



HAL
open science

SOFI: Finding point group symmetries in atomic clusters as finding the set of degenerate solutions in a shape-matching problem

M. Gunde, N. Salles, L. Grisanti, L. Martin-Samos, Anne Hémercyck

► **To cite this version:**

M. Gunde, N. Salles, L. Grisanti, L. Martin-Samos, Anne Hémercyck. SOFI: Finding point group symmetries in atomic clusters as finding the set of degenerate solutions in a shape-matching problem. *The Journal of Chemical Physics*, 2024, 161 (6), 10.1063/5.0215689 . hal-04677499

HAL Id: hal-04677499

<https://laas.hal.science/hal-04677499v1>

Submitted on 26 Aug 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

SOFI: Finding point group symmetries in atomic clusters as finding the set of degenerate solutions in a shape-matching problem

M. Gunde,^{1, a)} N. Salles,² L. Grisanti,^{1,2} L. Martin-Samos,² and A. Hemeryck³

¹⁾*Institute Ruder Bošković, Bijenička 54, 10000 Zagreb, Croatia*

²⁾*CNR-IOM/Democritos National Simulation Center, Istituto Officina dei Materiali, c/o SISSA, via Bonomea 265, IT-34136 Trieste, Italy*

³⁾*LAAS-CNRS, Université de Toulouse, CNRS, 7 avenue du Colonel Roche, 31000 Toulouse, France*

Point Group (PG) symmetries play a fundamental role in many aspects of theoretical chemistry and computational materials science. With the objective to automatize the search of PG symmetry operations of generic atomic clusters, we present a new algorithm called Symmetry Operation FINDER (SOFI). SOFI addresses the problem of identifying PG symmetry by framing it as a degenerate shape-matching problem, where the multiple solutions correspond to distinct symmetry operations. The developed algorithm is compared against three other algorithms dedicated to PG identification, on a large set of atomic clusters. The results, along with some illustrative use cases, showcase the effectiveness of SOFI. The SOFI algorithm is released as part of the IRA library, accessible at <https://github.com/mammasmias/IterativeRotationsAssignments>.

I. INTRODUCTION

In the rise of quantum-chemistry and first-principle methods, the identification of Point Group (PG) symmetry has been a necessary step to effectively downsize the numerical problem and reduce the cost of simulations. Besides the computational accelerations made possible by exploiting the symmetry, the presence and identification of a PG symmetry in a molecule, or rather a cluster, or a defect, can be used to address the fundamental properties of its electronic ground, and excited states¹. Assigning symmetry label to each state (i.e. vibrational or electronic), is desirable to connect to group theory, which is the formal context where PG symmetries are described. If the symmetry is assigned to all relevant quantum (eigen)states, group theory can be applied to specific quantum-mechanical operators to understand which matrix elements are expected to vanish based on symmetry rules. This is extremely useful for example in molecular spectroscopy: depending on the operator governing the interaction term, the symmetry-based *selection rules* will provide information on which transitions will actually be forbidden^{2,3}.

As the identification of PG symmetry is not a new problem, the computational community has made numerous attempts to automatize it. Typically, quantum chemistry software and some molecular builders/editors include capabilities related to the symmetries of the system (i.e. automatic identification, symmetrization of structure). The algorithmic approaches to identify the PG may vary based on how the atomic structures are handled, and the way symmetry elements are utilised/exploited within the software framework. Despite the fundamental nature of PG symmetries and the broad use of related algorithms, there seem to be relatively few standalone libraries devoted to their identifica-

tion. Moreover, it is often very difficult to trace which PG identification algorithm (and/or library) is integrated into the larger software environment, as the information is often poorly, or not at all documented. Many well-known quantum chemistry packages already date back to several years ago, when the modularity in programming was not necessarily a priority, so each developing group has created an "ad-hoc" method plugged into the code, often poorly updated through the years. For example by inspecting NWchem source code⁴ one can find a collection of old Fortran routines apparently written in the 90s, used for identifying symmetries. Besides, proprietary codes do not necessarily share information about their algorithm choice, even if some initiatives are more transparent, for example the IQmol builder⁵, which uses, and references the SymMol⁶ algorithm in its documentation.

In the subsequent paragraph we highlight a handful of approaches for the identification of PG symmetry, which have been devised as standalone algorithms/libraries, with some of them presently integrated into larger quantum chemistry codes, and/or computational "toolboxes".

The PG symmetry of a structure can be identified in several ways. A relatively common strategy, used in several available approaches, is to first find the principal axis of symmetry, and then test for possible operations along other axes in relation to the principal axis. The principal axes could be identified straight-forwardly by relying on the properties of the principal moments and axes of the inertia tensor, as done in SymMol⁶, and Symmetrizer⁷ software. Relying on the inertia tensor can however be problematic, since its principal axes can be ambiguous in some specific cases (spheroidal shapes). Alternatively, certain properties of the atomic structure itself (atomic positions, midpoints between atoms, etc.) can be used to identify the axes of symmetry, such as employed in SYMMAPS⁸, and SYVA⁹ software. Another idea is to partition the atomic structure into sets of symmetry-equivalent atoms, and search for symmetries within those

^{a)}Electronic mail: miha.gunde@gmail.com

sets. The partitioning can be done through the inspection of the distance matrix of the structure, such as in libmsym¹⁰, while the search for symmetries is conducted based on the properties of the inertia tensor of each equivalence set¹¹, and the PG of the entire structure is the subset of the symmetry operations common to all partitions. Some approaches based on arguments from graph theory also exist^{12,13}.

It is also common for PG-identification software to implement some other capabilities, such as precise symmetrization of structures^{7,9}, or similar, which can be thought of as extensions to the fundamental task of identifying the PG symmetry elements of a structure.

In the interest of our discussion and the algorithm we propose, we now briefly recall some fundamental notions of PG symmetry. If θ is a specific symmetry operation, which is an element of the PG of an atomic structure A , the equation:

$$A' \leftarrow \theta A, \quad (1)$$

represents the application of symmetry operation θ on structure A , which results in a symmetry-transformed structure A' . By definition of symmetry operations, A and A' are equivalent up to a permutation of atomic indices P_A :

$$A' = P_A A, \quad (2)$$

where P_A is a permutation matrix, which is unique for a given operation θ . Then, the relation:

$$P_A A = \theta A. \quad (3)$$

holds for any and every symmetry operation θ of the structure A . Each element (symmetry operation) of the PG of structure A corresponds to a unique solution of Eq. (3), in the form of a pair $\{\theta, P_A\}$.

It can be observed that the Eq. (3) is equivalent to the shape matching problem of two exactly congruent (equivalent) structures A and B , as:

$$P_B B = \mathbf{R} A + \mathbf{t}, \quad (4)$$

where \mathbf{R} is an orthonormal transformation matrix, and \mathbf{t} is a translation vector. By comparing Eq. (3) to Eq. (4), we see that \mathbf{R} is replaced by a symmetry operation θ , and the translation \mathbf{t} vanishes. When solving the general shape matching problem of Eq. (4), only one solution (the optimal one) is sought, whereas in the case of PG identification in Eq. (3), the set of all degenerate solutions represents the set of all possible symmetry operations of structure A . Finding the PG of a structure can therefore be addressed as a "degenerate" shape matching problem.

In practice, the "degeneracy" of these solutions is never exact, due to numerical limitations, or distortions in the atomic positions within the structure. An interpretation of "degeneracy" is thus needed, in terms of a threshold, such that all solutions within a given threshold are considered equivalent, degenerate. By the tuning of such

threshold, the notion of inexact symmetries can be covered.

In this work, we present the Symmetry Operation FINDER (SOFI) algorithm, and its implementation as a standalone library, aimed at solving the general problem of PG symmetry identification, in a way that is independent of specific properties of the atomic structure. SOFI finds the PG symmetry operations in the form of 3×3 matrices, and the corresponding permutations of atomic indices. The working principle of SOFI is fundamentally based on ideas and arguments of the shape-matching algorithm IRA^{14,15}. In contrast to other algorithms from the literature mentioned earlier, SOFI does not attempt to identify any special axes of symmetry, but is instead focused directly on generating the possible symmetry operations along *any* axis. SOFI takes as input an atomic structure, and returns the list of symmetry operation matrices, written in the specific form of the original reference frame, in which the structure was input/written. As such, the matrices can directly be used without any post processing. From this point of view, SOFI is a *solver* for symmetry elements, that does not modify the input structure in any way. In order to determine the PG of a structure, a separate routine is called, taking as input the list of symmetry operation matrices. Within this routine, each symmetry operation is first associated to the axis it acts on, and labelled according to the Schoenflies notation³. The PG is then deduced by inspection of the symmetry elements, following a standard flowchart. The search for non-exact symmetries in SOFI can be tuned via a threshold parameter, whose value is directly linked to the distortions in the atomic positions of the structure. Moreover, in case of an incomplete list of elements of a PG, basic arguments of group theory can be used to find the missing elements. An incomplete list of symmetry elements can be an indicator of broken symmetries in the system, which might happen when the atomic positions get distorted in a particular way. Due to the modularity of the algorithm implementation, external conditions acting on symmetry elements can easily be implemented through custom functions. As an example of such function, the action of an external arbitrary-direction magnetic field on the PG of a structure¹⁶ is included in the code. SOFI is written in Fortran, with a C-bound Application Programming Interface (API), making it a directly interoperable library. It can be used on its own, or easily inserted into other software.

Section II describes the details of our SOFI algorithm, and the Section III gives the implementation details. In the Section IV, SOFI algorithm is compared to three other algorithms for PG symmetry identification, namely SYVA⁹, SymMol⁶, and libmsym¹⁰. The discussion in Section V provides some possibly valuable insight into technical details of the algorithm, and the limits and tuning of its maximal "resolving" power. The Section VI gives some exemplary use cases of the SOFI algorithm, and its output. Further examples can be found in the online documentation.

II. THE ALGORITHM

We approach the problem of finding PG symmetries by solving the degenerate shape-matching problem of Eq. (3), but firstly we need to address the question of the origin point.

A fixed point is a point in space which remains unchanged under transformation by an arbitrary set of orthonormal matrices. When an atomic structure is transformed by symmetry operations belonging to some PG, we can observe one or more such fixed points. The points that remain fixed under transformation by all symmetry elements of the PG are the origin points of the entire PG, while points that are fixed only for a subset of the symmetry elements are origin points of the subgroups associated to that PG. For example, the *Oh* PG has six *C4v* subgroups oriented in three different directions, each with a different origin point, while in the case of *C*-groups, all points along the primary rotation axis are fixed. However, the only point of a generic structure, which is guaranteed to always remain fixed under transformation by any orthonormal matrix or symmetry element, is the geometric center (center of geometry, arithmetic mean, centroid). Therefore, picking the geometric center of a structure as the origin point, guarantees that the algorithm will parse the full PG of the structure. Indeed, most of the available algorithms assume this choice by default. However, alternative origin points are possible. For example when searching for symmetry operations about a given atom, the origin point should be the position of that specific atom (or some other specific point of choice). SOFI is agnostic to the choice of origin, which means it will find symmetry operations about the given point. This also means the origin has to be chosen from the outside (by the user, or another application).

After the origin point has been selected and the structure shifted accordingly, the atomic structure enters the main search loop of the SOFI algorithm. The key idea of SOFI is to parse the space of orthonormal transformations, at each step construct a trial transformation, evaluate Eq. (3) with the trial transformation, and keep track of all favourable solutions. Thus, the algorithm needs two main components, as follows.

The first component is a function which will evaluate Eq. (3). The idea employed in SOFI is to transform Eq. (3) into a distance $d(P_A A, \theta A)$, which should be equal to zero when P_A and θ correspond to a solution of Eq. (3), and greater than zero otherwise. In order to compute the distance $d(P_A A, \theta A)$, we employ the Hausdorff distance function^{14,15,17} with the generic form:

$$d_H(A, B) = \max_{a \in A} \min_{b \in B} d(a, b) \quad (5)$$

where A and B are structures with atomic vectors $a \in A$, $b \in B$, and $d(a, b)$ is a Cartesian distance. The Hausdorff distance is invariant to permutations, and variant to other rigid transformations of either structure (rotation, reflection, translation). The evaluation of Eq. (3) is

thus done by computing Eq. (6).

$$d(P_A A, \theta A) = d_H(A, \theta A) \quad (6)$$

If θ is a solution of Eq. (3), then $d_H(A, \theta A) = 0$. In order to compute d_H in SOFI, we use the CShDA algorithm, introduced in Ref.¹⁴. CShDA imposes a one-to-one assignment constraint on the atoms, and returns the permutation P_A and distance d_H .

The second component of the algorithm is a way of parsing the space of orthonormal transformations (3×3 orthonormal matrices θ). This parsing should be computationally efficient, but still broad enough to identify all possible solutions of Eq. (3). In SOFI, we define the parsing of transformation space in a similar way as in IRA¹⁴, in which the orthonormal transformations (bases) are constructed directly from atomic vectors present in the structure. It follows from Eq. (3), that for each atom $i \in A$ with the atomic position a_i , there exists a unique atom $j \in A$ with the atomic position a_j , such that:

$$a_i = \theta a_j. \quad (7)$$

The Eq. (7) can be seen as operation of change of basis, from the basis β defined by a_i , to basis γ defined by a_j , written:

$$\beta a_i = \gamma a_j, \quad (8)$$

from which we see that

$$\theta = \beta^T \gamma. \quad (9)$$

Therefore, the operation θ is completely defined by two orthonormal bases, namely β and γ , which are defined by atomic positions a_i and a_j , respectively.

However, an orthonormal basis set in 3D cannot be uniquely defined from a single 3D vector. This problem is resolved by adding another atomic vector $a_{i'}$ to the definition of the basis, as follows. The first basis vector \hat{e}_1 is the normalised atomic vector a_i , the second basis vector \hat{e}_2 is the component of the atomic vector $a_{i'}$ orthonormal to \hat{e}_1 , and the third basis vector is a cross product $\hat{e}_3 = \hat{e}_1 \times \hat{e}_2$. In this way, the basis β is defined by positions of atoms i and i' , and the basis γ by positions of atoms j and j' . Due to this, SOFI is restricted to structures containing at least 3 non-collinear atoms. The same procedure for constructing an orthonormal basis is used in IRA¹⁴.

According to the preceding considerations, in order to identify any symmetry operation θ , we need to find the quadruplet of atoms $\{i, i', j, j'\}$, which construct θ by Eq. (9), such that Eq. (3) is satisfied. In order to find all possible solutions of Eq. (3), we need in turn to list all unique matrices θ that can be constructed by such quadruplets of atoms, while also satisfying $d_H(A, \theta A) = 0$ by Eq. (5). The most naive algorithm to find all symmetry operations θ is then to brute-force search over all combinations of quadruplets of atoms in A , which give $d_H = 0$. Such algorithm is constructed by four nested

DO loops over all atoms, meaning an $\mathcal{O}(N^4)$ complexity, where N is the number of atoms in the structure. We can however quickly observe that such approach contains a huge degree of redundancy, which can be greatly reduced, or even avoided, by applying the following considerations.

A. Reduce complexity I: fix one basis

The Eq. (7) holds true for a given θ , for all atoms i simultaneously. Therefore, instead of looping over all i , i' , j , and j' , we can simply select any two different atoms i and i' , and loop over only j and j' . The atoms i and i' construct the basis β , which is kept fixed throughout the search, while the basis γ constructed by atoms j and j' keeps changing, effectively constructing different trial θ operations. The argument also holds for all possible operations θ , since each true symmetry operation θ is associated to a single permutation P_A . Thus the complexity of the search algorithm is reduced to $\mathcal{O}(N^2)$.

B. Reduce complexity II: preserve distances and atomic types

A PG symmetry operation θ is a unitary transformation, meaning it preserves distances. Thus, an atom after the transformation cannot have a position which is at a different distance from the origin than it was before the transformation. Moreover, upon transformation by θ , the position of atom i is effectively replaced by the atom j , which must have the same properties as atom i . Hence atom j should possess the same atomic type, and be positioned at the same distance from the origin, as atom i . This can be translated into a simple constraint for the choice of atom j – and also j' – namely the norms of atomic vectors (distances from the origin) must be equal, $\|a_i\| = \|a_j\|$ and $\|a_{i'}\| = \|a_{j'}\|$; and the atomic types of the pairs $\{i, j\}$ and $\{i', j'\}$ must be identical (see lines 4,5 and 7,8 of Alg. 1). Thus the complexity of the search algorithm is further reduced, to $\mathcal{O}(N_i N_{i'})$, where N_i and $N_{i'}$ are the number of atoms within the same radial distance and atomic type as atom i , and atom i' , respectively.

C. Reduce complexity III: pick atoms close to origin

The numbers N_i and $N_{i'}$ are further reduced by choosing i and i' which are as close as possible to the origin (avoiding the atoms at the origin, for numerical reasons), and are of the atomic type which has the least number of equivalent atoms in that radial shell. For close-packed structures (clusters), the worst-case scenario is a structure of a single atomic type, where N_i and $N_{i'}$ are 12 (hexagonal close-packed single-crystal structure), which

means 144 possible constructions of θ by the SOFI algorithm. For some types of structures however, these numbers can be a lot higher, the extreme case being fullerene-type structures where all the N atoms are at the same distance from the origin, $N_i = N_{i'} = N$. For those structures, the use of SOFI as described and implemented in the present paper is not efficient.

D. A word about (roto-)reflections

Notice that when the bases β and γ are generated with the procedure defined earlier, the matrix θ resulting from Eq. (9) is always such that its determinant is positive, $\det(\theta) = 1$. This means our search algorithm always produces trial transformations which correspond strictly to rotations. In order to mitigate this behaviour, we systematically flip each of the three vectors for each basis γ that is generated (see line 11 of Alg. 1). As such, the resulting trial transformations include proper rotations, reflections, and roto-reflections. This slightly raises the complexity of the algorithm by a factor of 4 (one for the original γ , and additional three for each flipped axis).

E. Inexact symmetries

A structure can exhibit inexact symmetries due to disorder or inaccuracies in atomic positions, which can be purely numerical, or due to physical reasons. In any case, this disorder can cause some symmetry elements to become inexact, *i.e.* the distance $d_H(A, \theta A)$ does not reach zero, but only a local minimum. For this reason we introduce the threshold `sym_thr` into SOFI, which modifies the condition for a transformation θ to be recognised as symmetry operation of a structure A into $d_H(A, \theta A) \leq \text{sym_thr}$, which is interpreted as the maximal deviation (in units of distance) between an atom of the non-transformed structure, and its associated atom of the transformed structure, must be below the threshold. The value of `sym_thr` is an input parameter to SOFI. For the same reason of possible disorder in the atomic structure, the transformations θ which are constructed from atomic vectors by Eq. (9), can potentially become inexact (in the sense of non-zero d_H). In order to obtain as precise transformation matrices θ as possible, we introduce the concept of matrix refinement, see Sec. III and Alg. 3.

It can still happen that SOFI's parsing procedure will not identify all the symmetry operations of a structure due to such atomic disorder. In that case, arguments of fundamental group theory can be used to generate the missing symmetry operations. The most important argument is the property of group closure, which states that if $\theta_i \in G$ and $\theta_j \in G$ then also the product $\theta_i * \theta_j \in G$, where G is the point group. Therefore, after the main parsing procedure of SOFI is performed, the list of found symmetry operations can be looped through itself, and

all possible products of the operations generated, then tested for the symmetry condition on the structure, and finally included in the set accordingly. Thus if any symmetry operation has been missed by the main search algorithm for any reason, it will be found and added by this (optional) procedure.

