Imago: open-source toolkit for 2D chemical structure image recognition

Viktor Smolov[*], Fedor Zentsev and Mikhail Rybalkin

GGA Software Services LLC

Abstract

Different chemical databases contain molecule structures with links to articles and patents, where such molecules are presented as images. Keeping such a database in a consistent state is a challenging problem, and, thus, efficient and accurate molecule image recognition algorithms are very important for solving this task.

We present an open-source toolkit for 2D chemical structure image recognition, called Imago. The main aim of this paper is to describe the algorithm approach implemented in Imago, and to share our ideas of possible improvements in the algorithm after we have summarized the results from the participation in the Image2Structure task at TREC-CHEM 2011.

1 Introduction

Significant amount of information about chemical structure properties can be found in the scientific literature. On the one hand, there are efficient modern full text search techniques that provide the ability to search for text over a huge amount of text data. For example, the biggest database, that provides a chemical-specific text search (with automatic synonyms detection) over more than 21 million citations for biomedical literature, is the PubMed database [1]. On the other hand, there are efficient methods, that are used to search for a chemical structure with required properties (that contains special query substructure) over o huge amount of molecule structures. For example, another popular database, that provides (sub)structure search functionality over more than 31 million chemical molecules, is the PubChem database [2]. But chemical articles contains both text and molecule structure images; we can only imagine what opportunities would we get by combining text data mining methods and cheminformatics search techniques.

This search problem is very similar to the so called Deep Web indexing problem: indexing the web content, that is not suitable for indexing by standard search engines. For example, Deep Web includes image and video files. According to rough estimates Deep Web is much larger than the web content, indexed by search engines.

Development of a universal chemical search engine, that could search by both text and (sub)structures, is a challenging problem. This problem involves subproblems from different scientific areas, which include natural language processing to find the meaning of a sentence describing the interaction between different structures, and optical image recognition to find

[*] Corresponding author. *E-mail address*: vsmolov@ggasoftware.com, smolovv@gmail.com

Pseudocode 1 Overall recognition procedure

```
Recognize(I)
    ▷ $I$ — depiction of a structure, $M$ — reconstructed molecule
    $M = \varnothing$

    ▷ Simple filtering steps (blur, binarization)
    ProcessFilters(I)

    ▷ Finding the molecule-like part of the image.
    ▷ All further steps use $I_M$.
    $I_M$ = SuperSegmentation(I)

    $S$ = Segmentation($I_M$)

    ▷ Locating single down stereo bonds. All corresponding segments are removed from $S$.
    $M \leftarrow$ ExtractSingleDownBonds(S)

    ▷ All image segments in $S$ are separated in two lists: symbols ($L_s$) and graphics ($L_g$).
    $L_s, L_g \leftarrow$ Separation(S)

    ▷ All symbols combine into groups (atom labels, superatoms). No recognition is done yet.
    $M \leftarrow$ GroupLabels($L_s$)

    ▷ Detecting line segments, aromatic circles in $L_g$. Constructing molecule skeleton.
    $M \leftarrow$ ExtractGraph($L_g$)

    ▷ Finding single-up bonds in $I_M$ and marking corresponding bonds in the molecule.
    $M \leftarrow$ FetchSingleUpBonds($I_M$)

    AttachAndRecognizeLabels(M)

    ▷ Repairing or figuring out stereo bonds orientation and aromatizing molecule if circles were presented.
    $M \leftarrow$ FixStereoCenters()
    Aromatize(M)
```

molecule structure by its image from an article. This paper is focused on the problem of recognizing a 2D chemical structure image and transforming this molecule image into a diagram graph structure, which can be used in different software applications, including search systems.

In this paper, we presents the description of the algorithm used in our open-source toolkit for 2D chemical structure image recognition — Imago. When we started to develop our chemical structure recognition toolkit, one of the main technical goals was to provide an open-source cross-platform library without any dependency on the third-party code in order to use this library in a wide range of applications, including mobile devices. The core part of Imago is written from scratch in modern C++ and does not use any third-party code. Our objective is to use the best known algorithms for optical recognition so that to guarantee Imago's outstanding portability and performance. Imago contains a GUI program and a command-line utility, as well as a documented API for developers.

2 Basic Recognition Routine

The current version of Imago implements a rather straightforward workflow with the following steps:

1. Binarize the whole image with a predefined threshold value.
2. Segmentate (divide image into the connected components) using the two steps below:
   (a) Perform global segmentation in the SuperSegmentation procedure: locate the actual chemical structure in the image by using 15 × 15 window to find the pixel's neighbors.

