Supporting information

A Robotics-Inspired Screening Algorithm for Molecular Caging Prediction

Oleksandr Kravchenko[1,*], Anastasiia Varava[2,*], Florian T. Pokorny[2], Didier Devaurs[3], Lydia E. Kavraki[4], and Danica Kragic[2]

[1]Department of Chemistry, School of Engineering Sciences in Chemistry, Biology and Health (CBH), KTH Royal Institute of Technology, 11428 Stockholm, Sweden

[2]Division of Robotics, Perception and Learning (RPL), School of Electrical Engineering and Computer Science (EECS), KTH Royal Institute of Technology, 10044 Stockholm, Sweden

[3]Univ. Grenoble Alpes, CNRS, Inria, Grenoble INP, LJK, 38000 Grenoble, France

[4]Department of Computer Science, Rice University, Houston, TX 77005, USA

[*]These authors contributed equally

1 Implementation details

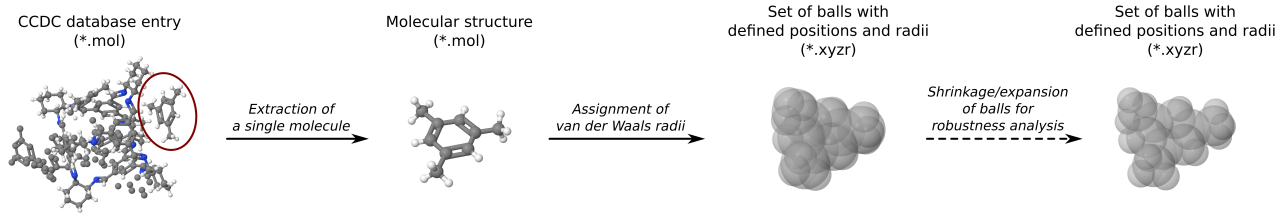

Figure S1: Preparation of input geometries for the algorithm from a publicly available CCDC database entry.

1.1 Proposed algorithm

The typical workflow for input preparation in our algorithm is shown in Figure S1. The obtained models in XYZR format (each line contains X, Y, Z coordinates of the ball center and its radius) can be directly used in the algorithm. In addition to two molecular models, the algorithm requires a discretization of the $SO(3)$ orientation space [1, 2] which defines the precision (ε-core) and computational time of the algorithm:

Input: models of the host and guest (*.xyzr);
discretization of SO(3) orientation space (set of intervals).

Output: number of bounded connected components of the free space of the guest.

In our experiments, we always use the same discretization of $SO(3)$, which was pre-computed using the algorithm proposed by Yershova et al. [1]. It is a uniform grid over $SO(3)$ that consists of 36,864 points; our configuration space approximation is thus based on 36,864 orientation slices.

If the algorithm reports 0 bounded connected components (BCCs), then the free space of the guest consists of only one unbounded connected component meaning that the guest is not caged by the host. Otherwise, a positive number of BCCs means that the guest is caged. Moreover, the existence of several BCCs indicates that the motion of the guest inside the host cavity is restricted.

For experimental purposes, algorithms were implemented in C++11, with the aid of the Computational Geometry Algorithms Library (CGAL) [3].

1.2 Other algorithms

1.2.1 pyWINDOW

The open-source Python package pyWINDOW (whose version 0.0.2 is publicly available at `https://github.com/marcinmiklitz/pywindow`) was used for the determination of pore-limiting diameters (PLDs) of single molecules [4]. The primary method `calculate_windows()` was used with parameters `processes=8` (parallelization on 8 CPUs) and `adjust=N`, where $N = 1, 5, 25$ for 250, 1250, and 6250 sample points on a sampling sphere, respectively. The best approximation of each window diameter was calculated as the minimum of the values obtained using $N = 1, 5, 25$ and various orientations of the input geometry. Calculations of circumcircle radii were performed using the built-in method `_circumcircle()` with manually selected atom sets.

1.2.2 Zeo++

The open-source C++ package Zeo++ (whose version 0.3 is publicly available at `http://www.zeoplusplus.org/download.html` upon registration) was used to determine pore diameters in crystal structures. Crystallographic Information Files (CIF) from corresponding CCDC entries (see Fig. S2) were used as input. In particular, the routine `network` with parameter `-ha` for improved accuracy was invoked.