The value given to the threshold `sym_thr` is directly linked to the level of disorder in atomic positions on the structure. Its extremal value can ultimately be related to the distance between nearest atoms in the structure, in the sense that if the value `sym_thr` is close to, or larger than that, the algorithm will accept operations that effectively exchange the positions of atoms, as valid symmetry operations. In order to avoid that, the rule of thumb is that `sym_thr` can sensibly be set to any values up to about half the distance between nearest atoms, anything beyond that should be carefully tested, and will most probably produce results that will be hard to interpret.

III. IMPLEMENTATION

The main loop of the SOFI search procedure is schematised in Alg. 1, which follows the considerations from Sec. II. Each iteration of the main loop consists of constructing a new orthonormal basis γ , and with it a new trial operation θ' . Each trial θ' is passed to the routine `try_sofi` (schematised in Alg. 2). This routine first determines if the matrix is already a known symmetry operation, and if not, computes the first $d_H(A, \theta'A)$ via the CShDA algorithm. If the first d_H is small enough, the trial θ' is considered as potential symmetry operation, and passed to the `refine` routine (schematised in Alg. 3). The `refine` routine returns a refined trial θ' and the associated $d_H(A, \theta'A)$. If this distance d_H is below the threshold `sym_thr`, the trial θ' is stored as a new symmetry operation of A . At the end of the main loop, the list of symmetry operations θ of structure A is known.

A trial matrix transformation θ' , whose action on a structure A is sufficiently close to some true symmetry operation θ , can be identified, and *refined* to correspond to the true symmetry θ , see Alg. 3. The matrix θ' is said to be sufficiently close to θ , when the distance $d_H(A, \theta'A)$ is below a given parameter value, but not zero (this value in SOFI corresponds to $5 * \text{sym_thr}$, see line 4 of Alg. 2). In this case, the permutation of atoms P_A that is returned when computing $d_H(A, \theta'A)$ with CShDA, is not exact, but can be said to be partially correct, meaning that there are at least some atoms for which the assignments are correct. This partially correct assignment can be applied to the structure A , and used as input to the well-known SVD-based algorithm described in Ref.¹⁸, which returns an optimal rotation given a fixed permutation of atoms. The small rotation ν which is found by SVD is added to the original transformation, $\theta'' = \nu\theta'$, and applied to the structure, $\theta''A$. At this point, the distance $d_H(A, \theta''A)$ and the corresponding assignments

Algorithm 1: The main search loop of SOFI, including the considerations from Sec. II A-II D. The called routine `try_sofi` is schematised in Alg. 2.

Input: structure A , threshold `sym_thr`.
Output: List of symmetry operations θ .

```

1 Select  $i$  and  $i'$  close to origin;           !! Sec II C
2  $\beta \leftarrow$  OrthonormalBasis(  $i, i'$  );   !! Sec II A
3 for  $j \in 1, N$  do
4   if(  $\text{abs}(|a_i| - |a_j|) > \text{sym\_thr}$  )cycle; !! Sec II B
5   if(  $\text{typ}(i) \neq \text{typ}(j)$  )cycle;
6   for  $j' \in 1, N$  do
7     if(  $\text{abs}(|a_{i'}| - |a_{j'}|) > \text{sym\_thr}$  )cycle;
8     if(  $\text{typ}(i') \neq \text{typ}(j')$  )cycle;
9      $\gamma \leftarrow$  OrthonormalBasis(  $j, j'$  );
10    for  $m \in 0, 3$  do
11      flip axis  $m$  of  $\gamma$ ;           !! Sec II D
12       $\theta' = \beta^T \gamma$ ; call try_sofi( $A, \theta', \text{sym\_thr}$ );
13    end
14  end
15 end
```

Algorithm 2: The schema of routine `try_sofi`. The called routine `refine` is schematised in Alg. 3.

Input: structure A , 3×3 matrix θ' , threshold `sym_thr`.
Output: Updated list of symmetry operations θ .

```

1 Compare  $\theta'$  to current list of all  $\theta$ ;
2 if(  $\theta' \in$  list )return;
3 Compute  $d = d_H(A, \theta'A)$  via CShDA algorithm;
4 if(  $d > 5.0 * \text{sym\_thr}$  )return;
5 call refine( $A, \theta', d_H$ );
6 if(  $d_H \leq \text{sym\_thr}$  )add  $\theta'$  to list;
```

P_A'' are recomputed. If P_A'' contains permutations, the cycle of SVD-CShDA is repeated. Once the new assignment does not contain any further permutation, the procedure is stopped, the refined matrix is $\theta'' = \nu\theta'$, and if $d_H(A, \theta''A) \leq \text{sym_thr}$, then the refined matrix θ'' is a true symmetry operation θ . The process is schematised in Algorithm 3.

Algorithm 3: The schema of routine `refine`.

Input: structure A , trial matrix θ' .
Output: refined matrix θ' , distance d_H .

```

1  $\theta'' \leftarrow \theta'$ ;
2  $P_A, d_H \leftarrow$  CShDA( $A, \theta''A$ );
3 for  $n \in 1, \text{max\_iter}$  do
4    $\nu \leftarrow$  SVDrotation( $A, \theta''A$ );
5    $\theta'' \leftarrow \nu\theta''$ ;
6    $P_A, d_H \leftarrow$  CShDA( $A, \theta''A$ );
7   if (no permutation in  $P_A$ ): exit loop;
8 end
9 return  $\theta'' \leftarrow \theta'', d_H$ 
```

The self-sufficient iteration scheme of Algorithm 3 is similar to the algorithm Iterative Closest Point (ICP)¹⁹, which is very well known in the computer vision community. ICP has some known pitfalls, and several proposed solutions to them²⁰. The most notable being getting stuck in local minima, or oscillating between two equivalent solutions. It should be stated that these pitfalls can in principle occur in our matrix refinement approach as well, however, they occur in situations when $d_H(A, \theta' A)$ is far from zero, and above any meaningful threshold, thus the trial transformation θ' in such situations is far from a suitable candidate for our purposes, and it is not of concern. Moreover, the number of iterations in Algorithm 3 has a hard limit at `max_iter=3`.

In order to determine the PG from a list of symmetry operations in the form of 3×3 matrices, as returned by SOFI, we have implemented a decision process based on a flowchart available in physical chemistry textbooks³. The decision tree is based on the identification and classification of specific symmetry elements. For our purpose, the simplest way of carrying it out is to employ the Schoenflies notation for the symmetry elements, in the form of $Op\ n^p$, where Op is the name of the operation (C for rotation, S for rotoreflection, I for inversion, and E for identity), n is the order of rotation, and p the multiple of rotation, for example $S\ 12^5$ corresponds to a rotoreflection with an angle of $\frac{5}{12}2\pi$. The analysis of a 3×3 matrix, and assignment of its Schoenflies label is done as described in the Appendix A. Once the list of symmetry operations is converted to the Schoenflies notation, and grouped together by the axis they act on, the decision tree to determine the PG is applied.

The SOFI software library is organized as a series of independent routines. For performing a default (full) search for PG of a structure, there is a wrapper routine which includes calls to all the needed routines. This includes shifting the structure to its geometric center before entering SOFI, and using the closure property of the groups to fully identify all symmetry operations (see Sec. IIE). A workflow for more specialized tasks can be constructed as-desired from the routines included in the library.

The source code repository also includes an API to the main SOFI routines, which is bound to the C-namespaces via the `iso_c_binding` intrinsic Fortran module. The API is compiled into a dynamic library (file `.so`), and thus the interface routines can easily be linked and called from either Fortran or C, or imported into Python. Example programs for each of these languages are included in the repository. An appropriate Python module is also proposed, which acts as an interface to the C-bound routines in the dynamic library, through the `ctypes` Python module.

IV. COMPARISON TO OTHER ALGORITHMS

The SOFI algorithm has been compared to the SYVA⁹, SymMol⁶, and libmsym¹⁰ software. The full details and results of the tests are available in the Supplementary Materials, and the associated Zenodo repository²¹ which contains all the structures, software, and scripts used in the test, and all results in plain text format. The goal of this comparison is to showcase the behaviour of SOFI in comparison to other established software.

All of the atomic structures included in our comparison test were obtained from the Cambridge Cluster Database (CCD)²². They include atomic cluster structures, where we have excluded any structures with number of atoms less than 3. The total number of structures included in the test is thus 3858, with the number of atoms varying between 3 and 380. There is only one set of structures with more than one atomic type, which has two atomic types.

The comparison tests were carried out by computing the PG of each input structure, at different values of the symmetry threshold. Changing the value of this threshold can result in the respective algorithm detecting a different number of possible symmetry operations, and hence a different assignment of the PG. The SYVA and SymMol algorithms have a single adjustable parameter for the tolerance, which, to our understanding, is more or less equivalent to the symmetry threshold `sym_thr` used in SOFI. That is, it specifies the maximal allowed displacement of any atom of the symmetry-transformed structure, with respect to the original structure. Therefore, it is possible to directly compare the results of SYVA, SymMol, and SOFI algorithms at the same value of tolerance. On the other hand, the libmsym algorithm does not have a single-valued parameter for the tolerance, but rather a set of parameters which act upon different segments of the algorithm, and it is therefore not straight-forward to compare the results. Any computation where a respective algorithm outputs an explicit message of error or failure, or the software crashes with no output, is recorded as a failure.

The values for the tolerance used in SYVA, SymMol, and SOFI were $tol = 0.001$ and $tol = 0.05$, while for libmsym we used three sets of parameters which are also used in the Avogadro software²³, where they are labelled as "default", "medium", and "loose".

To the best of our knowledge in the CCD database, the source/algorithm which was used for obtaining the listed PGs is not clearly specified. This is an issue since the claimed PG cannot be actually tested or revised. In other words, the PG assignment by CCD cannot be assumed to have a benchmarking value.

The Fig. 1 gives a correlation plot of the PGs as assigned by the different algorithms included in the test, excluding CCD. This plot has been made by combining the results of each algorithm in the test, using the value of tolerance for which that algorithm produced the least failures. These are explicitly SYVA($tol = 0.001$),

SymMol($tol = 0.05$), libmsym($tol = "default"$), and SOFI($tol = 0.05$). A point on the diagonal of the plot represents PG assignments of SOFI equal to any of the other algorithms, while an off-diagonal point represents different assignment of PGs, for the same structure. The distance of a point from the diagonal could be seen as proportional to the extent of the difference in the assignment, in terms of number of symmetry operations of a group. Similar plots of this type are available in the Supplementary Materials, resolved per each algorithm and tolerance value included in the test.

We show in the Supplementary Materials, that SOFI is fully capable of computing the PG symmetry in atomic clusters. Contrary to other algorithms in the test, there are zero cases of failure in SOFI, among all the structures and tolerance values included in the test. In all cases of disagreement between SOFI and another algorithm, SOFI is capable of finding a PG of the same order, or higher, when computed with an appropriate tolerance value.

The small differences in SYVA, SymMol, and SOFI most probably arise due to the different ways in which algorithms find the symmetry operations, or their axes. A small difference in the axis on which a symmetry element acts, can lead to a small difference in the atomic positions of the symmetry-transformed structure, which leads to a different value of distance to the original structure. When such small discrepancies lead to distance values that are comparable to the value of symmetry threshold used, they can be sufficient to cause an algorithm to detect PGs of a different (lower) order. These small variations can be observed in our tests, where in some cases SOFI detects less symmetry operations than SYVA and SymMol, however upon slight increase in the threshold, the higher-order PGs are always retrieved.

A more in-depth analysis of all the results would require a detailed inspection of the participating algorithms, and a case-to-case analysis of each result, which is well beyond the scope of the present article. Besides, our tests did not investigate the impact of the symmetry threshold in a systematic way, which makes drawing any general conclusions from this comparison test nontrivial.

The Dzugutov-mod clusters dataset was used to give some indicative timing of the algorithms. It contains 248 structures with number of atoms from 3 to 250, all of the same atomic type. The structures where an algorithm was not able to determine a PG were excluded from this measurement, since they take a disproportionate amount of time. The SYVA algorithm took 155.8 s to determine 248 PGs, SymMol took 2.1 s to determine 245 PGs, libmsym took 0.8 s to determine 245 PGs, and SOFI took 1.2 s for 248 PGs. All the timing measurements were done on single Intel Core i7-10700K processor.

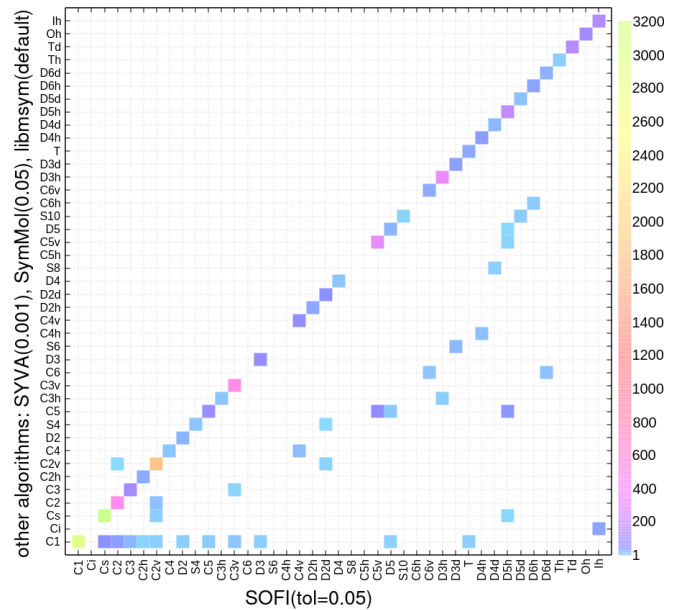


FIG. 1. Compact summary of the comparison test results: The horizontal axis represents PGs as assigned by SOFI, and the vertical axis PGs assigned by any other algorithms in the test (SYVA, SymMol, libmsym). Labels of PGs are sorted by the cardinality of the groups. The points below the diagonal represent cases where SOFI assigned a PG with higher order, and values above the diagonal represent cases where SOFI assigned a PG with lower order. The color of each point represents the number of structures where that particular combination of PG assignment happened.

V. DISCUSSION

Due to the SVD-based "matrix refinement" procedure introduced in SOFI (Sec. III, Alg. 3), the axis of a symmetry element is ultimately set such that the root-mean-square distance between the original, and the symmetry-transformed structure $d_{rmsd}(A, \theta A)$ is minimized. However it should be noted that, in general, the value of θ which minimizes a distance function D , in the sense of Eq. (10),

$$\arg \min_{\theta} \{D(\theta A, A)\} \quad (10)$$

can be slightly different for different distance functions D , such as the root-mean-square d_{rmsd} , and Hausdorff distance d_H , the latter of which is used to evaluate the operations θ in SOFI by Eq. (6). This is a small inconsistency in the SOFI algorithm, and it means the value of Eq. (6) can actually *increase* slightly after the matrix refinement procedure. We have found in our tests that this is the reason for SOFI to set the axis of some symmetry operation slightly different than the other algorithms, and that leads to rare cases when SOFI needs a slightly higher value of symmetry threshold, to obtain the same PG as SYVA or SymMol. In the set of structures used in our comparison test, the usefulness of the matrix refine-

ment procedure might not be apparent, however when SOFI is used on structures with more disorder in the atomic positions, the refinement procedure is key to the recognition of proper symmetry operations.

The slowest part of the SOFI algorithm is the evaluation of the distance $d_H(A, \theta'A)$ from Eq. (5), calculated by the CShDA algorithm, which is done for each trial θ' . It would make sense to spend some computational effort in trying to lower as much as possible the number of times CShDA is computed. This can be done by recognising when a trial θ' is very similar to some already known symmetry operation θ , meaning the trial θ' can be skipped, and thus avoid computing CShDA. In order to evaluate the similarity between two orthonormal matrices, we use a simple distance function $d(\theta^a, \theta^b)$, which is a sum of the distances between the matrix elements:

$$d(\theta^a, \theta^b) = \sqrt{\sum_{i,j} (\theta_{ij}^a - \theta_{ij}^b)^2} \quad (11)$$

which can be seen as a measure of the order of rotation needed to transform θ^a into θ^b . By default, SOFI considers two orthonormal matrices, namely θ' and θ , as equal when the order of rotation needed to transform one into the other is more than 200, that is an operation C200 in the Schoenflies notation, or rotation by angle $0.005 \cdot 2\pi$. This sets the maximal limit to the resolving power of SOFI, meaning that any symmetry operation of order higher than 200 can not be recognised. This default behaviour can be changed by tuning of some internal parameters, and recompiling the code. The procedure to do that is detailed in the online documentation, available through the front page of the GitHub repository.

With this modification to SOFI, a relatively big speed-up is achieved for structures with a large number of atoms close to the origin (number N_i in Sec. IIC), where the redundancy of the search space of matrices θ is large. The very extreme example of such structure being the perfect fullerene-type structures, where all atoms are at the same distance.

The search space of highly symmetric tetrahedral, octahedral, and icosahedral PGs in particular – T , O , and I variants – can be largely redundant. In SOFI, this redundancy can be reduced by an early-termination condition, which is triggered when both of the following criteria are met: *i*) either σ or I operations are found, and *ii*) there are two or more unique C_n axes with $n \geq 3$. The two criteria are sufficient to terminate the search procedure, and directly identify the corresponding PG by the standard procedure. In principle, this condition can be ambiguous, since a group can be a subgroup of another (for instance, all T variants are subgroups of Oh). However the ambiguity is lifted by performing the combinations of symmetry elements identified at the current step of the algorithm, which are at that point extremely unlikely to be cyclic, and will thus generate the missing operations. The overhead of computing the early-termination condition is largely negligible.

The redundancy of the search space however cannot be reduced by the mentioned arguments when the number of symmetry operations is low, as for instance in $C1$, Cs , and Ci PGs. The absolute worst-case performance for the SOFI algorithm as presented in this article is thus a structure with many atoms N , all of a single atomic type, where relatively many of them are close to the origin (N_i in Sec. IIC), and a low-order PG.

VI. EXEMPLARY USES

A. Generating PG by combination of group elements

In SOFI, the assignment of PG name is done by a separate routine, which acts only on the list of operations θ . Thus, we can design a small program which reads a number of input operations, generates all possible combinations $\theta' = \theta^i * \theta^j$ in a self-consistent manner, and assigns the PG name to the final list of operations. In the present example, we use this program to confirm that the D2h and D3d groups are indeed among the subgroups of D6h. In group theory a high-symmetry PG may lose certain elements to give rise to a new PG, which is a subgroup of the former. Such descent of symmetry can be useful in certain applications. The small program of this example is included in the examples directory of the git repository: `examples/SOFI/ex_D6h_subgroup`.

The upper part of Table I lists the symmetry operations θ of some atomic structure with the D6h PG (the precise structure is not important for this example). If we take only the four operations of labels 5 and 7, and generate all possible unique combinations $\theta' = \theta^i * \theta^j$, we obtain the list of operations in the middle part of Table I. It is straightforward to verify that the obtained list of operations corresponds to D2h PG, and that all the resulting operations are in fact part of D6h, and therefore D2h is a subgroup of D6h.

Similarly, by generating all possible combinations of the four operations labelled 6 and 7 on the upper part of Table I, we obtain the list shown in the lower part of Table I, which constitutes the D3d PG, and is a subgroup of D6h. It can be verified that if we take all unique operations from the middle and lower parts of Tables I, which is 16 operations, and input them to the combinations program, we obtain back the list of 24 operations of D6h from the upper section of Table I.

