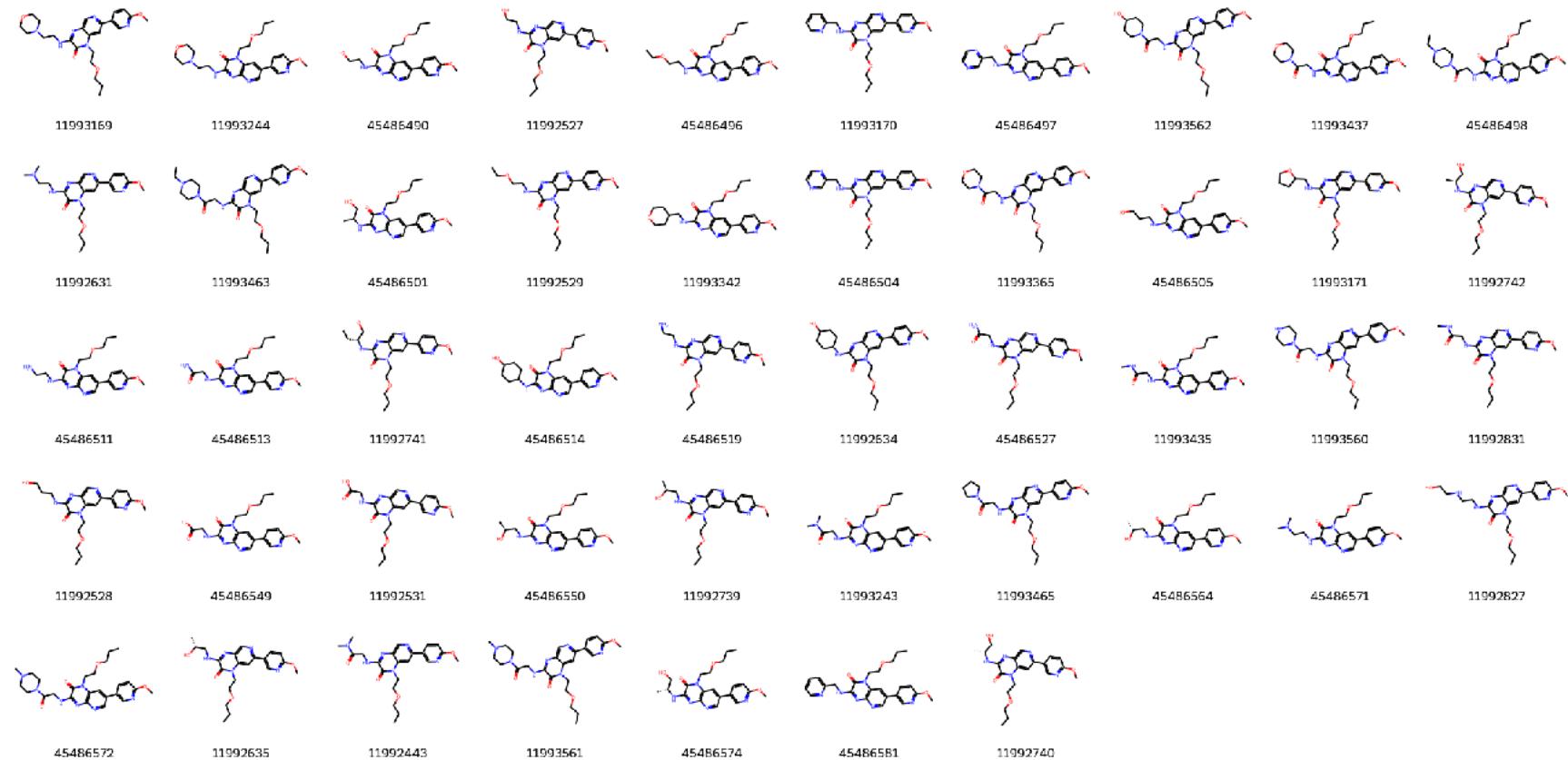
Method	Sildenafil	Vardenafil	Tadalafil
DCL SA	453.8	472.7	344.4
FreeSASA (L&R)	433.2	450.3	332.6
FreeSASA (S&R)	435.2	447.7	333.6
LabuteASA	192.8	199.2	166.4
DCLV	414.3	430.1	334.6
RDKit Vol	366.7	380.0	296.2

Surface areas in Å², Volumes in Å³

Rashid, A., Clin. Cornerstone, 7, 47-55 (2005)