2 Algorithm runtimes

All computations were performed on a desktop computer with Intel Core i7-4790K CPU and 32 GB RAM.

Table S1: Comparison of algorithm runtimes when determining the PLDs of **CC3** and analyzing whether Xe and Kr are caged (average over 515 conformations). pyWINDOW used parallel computations on 8 CPUs within a single run of the algorithm. The determination of PLD was performed by enumerating a spherical guest radius from 0.0 Å to a circumsphere radius of the host with 0.005 Å step (multiple runs of the algorithm were parallelized on 8 CPUs to ensure fair comparison).

Experiment	pyWINDOW			This work
	N = 250[a]	N = 1250	N = 6250	
PLD determination	846 ms[b]	2916 ms	11123 ms	405 ms
Analysis of Xe/Kr caging				1.1 ms

[a] Number of points on a sampling sphere (see Sec. 1.2.1). Running the algorithm with N = 250 did not produce correct results in many cases. Therefore increasing N up to 6250 was required for the comparison with our algorithm.

[b] Algorithms like pyWINDOW do not allow detecting caging complexes directly; they require the determination of PLD followed by a comparison with the diameter of a spherical guest; therefore runtimes are the same for both experiments.

Table S2: Runtimes of our algorithm during the analysis of the host **CC3** and guests **Mes**, **mX**, **4ET** (parallelized on 8 CPUs). For each run, a minimum, maximum, and an average (over 515 conformations) time is specified.

Guest	Time per run, s		
	minimum	average	maximum
Mes	271.5	488.8	741.7
mX	253.0	453.6	704.4
4ET	358.0	570.4	750.2

3 Structures

3.1 Hosts

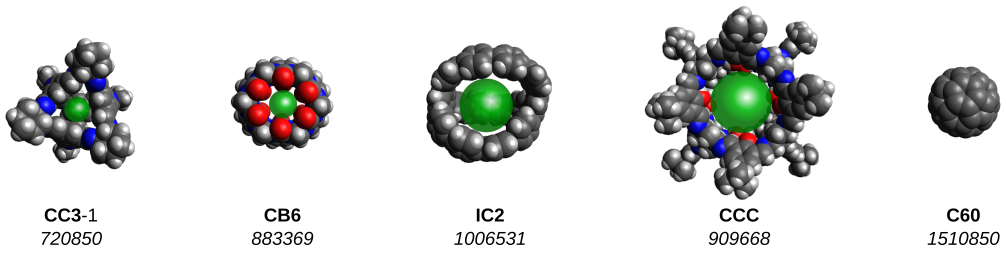

Figure S2: Hosts selected for PLD determination. Each entry presents a structure, an abbreviation given in the reference paper (**CCC** was used in lieu of abbreviation for chiral covalent cage [5]), and the identifier of the corresponding CCDC database entry. Each structure is accompanied by a green sphere with radius equal to PLD.