B. Obtaining symmetry-equivalent configurations

The power of knowing the set of local symmetries θ is demonstrated on the example system of heptamer island on Pt(111) surface, taken from the OptBench database²⁴, shown on Fig. 2.

A particular, known displacement of the atoms from the initial structure is shown on Fig. 3 in six symmetry-equivalent variants in each panel, with the displacement

	Label	Operations θ	Axis
D6h	0	E, I	
	1	$\sigma, C_2, C_3, C_6, S_3, S_6$	0.989 -0.094 -0.109
	2	σ, C_2	-0.052 -0.939 0.339
	3	σ, C_2	-0.134 -0.330 -0.934
	4	σ, C_2	-0.022 0.648 -0.761
	5	σ, C_2	-0.112 -0.978 -0.173
	6	σ, C_2	-0.142 -0.756 -0.639
	7	σ, C_2	0.090 -0.183 0.979
D2h	0	E, I	
	1	σ, C_2	-0.112 -0.978 -0.173
	2	σ, C_2	0.090 -0.183 0.979
	3	σ, C_2	0.989 -0.094 -0.109
D3d	0	E, I	
	1	$C_3, \overline{C_3}, S_6, \overline{S_6}$	0.989 -0.094 -0.109
	2	σ, C_2	0.090 -0.183 0.979
	3	σ, C_2	-0.142 -0.756 -0.639
	4	σ, C_2	-0.052 -0.939 0.339

TABLE I. Upper part: list of operations θ of some structure with the D6h PG, grouped together by the axis they act on. The transposes of applicable operations ($\overline{C_3}, \overline{C_6}, \overline{S_3}, \overline{S_6}$) on axis 1 are omitted for clarity.

Middle part: the resulting operations of all possible combinations of four operations on axes 5 and 7 from the upper part. The PG with these operations is D2h, and all the operations are also included in D6h.

Lower part: the resulting operations of all possible combinations of four operations on axes 6 and 7 from the upper part. The PG with these operations is D3d, and all the elements are also included in D6h.

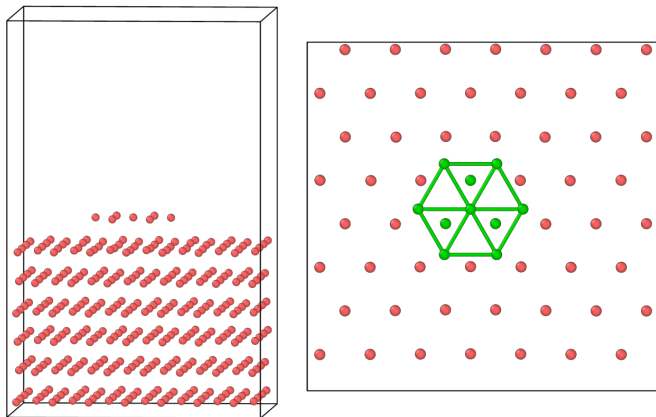


FIG. 2. Initial state configuration of the heptamer island on Pt(111) surface. Left: side view. Right: top view of the surface layer, with bonds drawn between the atoms of the island. Marked in green are the atoms of the local configuration that is used as input for SOFI.

vectors marked in cyan, each labelled Δ_j , for $j \in [1, 6]$. The displacement of the atoms by specifically these vectors actually corresponds to equivalent configurations at the saddle point of the potential energy surface (PES) of the system, the details of which are not important for the purpose of this example. By knowing only one of these displacements, say Δ_1 shown on the top left panel of Fig. 3, the other five can be generated by only using the information of local symmetry of the heptamer

island, which can be computed by SOFI.

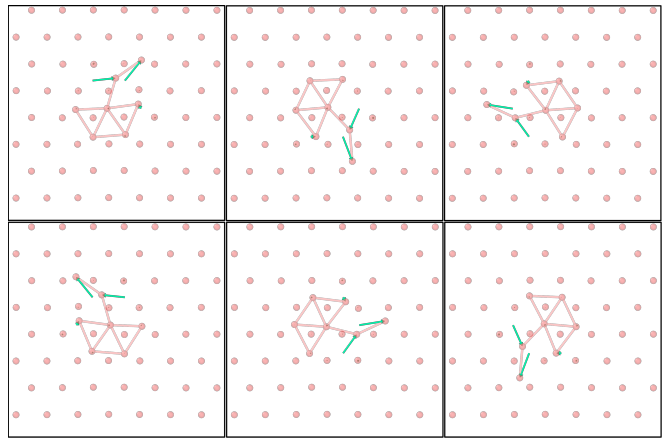


FIG. 3. First panel (top left) shows Δ_1 in cyan, which is on other panels transformed by the set $\{\theta_j, P_j\}$, and represents saddle points of the PES equivalent by local symmetry. The top row contains the rotations of Δ_1 about the axis through the origin atom (out of plane axis), and the bottom row contains reflections of Δ_1 over three mirror planes.

In the present example, this was done by first selecting the first-neighbour local configuration around the atom at the center of the heptamer, shown in green on Fig. 2, where the set of atomic indices contained in the local configuration is labelled $[s]$, and the atom at the center of the heptamer is taken as the origin point. This local configuration was then input to SOFI as an independent atomic structure, which returns the set of six symmetry operation matrices and the corresponding permutations $\{\theta_j, P_j\}$, in relation to the desired origin. The symmetry operations are three rotations about the vertical axis through the origin atom, and three mirror reflections. The displacement $\Delta_1[s]$ on the subset of atoms $[s]$ can then be transformed by applying each matrix θ_j , followed by permutation of the vector elements P_j , to obtain the other five displacements $\Delta_j[s]$,

$$\Delta_j[s] = P_j(\theta_j \Delta_1[s]) \quad (12)$$

which correspond to the displacement vectors of Fig. 3, and bring the system to the symmetry-equivalent saddle point configurations.

The example shown in this section illustrates a procedure which exploits the list of symmetry operations, and applies them on some property of the atomic structure to obtain the symmetry-equivalent representations of that property, in this case a displacement. By analogy, such procedure can be done to any vectorial property of a structure, for which symmetry-equivalent representations exist. The symmetrization of an atomic structure is an example of such procedure (with some modifications), acting on the atomic positions.

VII. CONCLUSION

We have developed the SOFI algorithm for finding PG symmetry operations of non-periodic atomic structures, based on the IRA shape matching algorithm¹⁴. The working principle of SOFI is to cast the problem of finding PGs as a shape-matching problem of a single structure with itself, and find all the degenerate solutions. The set of these solutions corresponds to the symmetry operations of the atomic structure.

Contrary to some other algorithms, SOFI is not limited to the number of atoms that a structure can contain. It is also agnostic to the choice of the origin point, which allows the identification of symmetry operations about any chosen point/atom, not only the geometric center.

We have tested SOFI on a large set of atomic structures where symmetry identification can be challenging, such as single-element clusters and close-packed structures. The former are critical because of the large number of possible permutations of atomic indices, and the latter due to the rather homogeneous spatial distribution of atoms, which prevents simple identification of any principal axes. Both these cases are of relevance for numerical applications, for instance in processing substructures obtained from other condensed phase (periodic) simulations, whenever a local symmetry assignment is required for clusters or slabs, and hence where an automatic assignment of symmetry is highly desirable. Due to the way SOFI parses through the space of possible operations, it can be advantageous in such critical cases.

We have compared the results of our SOFI algorithm to three other algorithms for PG identification from the literature, namely SYVA, SymMol, and libmsym, at two different values of the symmetry threshold. The results are given in a statistical manner that shows the general behaviour of SOFI, as compared to other algorithms. SOFI was able to successfully identify PGs across all sets, with a large degree of agreement with other algorithms. In most cases of disagreement between algorithms, SOFI has found a higher number of symmetry elements within the specified symmetry threshold, and thus a higher order point group. In all cases where SOFI found a lower order PG than another algorithm, we have been able to retrieve the higher order PG by tuning of the symmetry thresholds parameter of SOFI.

We also present two examples of how SOFI can be utilised, other than simple identification of PGs; in the first example as a tool to generate combinations of several matrices until group completeness; and in the second example as tool to aid automatic generation of structures, and their properties (in this case a displacement vector), that transform according to the local PG symmetry about a specific atom, *i.e.* any covariant property.

One of the core ideas of SOFI is to provide a modular, and interoperable set of computational routines, which could be used for retrieving the point group symmetry properties of a structure, as standalone, or within another application. The inclusion of SOFI into the IRA library

makes it a convenient, and accessible tool for purposes related to the precise analysis of atomic structures.

SUPPLEMENTARY MATERIALS

The supplementary material contains all results of the comparison tests done in this work. The results are reported in tables and images, where the tables report the result (PG) of each algorithm included in the test for each structure, and the images serve as an overall comparison between the algorithms.

ACKNOWLEDGMENTS

The authors are active members of the Multiscale And Multi-Model ApproacheS for Materials In Applied Science consortium (MAMMASMIAS consortium), and acknowledge the efforts of the consortium in fostering scientific collaboration. The work was partly funded by the Croatian Science Foundation (HrZZ), via the project IP-2020-02-7262 (HYMO4EXNOMOMA). All images of atomic structures in this article were generated with Ovito²⁵ software.

AUTHOR DECLARATIONS

Conflict of interest

The authors have no conflicts to disclose.

Author contributions

M. Gunde: Conceptualization, Data curation, Software, Writing - original draft, Writing - review & editing, Supervision, Visualization; **N. Salles:** Conceptualization, Writing - review & editing, Supervision, Visualization; **L. Grisanti:** Conceptualization, Writing - review & editing, Supervision, Visualization; **L. Martin-Samos:** Conceptualization, Writing - review & editing, Supervision, Visualization; **A. Hemeryck:** Conceptualization, Writing - review & editing, Supervision, Visualization;

DATA AND SOURCE CODE AVAILABILITY

SOFI is released as part of the IRA library, which is under double licensing, GPL v3 and Apache v2. The source code and example programs are available at <https://github.com/mammasmias/IterativeRotationsAssignments>. Link to the online documentation is available from the main GitHub page, which includes some tutorials, and detailed descriptions

of the source code, the C-bound API, and the python interface. All data used in the comparison tests is openly available from the CCD²². All datasets and scripts used in the comparison test, and plain-text results are available from Zenodo²¹.

Appendix A: Some properties of PG symmetry matrices

The matrices θ corresponding to PG symmetry operations are orthonormal ($\theta^{-1} = \theta^T$), non-symmetric matrices with real elements, and correspond to either pure rotations (symbol C), pure reflections (symbol σ), or combinations of both (rotoreflections, symbol S).

The matrices θ_R corresponding to pure rotations (symbol C) have a positive determinant, $\det(\theta_R) = 1$, the value of trace is $Tr(\theta_R) = 1 + 2 \cos(\alpha)$, and their eigenvalues are $\lambda_1 = 1$, and $\lambda_{2,3} = \exp(\pm i\alpha)$ where α is the angle of rotation about the axis given by the eigenvector corresponding to λ_1 . A pure rotation matrix can be constructed from knowing the angle and axis of rotation, for example by the well-known Euler-Rodrigues rotation formula.

Matrices θ_F which represent pure reflection (symbol σ) are characterised by the negative determinant, $\det(\theta_F) = -1$, the value of trace $Tr(\theta_F) = 1$, and their eigenvalues are $\lambda_1 = -1$, and $\lambda_{2,3} = 1$. The normal vector of the plane of reflection is given by the eigenvector of θ_F corresponding to λ_1 . A reflection matrix can be constructed from knowing the normal vector of the plane of reflection $|x\rangle$ by $\theta_F = E - 2|x\rangle\langle x|$, where E is the identity matrix, and $|\cdot\rangle\langle\cdot|$ indicates the outer product.

Matrices θ_{RF} corresponding to rotoreflections (symbol S) are characterised by the negative determinant, $\det(\theta_{RF}) = -1$, the value of trace $Tr(\theta_{RF}) = -1 + 2 \cos(\alpha)$, and their eigenvalues are $\lambda_1 = -1$, and $\lambda_{2,3} = \exp(\pm i\alpha)$, where α is the angle of rotation about the axis given by the eigenvector of λ_1 , which is at the same time also the normal vector of the plane of reflection. Matrices corresponding to rotoreflections can be constructed from combination of pure rotation and pure reflection matrices, $\theta_{RF} = \theta_F \theta_R$, where θ_F and θ_R have the same vector as the axis of rotation, and the normal of the plane of reflection.

The angle α in radians is transformed to integers n and p , such that $\alpha/2\pi = p/n$. For instance, the rotation for 5/12th of a circle (angle 150°), would be $n = 12$ and $p = 5$.

It can be observed that the use of σ symbol for pure reflections over some axis is redundant, since an equivalent notation is a rotoreflection S with angle $\alpha = 0$. We thus label pure reflections as " $S 0$ ".

The properties listed above are synthesised in the Table II.

The slight exceptions to the above are the identity $E = \text{diag}(1,1,1)$ and the point-inversion $I = \text{diag}(-1, -1, -1)$, which have the same matrix form regardless of the orientation of the reference frame. Never-

	identity E	inversion I	rotation C	rotoreflection S
$\det(\theta)$	1	-1	1	-1
$\lambda_{1,2,3}$	1, 1, 1	-1, -1, -1	1, $\exp(\pm i\alpha)$	-1, $\exp(\pm i\alpha)$
$Tr(\theta)$	3	-3	$1 + 2 \cos(\alpha)$	$-1 + 2 \cos(\alpha)$

TABLE II. The axis of rotation (plane of reflection) is given by the eigenvector corresponding to λ_1 .

theless, the identity matrix can be seen as rotation about any axis for an angle $\alpha = 0$ (operation $C 0$), and the point-inversion can be seen as rotoreflection about any axis for an angle $\alpha = \pi$ (operation $S 2^{\wedge}1$).

Due to the numerical procedure of matrix diagonalization, where any \pm direction is an equivalent solution for the eigenvectors (thus the axis), the relative orientation of the operation (sign of the angle α) cannot be trusted. For this reason, we follow the convention for orienting the axis such that its components are: $z > 0$, if $z = 0$ then $x > 0$, and if $x = 0$ then $y > 0$. The sign of α is then deduced with respect to this axis.

¹G. Herrero-Saboya, L. Martin-Samos, N. Richard, and A. Hemeryck, "Common defects in diamond lattices as instances of the general $t \otimes (e + t_2)$ jahn-teller effect," *Phys. Rev. Mater.* **6**, 034601 (2022).

²F. A. Cotton, *Chemical applications of group theory* (John Wiley & Sons., 1990.).

³P. Atkins and F. R., "Molecular quantum mechanics," (Oxford university Press, 2005) Chap. 5.

⁴E. Apra, E. J. Bylaska, W. A. de Jong, N. Govind, K. Kowalski, T. P. Straatsma, M. Valiev, H. J. J. van Dam, Y. Alexeev, J. Anchell, V. Anisimov, F. W. Aquino, R. Atta-Fynn, J. Autschbach, N. P. Bauman, J. C. Becca, D. E. Bernholdt, K. Bhaskaran-Nair, S. Bogatko, P. Borowski, J. Boschen, J. Brabec, A. Bruner, E. Cauët, Y. Chen, G. N. Chuev, C. J. Cramer, J. Daily, M. J. O. Deegan, J. Dunning, T. H., M. Dupuis, K. G. Dyall, G. I. Fann, S. A. Fischer, A. Fonari, H. Früchtl, L. Gagliardi, J. Garza, N. Gawande, S. Ghosh, K. Glaesemann, A. W. Goetz, J. Hammond, V. Helms, E. D. Hermes, K. Hirao, S. Hirata, M. Jacquelin, L. Jensen, B. G. Johnson, H. Jónsson, R. A. Kendall, M. Klemm, R. Kobayashi, V. Konkov, S. Krishnamoorthy, M. Krishnan, Z. Lin, R. D. Lins, R. J. Littlefield, A. J. Logsdail, K. Lopata, W. Ma, A. V. Marenich, J. Martin del Campo, D. Mejia-Rodriguez, J. E. Moore, J. M. Mullin, T. Nakajima, D. R. Nascimento, J. A. Nichols, P. J. Nichols, J. Nieplocha, A. Otero-de-la Roza, B. Palmer, A. Panyala, T. Pirojsirikul, B. Peng, R. Peverati, J. Pittner, L. Pollack, R. M. Richard, P. Sadayappan, G. C. Schatz, W. A. Shelton, D. W. Silverstein, D. M. A. Smith, T. A. Soares, D. Song, M. Swart, H. L. Taylor, G. S. Thomas, V. Tipparaju, D. G. Truhlar, K. Tsemekhman, T. Van Voorhis, Vázquez-Mayagoitia, P. Verma, O. Villa, A. Vishnu, K. D. Vogiatzis, D. Wang, J. H. Weare, M. J. Williamson, T. L. Windus, K. Woliński, A. T. Wong, Q. Wu, C. Yang, Q. Yu, M. Zacharias, Z. Zhang, Y. Zhao, and R. J. Harrison, "NWChem: Past, present, and future," *The Journal of Chemical Physics* **152**, 184102 (2020), https://pubs.aip.org/aip/jcp/article-pdf/doi/10.1063/5.0004997/16684361/184102_1_online.pdf.

⁵T. Pilati and A. Forni, "SYMMOL: a program to find the maximum symmetry group in an atom cluster, given a prefixed tolerance," *J. Appl. Crystallogr.* **31**, 503–504 (1998).

⁷R. J. Largent, W. F. Polik, and J. R. Schmidt, "Symmetrizer: Algorithmic determination of point groups in nearly symmetric molecules," *J. Comput. Chem.* **33**, 1637–1642 (2012), <https://onlinelibrary.wiley.com/doi/pdf/10.1002/jcc.22995>.