   (b) Apply the standard (with 3 × 3 window) segmentation to the largest segment from the step a), where the structure is assumed to be depicted.
3. Find and remove the single down (dashed triangle) stereo bonds in order to prevent incorrect recognition of that small line segments in the classification procedure.
4. The next and the most significant step is to classify the list of image segments into the two layers: symbols layer with the atom labels, and graphics layer with the bonds. After the separation, all symbols are grouped into atom labels and superatoms. All the graphics segments from the graphics layer are used to construct a molecule graph, called skeleton. Single up (solid triangle) stereo bond are processed after the skeleton is constructed.
5. For all atoms and superatoms, determine the appropriate skeleton nodes and conducts the optical character recognition. The resulting structure can now be saved in the MDL Molfile format.

The pseudocode for the whole algorithm is listed in Pseudocode 1.

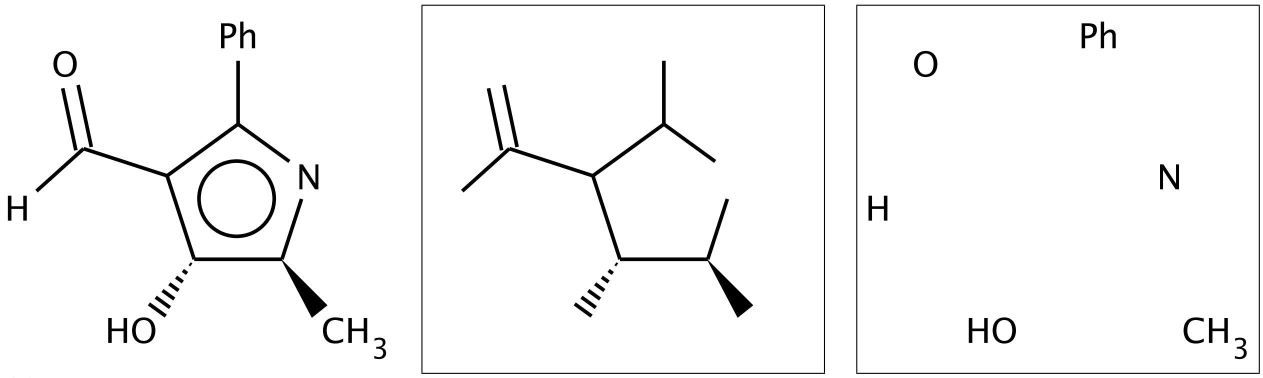
3 Detailed Description

In this section, we provide a detailed description of each step, used in our molecule image-to-structure recognition algorithm.

3.1 Separation

Separation of symbols and graphics is based on the approximate value of the capital letter height. If this value is found, then all segments are classified by this height with checking some exceptional cases, such as small single bonds. But the calculation of capital letter height is not a trivial procedure, if the image contains both bonds and text.

To solve this problem, we use a criterion based on the following approach: symbols form a set of image segments with the same height and with the aspect ratio (width to height ratio) in the experimentally-calculated range [MinSymRatio, MaxSymRatio]. Unfortunately, not only the symbols satisfy this criterion, but also some image segments, which contain only one line segment (e.g. part of the double bond). But such image segments with only one single line are checked with a trivial algorithm. Fig. 1 presents a sample molecule with 2 separated layers.



(a) Depicted molecule with up and down stereo bonds, and with pseudoatoms

(b) Graphics layer. Aromatic ring is excluded

(c) Symbols layer

Figure 1: Example of a molecule

This procedure of the symbols height estimation is listed in Pseudocode 2.

Pseudocode 2 Average symbols height estimation

```
EstimateCapHeight(L)
▷ $L$ – list of image segments

▷ Group segments by height, allowing some small deviation
$L_g$ = GroupByHeight(L)

repeat
        $l_g$ = LargestGroup($L_g$)
        ▷ Check if there is an image segment in the group, which contains single line segment
        if CheckSequence($l_g$)
            then $cap\_height$ = InterQuartileMeanHeight($l_g$)
                return
        Delete($l_g$)
    until true
```

3.2 Single down bonds

According to the IUPAC recommendations, a single down bond is a set of k ($k \geq 3$) parallel and equidistant to each other single line segments.

Pseudocode 3 contains the AddSingleDownBond procedure that calculates the coordinates of the bond ends and also the bond's orientation. If single-down bond was depicted not as a dashed triangle, but as a dashed rectangle, then the FixStereoCenters procedure (not included in this paper) will determine the orientation of this bond on a later stage of the algorithm execution.

3.3 Molecule skeleton construction

The goal of this step is to transform all raster graphics information into its vector representation. This operation is split into three substeps:

1. **Thinning.** All graphic elements are thinned using neighborhood maps [8], so that the thickness of any line in the image equals to 1.