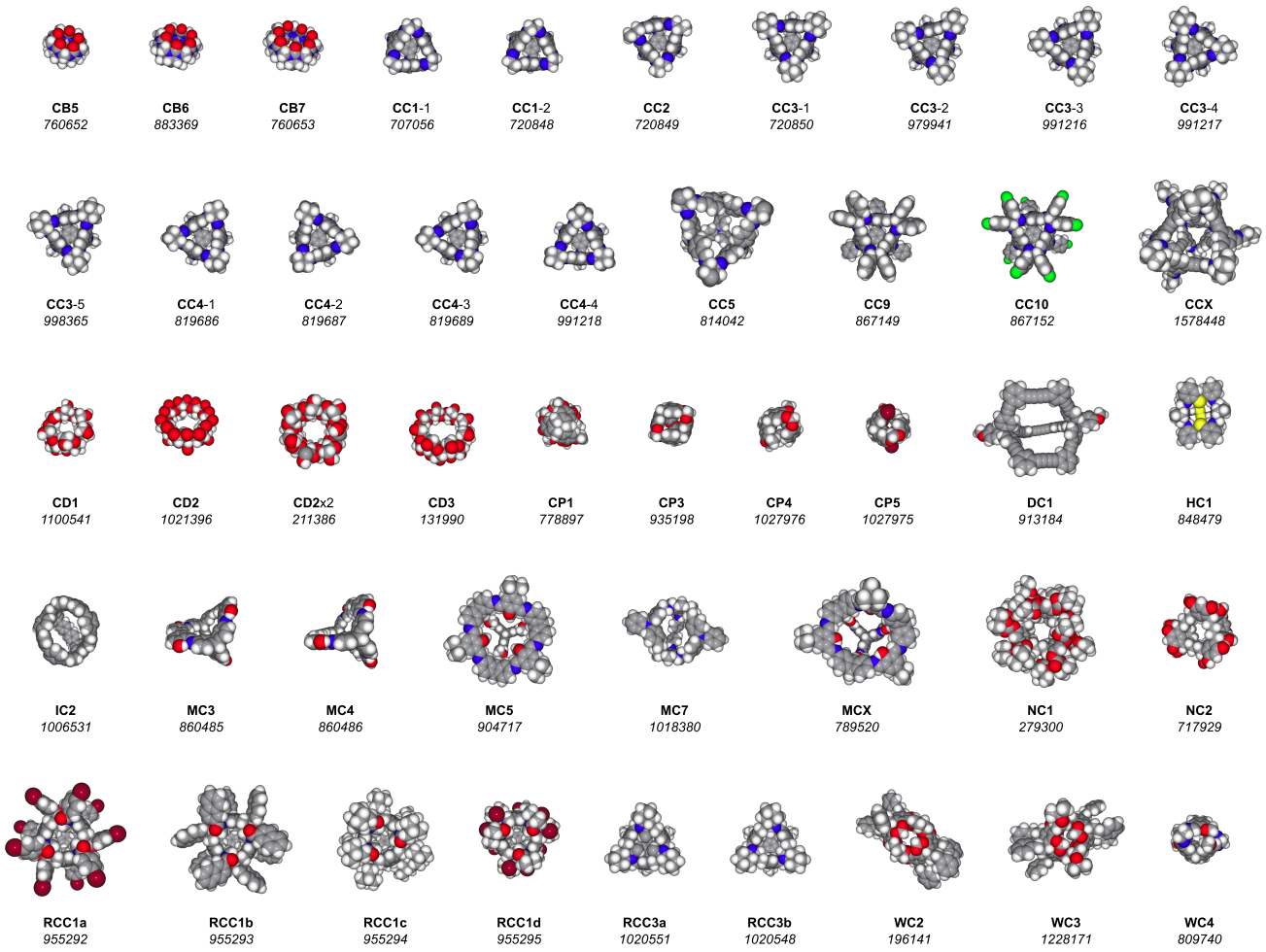

Figure S3: Hosts selected for the screening experiment. Each entry presents a structure, an abbreviation given in the reference paper (number replaced with X if the name was missing [6, 7]), and the identifier of the corresponding CCDC database entry. Detailed structures as well as references to the first reports of host molecules can be found in the work by Miklitz et al. [8]

3.2 Guests

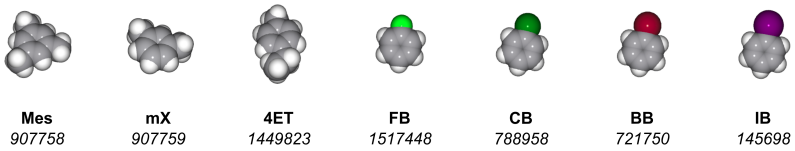

Figure S4: Guests used in the experiments. Each entry presents a structure, an abbreviation given in the reference paper or present work, and the identifier of the corresponding CCDC database entry.

3.3 Conformations

CC3 conformations were obtained from the published molecular dynamics trajectory [4]. Molecular dynamics trajectories for the guest molecules were generated using the Gromacs software [9]. Gromacs input was prepared using ACPYPE [10] and AMBER force field [11]. Simulations were run in the NVT ensemble using the modified Berendsen thermostat [12] set at temperature 300 K. 1 fs step was used in all simulations, with equilibration run of 10 ps, followed by a production run of 10 ns (snapshot taken every 1 ps). Root mean square displacement of atoms between conformations was calculated either with the built-in Gromacs RMSD routine `gmx rms`, or with the open-source RMSD utility (publicly available at `https://github.com/charnley/rmsd`). RMSD values for all guests did not exceed 0.05 Å (maximum value for **IB** is 0.05 Å); the RMSD value for **CC3** was determined to be 0.25 Å.