- ⁸K. Nedwed, K. Gatterer, and H. P. Fritzer, "Symapps 1.0: A software packet for group theoretical applications to molecular symmetry," *Comput. Chem.* **18**, 371–376 (1994).
- ⁹L. Gyevi-Nagy and G. Tasi, "Syva: A program to analyze symmetry of molecules based on vector algebra," *Comput. Phys. Commun.* **215**, 156–164 (2017).
- ¹⁰M. Johansson and V. Veryazov, "Automatic procedure for generating symmetry adapted wavefunctions," *J. Cheminformatics* **9**, 8 (2017).
- ¹¹O. Beruski and L. N. Vidal, "Algorithms for computer detection of symmetry elements in molecular systems," *J. Comput. Chem.* **35**, 290–299 (2014), <https://onlinelibrary.wiley.com/doi/pdf/10.1002/jcc.23493>.
- ¹²J. Ivanov and G. Schüürmann, "Simple algorithms for determining the molecular symmetry," *J. Chem. Inf. Comp. Sci.* **39**, 728–737 (1999), <https://doi.org/10.1021/ci990322q>.
- ¹³N. M. Vandewiele, R. Van de Vijver, K. M. Van Geem, M.-F. Reyniers, and G. B. Marin, "Symmetry calculation for molecules and transition states," *J. Comput. Chem.* **36**, 181–192 (2015), <https://onlinelibrary.wiley.com/doi/pdf/10.1002/jcc.23788>.
- ¹⁴M. Gunde, N. Salles, A. Hémercyck, and L. Martin-Samos, "Ira: A shape matching approach for recognition and comparison of generic atomic patterns," *J. Chem. Inf. Model.* **61**, 5446–5457 (2021), pMID: 34704748, <https://doi.org/10.1021/acs.jcim.1c00567>.
- ¹⁵M. Gunde, "Development of ira: a shape matching algorithm, its implementation, and utility in a general off-lattice kmc kernel (phd thesis)," (2021).
- ¹⁶A. Pausch, M. Gebele, and W. Klopper, "Molecular point groups and symmetry in external magnetic fields," *J. Chem. Phys.* **155**, 201101 (2021), <https://doi.org/10.1063/5.0069859>.
- ¹⁷T. Eiter and H. Mannila, "Distance measures for point sets and their computation," *Acta Inform.* **34**, 109–133 (1997).
- ¹⁸K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-squares fitting of two 3-d point sets," *IEEE T. Pattern Anal.* **PAMI-9**, 698–700 (1987).
- ¹⁹P. J. Besl and N. D. McKay, "A method for registration of 3-d shapes," *IEEE T. Pattern Anal.* **14**, 239–256 (1992).
- ²⁰H. Pottmann, Q.-X. Huang, Y.-L. Yang, and S.-M. Hu, "Geometry and convergence analysis of algorithms for registration of 3d shapes," *Int. J. Comput. Vision* **67**, 277–296 (2006).
- ²¹M. Gunde, N. Salles, L. Grisanti, L. MARTIN-SAMOS, and A. Hemeryck, "SOFI: Finding point group symmetries in atomic clusters as finding the set of degenerate solutions in a shape-matching problem," (2024).
- ²²D. J. Wales, J. P. K. Doye, A. Dullweber, M. P. Hodges, F. Y. N. F. Calvo, J. Hernández-Rojas, and T. F. Middleton, "The cambridge cluster database," <https://www.wales.ch.cam.ac.uk/CCD.html>.
- ²³M. D. Hanwell, D. E. Curtis, D. C. Lonie, T. Vandermeersch, E. Zurek, and G. R. Hutchison, "Avogadro: an advanced semantic chemical editor, visualization, and analysis platform," *Journal of Cheminformatics* **4**, 17 (2012).
- ²⁴S. T. Chill, J. Stevenson, V. Ruehle, C. Shang, P. Xiao, J. D. Farrell, D. J. Wales, and G. Henkelman, "Benchmarks for characterization of minima, transition states, and pathways in atomic, molecular, and condensed matter systems," *J. Chem. Theory Comput.* **10**, 5476–5482 (2014), pMID: 26583230, <https://doi.org/10.1021/ct5008718>.
- ²⁵A. Stukowski, "Visualization and analysis of atomistic simulation data with ovito—the open visualization tool," *Model. Simul. Mater. Sc.* **18**, 015012 (2009).

Supplementary Materials for: "SOFI: Finding point group symmetries in atomic clusters as finding the set of degenerate solutions in a shape-matching problem"

M. Gunde^a, N. Salles^b, L. Grisanti^{a,b}, L. Martin-Samos^b, A. Hemeryck^c

^a Institute Ruđer Bošković, Bijenička 54, 10000 Zagreb, Croatia

^b CNR-IOM/Democritos National Simulation Center, Istituto Officina dei Materiali, c/o SISSA, via Bonomea 265, IT-34136 Trieste, Italy

^c LAAS-CNRS, Université de Toulouse, CNRS, 7 avenue du Colonel Roche, 31000 Toulouse, France

1 Algorithms and datasets used in the comparison test

All the structures, software source codes, scripts, and plain-text format of all results from the test can be downloaded from Zenodo [1].

The SOFI algorithm presented in the main article has been compared to the SYVA [2], SymMol [3], and libmsym [4] software. SYVA has been downloaded from ¹. SymMol software has been extracted from IQmol3 repository, commit 32cbf13 at ², and libmsym v0.2.3 from ³.

In order to make the tests run smoother, we had to make some small modifications to some of the source code, mostly regarding their output. The SYVA code was used completely as-is, it is limited to max 250 atoms, and we were not able to modify that limit. The SymMol code is by default limited to maximum 100 atoms, but we were able to simply modify this to our needs. Its source was also modified in one instance of integer initialisation, which was causing a crash due to segmentation fault. The output of SymMol uses the label "C3i", and "T23" for the groups "S6" and "T" respectively. The former is a possible alternative nomenclature, while we are not sure of the latter, it might simply be a typo in the code. We change all instances of these labels to "S6", and "T", after SymMol outputs them. The libmsym software has been modified as suggested in the GitHub issue #17 ⁴ to properly output the group C1, instead of an error message, when that is the case. All modifications to the source codes done for the purpose of this test are available as diff patches in the Zenodo reference [1].

The data sets containing the atomic structures used for testing the code are taken from the Cambridge Cluster Database (CCD) website [5], more specifically the sets where PG information is available. The ones used in our test are listed below, referred to by their name as it appears on the CCD website:

- Lennard-Jones clusters N=3-150 [6],
- Quantum Lennard-Jones clusters [7],
- Sutton-Chen clusters 10-8 , 9-6 , and 12-6 [8],
- Glue clusters Al, [9] , and Pb [10],
- Gupta clusters (Zn, Cd, Pb, Na) [11, 12, 13],
- Murrell-Mottram clusters (Na) [12],
- Finnis-Sinclair clusters Mo and Fe [14],
- C60 clusters [15],
- C60-Pacheco clusters [16],
- Morse clusters [17, 18, 19],

¹<https://data.mendeley.com/datasets/xd4g8hyjf2/1>

²<https://github.com/nutjunkie/IQmol3/blob/09e9f6b832e8a7cb6bde1e787a93a03993023713/src/Fortran/symmol.f90>

³<https://github.com/mcodev31/libmsym/releases/tag/v0.2.3>

⁴<https://github.com/mcodev31/libmsym/issues/17>

- Dzugutov [20], and Dzugutov-mod clusters [21],
- $(\text{NaCl})_N\text{Cl}^-$ for Coulomb/Born-Meyer and Welch potentials [22],
- binary Lennard-Jones clusters [23].

The format of the files containing atomic structures was changed into the regular xyz format, and all atomic types have been assigned to character (string) symbol values. In cases when the atomic type is not obvious (i.e. in the binary LJ, or morse dataset), H and/or He were used as surrogate symbols.

The total dataset of atomic structures thus contains 3858 structures, with number of atoms ranging from 3 to 380. The only set of structures containing more than one atomic type is the bin_LJ set, which contains two atomic types.

2 Methodology

2.1 The tolerance values

The algorithms SYVA [2], SymMol [3], and SOFI have a more or less equivalent definition of the tolerance parameter (symmetry threshold in SOFI), which is that the maximal displacement of any atom of the symmetry-transformed structure, with respect to the original structure, should be below the tolerance. Due to the equivalent definition and meaning of the tolerance value, the results of these three algorithms can be directly compared at the same value of tolerance.

We have performed the tests with the tolerance at two values, (i) with $\text{tol}=0.001$, which is the default value for the tolerance in the SYVA code, and (ii) at the value $\text{tol}=0.05$. These values were applied to the SYVA, SymMol, and SOFI algorithms. The software libmsym [4] does not have a single value for the tolerance parameter, but instead several values related to different aspects of the respective algorithm. For the present test, we took three different sets of values for the tolerance in libmsym, which are also used in the avogadro software [24], in which they are labelled as "tight" or "default", "medium", and "loose" (see link ⁵).

2.2 Reported values

For each input structure we compute the PG with a given tolerance, using each algorithm. The symbol "-" in any entry indicates a failure of the software to find any PG, where the error or failure was either reported by the code, the code has crashed with no output, or in the case of SymMol the computation has timed out.

The leading hypothesis for differences in the results of the compared algorithms is that the use of principal axes of inertia can be ambiguous when its eigenvalues are very similar. For this reason, we explicitly compute the matrix of inertia for each structure, and diagonalize it to obtain the principal axes. We then perform the scalar product between each of the principal axes of inertia, and each of the principal axes of symmetry as found by SOFI, and store the highest value of the dot product among all the possible combinations. The absolute value 1.0 of this dot product indicates that any one of the principal axes of inertia is indeed along any one of the principal axes of symmetry, while a value significantly lower than 1.0 indicates that none of the principal axes of inertia are aligned with any of the principal axes of symmetry. The value is reported in the tables under the column "dot product". For the C1 point groups, the reported value is irrelevant since the C1 group has no principal axis of symmetry.

The columns labelled N_{op} report the number of symmetry operations associated to the respective PG.

3 Results

For a more compact representation, the results are plotted in form of 2D histograms (*i.e.* Fig. S1), where each of the axes represents the result for a given atomic structure by a given algorithm, at a given tolerance value. All cases of failure are omitted from the 2D histograms. The values inside the plot are positioned such that their (x, y) position encodes the PGs resulting from each of the two algorithms, and their numerical value represents the number of input structures where such combination of PGs occurred. The values on the diagonal of the plot represent cases when the two compared algorithms returned the same result for the same structure. The off-diagonal values represent cases where the two algorithms have returned a different result.

The point groups are ordered on the axes by their order/cardinality (number of elements). All values appearing on the upper side of the diagonal indicate that the resulting PG by algorithm reported on the y -axis has a higher order than the same result with the algorithm on the x -axis, and the opposite for values below the diagonal.

⁵<https://github.com/OpenChemistry/avogadrolibs/blob/master/avogadro/qtplugins/symmetry/symmetrywidget.cpp>

In the big majority of structures, the algorithms agree among each other in the resulting PG, there are however some cases of different results. In the following sections we analyze the cases of differences, mainly from the point of view of SOFI. That is, we are not interested in analysing and resolving each particular problematic case of every algorithm, but rather only those where there is a difference with respect to SOFI. In cases where SOFI finds a lower-order PG, we find the value of tolerance needed by SOFI to obtain the same PG. In the opposite case, we do not perform any further computation. Due to this, the comparison tests performed here cannot be viewed as fair benchmarking of algorithms, but merely as a way to expose the performance of SOFI, by means of comparison to other established algorithms.

The Sections 3.1, 3.2, and 3.3 contain the comparisons to SYVA, SymMol, and libmsysm, respectively. The Section 5 lists the results of all algorithms and structures used in the test. The summary of reported failures is provided in the Tab. S8.

3.1 Comparison with SYVA

The SYVA software is by implementation limited to max 250 atoms, we thus ignore all structures with more than 250 atoms, they are not included in the following analysis.

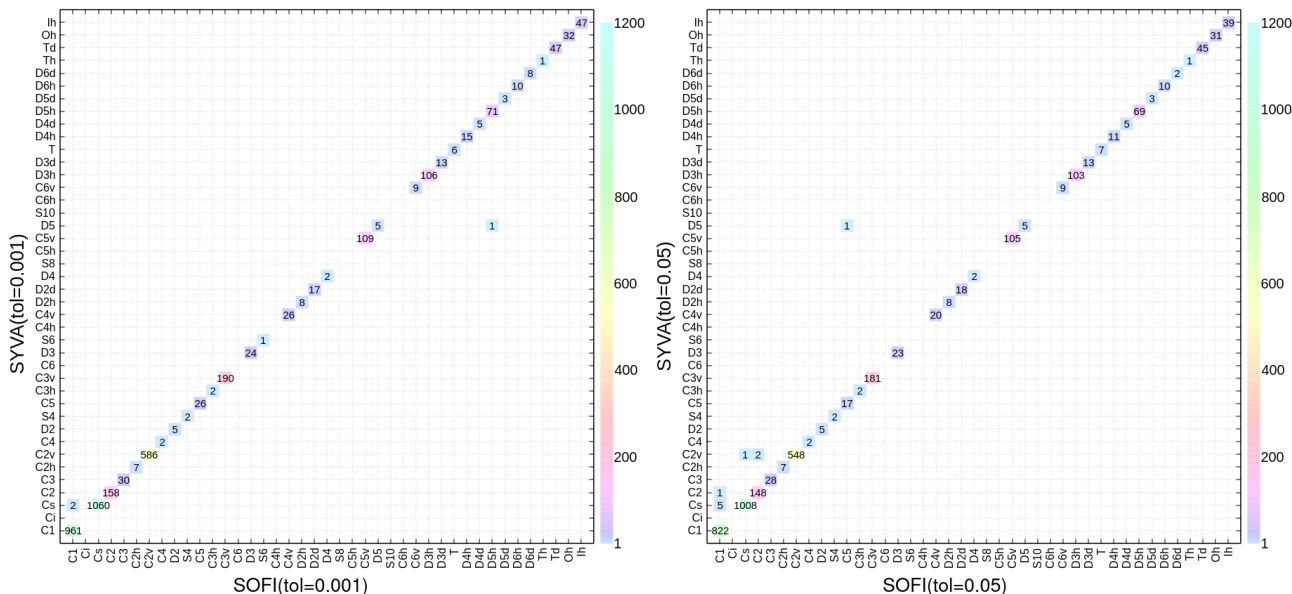


Figure S1: Comparison of SYVA and SOFI at (left) $tol = 0.001$, and (right) $tol = 0.05$.

3.1.1 Value $tol = 0.001$

The case with $tol = 0.001$ is shown on Fig. S1(left). All the resulting PGs agree with the result of SYVA($tol = 0.001$), except the five cases listed in Tab. S1, which we analyzed further:

set	struc name	SYVA	N_{op}	SOFI	N_{op}	dot product	needed tol
glue_Pb	54	–	0	S10	10	1.00000	
gupta/Co	174	Cs	2	C1	1	1.00000	0.00131
gupta/Co	183	–	0	C2	2	1.00000	
gupta/Co	190	D5	10	D5h	20	1.00000	
QLJ	75A	Cs	2	C1	1	0.96203	0.0014

Table S1: Cases of different results between SYVA and SOFI at $tol = 0.001$. There are two cases of failure by SYVA. In cases where SYVA has found a PG with a higher order, the column labelled "needed tol" indicates the value of tol such that SOFI(tol) retrieves the same PG as SYVA.

- structure glue_Pb:54, SYVA reports "unable to determine PG", however the symmetry operations that it reports correspond to point group S10. SOFI reports group S10.

- structure Gupta/Co:174, SYVA reports group Cs, while SOFI reports group C1. The S0 symmetry operation that is missed by SOFI($tol = 0.001$) is found with SOFI($tol = 0.00131$).
- structure Gupta/Co:183, SYVA reports 3 symmetry operations, but reports a failure to find PG. SOFI($tol = 0.001$) reports 2 symmetry operations and the group C2. The extra symmetry operation found by SYVA($tol = 0.001$) is found by SOFI($tol = 0.0012$), which finds an additional symmetry for the group C2v.
- structure Gupta/Co:190, SYVA find D5, SOFI finds D5d-, and D5h after combinations, with maximal distortion value 0.0012.
- structure QLJ:71A, SYVA finds Cs, SOFI finds C1. The S0 symmetry missed by SOFI($tol = 0.001$) is retrieved with SOFI($tol = 0.0014$).

Upon some analysis of the SYVA source code, we find a table of hard-coded point groups, and the group S10 is not among them (see file `blockdata.f` in SYVA source). It also seems like SYVA does not attempt to deal with slightly broken symmetries (as in structure Gupta/Co:183), but instead directly reports a failure.

Henceforth we skip listing and analysing any of the reported failures.

3.1.2 Value $tol = 0.05$

set	struc name	SYVA	N_{op}	SOFI	N_{op}	dot product	needed tol
glue_Al	134	C2v	4	C2	2	1.00000	0.059
glue_Al	181	C2v	4	Cs	2	1.00000	0.062
glue_Al	183	Cs	2	C1	1	1.00000	0.061
gupta/Co	37	C2	2	C1	1	0.99998	0.051
gupta/Na	246	D5	10	C5	5	1.00000	0.051
murrel-mottram/Na	109	Cs	2	C1	1		
murrel-mottram/Na	112	Cs	2	C1	1		
murrel-mottram/Na	113	Cs	2	C1	1		
murrel-mottram/Na	120	C2v	4	C2	2	1.00000	
NaCl_N/CBM	15	Cs	2	C1	1		

Table S2: Cases of different results between SYVA and SOFI at $tol = 0.05$. There are 290 cases of failure by SYVA. In cases where SYVA has found a PG with a higher order, the column labelled "needed tol" indicates the value of tol such that SOFI(tol) retrieves the same PG as SYVA.

In the case of $tol = 0.05$, the number of differences between SYVA and SOFI is 10. The number of failures reported by SYVA($tol = 0.05$) is 290, without counting the structures with more than 250 atoms. The cases of different results are given in the Tab. S2.

In the last 5 cases of the Tab. S2 (the murrel-mottram and NaCl_N cases), SYVA has returned a PG of higher order than SOFI, and we were not able to determine the value of tol for which SOFI(tol) would retrieve the same groups. Upon further inspection into these cases, the symmetry operations in the SYVA output seem unconventional, for example the reported axes of symmetry are far from normalized (normally the axes output by SYVA are normalized), the angles of rotation have peculiar values such as 179.95 degrees (which we have not observed in other SYVA outputs). These observations make us suspect a possible (unknown) software bug in SYVA, which might be further supported by the fact that none of the included software reports the groups found by SYVA for these structures, at any tolerance value (see the complete results in Sec. 5, namely the corresponding entries in Tables S45, S46, and S50). In fact, all other software agree in the resulting PG.

3.2 Comparison with SymMol

In some cases, the SymMol software seems to get trapped in what appears to be an infinite loop (a process that never terminates). To mitigate this, we constrain every call to SymMol with a timeout value of 10 seconds. If a computation takes more than 10 seconds, the call is terminated and result recorded as failure.

SymMol software labels the S6 group as C3i. We have changed all occurrences of C3i in the results into S6, a-posteriori.

3.2.1 Value $tol = 0.001$

The number of different results between SymMol($tol = 0.001$) and SOFI($tol = 0.001$) is 225, where in all cases SOFI has found a higher order PG than SymMol. On top of that, the number of cases where SymMol fails due to the timeout constraint is 112. We skip listing all these cases here, but they could easily be filtered from all the results.

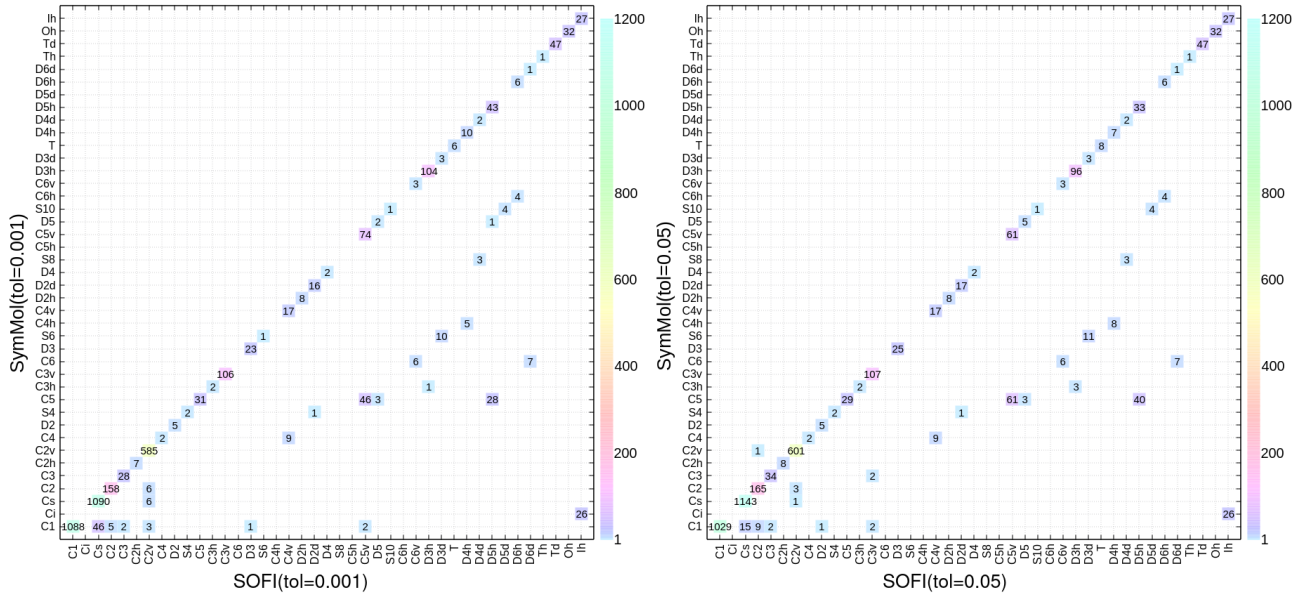


Figure S2: Comparison of SymMol and SOFI at (left) $tol = 0.001$, and (right) $tol = 0.05$.