Pseudocode 3 Single down bonds recognition

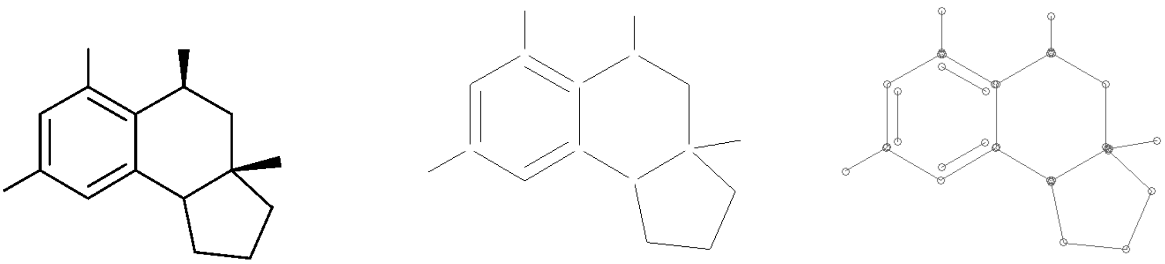
```
ExtractSingleDownBonds($S$)
▷ $Q$ — list of single line segments
$Q = \varnothing$

▷ $S$ — list of image segments
for $s \in S$
      do if IsSingleEdge($s$)
            then $Q \leftarrow s$

for $a \in Q$
      do $L_a = \emptyset$
         for $b \in Q, b \neq a$
              do if $a \parallel b$
                    then $L_a \leftarrow b$

         ▷ Now $L_a$ contains all segments, parallel to $a$. Need to check distances.
         if CheckDistance($L_a$)
            then AddSingleDownBond($L_a$)
                 DeleteSegments($L_a$)
```

2. **Removing crossings.** Each black pixel with more than 2 black pixels in its 8-neighbors becomes white. After this step, only polylines are present on the image.
3. **Vectorization.** Each polyline is smoothed using the Douglas–Peucker algorithm [5].



(a) Example of the graphics layer (b) Thinned image after step 2 (c) Vectorized image

Figure 2: Skeleton construction

After these 3 substeps, all vectorized line segments form an Euclidean graph – the molecule skeleton. The Imago algorithm substantially modifies this graph in the following manner: first, the close vertices are merged, and then parallel and also close to each other edges are replaced by molecule bonds with the corresponding bond order. The atom and bond closeness criterion depends on the average bond length.

Sometimes, when the molecule is poorly drawn, thick straight lines might become wavy after thinning. The vectorization will then give an excessive and wavy result. It this case, Imago conducts one additional modification step on the skeleton: some edges can be removed or shrunk depending on their length, lengths of adjacent vertices, and angles between these vertices and the bond.

3.4 Superatoms and character recognition

To group symbols (image segments) into superatoms, we build a relative neighborhood graph [4] of segments and then remove edges that are longer than the space size from this graph. All connected components in the graph are considered as superatom groups.

For character recognition, we use a simple classification procedure based on Fourier descriptors [3, 7] of symbols' outer and inner contours. Also, because we start with the grouped symbols, some heuristics can be used to predict the next character and to improve the recognition rate.

3.5 Single up bonds

To find single up bonds in the image, we exploit one side effect of thinning – solid triangles are thinned to the line segments (see Fig. 2b). So, for each single bond in the molecule skeleton, we check whether it was a triangle after thinning. The code for this operation is listed in Pseudocode 4.

Pseudocode 4 Single up bonds recognition

```
FetchSingleUpBonds(I, M)
▷ $I$ – image of the structure, $M$ – skeleton

for B $\in$ M.bonds
    do if (B.type = SINGLE) $\wedge$ (IsTriangle(I, b.begin, b.end))
        then B.type $\leftarrow$ SINGLE-UP
return $M$
```

4 Abbreviation Expansion

Usually, the depicted molecule images in the articles contain abbreviations for the well-known substructures: COOH, CN, Ph, etc. But chemical databases should contain only atoms without any abbreviations. To solve the problem of abbreviation expansion, Imago contains a list with the common abbreviations and expanded structures in SMILES notation. After recognizing the image, Imago replaces such superatoms with the expanded structures. To find positions of the expanded atoms, Imago uses the algorithm of 2D coordinates generation from the open-source organic chemistry toolkit Indigo [9] developed by GGA Software. This algorithm provides functionality to find the position of the set of atoms without changing positions of other atoms.

5 Discussion

One of the aim of the Image2Structure task at TREC-CHEM 2011 is to share the ideas about 2D chemical structure image recognition. In this section we will summarize pros and cons of our implementation, and will share our ideas on how the image recognition can be done in more efficient way.

