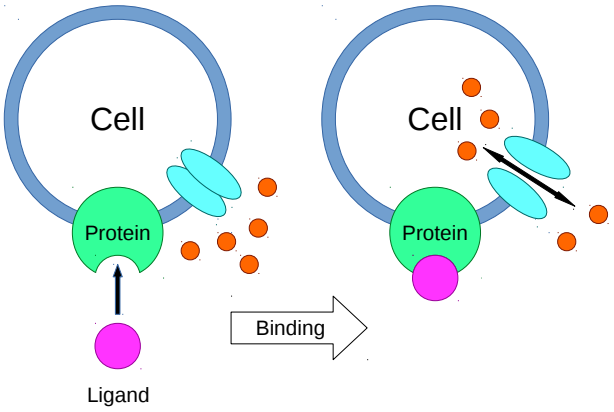
Ligand Affinity Prediction with Multi-Pattern Kernels

Katrin Ullrich, Jennifer Mack, Pascal Welke

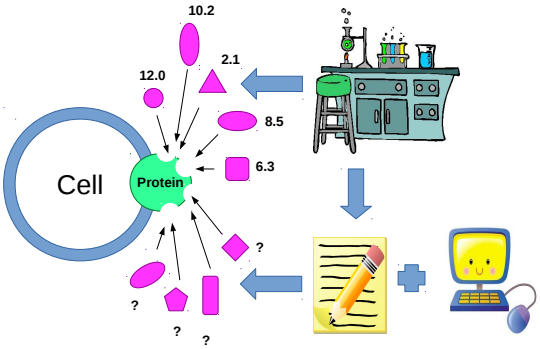
DS 2016

universitätbonn

Practical Problem

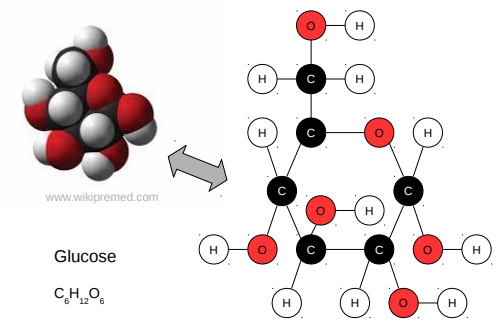


- small molecules (*ligands*) bind to proteins
- protein ligand binding triggers many biochemical processes
- $\Rightarrow$ starting point for *drug discovery* and *design*
- strength of bond characterized via real-valued *affinity*



- Affinity values can be determined practically
- This process is still time- and cost-intensive
  ⇒ We want to predict unknown affinities with machine learning tools!

Molecular Graphs



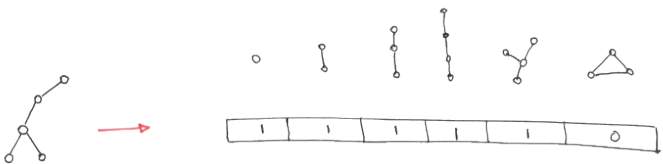
- Ligands can be represented as *labeled undirected graphs*
- Vertices correspond to atoms, edges to bonds
- Vertex labels: C, O, H, N, S, ...
- Edge labels: single, double, aromatic bond

How to Learn From Graphs?

- Graphs are nice data structures
  - they capture a lot of information about chemical molecules
- But how can we access the contained information with machine learning algorithms?
  - by finding a feature representation for each graph

How to Learn From Graphs?

- Graphs are nice data structures
  - they capture a lot of information about chemical molecules
- But how can we access the contained information with machine learning algorithms?
  - by finding a feature representation for each graph