3.2.2 value $tol = 0.05$

set	struc name	SymMol	N_{op}	SOFI	N_{op}	dot product	needed tol
glue_Al	134	C2v	4	C2	2	1.00000	0.059

Table S3: Case of higher-order symmetry obtained by SymMol than SOFI at $tol = 0.05$. The column labelled "needed tol" indicates the value of tol such that $SOFI(tol)$ retrieves the same PG as SymMol.

The number of different results between $SymMol(tol = 0.05)$ and $SOFI(tol = 0.05)$ is 221, where in all cases except for one, SOFI has found a higher order PG than SymMol. The single case where $SymMol(tol = 0.05)$ found a higher order symmetry is reported in Tab. S3. With the modified tolerance, $SOFI(tol = 0.059)$ retrieves the C2v group reported by $SymMol(tol = 0.05)$. On top of that, the number of cases where SymMol fails due to the timeout constraint is 112.

3.2.3 General observations

set	name	SYVA		SymMol		libmsym			SOFI		dot product
		0.001	0.05	0.001	0.05	'default'	'medium'	'loose'	0.001	0.05	
bin_LJ/1.05	55	Ih	Ih	Ci	Ci	Ih	Ih	Ih	Ih	Ih	0.98152
bin_LJ/1.1	45	Ih	Ih	Ci	Ci	Ih	Ih	Ih	Ih	Ih	0.99961
bin_LJ/1.15	45	Ih	Ih	Ci	Ci	Ih	Ih	Ih	Ih	Ih	0.99898
bin_LJ/1.2	45	Ih	Ih	Ci	Ci	Ih	Ih	Ih	Ih	Ih	0.97000
bin_LJ/1.25	45	Ih	Ih	Ci	Ci	Ih	Ih	Ih	Ih	Ih	0.95774
bin_LJ/1.3	45	Ih	Ih	Ci	Ci	Ih	Ih	Ih	Ih	Ih	0.96335
dzugutov	13	Ih	Ih	Ci	Ci	Ih	Ih	Ih	Ih	Ih	0.99682
dzugutov_mod	13	Ih	Ih	Ci	Ci	Ih	Ih	Ih	Ih	Ih	0.93269
gupta/Co	13	Ih	-	Ci	Ci	Ih	Ih	Ih	Ih	Ih	0.97125
gupta/Na	13	Ih	-	Ci	Ci	Ih	Ih	Ih	Ih	Ih	0.93316
gupta/Na	297	-	-	Ci	Ci	Ih	Ih	Ih	Ih	Ih	1.00000
gupta/Pb	13	Ih	-	Ci	Ci	Ih	Ih	Ih	Ih	Ih	0.97225
gupta/Pb	54	Ih	Ih	Ci	Ci	Ih	Ih	Ih	Ih	Ih	0.95000
gupta/Pb	55	Ih	Ih	Ci	Ci	Ih	Ih	Ih	Ih	Ih	0.95450
LJ	13	Ih	Ih	Ci	Ci	Ih	Ih	Ih	Ih	Ih	1.00000
LJ	135	Ih	Ih	Ci	Ci	Ih	Ih	Ih	Ih	Ih	0.91305
LJ	147	Ih	Ih	Ci	Ci	Ih	Ih	Ih	Ih	Ih	0.89257
morse	13A	Ih	-	Ci	Ci	Ih	Ih	Ih	Ih	Ih	0.96634
morse	45A	Ih	Ih	Ci	Ci	Ih	Ih	Ih	Ih	Ih	0.98813
morse	55B	Ih	Ih	Ci	Ci	Ih	Ih	Ih	Ih	Ih	0.94555
morse	147B	Ih	Ih	Ci	Ci	Ih	Ih	Ih	Ih	Ih	0.97282
sutton_9_6	13	Ih	-	Ci	Ci	Ih	Ih	Ih	Ih	Ih	0.98001
sutton_9_6	55	Ih	Ih	Ci	Ci	Ih	Ih	Ih	Ih	Ih	0.98488
sutton_10_8	13	Ih	-	Ci	Ci	Ih	Ih	Ih	Ih	Ih	0.99522
sutton_12_6	13	Ih	-	Ci	Ci	Ih	Ih	Ih	Ih	Ih	0.94451
sutton_12_6	55	Ih	Ih	Ci	Ci	Ih	Ih	Ih	Ih	Ih	0.97111

Table S4: Cases where SymMol fails to recognize the Ih point group.

It is worth noting that the SymMol algorithm seems to have some trouble in recognising the Ih group. Out of 53 structures in the dataset where Ih group appears, SymMol only recognizes 27 of them correctly, even when all other algorithms seem to agree on the Ih group. Moreover, SymMol systematically labels these as the Ci group. All such

cases are noted in the Tab. S4. It is also worth noting that in many of these cases, the value of "dot product", which is the maximal scalar product between the symmetry axes and the principal axes of inertia, is below 1.0. This is significant since there are 6 principal axes of symmetry in the Ih group, and none of them seem to perfectly align with any of the principal axes of inertia. Based on this data, one could make a statement about the idea of basing the identification of symmetry axes solely on the inertial axes, or more precisely a statement about the ambiguity of such procedure.

3.3 Comparison with libmsym

Since the libmsym software has a completely different definition of tolerance, we cannot directly compare the obtained value. We thus compare each of the three sets of tolerance values used in libmsym, with each of the two tolerance values used in SOFI.

3.3.1 libmsym(tol = "default")

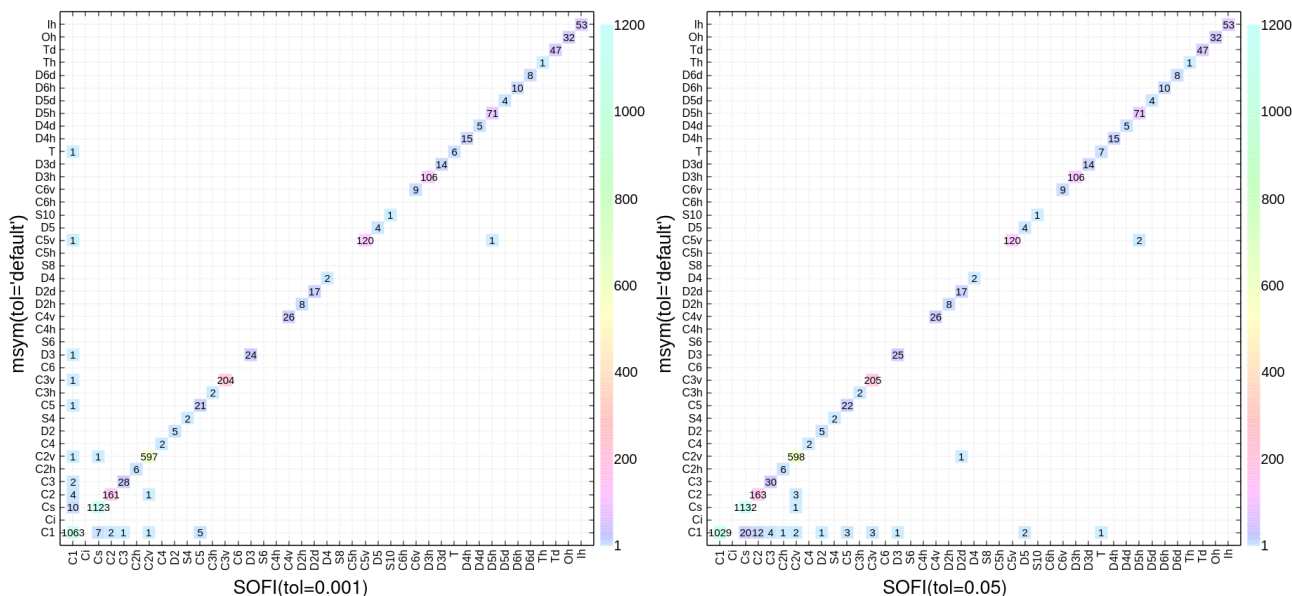


Figure S3: Comparison of libmsym(tol = "default") with (left) SOFI(tol = 0.001), and (right) SOFI(tol = 0.05).

There are 20 cases of failure of libmsym(tol = "default"), the number of cases where SOFI(tol = 0.001) found a higher-order PG than libmsym(tol = "default") is 18, while in 23 cases libmsym(tol = "default") found a higher-order PG. The latter 23 cases are listed in Tab. S5. We can observe that SOFI can retrieve the PG found by libmsym in all the cases, by setting a slightly higher tolerance value.

In case of SOFI(tol = 0.05), there are 57 differences with respect to libmsym(tol = "default"), in all of which SOFI has found a higher-order PG than libmsym.

3.3.2 libmsym(tol = "medium")

At the "medium" tolerance level, libmsym reports a failure for 227 cases. The number of cases where SOFI(tol = 0.001) produces a higher-order PG than libmsym(tol = "medium") is 33, while vice-versa the number of cases is 28. The latter 28 cases are reported in Tab. S6.

With SOFI(tol = 0.05) there are 58 cases where SOFI finds a higher-order PG, and 3 cases where libmsym(tol = "medium") finds a higher-order PG. These 3 cases can be inferred from Tab. S6, as the cases where "needed tol" is beyond 0.05, namely structures in the glue_Al set, 79, 134, and 183. It can be observed that all of those can be retrieved in SOFI by raising the tolerance.

3.3.3 libmsym(tol = "loose")

With the libmsym(tol = "loose"), there are 782 failures reported by libmsym. In 82 cases SOFI(tol = 0.001) has found higher-order PG, while in 37 cases libmsym(tol = "loose") found a high-order PG.

set	struc name	libmsym	N_{op}	SOFI	N_{op}	dot product	needed tol
gupta/Co	174	Cs	2	C1	1	1.00000	0.0014
gupta/Co	175	Cs	2	C1	1	1.00000	0.0035
gupta/Co	179	Cs	2	C1	1	1.00000	0.0021
gupta/Co	184	Cs	2	C1	1	1.00000	0.0020
murrel-mottram/Na	40	Cs	2	C1	1	1.00000	0.0032
murrel-mottram/Na	99	C2	2	C1	1	0.99994	0.0051
murrel-mottram/Na	107	C2	2	C1	1	1.00000	0.0067
murrel-mottram/Na	111	C5	5	C1	1	1.00000	0.0015
murrel-mottram/Na	132	Cs	2	C1	1	1.00000	0.0078
murrel-mottram/Na	152	Cs	2	C1	1	1.00000	0.0026
murrel-mottram/Na	153	C3v	6	C1	1	1.00000	0.0026
murrel-mottram/Na	154	Cs	2	C1	1	1.00000	0.0015
murrel-mottram/Na	157	C2v	4	C1	1	1.00000	0.0038
murrel-mottram/Na	158	Cs	2	C1	1	1.00000	0.002
murrel-mottram/Na	182	C2	2	C1	1	1.00000	0.0026
murrel-mottram/Na	207	C3	3	C1	1	1.00000	0.0035
murrel-mottram/Na	216	T	12	C1	1	0.95174	0.0032
murrel-mottram/Na	222	C3	3	C1	1	1.00000	0.0024
murrel-mottram/Na	232	D3	6	C1	1	1.00000	0.0014
murrel-mottram/Na	242	C2	2	C1	1	1.00000	0.0029
murrel-mottram/Na	321	Cs	2	C1	1	1.00000	0.0017
murrel-mottram/Na	327	C2v	4	Cs	2	1.00000	0.002
QLJ	75A	C5v	10	C1	1	1.00000	0.0053

Table S5: Cases of higher-order symmetry obtained by libmsym($tol = "default"$) than SOFI($tol = 0.001$). The column labelled "needed tol" indicates the value of tol such that SOFI(tol) retrieves the same PG as libmsym.

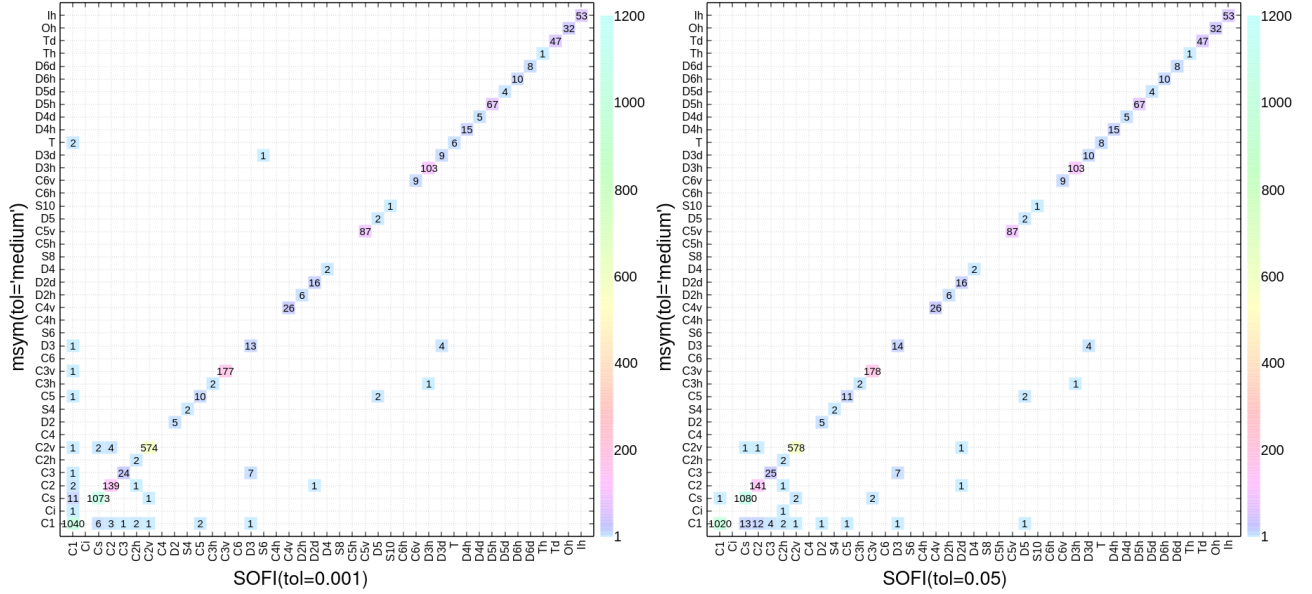


Figure S4: Comparison of libmsym($tol = "medium"$) with (left) SOFI($tol = 0.001$), and (right) SOFI($tol = 0.05$).

When using SOFI($tol = 0.05$), there are 93 cases where SOFI finds a higher-order PG, and 13 cases where libmsym($tol = "loose"$) finds a higher-order PG. All these 13 cases can be inferred from Tab. S7, as cases where the "needed tol" exceeds 0.05. It can also be observed that for all these cases, SOFI can retrieve the higher-order PG with an appropriate value of tol .

set	struc name	libsym	N_{op}	SOFI	N_{op}	dot product	needed tol
bin_LJ/1.15	71	Cs	2	C1	1	0.95403	0.27549
C60	12	C5v	10	Cs	2	1.00000	0.2268
finnis-sinclair/Mo	8	D2d	8	C2v	4	1.00000	0.0590
finnis-sinclair/Mo	26	Td	24	S4	4	0.81061	0.1886
finnis-sinclair/Mo	27	Cs	2	C1	1	1.00000	0.1999
finnis-sinclair/Mo	29	C2v	4	C2	2	1.00000	0.2743
finnis-sinclair/Mo	31	C3	3	C1	1	0.99905	0.0409
finnis-sinclair/Mo	41	Cs	2	C1	1	0.99977	0.2409
finnis-sinclair/Mo	48	C3v	6	C3	3	1.00000	0.1860
glue_Pb	8	D2d	8	C2v	4	1.00000	0.0426
glue_Pb	26	C2v	4	Cs	2	1.00000	0.5462
glue_Pb	98	Cs	2	C1	1	0.76817	0.2175
glue_Al	70	C2v	4	C2	2	1.00000	0.0047
glue_Al	77	C6v	12	C3v	6	1.00000	0.0905
glue_Al	96	C2v	4	C2	2	1.00000	0.0432
gupta/Co	179	Cs	2	C1	1	1.00000	0.0021
gupta/Co	184	Cs	2	C1	1	1.00000	0.0020
gupta/Co	189	Cs	2	C1	1	1.00000	0.0011
gupta/Na	38	Oh	48	Th	24	0.70421	0.1677
morse	11D	C2v	4	Cs	2	0.99983	0.0463
morse	12C	C5v	10	Cs	2	1.00000	0.0767
murrel-mottram/Na	89	C3	3	C1	1	0.99991	0.0110
murrel-mottram/Na	92	T	12	C1	1	0.98969	0.0195
murrel-mottram/Na	104	C2	2	C1	1	0.99999	0.0267
murrel-mottram/Na	119	Cs	2	C1	1	0.99992	0.0216
murrel-mottram/Na	132	Cs	2	C1	1	1.00000	0.0078
murrel-mottram/Na	151	Cs	2	C1	1	0.99839	0.0087
murrel-mottram/Na	152	Cs	2	C1	1	1.00000	0.0026
murrel-mottram/Na	153	C3v	6	C1	1	1.00000	0.0026
murrel-mottram/Na	154	Cs	2	C1	1	1.00000	0.0015
murrel-mottram/Na	156	Cs	2	C1	1	0.99810	0.0095
murrel-mottram/Na	207	C3	3	C1	1	1.00000	0.0035
murrel-mottram/Na	216	T	12	C1	1	0.95174	0.0033
murrel-mottram/Na	318	Cs	2	C1	1	1.00000	0.0012
murrel-mottram/Na	319	C3v	6	Cs	2	1.00000	0.0434
murrel-mottram/Na	321	Cs	2	C1	1	1.00000	0.0017
murrel-mottram/Na	324	Cs	2	C1	1	0.99996	0.0056

Table S7: Cases of higher-order PG obtained by libmsym($tol = "loose"$) than SOFI($tol = 0.001$). The column labelled "needed tol" indicates the value of tol such that SOFI(tol) retrieves the same PG as libmsym.

set	SYVA		SymMol		libmsym			SOFI	
	0.001	0.05	0.001	0.05	'default'	'medium'	'loose'	0.001	0.05
bin_LJ/1.05	0	0	3	3	0	1	6	0	0
bin_LJ/1.1	0	0	2	2	1	2	11	0	0
bin_LJ/1.15	0	0	2	2	0	2	13	0	0
bin_LJ/1.2	0	0	4	3	0	0	11	0	0
bin_LJ/1.25	0	0	3	4	0	3	21	0	0
bin_LJ/1.3	0	0	2	3	0	2	18	0	0
C60	0	0	3	2	0	4	7	0	0
C60_pacheco	0	92	1	1	0	3	24	0	0
dzugutov	0	0	1	0	0	0	16	0	0
dzugutov_mod	0	0	4	3	2	20	64	0	0
finnis-sinclair/Fe	0	0	2	2	0	1	9	0	0
finnis-sinclair/Mo	0	0	1	1	0	5	18	0	0
glue_Pb	1	13	5	3	0	5	19	0	0
glue_Al	0	0	4	3	1	12	37	0	0
gupta/Cd	0	1	0	0	0	2	8	0	0
gupta/Co	1	1	9	9	1	20	62	0	0
gupta/Na	0	2	20	21	6	48	142	0	0
gupta/Pb	0	2	4	3	0	5	31	0	0
gupta/Zn	0	1	1	0	0	1	8	0	0
LJ	0	0	8	8	0	11	36	0	0
morse	0	2	12	12	0	13	58	0	0
murrel-mottram/Na	0	132	13	12	9	57	100	0	0
NaCl_N/welch	0	21	0	0	0	0	6	0	0
NaCl_N/CBM	0	19	0	0	0	0	4	0	0
QLJ	0	0	0	0	0	3	6	0	0
sutton_9_6	0	2	2	3	0	3	17	0	0
sutton_10_8	0	1	1	1	0	2	16	0	0
sutton_12_6	0	1	5	6	0	2	14	0	0
total	2	290	112	107	20	227	782	0	0

Table S8: The count of failures of each algorithm and tolerance value used. Failures are counted only for cases when the output of an algorithm actually reports a failure, or the program crashes. For SYVA, the count does not include structures with more than 250 atoms. For SymMol the count includes cases of timed-out computations.

of group completeness), but rather just report a failure. On the other hand, it is possible that SymMol performs combinations of elements with the goal of group completeness. We have not found a routine explicitly dedicated to that in SymMol, but we also did not spend much time investigating the source code. In SOFI, the incomplete groups are explicitly labelled with + or - signs, which are related to the number of actual elements, and the expected number of elements for a PG identified via flowchart. These cases can optionally be explicitly made into complete groups, via combinations of the symmetry elements.

