Data Mining 2: Frequent Pattern Mining — Summer 2017
Christian Borgelt — from 04.07.2017

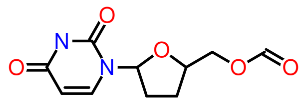
Exercise Sheet 5

Exercise 27 — Molecular Description Languages

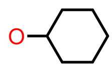
a) Draw the molecule that is described by the following SMILES string:
`S2c1c4c(ccc1N(c3c2cccc3)C(=O)C)cccc4`

b) Draw the molecule that is described by the following SLN string:
`NH2C[1]:N:C(:C(:N:C:@1C(=O)OCH3)C(=O)C[2]:CH:CH:CH:CH:CH:@2)NH2`

c) Construct a SMILES description of this molecule:



d) Find at least three different SMILES descriptions of phenol, that is, of this molecule:



Exercise 28 — Molecular Description Languages

Draw the molecule that is described by the following SDfile:

```
4728
          1016 111283D                             4728

  9  9        0                  1 V2000
   -0.0965    1.3884    0.0104 N   0  0  0  0  0  0  0  0  0  0
    1.8297   -0.2821   -0.0166 S   0  0  0  0  0  0  0  0  0  0
    1.0302    2.1371    0.0038 N   0  0  0  0  0  0  0  0  0  0
    0.1607    0.0664    0.0009 C   0  0  0  0  0  0  0  0  0  0
    2.1519    1.3924   -0.0108 C   0  0  0  0  0  0  0  0  0  0
    3.3971    1.9238   -0.0196 N   0  0  0  0  0  0  0  0  0  0
   -0.5873   -0.6698    0.0040 H   0  0  0  0  0  0  0  0  0  0
    3.5072    2.8871   -0.0156 H   0  0  0  0  0  0  0  0  0  0
    4.1746    1.3453   -0.0300 H   0  0  0  0  0  0  0  0  0  0
  1  4  2  0  0  0  0
  1  3  1  0  0  0  0
  2  4  1  0  0  0  0
  2  5  1  0  0  0  0
  3  5  2  0  0  0  0
  4  7  1  0  0  0  0
  5  6  1  0  0  0  0
  6  8  1  0  0  0  0
  6  9  1  0  0  0  0
M  END
$$$$
```

Exercise 29 — Subgraph Isomorphisms

a) How often does the fragment `C-C-C` occur in phenol (see Exercise 27d)?
In other words: How many subgraph isomorphisms exist?

b) Does phenol (see Exercise 27d) possess an automorphism that is not the identity?
How many different automorphisms does phenol possess?

c) Construct the set of all connected subgraphs of phenol (see Exercise 27d)!

d) Find all bonds in phenol (see Exercise 27d) that are bridges!
Are any of these bridges proper bridges?

e) Find all bonds in the graph/molecule from Exercise 27c) that are bridges!
Which of these bridges are proper bridges?

Exercise 30 Searching for Frequent Subgraphs

a) Find all frequent (sub)graphs containing sulphur for a minimum support $s_{\text{min}} = 3$ in the graph database (SMILES format) shown on the right:

`CCS(O)(O)N`	`CCS(=N)N`
`CCS(O)(C)N`	`CS(=N)N`
`CS(O)(C)N`	`CS(=N)O`

b) Why is it more difficult to avoid redundant search when searching for frequent (sub)graphs than for frequent item sets? What are the main problems?

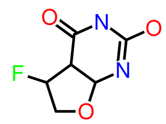
c) What is the purpose of constructing code words of (sub)graphs?
What is a canonical code word? What do we actually need?

d) How do we assign a unique parent (sub)graph based on a canonical code words for a (sub)graph? What information from the canonical code word do we use?

Exercise 31 Searching for Frequent Subgraphs

a) Why do we exclude the removal of proper bridges when assigning unique parents?

b) How many different spanning trees does the graph / molecule shown on the right possess? Justify your answer!



c) Why is the number of possible code words usually (much) larger than the number of spanning trees? What additional choices does one have?

d) Why does a coding scheme based on spanning trees in which edges closing cycles are listed after the spanning tree edges ensure that we can always the last edge? (In other words: why can we spare ourselves the check that this edge is not a proper bridge? What must the last edge rather be?)

e) How many different (extended) adjacency matrices does the graph/molecule in part b) of this exercise possess?

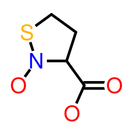
Exercise 32 Canonical Code Words

a) Check whether the code word

`S 10-N 21-O 31-C 43-C 54-O 64=O 73-C 87-C 80-C`

is the canonical code word, based on a depth-first search spanning tree, for this graph/molecule:

Use the order `S` $\prec$ `N` $\prec$ `O` $\prec$ `C` for the atoms and the order `-` $\prec$ `=` for the bonds!



b) Check whether the code word

`S 0-N1 0-C2 1-O3 1-C4 2-C5 4-C5 4-C6 6-O7 6=O8`

is the canonical code word, based on a breadth-first search spanning tree, for the graph/molecule of part a)! Use the same orders as in part a)!