BENCHMARKING - PUBCHEM 446781

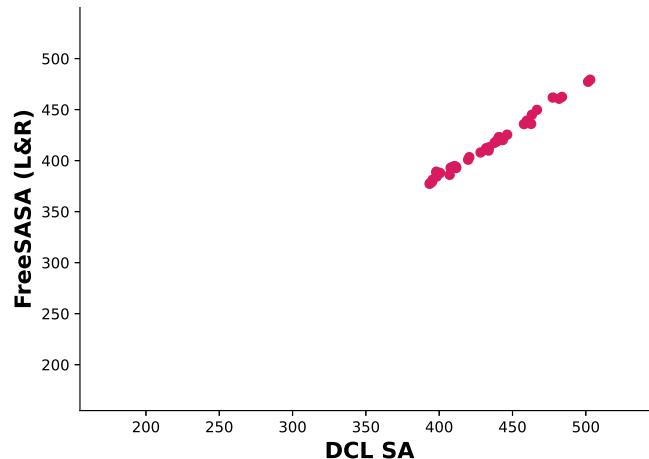


47 other known inhibitors of PDE5

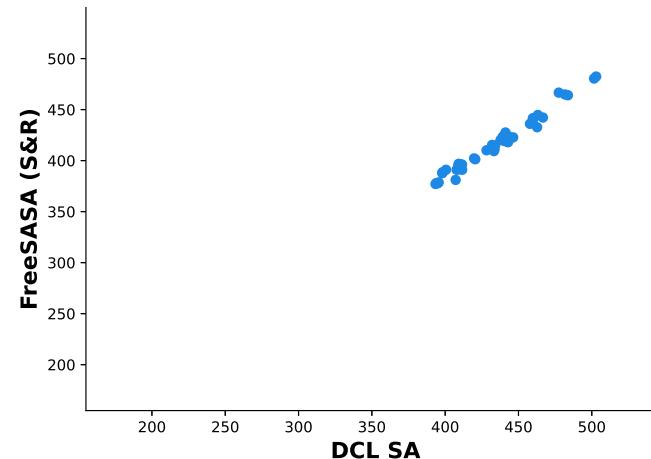


BENCHMARKING - PUBCHEM 446781

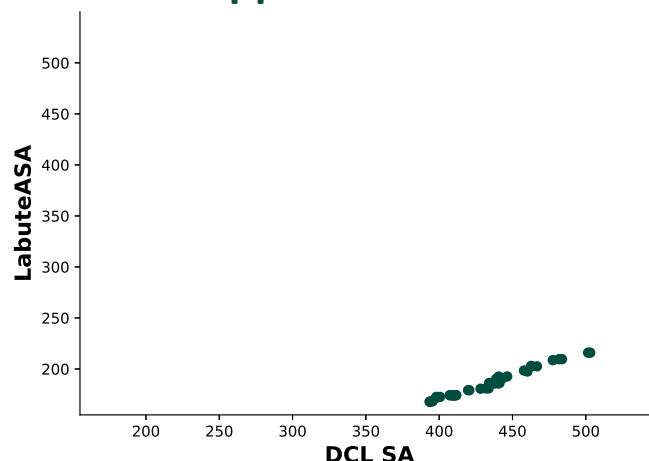
DCL Surface Area vs
FreeSASA Lee and Richards



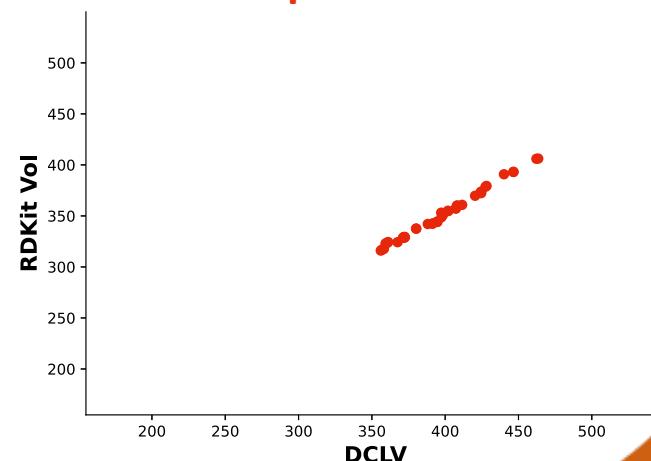
DCL Surface Area vs
FreeSASA Shrike and Rupley



DCL Surface Area vs
Labute Approx. Surface Area



DCLV_{VDW} VS
RDKit ComputeMolVolume



BENCHMARKING - PUBCHEM 446781

	DCLSA	FreeSASA (L&R)	FreeSASA (S&R)	LabuteASA
Time (s)	0.743	0.555	0.623	0.453
Mol/s	63.26	84.68	75.44	103.75

	DCLV	RDKit ComputeMolVolume
Time (s)	0.855	4.169
Mol/s	54.97	11.27



CONCLUSIONS



- New method in RDKit for calculating volume, surface area, packing density and compactness.
- Compares well to existing FreeSASA surface area and ComputeMolVolume methods in RDKit
- Faster than existing ComputeMolVolume + offers ability to compute solvent accessible volume.



FUTURE WORK



Polar Surface Area (given a list of which atoms are polar) - **in progress**



Other potentially useful 3D descriptors (from Greg):

- Contribution of individual atoms to surface area and volume
- Surface points
- Number of surface points per atom



ACKNOWLEDGEMENTS



Roger Sayle
Team @ NextMove
Greg Landrum



Contact

rachael@nextmovesoftware.com