The comparison tests performed in this paper show that the developed SOFI algorithm is capable of identifying the PG symmetry in atomic structures. Among all the structures and tolerance values included in the test, there is no case of failure of SOFI, contrary to other algorithms (See Tab. S8). In all cases of disagreement between algorithms, SOFI is capable to find a PG of the same order, or higher, when computed with an appropriate tolerance value.

5 All results

The following Tables contain the results by each algorithm used, at each tolerance value, for each structure in the dataset. The first column reports the exact name of the structure as it appears in the dataset, the column marked "diff" contains an exclamation mark (!) if there is a disagreement between the algorithms, and the letter "M" if the PG assignment changes with raising the threshold value.

They are separated by dataset, as follows. Bin LJ results in Tab. [S9](#), [S10](#), [S11](#), [S12](#), [S13](#), and [S14](#). C60 and C60-Pacheco results in Tab [S15](#), and [S16](#). Dzugutov and Dzugutov_mod results in Tab. [S17](#), [S18](#), [S19](#), and [S20](#). Finnis-Sinclair results in Tab [S21](#) and [S22](#). Glue results in Tab [S23](#), and [S25](#). Gupta results in Tab [S27](#), [S28](#), [S29](#), [S30](#), [S31](#), [S32](#), [S33](#), [S34](#), [S35](#), [S36](#), [S37](#), and [S38](#). LJ results in Tab. [S39](#), and [S40](#). Morse results in Tab [S41](#), [S42](#), [S43](#), and [S44](#). Murrel-Mottram in Tab [S45](#), [S46](#), [S47](#), [S48](#). NaClN results in Tab. [S49](#), and [S50](#). Quantum LJ results in Tab. [S51](#). Sutton results in Tab [S52](#), [S53](#), and [S54](#).

name	SYVA		SymMol		libmsym			SOFI		diff	dot	product
	0.001	0.05	0.001	0.05	'default'	'medium'	'loose'	0.001	0.05			
5	D3h	D3h	D3h	D3h	D3h	D3h	D3h	D3h	D3h			1.00000
6	D4h	D4h	D4h	D4h	D4h	D4h	D4h	D4h	D4h			1.00000
7	D5h	D5h	D5h	D5h	D5h	D5h	D5h	D5h	D5h			1.00000
8	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
9	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v			1.00000
10	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v			1.00000
11	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v			1.00000
12	C5v	C5v	C5v	C5v	C5v	C5v	C5v	C5v	C5v			1.00000
13	Ih	Ih	Ih	Ih	Ih	Ih	Ih	Ih	Ih			0.99906
14	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v			1.00000
15	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
16	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
17	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v			1.00000
18	C5v	C5v	C5v	C5v	C5v	C5v	C5v	C5v	C5v			1.00000
19	D5h	D5h	D5h	D5h	D5h	D5h	D5h	D5h	D5h			1.00000
20	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v			1.00000
21	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
22	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
23	D3h	D3h	D3h	D3h	D3h	D3h	D3h	D3h	D3h			1.00000
24	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
25	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
26	Td	Td	Td	Td	Td	Td	Td	Td	Td			0.89941
27	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
28	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
29	D3h	D3h	D3h	D3h	D3h	D3h	-	D3h	D3h			1.00000
30	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v			1.00000
31	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
32	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v			1.00000
33	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
34	D5h	D5h	D5h	D5h	D5h	D5h	D5h	D5h	D5h			1.00000
35	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v			1.00000
36	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
37	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
38	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
39	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v			1.00000
40	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
41	C2	C2	C2	C2	C2	C2	C2	C2	C2			1.00000
42	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
43	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
44	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
45	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
46	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
47	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
48	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
49	Cs	Cs	Cs	Cs	Cs	Cs	-	Cs	Cs			1.00000
50	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
51	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v			1.00000
52	C3v	C3v	-	-	C3v	C3v	C3v	C3v	C3v			1.00000
53	C2v	C2v	C2v	C2v	C2v	C2v	-	C2v	C2v			1.00000
54	C5v	C5v	C5	C5	C5v	C5v	-	C5v	C5v	!		1.00000
55	Ih	Ih	Ci	Ci	Ih	Ih	Ih	Ih	Ih	!		0.98791
56	C3v	C3v	-	-	C3v	-	C3v	C3v	C3v			1.00000
57	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
58	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v			1.00000
59	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v			1.00000
60	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
61	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v			1.00000
62	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
63	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
64	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
65	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
66	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
67	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
68	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
69	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
70	C5v	C5v	C5	C5	C5v	C5v	-	C5v	C5v	!		1.00000
71	C5v	C5v	C5	C5	C5v	C5v	-	C5v	C5v	!		1.00000
72	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
73	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
74	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
75	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
76	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
77	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
78	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
79	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
80	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
81	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v			1.00000
82	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v			1.00000
83	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
84	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
85	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
86	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
87	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
88	C3v	C3v	-	-	C3v	C3v	C3v	C3v	C3v			1.00000
89	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
90	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
91	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
92	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
93	C2v	C2v	C2v	C2v	C2v	C2v	C1	C2v	C2v	!		1.00000
94	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
95	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
96	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
97	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
98	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
99	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
100	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000

Table S9: Dataset: bin_LJ/1.05, part 1.

name	SYVA		SymMol		libmsym			SOFI		diff	dot	product
	0.001	0.05	0.001	0.05	'default'	'medium'	'loose'	0.001	0.05			
5	D3h	D3h	D3h	D3h	D3h	D3h	D3h	D3h	D3h			1.00000
6	D4h	D4h	D4h	D4h	D4h	D4h	D4h	D4h	D4h			1.00000
7	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v			1.00000
8	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v			1.00000
9	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
10	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v			1.00000
11	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v			1.00000
12	C5v	C5v	C5v	C5v	C5v	C5v	C5v	C5v	C5v			1.00000
13	Ih	Ih	Ih	Ih	Ih	Ih	Ih	Ih	Ih			0.95688
14	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v			1.00000
15	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
16	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
17	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v			1.00000
18	C5v	C5v	C5v	C5v	C5v	C5v	C5v	C5v	C5v			1.00000
19	D5h	D5h	D5h	D5h	D5h	D5h	D5h	D5h	D5h			1.00000
20	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v			1.00000
21	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v			1.00000
22	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
23	C3v	C3v	C3v	C3v	C3v	C3v	-	C3v	C3v			1.00000
24	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
25	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
26	Td	Td	Td	Td	Td	Td	Td	Td	Td			0.99856
27	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
28	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
29	D3h	D3h	D3h	D3h	D3h	D3h	-	D3h	D3h			1.00000
30	C2v	C2v	C2v	C2v	C2v	C2v	-	C2v	C2v			1.00000
31	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
32	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v			1.00000
33	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
34	D5h	D5h	D5h	D5h	D5h	D5h	D5h	D5h	D5h			1.00000
35	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v			1.00000
36	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
37	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
38	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
39	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v			1.00000
40	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
41	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v			1.00000
42	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
43	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v			1.00000
44	C5v	C5v	C5v	C5v	C5v	C5v	C5v	C5v	C5v			1.00000
45	Ih	Ih	Ci	Ci	Ih	Ih	Ih	Ih	Ih	!		0.99949
46	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
47	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
48	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v			1.00000
49	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
50	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
51	D6d	D6d	C6	C6	D6d	D6d	-	D6d	D6d	!		1.00000
52	Cs	Cs	Cs	Cs	Cs	Cs	-	Cs	Cs			1.00000
53	C2	C2	C2	C2	C2	C2	-	C2	C2			1.00000
54	D3h	D3h	D3h	D3h	D3h	D3h	D3h	D3h	D3h			1.00000
55	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
56	C2	C2	C2	C2	C2	C2	C2	C2	C2			1.00000
57	Td	Td	Td	Td	Td	Td	Td	Td	Td			0.86483
58	C3v	C3v	-	-	C3v	C3v	C3v	C3v	C3v			1.00000
59	C2v	C2v	C2v	C2v	C2v	C2v	-	C2v	C2v			1.00000
60	C3v	C3v	-	-	C3v	C3v	-	C3v	C3v			1.00000
61	Td	Td	Td	Td	Td	Td	Td	Td	Td			0.88139
62	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
63	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
64	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
65	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
66	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
67	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
68	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
69	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
70	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
71	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
72	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
73	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
74	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
75	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
76	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
77	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
78	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
79	Cs	Cs	Cs	Cs	Cs	Cs	-	Cs	Cs			1.00000
80	C2	C2	C2	C2	C2	C2	-	C2	C2			1.00000
81	Cs	Cs	Cs	Cs	Cs	Cs	-	Cs	Cs			1.00000
82	S6	D3d	S6	S6	-	D3d	-	S6	D3d	!M		1.00000
83	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
84	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
85	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
86	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
87	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
88	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
89	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
90	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
91	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
92	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
93	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
94	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
95	C2v	C2v	C2v	C2v	C2v	-	C2v	C2v	C2v			1.00000
96	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
97	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
98	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
99	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
100	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000

Table S10: Dataset: bin_LJ/1.1, part 1.

name	SYVA		SymMol		libmsym			SOFI		diff	dot	product
	0.001	0.05	0.001	0.05	'default'	'medium'	'loose'	0.001	0.05			
5	D3h	D3h	D3h	D3h	D3h	D3h	D3h	D3h	D3h			1.00000
6	D4h	D4h	D4h	D4h	D4h	D4h	D4h	D4h	D4h			1.00000
7	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v			1.00000
8	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v			1.00000
9	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v			1.00000
10	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
11	C3v	C3v	-	-	C3v	C3v	-	C3v	C3v			1.00000
12	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
13	Ih	Ih	Ih	Ih	Ih	Ih	Ih	Ih	Ih			0.91992
14	C3v	C3v	C3v	C3v	C3v	-	C3v	C3v	C3v			1.00000
15	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v			1.00000
16	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
17	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v			1.00000
18	C5v	C5v	C5v	C5v	C5v	C5v	C5v	C5v	C5v			1.00000
19	D5h	D5h	D5h	D5h	D5h	D5h	D5h	D5h	D5h			1.00000
20	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v			1.00000
21	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v			1.00000
22	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
23	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v			1.00000
24	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
25	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
26	Td	Td	Td	Td	Td	Td	Td	Td	Td			0.94640
27	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
28	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
29	D3h	D3h	D3h	D3h	D3h	D3h	-	D3h	D3h			1.00000
30	C2v	C2v	C2v	C2v	C2v	C2v	-	C2v	C2v			1.00000
31	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
32	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v			1.00000
33	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
34	D5h	D5h	D5h	D5h	D5h	D5h	D5h	D5h	D5h			1.00000
35	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v			1.00000
36	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
37	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
38	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
39	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v			1.00000
40	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
41	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v			1.00000
42	C3v	C3v	-	-	C3v	C3v	C3v	C3v	C3v			1.00000
43	C2v	C2v	C2v	C2v	C2v	C2v	-	C2v	C2v			1.00000
44	C5v	C5v	C5	C5	C5v	C5v	-	C5v	C5v	!		1.00000
45	Ih	Ih	Ci	Ci	Ih	Ih	Ih	Ih	Ih	!		0.99851
46	C2v	C2v	C2v	C2v	C2v	C2v	-	C2v	C2v			1.00000
47	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
48	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v			1.00000
49	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
50	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v			1.00000
51	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
52	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
53	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
54	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
55	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
56	C2	C2	C2	C2	C2	C2	C2	C2	C2			1.00000
57	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
58	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
59	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
60	C2	C2	C2	C2	C2	C2	C2	C2	C2			1.00000
61	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
62	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
63	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
64	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
65	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
66	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
67	Cs	Cs	Cs	Cs	Cs	Cs	-	Cs	Cs			1.00000
68	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
69	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
70	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
71	C1	C1	C1	C1	C1	C1	Cs	C1	C1	!		0.00000
72	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
73	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
74	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
75	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
76	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
77	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
78	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
79	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
80	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
81	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
82	D2h	D2h	D2h	D2h	D2h	D2h	D2	D2h	D2h	!		1.00000
83	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
84	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
85	Cs	Cs	Cs	Cs	Cs	Cs	-	Cs	Cs			1.00000
86	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
87	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
88	Cs	Cs	Cs	Cs	Cs	Cs	-	Cs	Cs			1.00000
89	Cs	Cs	Cs	Cs	Cs	Cs	-	Cs	Cs			1.00000
90	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
91	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
92	C2h	C2h	C2h	C2h	C2h	-	-	C2h	C2h			1.00000
93	C1	C1	C1	C1	C1	C1	-	C1	C1			0.00000
94	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
95	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
96	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
97	C1	C1	C1	C1	C1	C1	-	C1	C1			0.00000
98	C2	C2	C2	C2	C2	C2	-	C2	C2			1.00000
99	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
100	D3	D3	D3	D3	D3	C3	C1	D3	D3	!		1.00000

Table S11: Dataset: bin_LJ/1.15, part 1.

name	SYVA		SymMol		libmsym			SOFI		diff	dot	product
	0.001	0.05	0.001	0.05	'default'	'medium'	'loose'	0.001	0.05			
5	D3h	D3h	D3h	D3h	D3h	D3h	D3h	D3h	D3h			1.00000
6	D4h	D4h	D4h	D4h	D4h	D4h	D4h	D4h	D4h			1.00000
7	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v			1.00000
8	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v			1.00000
9	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
10	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
11	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
12	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
13	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v			1.00000
14	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v			1.00000
15	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v			1.00000
16	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
17	C2	C2	C2	C2	C2	C2	C2	C2	C2			1.00000
18	C5v	C5v	C5v	C5v	C5v	C5v	C5v	C5v	C5v			1.00000
19	D5h	D5h	D5h	D5h	D5h	D5h	D5h	D5h	D5h			1.00000
20	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
21	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v			1.00000
22	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
23	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v			1.00000
24	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
25	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
26	C3v	C3v	-	-	C3v	C3v	C3v	C3v	C3v			1.00000
27	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
28	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
29	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
30	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
31	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
32	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
33	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
34	C5v	C5v	C5v	C5v	C5v	C5v	C5v	C5v	C5v			1.00000
35	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
36	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
37	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
38	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
39	C5v	C5v	C5	C5	C5v	C5v	C5v	C5v	C5v	!		1.00000
40	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
41	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v			1.00000
42	C3v	C3v	-	-	C3v	C3v	C3v	C3v	C3v			1.00000
43	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v			1.00000
44	C5v	C5v	C5	C5	C5v	C5v	-	C5v	C5v	!		1.00000
45	Ih	Ih	Ci	Ci	Ih	Ih	Ih	Ih	Ih	!		0.95604
46	C2v	C2v	C2v	C2v	C2v	C2v	-	C2v	C2v	!		1.00000
47	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
48	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v			1.00000
49	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
50	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v			1.00000
51	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
52	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
53	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
54	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
55	C5v	C5v	C5	C5	C5v	C5v	-	C5v	C5v	!		1.00000
56	D5h	D5h	C5	C5	D5h	D5h	-	D5h	D5h	!		1.00000
57	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
58	C2v	C2v	C2v	C2v	C2v	C2v	-	C2v	C2v			1.00000
59	Cs	Cs	Cs	Cs	Cs	Cs	-	Cs	Cs			1.00000
60	C1	C1	-	C1	C1	C1	C1	C1	C1			0.00000
61	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
62	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v			1.00000
63	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
64	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
65	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
66	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
67	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
68	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
69	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
70	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
71	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
72	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
73	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
74	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
75	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
76	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
77	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
78	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
79	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
80	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
81	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
82	D2h	D2h	D2h	D2h	D2h	D2h	D2	D2h	D2h	!		1.00000
83	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
84	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
85	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
86	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
87	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
88	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
89	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
90	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
91	C3v	C3v	-	-	C3v	C3v	-	C3v	C3v			1.00000
92	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
93	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
94	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
95	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
96	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
97	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
98	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
99	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
100	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000

Table S12: Dataset: bin_LJ/1.2, part 1.

