Two Parameter Persistence for Virtual Ligand Screening

(joint work with Michael Lesnick and Ted Willke)

Bryn Keller
Intel Labs
Institute for Mathematics and its Applications,
University of Minnesota
15 Aug 2018
Drug Discovery
The problem

- Drug companies have massive databases of chemicals
- They don’t know what most of them do. Some are good (future) drugs. Some are poison, or just useless. Almost all of them have never been studied
- They want a way to take a known good drug and find other things similar to it in their databases, so they can see if those would be good drugs too
It costs about $2.6 billion to develop a single drug!
Some medicinal chemistry vocabulary

Virtual Screening is the process of finding candidate drugs using a computer

Ligands (drugs and other substances) fit into binding pockets (aka targets) in proteins like a key in a lock

Two approaches:
• Structure-based: use the lock to find a key that fits
• Ligand-based: use the key to find other keys that might fit the same lock

Our task is to create a superior ligand-based method
An aside: molecules have complicated shapes

• Compounds have multiple *conformations* that have same atoms, but are rotated or twisted differently

• How to account for these rotational conformations… without having to store them all in the database?

• How to match them as if they were different compounds?

• These rotations matter – up to 100x difference in drug effectiveness!’

• We’re not even talking about other forms of (structural) isomerism, those kinds of isomers are stored as different molecules.
**Persistent Homology for virtual Screening (PHoS)**

- Essence of the problem is nearest neighbor search
- Given a target substance, return the top N best matches for that in the database
- Use 2-parameter persistent homology to capture essential features of the 3D shape of molecules, and generate molecular signatures
- Store these signatures in a database
- Use smart metric data structures to minimize number of comparisons
- Parallelize and distribute!
Molecular Signatures
Idea

• Use 2-parameter persistence to capture three persistence modules $(H_0, H_1, H_2)$ for each molecule

• First parameter: Euclidean distance between atoms

• Second parameter: Some kind of chemical property of atoms, e.g. partial charge, mass, hybridization, aromaticity, etc.
1-parameter persistence on Euclidean distance
Two parameters

CHANGE

DISTANCE (ε)
Choice of parameters matters
RIVET

• Rank Invariant Visualization and Exploration Tool

• Tool for calculating 2-parameter persistence modules from data and visualizing them

• Invented by Mike Lesnick & Matthew Wright in about 2013, with help for the last couple of years from me and a growing number of contributors


• Get it at http://rivet.online

• Python API available at https://github.com/rivettda/rivet-python
RIVET in action
2 views of aspirin

How barcodes vary as we vary distance

How barcodes vary as we vary partial charge
Distances
2 notions of distance

- **(Approximate) matching distance** - more accurate, much more expensive

- **$L^2$ distance on the restricted Hilbert function** - fairly accurate, much faster

- In both cases, we take the total distance between molecules A and B to be the sum of the distances between the $i^{th}$ persistence modules of A and B, $i = 0,1,2$
Matching distance

- Each choice of angle and offset produces a (potentially) different barcode.

- We call these the fibered barcodes

- We want to compare 2-parameter persistence modules by considering (a subset of) all possible fibered barcodes
Matching distance

\[ d_M(B, C) := \max_L w_L d_b(B(L), C(L)) \]

- B, C are 2-parameter persistence modules
- L ranges over affine lines of positive slope
- B(L) is the fibered barcode of B along L
- \( w_L \) is a weight that depends only on the slope of L
- \( d_b \) is the bottleneck distance
Hilbert Function

\[ \text{Hil}_M(a) = \dim M_a \]

For \( M \) a persistence module, and \( a \) a bigrade
Hilbert function visualization with RIVET

Aspirin

Tylenol

Doxorubicin
Restricted Hilbert Function

• Informally, the restricted Hilbert function is the Hilbert function within the bounds shown in RIVET, and 0 elsewhere.

• Formally, for $i \geq 0$, let $M^i$ denote the $i^{th}$ module in a minimal free resolution of $M$, and let $R(M)$ be the minimal rectangle containing all bigrades of elements in bases for $M^0$ and $M^1$. Then:

$$\text{RHil}_M(a) := \begin{cases} 
\text{Hil}_M(a) & \text{for } a \in R(M), \\
0 & \text{otherwise}. 
\end{cases}$$
$d^2(f, g) := \sqrt{\int (f - g)^2 \, dA}$. 
$d^2(\cdot, \cdot) = \sqrt{\sum \cdot 2}$

$L^2$ distance on restricted Hilbert functions (discretely, concretely)
An Example
Results
Scoring

Test databases have two kinds of molecules in them:
- **Active** molecules are (probably) good drugs
- **Decoy** molecules are poisons, inert, or otherwise bad, but have many similar properties to the actives

\[
EF_\alpha = \frac{N_{\text{actives, } \alpha\%} / N_{\text{database, } \alpha\%}}{N_{\text{actives}} / N_{\text{database}}}
\]
Datasets

• Two databases tested

• Cleves & Jain dataset is small: 979 compounds, about 850 of which are decoys. The same decoys are used for each protein target. Used for comparison with Shin et al. (2015) study.