The Imago algorithm described above is straightforward, which leads to an efficient fast implementation. If the structure was rendered according to the common rules, and if the image resolution is high, then the result structure is correct in the most cases. But if there is a lot of noise in the image, or if the resolution is low, then such straightforward implementation gives incorrect results. The following situations should be taken into account for developing an advanced chemical structure image recognition algorithm:

1. When the image resolution is low, then it is common that atom label(s) are not separated from the bonds. Our separation method does not work well in such case.
2. When there are few symbols in the image, the average bond length can be calculated with large error. This may lead to the situation when specific atom symbols are treated as a bonds chain.
3. If the atom label contains more than one symbol, these symbols can be rendered without space pixels between then. Our implementation also does not give good results in this case.

We also believe that the image recognition program should output a degree of confidence of the image being recognized correctly. If the recognition program outputs only the recognized molecule, then we have to check all molecules manually to ensure that recognition was correct. But if the program outputs the degree of confidence, then we have to check only images with low level of confidence. The presence of such an essential feature would be a basis for automatic batch image recognition in a production environment. How to estimate this level of confidence is an open question though. This question is very similar to the problem of verifying the results provided by an arbitrary chemical structure image recognition program.

Considering all the problems listed above, and all our experience in molecule image recognition, we believe, that the following ideas and techniques should be used for designing an advanced recognition algorithm:

1. The overall recognition procedure should be designed as an optimization procedure to maximize the level of confidence that the recognized structure is correct.
2. The whole image should be divided into primitives, and each primitive or set of primitives should be mapped onto a set of chemical primitives with some probability. For example, a Iodine atom can be mapped to an atom with probability 80%, and to a single bond with probability 20%, depending on the relative size of the primitive. Or, a set of atoms can be represented as a superatom with some probability.
3. Some probabilistic criteria should be developed to estimate the validity of the recognized structure.
4. When all the primitives are collected, the most well-recognized ones should be processed first. The first step in our current implementation is separation of the image into bonds and atoms layers based on labels height, but it seems that such separation should be done by splitting bonds first.
5. The current implementation of Imago has strict straightforward structure: filtration, segmentation, and recognition. These steps should be dependent, and if some set of parameters leads to low level of confidence, then this set of parameters should be adjusted. For example, the recognition process can be organized as a multi-pass workflow, where each iteration attempts to adjust parameters in order to maximize the confidence level.

These points relate only to the image recognition algorithm, but when dealing with chemical articles and patents, the recognition problem becomes much more complicated: it is very important to separate the molecule images from the text. This problem becomes very complicated for recognizing combinatorial libraries, where a single combinatorial library is represented as separate molecules with captions.

6 Conclusions

We presented the description of the algorithm implemented in our open-source toolkit for 2D chemical structure image recognition — Imago, licensed under GPLv3. It has both GUI application, that can used to open images and to extract molecules from images, and a command-line utility to process molecule images in a batch mode. The open-source library has a well documented API for developers.

Automatic image-to-structure conversion tools are very important for the automatic retrieval of chemical information. Huge amount of chemical papers and patents contains information that can be used in the next-generation search databases and expert systems. But development of a such search system is a challenging problem. To solve it, methods from the different areas should be used, including natural language processing, text searching, and optical character recognition. GGA's Imago implementation is a step towards achieving this goal.

References

[1] PubMed database
`http://www.ncbi.nlm.nih.gov/pubmed/`

[2] PubChem database
`http://pubchem.ncbi.nlm.nih.gov/`

[3] Ř Due Trier, AK Jain, T Tax, *Feature extraction methods for character recognition - a survey*, Pattern recognition, 1996

[4] G. T. Toussaint, *The relative neighborhood graph of a finite planar set*, Pattern Recognition 12 (4): 261–268

[5] D. Douglas, T. Peucker, *Algorithms for the reduction of the number of points required to represent a digitized line or its caricature*, The Canadian Cartographer 10(2), 112–122

[6] *Graphical Representation Standards For Chemical Structure Diagrams (IUPAC Recommendations 2008)*
`http://www.iupac.org/publications/pac/80/2/0277/`

[7] Charles T. Zahn & Ralph Z. Roskies, *Fourier Descriptors for Plane Closed Curves*, IEEE Transactions on computers, Vol. c-21, No. 3, march 1972

[8] Joseph M. Cychosz, *"Efficient Binary Image Thinning using Neighborhood Maps"* Graphics Gems IV, Academic Press, 1994

[9] Indigo: Universal cheminformatics toolkit
`http://ggasoftware.com/opensource/indigo`