

## Supporting Information

### Break Down in Order to Build Up: Decomposing Small Molecules for Fragment-Based Drug

#### Design with eMolFrag

Tairan Liu<sup>1,§</sup> Misagh Naderi<sup>2,§</sup> Chris Alvin<sup>3</sup>, Supratik Mukhopadhyay<sup>4</sup>, and Michal Brylinski<sup>2,5,\*</sup>

<sup>1</sup>Department of Mechanical Engineering, Louisiana State University, Baton Rouge, LA, 70803, USA

<sup>2</sup>Department of Biological Sciences, Louisiana State University, Baton Rouge, LA, 70803, USA

<sup>3</sup>Department of Computer Science and Information Systems, Bradley University, Peoria, IL, 61625, USA

<sup>4</sup>Department of Computer Science, Louisiana State University, Baton Rouge, LA, 70803, USA

<sup>5</sup>Center for Computation & Technology, Louisiana State University, Baton Rouge, LA, 70803, USA

<sup>§</sup> These authors contributed equally to this work

\* To whom correspondence should be addressed. Michal Brylinski, phone: +1-225-5782791; email: [michal@brylinski.org](mailto:michal@brylinski.org)

#### Content:

##### 1. Algorithms

- **Algorithm S1.** Molecular fragmentation.
- **Algorithm S2.** Linker extraction.
- **Algorithm S3.** Removal of redundant fragments.

##### 2. Tables

- **Table S1.** Computing speed of eMolFrag and molBLOCKS.

##### 3. Examples

- **Example S1.** Brick in SDF format.
- **Example S2.** Linker in SDF format.

**Algorithm S1.** Molecular fragmentation.

```
1: procedure Fragment(Set<Molecule>  $M$ )
2:   List<Brick>  $B := \emptyset$ 
3:   List<Linker>  $L := \emptyset$ 
4:   for each  $m \in M$ 
5:     for each  $f \in \text{FragmentOnBRICSBonds}(m)$ 
6:       if  $f.\text{isBrick}()$  then
7:          $I := f.\text{RemoveDummyAtoms}()$ 
8:          $f.\text{RemoveHydrogen}()$ 
9:          $f.\text{AddAppendix}(I)$ 
10:         $B_m := B_m \cup \{f\}$ 
11:       end if
12:     end for
13:     $B := B \cup B_m$ 
14:     $L := L \cup \text{ComputeLinkers}(m, B_m)$ 
15:  end for
16:  return  $\langle B, L \rangle$ 
17: end procedure
```

**Algorithm S2.** Linker extraction.

```
1: procedure ComputeLinkers(Molecule  $m$ , List<Brick>  $B_m$ )  
2:   for each  $b \in B_m$   
3:      $m.removeBrick(b)$   
4:   end for  
5:   List<Linker>  $\ell := m.RemainingFragments()$   
6:   for each  $l \in \ell$   
7:      $l.AddAppendix(m)$   
8:   end for  
9:   return  $\ell$   
10: end procedure
```

**Algorithm S3.** Removal of redundant fragments.

```
1: procedure RemoveRedundancy(List<Fragment>  $F$ )
2:   List<Fragment>  $U := \emptyset$  // Unique fragment set
3:   List<Set<Fragment>>  $\mathcal{P}/\sim := \text{Partition}(F)$ 
4:   for each  $P \in \mathcal{P}/\sim$ 
5:     while  $P \neq \emptyset$ 
6:        $f_0 := P.\text{removeFirst}()$ 
7:        $U := U \cup \{f_0\}$ 
8:       for each  $f \in P$ 
9:         if  $f_0 = f$  then  $P := P \setminus \{f\}$ 
10:      end for
11:    end while
12:  end for
13:  return  $U$ 
14: end procedure
15: procedure RemoveRedundancy(List<Brick>  $\mathcal{B}$ , List<Linker>  $\mathcal{L}$ )
16:   $\mathcal{B} := \text{RemoveRedundancy}(\mathcal{B})$ 
17:   $\mathcal{L} := \text{RemoveRedundancy}(\mathcal{L})$ 
18: end procedure
```

**Table S1.** Computing speed of eMolFrag and molBLOCKS measured by the number of compounds fragmented per second. Serial and parallel versions of eMolFrag and a serial version of molBLOCKS were tested against several datasets whose size ranges from 100 to 12,800 molecules randomly selected from the DUD-E library.