4 Screening results

	CB5	CB6	CB7	CC1-1	CC1-2	CC2	CC3-1	CC3-2	CC3-3	CC3-4	CC3-5	CC4-1	CC4-2	CC4-3	CC4-4	CC5
FB	n	s	n	n	n	n	s	w	w	w	s	w	w	w	w	n
CB	n	s	n	n	n	n	w	n	w	w	w	n	n	n	n	n
BB	n	s	n	n	n	n	w	w	w	w	w	n	n	n	n	n
IB	n	s	n	n	n	n	w	w	w	w	w	n	n	n	n	n

	CC9	CC10	CCX	CD1	CD2	CD2x2	CD3	CP1	CP3	CP4	CP5	DC1	HC1	IC2	MC3
FB	w	n	n	n	n	n	n	n	n	n	n	n	w	n	n
CB	w	n	n	n	n	n	n	n	n	n	n	n	w	n	n
BB	w	n	n	n	n	n	n	n	n	n	n	n	n	n	n
IB	w	n	n	n	n	n	n	n	n	n	n	n	n	n	n

	MC4	MC5	MC7	MCX	NC1	NC2	RCC1a	RCC1b	RCC1c	RCC1d	RCC3a	RCC3b	WC2	WC3	WC4
FB	n	n	n	n	s	w	w	w	n	n	n	s	s	s	s
CB	n	n	n	n	w	w	n	n	n	n	n	s	s	s	w
BB	n	n	n	n	w	w	n	n	n	n	n	s	s	s	w
IB	n	n	n	n	w	w	n	n	n	n	n	s	s	s	n

Figure S5: Results of the screening of 184 host-guest pairs: s – strong caging complex, w – weak caging complex, n – not a caging complex. **FB** – fluorobenzene, **CB** – chlorobenzene, **BB** – bromobenzene, **IB** – iodobenzene.

References

[1] Yershova, A.; Jain, S.; Lavalle, S. M.; Mitchell, J. C. *The International journal of robotics research* **2010**, *29*, 801–812.

[2] Varava, A.; Carvalho, J. F.; Pokorny, F. T.; Kragic, D. Free Space of Rigid Objects: Caging, Path Non-Existence, and Narrow Passage Detection. The Workshop on the Algorithmic Foundations of Robotics. 2018.

[3] The CGAL Project, *CGAL User and Reference Manual*, 4.7 ed.; CGAL Editorial Board, 2015.

[4] Miklitz, M.; Jelfs, K. E. *J. Chem. Inf. Model.* **2018**, *58*, 2387–2391.

[5] Skowronek, P.; Warżajtis, B.; Rychlewska, U.; Gawroński, J. *Chem. Commun.* **2013**, *49*, 2524–2526.

[6] Pugh, C. J.; Santolini, V.; Greenaway, R. L.; Little, M. A.; Briggs, M. E.; Jelfs, K. E.; Cooper, A. I. *Cryst. Growth Des.* **2018**, *18*, 2759–2764.

[7] Mastalerz, M.; Schneider, M. W.; Oppel, I. M.; Presly, O. *Angew. Chem., Int. Ed.* **2011**, *50*, 1046–1051.

[8] Miklitz, M.; Jiang, S.; Clowes, R.; Briggs, M. E.; Cooper, A. I.; Jelfs, K. E. *J. Phys. Chem. C* **2017**, *121*, 15211–15222.

[9] Abraham, M. J.; Murtola, T.; Schulz, R.; Páll, S.; Smith, J. C.; Hess, B.; Lindahl, E. *SoftwareX* **2015**, *1-2*, 19–25.

[10] Sousa da Silva, A. W.; Vranken, W. F. *BMC Research Notes* **2012**, *5*, 367.

[11] Case, D. et al. AMBER. 2018.

[12] Bussi, G.; Donadio, D.; Parrinello, M. *J. Chem. Phys.* **2007**, *126*, 014101.