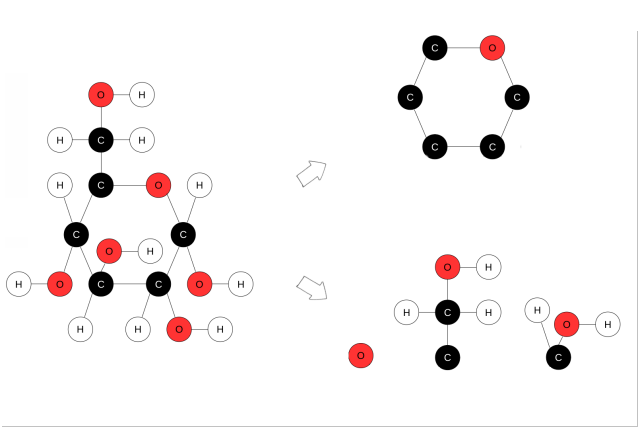
Fingerprints: the State of the Art

- Different publicly available or commercial feature representations for small molecules exist, so-called *molecular fingerprints*
  - Structural and/or physico-chemical information
  - Binary, counting, or, real-valued format
  - *MACCS Keys*: 166 binary molecular features
    ECFP Fingerprints: binary subtree patterns
    Graph Kernel Features: (soon)
- *State-of-the-art* for ligand affinity prediction:
  support vector regression (SVR) using one of the available molecular fingerprints

Our Contribution

- Question: Can we take profit from the diversity of these descriptors and how?
- Idea: Instead of choosing one descriptor in an expensive procedure we use several of them in a clever way
- We show that affinity prediction benefits from *supervised multi-view* machine learning approaches

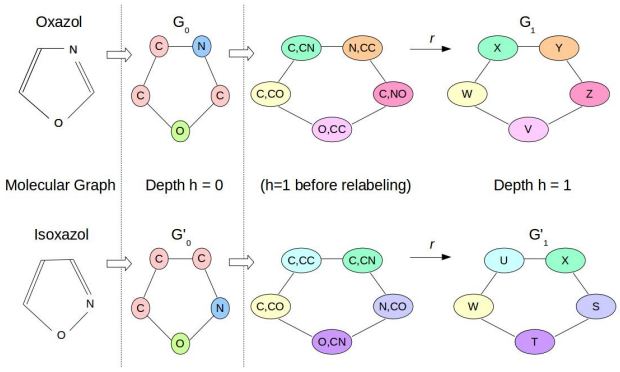
Graph Kernel Features – Two Examples



- Cyclic Patterns:
  - All simple cycles in the graph
- Tree Patterns:
  - Remaining trees after edges of cycles have been removed

...up to isomorphism

Graph Kernel Features – Weisfeiler-Lehman Labeling



Runs in iterations

- Vertices get relabeled based on their own and their neighbors labels
- A compression step is applied

We can combine WL and the previous graph kernel features

Combinatorial Explosion

- We now have available
  - Cyclic patterns ($\mathcal{C}$)
  - Tree patterns ($\mathcal{T}$)
  - Shortest path patterns ($\mathcal{P}$)
  - Vertex label patterns ($\mathcal{L}$)

  each for several iterations of the Weisfeiler-Lehman labeling
- This gives us 2^{4^h} possible ways of selecting a combination of these fingerprints for h iterations
- For only 2 iterations of WL, this results in 65536 possible combined fingerprints...

Approach (1/3): Weighted Concatenation of Views

- We consider multiple feature representations of molecular graphs
- The novel fingerprint should be a *weighted concatenation* of the single views
- In our setting, the *view* represents one of the pattern classes $\mathcal{C}$, $\mathcal{T}$, $\mathcal{P}$, or, $\mathcal{L}$ defined above for some WL iteration h

Approach (2/3): Multi-Pattern Kernel

- The linear approach can be generalized to *kernel functions* $k_v : \mathcal{G} \times \mathcal{G} \to \mathbb{R}$ and corresponding feature spaces (*representer theorem*)
- We define the *multi-pattern kernel* for some $h \in \mathbb{N}$

$$k_{MPK}(G, G') = \sum_{i=0}^{h} \sum_{v \in \{\mathcal{C}, \mathcal{T}, \mathcal{P}, \mathcal{L}\}} b_{vi} \cdot k_v(G_i, G'_i) \quad , \quad b_{vi} \in \mathbb{R}$$

for G_i, G'_i being the Weisfeiler-Lehman labeled graphs of depth i

Approach (3/3): Multi-View Learning

Simultaneous calculation of functions $f_1, \ldots, f_M$ and linear coefficients $b = (b_1, \ldots, b_M)$ via *multi-view learning* (MVL) utilizing training examples $(x_1, y_1), \ldots, (x_n, y_n)$ with