| Dataset size <sup>a</sup> | eMolFrag <sup>b</sup> |                              | eMolFrag (Part I) <sup>c</sup> |                              | molBLOCKS<br>ext. <sup>e</sup> |
|---------------------------|-----------------------|------------------------------|--------------------------------|------------------------------|--------------------------------|
|                           | <i>Serial</i>         | <i>Parallel</i> <sup>d</sup> | <i>Serial</i>                  | <i>Parallel</i> <sup>d</sup> |                                |
| 100                       | 8.7                   | 24.0                         | 9.8                            | 34.6                         | 6.6                            |
| 200                       | 8.3                   | 22.2                         | 11.3                           | 52.9                         | 8.7                            |
| 400                       | 7.7                   | 20.3                         | 12.9                           | 69.2                         | 10.9                           |
| 800                       | 7.3                   | 17.7                         | 22.6                           | 66.7                         | 11.8                           |
| 1,600                     | 6.8                   | 16.8                         | 19.6                           | 62.9                         | 12.2                           |
| 3,200                     | 6.4                   | 15.4                         | 21.8                           | 59.6                         | 14.6                           |
| 6,400                     | 6.0                   | 13.9                         | 23.9                           | 49.0                         | 13.0                           |
| 12,800                    | 4.8                   | 11.8                         | 23.2                           | 61.1                         | 12.5                           |

<sup>a</sup> Number of input molecules.

<sup>b</sup> Full eMolFrag including fragmentation (Part I) and removing redundancy (Part II).

<sup>c</sup> eMolFrag including fragmentation (Part I) only.

<sup>d</sup> Executed on 16 computing cores.

<sup>e</sup> Executed in the “extensive fragmentation” mode with a minimum fragment size of 4 atoms.

### Example S1. Brick in SDF format.

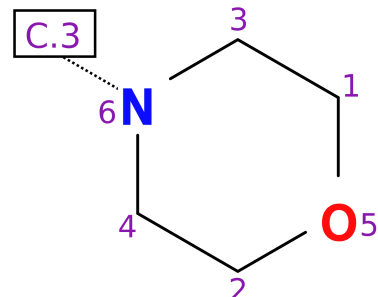
```
b-CHEMBL175476.mol2-000.sdf
RDKit          3D
```

```
6 6 0 0 0 0 0 0 0 0999 V2000
  1.2268 -10.0020 -0.0554 C  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 -0.4759 -10.3733  1.5879 C  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  0.8909  -8.5086 -0.0838 C  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 -0.8291  -8.8837  1.5761 C  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  0.0595 -10.7412  0.3138 O  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  0.3682  -8.1042  1.2294 N  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  1  3  1  0
  1  5  1  0
  2  4  1  0
  2  5  1  0
  3  6  1  0
  4  6  1  0
M  END

> <ATOMTYPES>
C.3
C.3
C.3
C.3
O.3
N.3

> <BRANCH @atom-number eligible-atmtype-to-connect>
6 C.3

> <fragments similar>
/tmp/michal/R0G1XHYZUN-17313/output/output-chop-comb/b-test1.mol2-000.sdf
$$$$
```



The auxiliary information included in brick SDF files:

<ATOMTYPES> Atom types according to SYBYL ordered according to the atom section containing Cartesian coordinates.

<BRANCH @atom-number eligible-atmtype-to-connect> List of all possible bonds for this brick. The 1<sup>st</sup> column is the atom index followed by atom types allowed to be connected at this position. For example, the 6<sup>th</sup> atom in the brick fragment shown above, which is N.3, can connect to a C.3 atom.

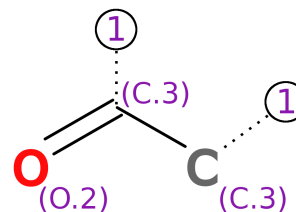
<fragments similar> After removing redundancy, only one construct is kept for each unique fragment. This section tracks back all similar fragments that have been consolidated.

### Example S2. Linker in SDF format.

```
l-test1.mol2-001.sdf
      RDKit          3D
```

```
  3  2  0  0  0  0  0  0  0  0  0999 V2000
    2.4295   -6.4901    1.0459 O   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
    0.0858   -6.6629    1.2625 C   0  0  0  0  0  0  0  0  0  0  0  0  0  0
    1.3774   -5.8952    1.1464 C   0  0  0  0  0  0  0  0  0  0  0  0  0  0
  1  3  2  0
  2  3  1  0
M  END
```

```
> <MAX-NUMBER-Of-CONTACTS ATOMTYPES>
0 O.2
1 C.3
1 C.2
$$$$
```



The auxiliary information included in linker SDF files:

< MAX-NUMBER-Of-CONTACTS ATOMTYPES> The 1<sup>st</sup> column shows the maximum number of connections allowed at every atom, which are ordered according to the atom section containing Cartesian coordinates. Atom types are listed in the 2<sup>nd</sup> column. For example, the second line in this section in the linker fragment shown above (1 C.3) means that the 2<sup>nd</sup> atom is C.3 and it can form only one connection to other atoms.