name	SYVA		SymMol		libmsym			SOFI		diff	dot	product
	0.001	0.05	0.001	0.05	'default'	'medium'	'loose'	0.001	0.05			
5	D3h	D3h	D3h	D3h	D3h	D3h	D3h	D3h	D3h			1.00000
6	D4h	D4h	D4h	D4h	D4h	D4h	D4h	D4h	D4h			1.00000
7	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v			1.00000
8	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v			1.00000
9	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
10	D3h	D3h	D3h	-	D3h	-	D3h	D3h	D3h			1.00000
11	C2	C2	C2	C2	C2	C2	C2	C2	C2			1.00000
12	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
13	C5v	C5v	C5v	C5v	C5v	C5v	C5v	C5v	C5v			1.00000
14	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
15	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v			1.00000
16	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
17	C2	C2	C2	C2	C2	C2	-	C2	C2			1.00000
18	C5v	C5v	C5v	C5v	C5v	C5v	C5v	C5v	C5v			1.00000
19	D5h	D5h	D5h	D5h	D5h	D5h	D5h	D5h	D5h			1.00000
20	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
21	C2	C2	C2	C2	C2	C2	-	C2	C2			1.00000
22	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
23	C2v	C2v	C2v	C2v	C2v	C2v	-	C2v	C2v			1.00000
24	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
25	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
26	C3v	C3v	-	-	C3v	C3v	C3v	C3v	C3v			1.00000
27	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
28	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
29	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v			1.00000
30	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
31	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
32	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
33	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
34	C5v	C5v	C5v	C5v	C5v	C5v	C5v	C5v	C5v			1.00000
35	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
36	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
37	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
38	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
39	C5v	C5v	C5	C5	C5v	C5v	C5v	C5v	C5v	!		1.00000
40	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
41	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v			1.00000
42	C3v	C3v	-	-	C3v	C3v	C3v	C3v	C3v			1.00000
43	C2v	C2v	C2v	C2v	C2v	C2v	-	C2v	C2v			1.00000
44	C5v	C5v	C5	C5	C5v	C5v	-	C5v	C5v	!		1.00000
45	Ih	Ih	Ci	Ci	Ih	Ih	Ih	Ih	Ih	!		0.95488
46	C2v	C2v	C2v	C2v	C2v	C2v	-	C2v	C2v			1.00000
47	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
48	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v			1.00000
49	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
50	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v			1.00000
51	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
52	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
53	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
54	D5h	D5h	C5	C5	D5h	D5h	-	D5h	D5h	!		1.00000
55	C5v	C5v	C5	C5	C5v	C5v	-	C5v	C5v	!		1.00000
56	D5h	D5h	C5	C5	D5h	D5h	-	D5h	D5h	!		1.00000
57	Cs	Cs	Cs	Cs	Cs	Cs	-	Cs	Cs			1.00000
58	C2v	C2v	C2v	C2v	C2v	C2v	-	C2v	C2v			1.00000
59	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
60	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
61	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
62	D3h	D3h	D3h	D3h	D3h	D3h	-	D3h	D3h			1.00000
63	C3v	C3v	-	-	C3v	C3v	-	C3v	C3v			1.00000
64	D3h	D3h	D3h	D3h	D3h	D3h	-	D3h	D3h			1.00000
65	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
66	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
67	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
68	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
69	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
70	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
71	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
72	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
73	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
74	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
75	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
76	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
77	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
78	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v			1.00000
79	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
80	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
81	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
82	Cs	Cs	Cs	Cs	Cs	Cs	-	Cs	Cs			1.00000
83	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
84	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
85	C2v	C2v	C2v	C2v	C2v	-	-	C2v	C2v			1.00000
86	Cs	Cs	Cs	Cs	Cs	Cs	-	Cs	Cs			1.00000
87	C2v	C2v	C2v	C2v	C2v	C2v	-	C2v	C2v			1.00000
88	Cs	Cs	Cs	Cs	Cs	Cs	-	Cs	Cs			1.00000
89	C2v	C2v	C2v	C2v	C2v	C2v	-	C2v	C2v			1.00000
90	Cs	Cs	Cs	Cs	Cs	Cs	-	Cs	Cs			1.00000
91	C2v	C2v	C2v	C2v	C2v	-	-	C2v	C2v			1.00000
92	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
93	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
94	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
95	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
96	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
97	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
98	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
99	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
100	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000

Table S13: Dataset: bin_LJ/1.25, part 1.

name	SYVA		SymMol		libmsym			SOFI		diff	dot	product
	0.001	0.05	0.001	0.05	'default'	'medium'	'loose'	0.001	0.05			
5	D3h	D3h	D3h	D3h	D3h	D3h	D3h	D3h	D3h			1.00000
6	D4h	D4h	D4h	D4h	D4h	D4h	D4h	D4h	D4h			1.00000
7	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v			1.00000
8	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v			1.00000
9	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v			1.00000
10	D3h	D3h	D3h	D3h	D3h	D3h	D3h	D3h	D3h			1.00000
11	D4d	D4d	S8	S8	D4d	D4d	D4d	D4d	D4d	!		1.00000
12	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
13	C5v	C5v	C5v	C5v	C5v	C5v	C5v	C5v	C5v			1.00000
14	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
15	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
16	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v			1.00000
17	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
18	C5v	C5v	C5v	C5v	C5v	C5v	C5v	C5v	C5v			1.00000
19	D5h	D5h	D5h	D5h	D5h	D5h	D5h	D5h	D5h			1.00000
20	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
21	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
22	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
23	D3h	D3h	D3h	D3h	D3h	D3h	D3h	D3h	D3h			1.00000
24	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
25	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
26	C3v	C3v	-	-	C3v	C3v	C3v	C3v	C3v			1.00000
27	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
28	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
29	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v			1.00000
30	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
31	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
32	C2	C2	C2	C2	C2	C2	C2	C2	C2			1.00000
33	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
34	C5v	C5v	C5v	C5v	C5v	C5v	C5v	C5v	C5v			1.00000
35	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
36	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
37	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
38	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
39	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v			1.00000
40	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
41	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
42	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v			1.00000
43	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v			1.00000
44	C5v	C5v	C5	C5	C5v	-	-	C5v	C5v	!		1.00000
45	Ih	Ih	Ci	Ci	Ih	Ih	Ih	Ih	Ih	!		0.96564
46	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
47	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
48	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v			1.00000
49	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
50	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
51	Cs	Cs	Cs	Cs	Cs	Cs	-	Cs	Cs			1.00000
52	Cs	Cs	Cs	Cs	Cs	Cs	-	Cs	Cs			1.00000
53	Cs	Cs	Cs	Cs	Cs	Cs	-	Cs	Cs			1.00000
54	D5h	D5h	C5	C5	D5h	D5h	-	D5h	D5h	!		1.00000
55	C5v	C5v	C5	C5	C5v	C5v	-	C5v	C5v	!		1.00000
56	D5h	D5h	C5	C5	D5h	D5h	-	D5h	D5h	!		1.00000
57	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
58	C2v	C2v	C2v	C2v	C2v	C2v	-	C2v	C2v			1.00000
59	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
60	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v			1.00000
61	Cs	Cs	Cs	Cs	Cs	Cs	-	Cs	Cs			1.00000
62	D3h	D3h	D3h	D3h	D3h	D3h	-	D3h	D3h			1.00000
63	C3v	C3v	-	-	C3v	C3v	-	C3v	C3v			1.00000
64	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
65	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
66	Td	Td	Td	Td	Td	Td	Td	Td	Td			0.89647
67	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v			1.00000
68	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
69	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
70	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
71	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
72	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
73	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
74	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v			1.00000
75	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
76	C2	C2	C2	C2	C2	C2	C2	C2	C2			1.00000
77	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
78	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v			1.00000
79	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
80	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v			1.00000
81	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
82	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
83	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
84	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
85	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
86	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v			1.00000
87	Cs	Cs	Cs	Cs	Cs	Cs	-	Cs	Cs			1.00000
88	D6h	D6h	D6h	D6h	D6h	D6h	-	D6h	D6h			1.00000
89	Cs	Cs	Cs	Cs	Cs	Cs	-	Cs	Cs			1.00000
90	C2v	C2v	C2v	C2v	C2v	C2v	-	C2v	C2v			1.00000
91	C2v	C2v	C2v	C2v	C2v	C2v	-	C2v	C2v			1.00000
92	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
93	C2	C2	C2	C2	C2	C2	-	C2	C2			1.00000
94	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
95	C2v	C2v	C2v	C2v	C2v	-	-	C2v	C2v			1.00000
96	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
97	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
98	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
99	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
100	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000

Table S14: Dataset: bin_LJ/1.3, part 1.

name	SYVA		SymMol		libmsym			SOFI		diff	dot product
	0.001	0.05	0.001	0.05	'default'	'medium'	'loose'	0.001	0.05		
5	D3h	D3h	D3h	D3h	D3h	D3h	D3h	D3h	D3h		1.00000
6	Oh	Oh	Oh	Oh	Oh	Oh	Oh	Oh	Oh		1.00000
7	D5h	D5h	D5h	D5h	D5h	D5h	D5h	D5h	D5h		1.00000
8	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
9	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
10	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v		1.00000
11	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
12	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	!	1.00000
13	Ih	Ih	Ih	Ih	Ih	Ih	Ih	Ih	Ih		0.98239
14	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
15	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
16	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
17	D2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
18	D5h	D5h	C5	C5	D5h	D5h	D5h	D5h	D5h	!	1.00000
19	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
20	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
21	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
22	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
23	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
24	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
25	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v		1.00000
26	D3h	D3h	D3h	D3h	D3h	D3h	D3h	D3h	D3h		1.00000
27	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
28	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
29	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
30	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
31	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
32	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
33	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
34	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
35	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
36	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
37	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
38	Oh	Oh	Oh	Oh	Oh	Oh	Oh	Oh	Oh		0.99846
39	C4v	C4v	C4	C4	C4v	C4v	C4v	C4v	C4v	!	1.00000
40	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
41	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v		1.00000
42	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
43	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
44	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
45	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
46	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
47	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
48	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
49	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
50	D3h	D3h	D3h	D3h	D3h	D3h	D3h	D3h	D3h		1.00000
51	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
52	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
53	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
54	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
55	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
56	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
57	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
58	D3h	D3h	D3h	D3h	D3h	D3h	-	D3h	D3h		1.00000
59	Td	Td	Td	Td	Td	Td	Td	Td	Td		0.99983
60	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
61	C3v	C3v	C3v	C3v	C3v	-	-	C3v	C3v		1.00000
62	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
63	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
64	C2v	C2v	C2v	C2v	C2v	C2v	-	C2v	C2v		1.00000
65	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
66	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
67	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
68	C3v	C3v	-	-	C3v	C3v	C3v	C3v	C3v		1.00000
69	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
70	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
71	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
72	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
73	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
74	C5v	C5v	C5	C5	C5v	-	-	C5v	C5v	!	1.00000
75	D5h	D5h	D5h	C5	D5h	-	-	D5h	D5h	!	1.00000
76	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
77	C2v	C2v	C2v	C2v	C2v	C2v	Cs	C2v	C2v	!	1.00000
78	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
79	D3h	D3h	-	-	D3h	D3h	D3h	D3h	D3h		1.00000
80	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000

Table S15: Dataset: C60, part 1.

name	SYVA		SymMol		libmsym			SOFI		diff	dot product
	0.001	0.05	0.001	0.05	'default'	'medium'	'loose'	0.001	0.05		
3	D3h	D3h	D3h	D3h	D3h	D3h	D3h	D3h	D3h		1.00000
4	Td	Td	Td	Td	Td	Td	Td	Td	Td		0.99800
5	D3h	D3h	D3h	D3h	D3h	D3h	D3h	D3h	D3h		1.00000
6	Oh	Oh	Oh	Oh	Oh	Oh	Oh	Oh	Oh		0.97056
7	D5h	D5h	D5h	D5h	D5h	D5h	D5h	D5h	D5h		1.00000
8	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
9	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
10	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v		1.00000
11	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
12	C5v	C5v	C5v	C5v	C5v	C5v	C5v	C5v	C5v		1.00000
13	Ih	Ih	Ih	Ih	Ih	Ih	Ih	Ih	Ih		0.97445
14	C3v	-	C3v	C3v	C3v	C3v	C3v	C3v	C3v		1.00000
15	C2v	-	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
16	C2v	-	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
17	C2v	-	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
18	D5h	-	C5	C5	D5h	D5h	D5h	D5h	D5h	!	1.00000
19	C2v	-	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
20	C2v	-	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
21	Cs	-	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
22	C1	-	C1	C1	C1	C1	C1	C1	C1		0.00000
23	Cs	-	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
24	C1	-	C1	C1	C1	C1	C1	C1	C1		0.00000
25	C3v	-	C3v	C3v	C3v	C3v	C3v	C3v	C3v		1.00000
26	C2v	-	C2v	C2v	C2v	C2v	-	C2v	C2v		1.00000
27	Cs	-	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
28	Cs	-	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
29	D5h	-	C5	C5	D5h	D5h	D5h	D5h	D5h	!	1.00000
30	C2v	-	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
31	C2v	-	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
32	C1	-	C1	C1	C1	C1	C1	C1	C1		0.00000
33	C2v	-	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
34	C1	-	C1	C1	C1	C1	C1	C1	C1		0.00000
35	C2v	-	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
36	C1	-	-	C1	C1	C1	C1	C1	C1		0.00000
37	C2v	-	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
38	Oh	-	Oh	Oh	Oh	Oh	Oh	Oh	Oh		1.00000
39	C4v	-	C4	C4	C4v	C4v	C4v	C4v	C4v	!	1.00000
40	C2v	-	C2v	C2v	C2v	C2v	-	C2v	C2v		1.00000
41	C2v	-	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
42	Cs	-	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
43	C2v	-	C2v	C2v	C2v	C2v	Cs	C2v	C2v	!	1.00000
44	Cs	-	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
45	C2v	-	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
46	Cs	-	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
47	C2v	-	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
48	C2v	-	C2v	C2v	C2v	C2v	-	C2v	C2v		1.00000
49	Cs	-	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
50	D3h	-	D3h	D3h	D3h	D3h	D3h	D3h	D3h		1.00000
51	Cs	-	Cs	Cs	Cs	Cs	-	Cs	Cs		1.00000
52	C2v	-	C2v	C2v	C2v	C2v	-	C2v	C2v		1.00000
53	Cs	-	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
54	C2v	-	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
55	C2v	-	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
56	Cs	-	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
57	C2v	-	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
58	D3h	-	D3h	D3h	D3h	D3h	-	D3h	D3h		1.00000
59	Td	-	Td	Td	Td	Td	Td	Td	Td		0.99667
60	Cs	-	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
61	C3v	-	C3v	C3v	C3v	-	-	C3v	C3v		1.00000
62	C1	-	C1	C1	C1	C1	C1	C1	C1		0.00000
63	Cs	-	Cs	Cs	Cs	Cs	-	Cs	Cs		1.00000
64	C2v	-	C2v	C2v	C2v	C2v	-	C2v	C2v		1.00000
65	C2v	-	C2v	C2v	C2v	C2v	-	C2v	C2v		1.00000
66	Cs	-	Cs	Cs	Cs	Cs	-	Cs	Cs		1.00000
67	C2v	-	C2v	C2v	C2v	C2v	-	C2v	C2v		1.00000
68	C1	-	C1	C1	C1	C1	C1	C1	C1		0.00000
69	C1	-	C1	C1	C1	C1	C1	C1	C1		0.00000
70	Cs	-	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
71	C2v	-	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
72	C1	-	C1	C1	C1	C1	C1	C1	C1		0.00000
73	Cs	-	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
74	C5v	-	C5	C5	C5v	-	-	C5v	C5v	!	1.00000
75	D5h	-	D5h	D5h	D5h	-	-	D5h	D5h		1.00000
76	Cs	-	Cs	Cs	Cs	Cs	-	Cs	Cs		1.00000
77	C2v	-	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
78	C1	-	C1	C1	C1	C1	C1	C1	C1		0.00000
79	D3h	-	D3h	-	D3h	D3h	-	D3h	D3h		1.00000
80	Cs	-	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
81	C2v	-	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
82	C2v	-	C2v	C2v	C2v	C2v	-	C2v	C2v		1.00000
83	C2v	-	C2v	C2v	C2v	C2v	-	C2v	C2v		1.00000
84	C2v	-	C2v	C2v	C2v	C2v	-	C2v	C2v		1.00000
85	C3v	-	C3v	C3v	C3v	C3v	C3v	C3v	C3v		1.00000
86	C3v	-	C3v	C3v	C3v	C3v	C3v	C3v	C3v		1.00000
87	Cs	-	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
88	Cs	-	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
89	Cs	-	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
90	Cs	-	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
91	C3v	-	C3v	C3v	C3v	C3v	C3v	C3v	C3v		1.00000
92	Cs	-	Cs	Cs	Cs	Cs	-	Cs	Cs		1.00000
93	Cs	-	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
94	Cs	-	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
95	C2v	-	C2v	C2v	C2v	C2v	-	C2v	C2v		1.00000
96	Cs	-	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
97	C1	-	C1	C1	C1	C1	C1	C1	C1		0.00000
98	Cs	-	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
99	Cs	-	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
100	Td	-	Td	Td	Td	Td	Td	Td	Td		0.97270
101	D5h	-	D5h	C5	D5h	D5h	-	D5h	D5h	!	1.00000
102	C2v	-	C2v	C2v	C2v	C2v	-	C2v	C2v		1.00000
103	Cs	-	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
104	C2v	-	C2v	C2v	C2v	C2v	-	C2v	C2v		1.00000
105	Cs	-	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000

Table S16: Dataset: C60_pacheco, part 1.

name	SYVA		SymMol		libmsym			SOFI		diff	dot product
	0.001	0.05	0.001	0.05	'default'	'medium'	'loose'	0.001	0.05		
3	D3h	D3h	D3h	D3h	D3h	D3h	D3h	D3h	D3h		1.00000
4	Td	Td	Td	Td	Td	Td	Td	Td	Td		0.99481
5	D3h	D3h	D3h	D3h	D3h	D3h	D3h	D3h	D3h		1.00000
6	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
7	D5h	D5h	D5h	D5h	D5h	D5h	D5h	D5h	D5h		1.00000
8	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
9	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
10	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v		1.00000
11	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
12	C5v	C5v	C5v	C5v	C5v	C5v	C5v	C5v	C5v		1.00000
13	Ih	Ih	Ci	Ci	Ih	Ih	Ih	Ih	Ih	!	0.99682
14	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v		1.00000
15	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
16	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
17	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
18	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
19	D5h	D5h	C5	C5	D5h	D5h	D5h	D5h	D5h	!	1.00000
20	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
21	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
22	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
23	D3h	D3h	D3h	D3h	D3h	D3h	-	D3h	D3h		1.00000
24	Cs	Cs	Cs	Cs	Cs	Cs	-	Cs	Cs		1.00000
25	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
26	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
27	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
28	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
29	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
30	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
31	D5h	D5h	D5h	C5	D5h	D5h	-	D5h	D5h	!	1.00000
32	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
33	D3d	D3d	S6	S6	D3d	D3d	-	D3d	D3d	!	1.00000
34	C1	C1	C1	C1	C1	C1	-	C1	C1		0.00000
35	C2	C2	C2	C2	C2	C2	C2	C2	C2		1.00000
36	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
37	D5d	D5d	S10	S10	D5d	D5d	-	D5d	D5d	!	1.00000
38	D6h	D6h	D6h	D6h	D6h	D6h	-	D6h	D6h		1.00000
39	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
40	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
41	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
42	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
43	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
44	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
45	C2	C2	C2	C2	C2	C2	C1	C2	C2	!	1.00000
46	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
47	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
48	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
49	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
50	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
51	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
52	C2	C2	C2	C2	C2	C2	-	C2	C2		1.00000
53	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
54	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
55	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
56	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
57	Td	Td	Td	Td	Td	Td	Td	Td	Td		0.81200
58	C2h	C2h	-	C2h	C2h	C2h	-	C2h	C2h		1.00000
59	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
60	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
61	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
62	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
63	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
64	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
65	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
66	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
67	D2d	D2d	S4	S4	D2d	D2d	-	D2d	D2d	!	1.00000
68	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
69	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
70	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
71	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
72	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
73	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
74	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
75	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
76	D3d	D3d	S6	S6	D3d	D3d	-	D3d	D3d	!	1.00000
77	Cs	Cs	Cs	Cs	Cs	Cs	-	Cs	Cs		1.00000
78	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
79	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
80	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
81	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
82	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
83	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
84	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
85	C2	C2	C2	C2	C2	C2	-	C2	C2		1.00000
86	D3	D3	D3	D3	D3	D3	-	D3	D3		1.00000
87	C1	C1	-	C1	C1	C1	C1	C1	C1		0.00000
88	C2	C2	C2	C2	C2	C2	-	C2	C2		1.00000
89	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
90	C2	C2	C2	C2	C2	C2	-	C2	C2		1.00000
91	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
92	D3	D3	D3	D3	D3	D3	C2	D3	D3	!	1.00000
93	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
94	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
95	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
96	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
97	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
98	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
99	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
100	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000

