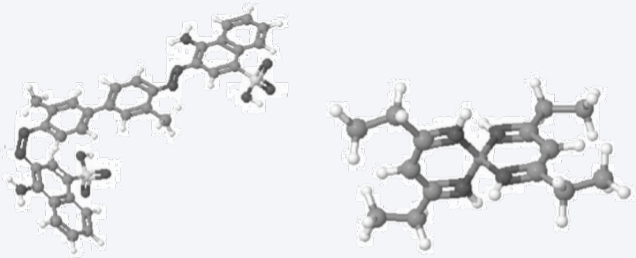
Task 5: Molecules

AIDS Antiviral Screen Database of Active Compounds



(a) Active (b) Inactive

250 training, 250 validation molecules

- Two classes active ‘a’ and inactive ‘i’
- Annotated in `train.txt` and `valid.txt`

Task

- Classify the molecules of the validation set using KNN
  Distance: approximate Graph Edit Distance (GED)

Input: Graph xml (gxl files)

XMLs with a lot of information

→ we only need nodes labeled with their chemical `symbol` and the unlabeled, undirected edges:

```
<node id="_1"><attr name="symbol"><string>C</string></attr>
…
<edge from="_1" to="_2">
…
```

Further info about the database (AIDS):

See riesen08graphdb.pdf → section 2.8 (there are also images of molecules)

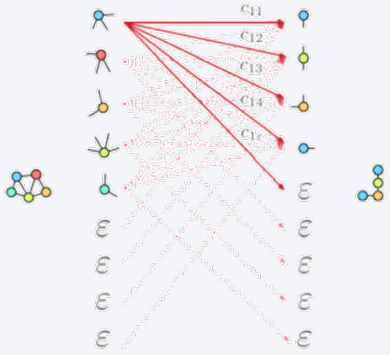
Hint: there are python libraries to parse XML files

Task 5: Molecules

Compute approximate GED between pairs of molecules with **bipartite graph matching**
(lecture 10, slide 21)

Build cost matrix (*Dirac*)

$$\mathbf{C} = \left[\begin{array}{cccc|cccc} c_{11} & c_{12} & \cdots & c_{1m} & c_{1\varepsilon} & \infty & \cdots & \infty \\ c_{21} & c_{22} & \cdots & c_{2m} & \infty & c_{2\varepsilon} & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \ddots & \infty \\ c_{n1} & c_{n2} & \cdots & c_{nm} & \infty & \cdots & \infty & c_{n\varepsilon} \\ \hline c_{\varepsilon 1} & \infty & \cdots & \infty & 0 & 0 & \cdots & 0 \\ \infty & c_{\varepsilon 2} & \ddots & \vdots & 0 & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \infty & \vdots & \ddots & \ddots & 0 \\ \infty & \cdots & \infty & c_{\varepsilon m} & 0 & \cdots & 0 & 0 \end{array}\right]$$



Hungarian Algorithm
To find optimal assignment

Derive Edit Path costs from the result
(distance for classification)

KNN for classification (optimize for `K`)

Task 5: Molecules

Recommendation

Use *Dirac* cost function for GED (optimize `Cn` and `Ce`)
(lecture 9, slide 36)

- Node substitution: `2*Cn` if symbols ≠, 0 otherwise
- Node deletion/insertion: `Cn`
- Edge deletion/insertion: `Ce`

Use an existing framework for the Hungarian algorithm