- ε-insensitive loss: *Multiple Kernel Learning* (MKL)

$$\underset{f_v, b_v \geq 0}{\operatorname{argmin}} \tfrac{1}{2} \sum_{v=1}^{M} \|f_v\|^2 + C \sum_{i=1}^{n} \max\{0, |f(x_i) - y_i| - \varepsilon\} + \tfrac{\Lambda}{2} \|b\|_p^2]$$

- squared loss: *Learning Kernel Ridge Regression* (LKRR)

$$\underset{f_v, b_v \geq 0}{\operatorname{argmin}} \sum_{v=1}^{M} \|f_v\|^2 + C \sum_{i=1}^{n} |f(x_i) - y_i|^2 \quad , \quad \text{s.t.} \|b - b_0\| \leq \Lambda$$

Datasets

- 20 datasets, each corresponds to a human protein
- each set comprises of 90 to 986 ligands with affinity annotations for the respective protein (pK_d-values)
- representation formats for ligands
  - standard molecular fingerprints MACCS and ECFP6
  - all graph pattern feature representations $\mathcal{C}, \mathcal{T}, \mathcal{P}$, and, $\mathcal{L}$
    ... based on all Weisfeiler-Lehman iterations $i \in \{0, \ldots, 6\}$
    ... in binary and counting version

Experimental Settings

We investigate the ε-insensitive and squared loss scenario for regression

ε-insensitive loss:

- single-view baseline: SVR
- multi-view approach: MKL
- SMO-MKL software

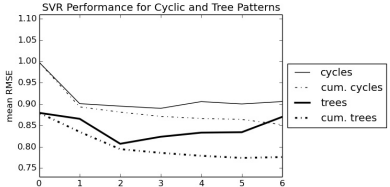
squared loss:

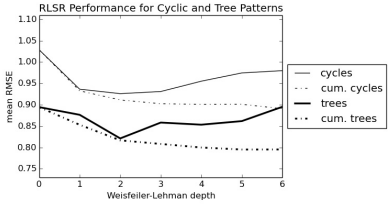
- single-view baseline: RLSR
- multi-view approach: LKRR
- own implementation

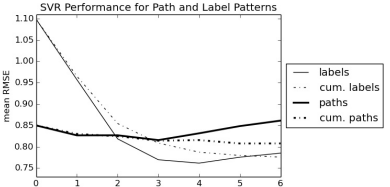
Preliminary experiments: using the single-view approaches we search for optimal WL depths of cumulative and non-cumulative pattern feature vectors

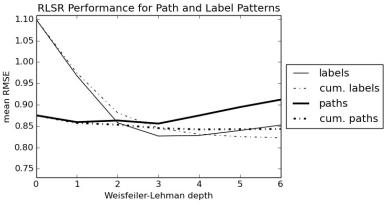
Main experiments: we compare single-view baselines applying standard fingerprints and graph kernels with multi-view approaches for optimal WL depths from the preliminary experiments

SVR and RLSR results for the counting version

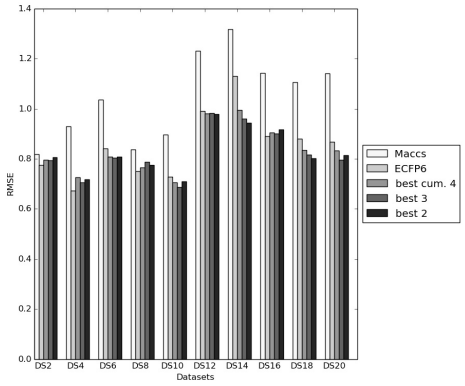


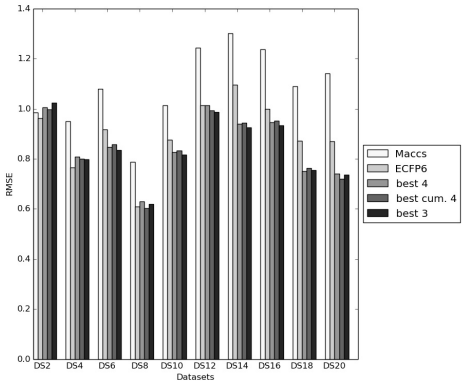






MKL and LKRR results for the counting version





Conclusion

- We described a way to leverage the variety of available molecular fingerprints for ligand affinity prediction
  - profit from different information
  - while managing the combinatorial complexity
- As a result, we found that a combination of fingerprints outperform single view methods