• DUD-E (Directory of Useful Decoys - Extended) is a large publicly available testing dataset. We used a 1.5M compound drug-like subset, with samples of about 1000 substances per protein target, with a similarly high percentage of decoys. More realistic than Cleves & Jain.
Comparison with industry leaders

- OpenEye ROCS is the tool to beat
  - Expensive ($60,000 per user), so we couldn’t run it ourselves
- Ultrafast Shape Recognition (USR) (Ballester 2007) is an open source tool that is fast and simple
- We’ll use USR as a reference to estimate our performance vs. ROCS, using the Shin et al. 2015 study
One small caveat

- Even to load the Cleves & Jain dataset, one needs to use the $60,000 package.
- Or you can use open source: RDKit, which we did
- It fails to load a few, so our dataset is slightly smaller than the Shin et al. one
- Based on performance of USR (10.16 on our dataset, 8.8 on Shin et al.), calculate adjustment factor of 1.155
- PHoS best average result: 18.598 / 1.155 = estimated 16.10 vs. ROCS 15.9.
- We judge PHoS is likely about as effective as ROCS, an industry leading system with > 10 years’ history and a company behind it
2-parameter persistence matters (CJ)
Choice of parameters matters (DUD-E)
Different targets like different parameters (CJ)

<table>
<thead>
<tr>
<th>Target</th>
<th>partial charge</th>
<th>atomic no.</th>
<th>total degree</th>
<th>aromatic</th>
<th>hybrid</th>
<th>1-param</th>
<th>USR</th>
</tr>
</thead>
<tbody>
<tr>
<td>d</td>
<td>32.593</td>
<td>12.914</td>
<td>15.989</td>
<td>17.834</td>
<td>16.604</td>
<td>11.684</td>
<td>16.521</td>
</tr>
<tr>
<td>e</td>
<td>15.21</td>
<td>20.281</td>
<td>19.267</td>
<td>15.21</td>
<td>13.182</td>
<td>11.154</td>
<td>11.098</td>
</tr>
<tr>
<td>f</td>
<td>35.84</td>
<td>27.569</td>
<td>19.299</td>
<td>13.785</td>
<td>33.083</td>
<td>19.299</td>
<td>16.458</td>
</tr>
<tr>
<td>g</td>
<td>32.449</td>
<td>12.168</td>
<td>17.239</td>
<td>17.239</td>
<td>16.224</td>
<td>11.154</td>
<td>9.08</td>
</tr>
<tr>
<td>j</td>
<td>6.79</td>
<td>10.097</td>
<td>6.964</td>
<td>10.097</td>
<td>8.182</td>
<td>5.919</td>
<td>7.622</td>
</tr>
<tr>
<td>o</td>
<td>12.204</td>
<td>11.356</td>
<td>11.45</td>
<td>11.921</td>
<td>14.136</td>
<td>8.01</td>
<td>9.519</td>
</tr>
<tr>
<td>p</td>
<td>7.902</td>
<td>5.569</td>
<td>4.139</td>
<td>4.064</td>
<td>5.72</td>
<td>5.644</td>
<td>4.643</td>
</tr>
<tr>
<td>r</td>
<td>7.655</td>
<td>23.162</td>
<td>11.188</td>
<td>12.759</td>
<td>27.873</td>
<td>12.759</td>
<td>16.797</td>
</tr>
<tr>
<td>u</td>
<td>19.444</td>
<td>25.0</td>
<td>20.139</td>
<td>19.792</td>
<td>19.792</td>
<td>13.889</td>
<td>7.946</td>
</tr>
<tr>
<td>v</td>
<td>15.96</td>
<td>23.94</td>
<td>17.456</td>
<td>17.456</td>
<td>18.454</td>
<td>10.474</td>
<td>9.429</td>
</tr>
</tbody>
</table>
Accuracy: $L^2$/Hilbert vs Matching (DUD-E)

Accuracy ($EF_{25\%}$) for $L^2$ Distance

Accuracy ($EF_{25\%}$) for Matching Distance
Runtime Performance
L² vs Matching (DUD-E)
Summary

• Estimated performance competitive with the best in the industry

• 2-parameter persistence quite a bit better than 1-parameter persistence in this context

• Different choices for 2nd parameter win on different protein targets.

• $L^2$ distance on restricted Hilbert function is surprisingly effective, and much faster than matching distance

• Preprint now available: https://chemrxiv.org/articles/PHoS_Persistent_Homology_for_Virtual_Screening/6969260
Side effects

• RIVET ([http://rivet.online](http://rivet.online)) enhanced:
  • C, C++ API (included in standard RIVET)
  • Python API ([https://github.com/rivettda/rivet-python](https://github.com/rivettda/rivet-python)) (now available!)
  • Rust API ([https://github.com/rivettda/rivet-rust](https://github.com/rivettda/rivet-rust)) (coming soon)
  • RIVET console application & APIs can generate Betti information / Hilbert function, bounds information, barcode queries.
    • Python & Rust APIs also support calculating both distances described today
  • Hera enhanced with C API (to be donated soon if desired)
Future Work

• Understand variation in effectiveness of different 2nd parameters on different protein targets

• Ways to handle conformations directly (e.g. treat molecules as configuration spaces)

• Combinations of signatures (e.g. include both partial charge and hybridization)

• Complexes other than VR (e.g. alpha, cubical)

• More fine tuning of distance metrics
  • Possibly using machine learning, e.g. to learn weights for different persistence modules

• Performance tuning
Thank you!

- Thanks to people who improved RIVET in ways that mattered for this work:
  - Mike Lesnick, Matthew Wright, Simon Segert, Roy Zhao

- Thanks to the Hera developers:
  - Michael Kerber, Dmitriy Morozov, and Arnur Nigmetov

- bryn.keller@intel.com
- @BrynKeller on Twitter
- https://linkedin.com/in/brynkeller
- https://www.xoltar.org