Table S17: Dataset: dzugutov, part 1.

name	SYVA		SymMol		libmsym			SOFI		diff	dot	product
	0.001	0.05	0.001	0.05	'default'	'medium'	'loose'	0.001	0.05			
222	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
223	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
224	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
225	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
226	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
227	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
228	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
229	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
230	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
231	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
232	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
233	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
234	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
235	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
236	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
237	C2v	C2v	C2v	C2v	C2v	C2v	-	C2v	C2v			1.00000
238	Cs	Cs	Cs	Cs	Cs	Cs	-	Cs	Cs			1.00000
239	C2v	C2v	C2v	C2v	C2v	C2v	-	C2v	C2v			1.00000
240	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
241	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
242	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
243	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
244	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
245	Cs	Cs	Cs	Cs	Cs	Cs	-	Cs	Cs			1.00000
246	Cs	Cs	Cs	Cs	Cs	Cs	-	Cs	Cs			1.00000
247	Cs	Cs	Cs	Cs	Cs	Cs	-	Cs	Cs			1.00000
248	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
249	Cs	Cs	Cs	Cs	Cs	Cs	-	Cs	Cs			1.00000
250	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000

Table S20: Dataset: dzugutov_mod, part 3.

name	SYVA		SymMol		libmsym			SOFI		diff	dot product
	0.001	0.05	0.001	0.05	'default'	'medium'	'loose'	0.001	0.05		
112	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
113	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
114	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
115	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
116	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
117	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
118	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
119	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
120	C2	C2	C2	C2	C2	-	C2	C2	C2		1.00000
121	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
122	C2	C2	C2	C2	C2	-	C2	C2	C2		1.00000
123	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
124	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
125	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
126	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
127	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
128	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
129	C3	C3	C3	C3	-	-	C3	C3	C3		1.00000
130	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
131	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
132	C2	C2	C2	C2	C2	C2	C1	C2	C2	!	1.00000
133	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
134	C2	C2v	C2	C2v	C2	C2v	-	C2	C2	!	1.00000
135	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
136	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
137	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
138	C2v	C2v	C2v	C2v	C2v	C2v	-	C2v	C2v		1.00000
139	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
140	Cs	Cs	Cs	Cs	Cs	Cs	-	Cs	Cs		1.00000
141	Cs	Cs	Cs	Cs	Cs	Cs	-	Cs	Cs		1.00000
142	C2v	C2v	C2v	C2v	C2v	-	C2v	C2v	C2v		1.00000
143	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
144	C2v	C2v	C2v	C2v	C2v	C2v	-	C2v	C2v		1.00000
145	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
146	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
147	Ih	Ih	Ih	Ih	Ih	Ih	Ih	Ih	Ih		0.98283
148	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
149	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
150	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
151	C2v	C2v	C2v	C2v	C2v	-	-	C2v	C2v		1.00000
152	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
153	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
154	Cs	Cs	Cs	Cs	Cs	-	-	Cs	Cs		1.00000
155	Cs	Cs	Cs	Cs	Cs	Cs	-	Cs	Cs		1.00000
156	C3v	C3v	-	-	C3v	C3v	-	C3v	C3v		1.00000
157	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
158	C2	C2	C2	C2	C2	-	-	C2	C2		1.00000
159	Cs	Cs	Cs	Cs	Cs	-	Cs	Cs	Cs		1.00000
160	D3h	D3h	D3h	D3h	D3h	D3h	-	D3h	D3h		1.00000
161	C1	C1	C1	C1	C1	C1	-	C1	C1		0.00000
162	Cs	Cs	Cs	Cs	Cs	Cs	-	Cs	Cs		1.00000
163	Cs	Cs	Cs	Cs	Cs	Cs	-	Cs	Cs		1.00000
164	C2	C2	C2	C2	C2	C2	-	C2	C2		1.00000
165	Cs	Cs	Cs	Cs	Cs	-	-	Cs	Cs		1.00000
166	Cs	Cs	Cs	Cs	Cs	-	-	Cs	Cs		1.00000
167	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
168	C2	C2	C2	C2	C2	C2	-	C2	C2		1.00000
169	C1	C1	C1	C1	C1	C1	-	C1	C1		0.00000
170	Cs	Cs	Cs	Cs	Cs	Cs	-	Cs	Cs		1.00000
171	Cs	Cs	Cs	Cs	Cs	Cs	-	Cs	Cs		1.00000
172	D3h	D3h	D3h	D3h	D3h	C3h	-	D3h	D3h	!	1.00000
173	C2	C2	C2	C2	C2	C2	-	C2	C2		1.00000
174	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
175	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
176	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
177	C2	C2	C2	C2	C2	C2	-	C2	C2		1.00000
178	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
179	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
180	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
181	Cs	C2v	Cs	Cs	Cs	Cs	-	Cs	Cs	!	1.00000
182	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
183	C1	Cs	C1	C1	C1	Cs	-	C1	C1	!	0.00000
184	C2	C2	C2	C2	C2	C2	C2	C2	C2		1.00000
185	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
186	C2	C2	C2	C2	C2	C2	C2	C2	C2		1.00000
187	C1	C1	C1	C1	C1	C1	-	C1	C1		0.00000
188	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
189	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
190	C2	C2	C2	C2	C2	C2	-	C2	C2		1.00000

Table S24: Dataset: glue_Al, part 2.

name	SYVA		SymMol		libmsym			SOFI		diff	dot product
	0.001	0.05	0.001	0.05	'default'	'medium'	'loose'	0.001	0.05		
112	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
113	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
114	C2	C2	C2	C2	C2	C2	C1	C2	C2	!	1.00000
115	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
116	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
117	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
118	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
119	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
120	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
121	C2	C2	C2	C2	C2	C2	C2	C2	C2		1.00000
122	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
123	C2	C2	C2	C2	C2	-	C2	C2	C2		1.00000
124	C2	C2	C2	C2	C2	C2	-	C2	C2		1.00000
125	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
126	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
127	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
128	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
129	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
130	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
131	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
132	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
133	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
134	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
135	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
136	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
137	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
138	C2	C2	C2	C2	C2	C2	C1	C2	C2	!	1.00000
139	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
140	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
141	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
142	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
143	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
144	C2	C2	C2	C2	C2	C2	-	C2	C2		1.00000
145	Cs	Cs	Cs	Cs	Cs	Cs	-	Cs	Cs		1.00000
146	C2	C2	C2	C2	C2	C2	-	C2	C2		1.00000
147	Cs	Cs	Cs	Cs	Cs	Cs	C1	Cs	Cs		1.00000
148	D3d	D3d	S6	S6	D3d	D3d	-	D3d	D3d	!	1.00000
149	C1	-	C1	C1	C1	C1	C1	C1	C1		0.00000
150	Cs	-	Cs	Cs	Cs	Cs	-	Cs	Cs		1.00000
151	C1	-	C1	C1	C1	C1	C1	C1	C1		0.00000
152	C2	-	C2	C2	C2	C2	-	C2	C2		1.00000
153	C1	-	C1	C1	C1	C1	C1	C1	C1		0.00000
154	C2	-	C2	C2	C2	C2	-	C2	C2		1.00000
155	C1	-	C1	C1	C1	C1	C1	C1	C1		0.00000
156	C2	-	C2	C2	C2	C2	C2	C2	C2		1.00000
157	C1	-	C1	C1	C1	C1	C1	C1	C1		0.00000
158	C2	-	C2	C2	C2	C2	C2	C2	C2		1.00000
159	C1	-	C1	C1	C1	C1	C1	C1	C1		0.00000
160	C1	-	C1	C1	C1	C1	C1	C1	C1		0.00000

Table S26: Dataset: glue_Pb, part 2.

name	SYVA		SymMol		libmsym			SOFI		diff	dot	product
	0.001	0.05	0.001	0.05	'default'	'medium'	'loose'	0.001	0.05			
112	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
113	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
114	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
115	C2	C2	C2	C2	C2	C2	C2	C2	C2			1.00000
116	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
117	C1	C1	C1	C1	C1	C1	C1	-	C1			0.00000
118	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
119	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
120	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
121	C2	C2	C2	C2	C2	C2	-	C2	C2			1.00000
122	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
123	C2v	C2v	C2v	C2v	C2v	C2v	-	C2v	C2v			1.00000
124	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
125	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000

Table S28: Dataset: gupta/Cd, part 2.

name	SYVA		SymMol		libmsym			SOFI		diff	dot	product
	0.001	0.05	0.001	0.05	'default'	'medium'	'loose'	0.001	0.05			
112	Cs	Cs	Cs	Cs	Cs	Cs	-	Cs	Cs		1.00000	
113	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000	
114	Cs	Cs	C1	Cs	Cs	Cs	Cs	Cs	Cs	!	1.00000	
115	C5v	C5v	C5v	C5v	C5v	-	-	C5v	C5v		1.00000	
116	C5v	C5v	C5v	C5v	C5v	-	-	C5v	C5v		1.00000	
117	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000	
118	Cs	Cs	C1	Cs	Cs	Cs	-	Cs	Cs	!	1.00000	
119	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000	
120	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000	
121	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000	
122	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000	
123	Cs	Cs	Cs	Cs	Cs	C1	Cs	Cs	Cs	!	1.00000	
124	Cs	Cs	C1	Cs	Cs	Cs	Cs	Cs	Cs	!	1.00000	
125	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000	
126	Cs	Cs	C1	Cs	Cs	Cs	Cs	Cs	Cs	!	1.00000	
127	C2v	C2v	C2	C2v	C2v	C2v	-	C2v	C2v	!	1.00000	
128	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000	
129	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000	
130	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000	
131	C2v	C2v	C2v	C2v	C2v	C2v	-	C2v	C2v		1.00000	
132	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000	
133	Cs	Cs	C1	Cs	Cs	Cs	Cs	Cs	Cs	!	1.00000	
134	C3v	C3v	-	-	C3v	C3v	C3v	C3v	C3v		1.00000	
135	Cs	Cs	Cs	Cs	Cs	Cs	-	Cs	Cs		1.00000	
136	Cs	Cs	C1	Cs	Cs	Cs	-	Cs	Cs	!	1.00000	
137	C3v	C3v	C3v	C3v	C3v	C3v	-	C3v	C3v		1.00000	
138	C3v	C3v	-	-	C3v	C3v	-	C3v	C3v		1.00000	
139	C2v	C2v	C1	C2v	C2v	C2v	-	C2v	C2v	!	1.00000	
140	Cs	Cs	C1	Cs	Cs	Cs	-	Cs	Cs	!	0.99991	
141	C5v	C5v	C1	C5v	C5v	-	-	C5v	C5v	!	0.99999	
142	Cs	Cs	C1	C1	Cs	Cs	C1	Cs	Cs	!	0.99983	
143	C2v	C2v	C2v	C2v	C2v	-	-	C2v	C2v		1.00000	
144	C3v	C3v	-	-	C3v	-	C3v	C3v	C3v		1.00000	
145	C2v	C2v	C1	C2v	C2v	C2v	-	C2v	C2v	!	1.00000	
146	C5v	C5v	C1	C5	C5v	-	-	C5v	C5v	!	1.00000	
147	Ih	Ih	Ih	Ih	Ih	Ih	Ih	Ih	Ih		0.98344	
148	Cs	Cs	C1	C1	Cs	-	-	Cs	Cs	!	0.99509	
149	Cs	Cs	C1	Cs	C1	Cs	-	Cs	Cs	!	1.00000	
150	C3v	C3v	-	-	C3v	C3v	-	C3v	C3v		1.00000	
151	Cs	Cs	C1	Cs	Cs	-	Cs	Cs	Cs	!	0.99998	
152	Cs	Cs	Cs	Cs	Cs	-	-	Cs	Cs		1.00000	
153	C3v	C3v	-	-	C3v	-	C3v	C3v	C3v		1.00000	
154	C2v	C2v	C2	C2v	C2v	C2v	-	C2v	C2v	!	1.00000	
155	Cs	Cs	C1	Cs	Cs	Cs	-	Cs	Cs	!	1.00000	
156	Cs	Cs	C1	Cs	Cs	Cs	Cs	Cs	Cs	!	1.00000	
157	C2v	C2v	C2	C2v	C2v	C2v	-	C2v	C2v	!	1.00000	
158	Cs	Cs	C1	Cs	Cs	Cs	Cs	Cs	Cs	!	1.00000	
159	C2v	C2v	C2v	C2v	C2v	C2v	-	C2v	C2v		1.00000	
160	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000	
161	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000	
162	Cs	Cs	C1	Cs	Cs	Cs	Cs	Cs	Cs	!	1.00000	
163	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000	
164	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000	
165	Cs	Cs	C1	Cs	Cs	Cs	Cs	Cs	Cs	!	1.00000	
166	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000	
167	Cs	Cs	Cs	Cs	Cs	Cs	-	Cs	Cs		1.00000	
168	C3v	C3v	C3v	C3v	C3v	C3v	-	C3v	C3v		1.00000	
169	C2	C2	C1	C2	C2	C2	-	C2	C2	!	1.00000	
170	C2v	C2v	C2v	C2v	C2	C2v	C2v	C2v	C2v	!	1.00000	
171	Cs	Cs	C1	Cs	Cs	Cs	-	Cs	Cs	!	1.00000	
172	C5v	C5v	C5	C5	C5v	C5v	-	C5v	C5v	!	1.00000	
173	C5v	C5v	C5v	C5v	C5v	C5v	-	C5v	C5v	!	1.00000	
174	Cs	Cs	C1	Cs	Cs	Cs	-	C1	Cs	!M	1.00000	
175	C1	Cs	C1	Cs	Cs	Cs	-	C1	Cs	!M	1.00000	
176	Cs	Cs	C1	Cs	Cs	Cs	-	Cs	Cs	!	1.00000	
177	C5v	C5v	C5v	C5v	C5v	-	-	C5v	C5v		1.00000	
178	C5v	C5v	C5	C5	C5v	-	-	C5v	C5v	!	1.00000	
179	C1	Cs	C1	Cs	Cs	Cs	Cs	C1	Cs	!M	1.00000	
180	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000	
181	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000	
182	Cs	Cs	Cs	Cs	Cs	C1	-	Cs	Cs	!	1.00000	
183	-	C2v	C1	C2v	C1	C2v	-	C2	C2v	!M	1.00000	
184	C1	Cs	C1	Cs	Cs	Cs	Cs	C1	Cs	!M	1.00000	
185	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000	
186	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000	
187	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000	
188	Cs	Cs	C1	Cs	C1	Cs	Cs	Cs	Cs	!	1.00000	
189	C1	Cs	C1	Cs	C1	C1	Cs	C1	Cs	!M	1.00000	
190	D5	D5h	D5	C5	C5v	D5h	-	D5h	D5h	!	1.00000	
191	C5v	C5v	C5v	C5	-	-	-	C5v	C5v	!	1.00000	
192	D5h	D5h	D5h	C5	D5h	D5h	-	D5h	D5h	!	1.00000	
193	C2v	C2v	C2v	C2v	C2v	-	-	C2v	C2v		1.00000	
194	Cs	Cs	Cs	Cs	C1	C1	-	Cs	Cs	!	1.00000	
195	C2v	C2v	C2v	C2v	C2v	C2v	-	C2v	C2v		1.00000	
196	C2v	C2v	Cs	C2v	C2v	C1	-	C2v	C2v	!	1.00000	
197	Cs	Cs	Cs	Cs	Cs	Cs	-	Cs	Cs		1.00000	
198	C2	C2	C1	C2	C2	C1	-	C2	C2	!	1.00000	
199	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000	
200	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000	

Table S30: Dataset: gupta/Co, part 2.

name	SYVA		SymMol		libmsym			SOFI		diff	dot product
	0.001	0.05	0.001	0.05	'default'	'medium'	'loose'	0.001	0.05		
332	-	-	Cs	Cs	Cs	Cs	-	Cs	Cs		1.00000
333	-	-	C1	C1	C1	C1	C1	C1	C1		0.00000
334	-	-	Cs	Cs	Cs	Cs	-	Cs	Cs		1.00000
335	-	-	C1	C1	C1	C1	C1	C1	C1		0.00000
336	-	-	Cs	Cs	Cs	Cs	-	Cs	Cs		1.00000
337	-	-	C1	C1	C1	C1	C1	C1	C1		0.00000
338	-	-	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
339	-	-	Cs	Cs	Cs	Cs	-	Cs	Cs		1.00000
340	-	-	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
341	-	-	C1	C1	C1	C1	C1	C1	C1		0.00000
342	-	-	C1	C1	C1	C1	C1	C1	C1		0.00000
343	-	-	C1	C1	C1	C1	C1	C1	C1		0.00000
344	-	-	Cs	Cs	Cs	Cs	-	Cs	Cs		1.00000
345	-	-	Cs	Cs	Cs	Cs	-	Cs	Cs		1.00000
346	-	-	C1	C1	C1	C1	C1	C1	C1		0.00000
347	-	-	Cs	Cs	Cs	-	-	Cs	Cs		1.00000
348	-	-	C5v	C5v	C5v	-	-	C5v	C5v		1.00000
349	-	-	Cs	Cs	Cs	Cs	-	Cs	Cs		1.00000
350	-	-	Cs	Cs	Cs	Cs	-	Cs	Cs		1.00000
351	-	-	Cs	Cs	Cs	-	-	Cs	Cs		1.00000
352	-	-	C1	C1	C1	C1	-	C1	C1		0.00000
353	-	-	Cs	Cs	Cs	Cs	-	Cs	Cs		1.00000
354	-	-	Cs	Cs	Cs	Cs	-	Cs	Cs		1.00000
355	-	-	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
356	-	-	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
357	-	-	C1	C1	C1	C1	C1	C1	C1		0.00000
358	-	-	C1	C1	C1	C1	C1	C1	C1		0.00000
359	-	-	C1	C1	C1	C1	C1	C1	C1		0.00000
360	-	-	C1	C1	C1	C1	C1	C1	C1		0.00000
361	-	-	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
362	-	-	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
363	-	-	C1	C1	C1	C1	C1	C1	C1		0.00000
364	-	-	C1	C1	C1	C1	C1	C1	C1		0.00000
365	-	-	C1	C1	C1	C1	C1	C1	C1		0.00000
366	-	-	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
367	-	-	C1	C1	C1	C1	C1	C1	C1		0.00000
368	-	-	C1	C1	C1	C1	C1	C1	C1		0.00000
369	-	-	C2v	C2v	C2v	C2v	-	C2v	C2v		1.00000
370	-	-	C1	C1	C1	C1	C1	C1	C1		0.00000
371	-	-	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
372	-	-	C1	C1	C1	C1	C1	C1	C1		0.00000
373	-	-	-	-	C3v	-	C3v	C3v	C3v		1.00000
374	-	-	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
375	-	-	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
376	-	-	-	-	C3v	C3v	C3v	C3v	C3v		1.00000
377	-	-	C1	C1	C1	C1	C1	C1	C1		0.00000
378	-	-	C1	C1	C1	C1	C1	C1	C1		0.00000
379	-	-	C1	C1	C1	C1	C1	C1	C1		0.00000

Table S34: Dataset: gupta/Na, part 4.

name	SYVA		SymMol		libmsym			SOFI		diff	dot product
	0.001	0.05	0.001	0.05	'default'	'medium'	'loose'	0.001	0.05		
112	C2v	C2v	C2v	C2v	C2v	C2v	-	C2v	C2v		1.00000
113	C2v	C2v	C2v	C2v	C2v	C2v	-	C2v	C2v		1.00000
114	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
115	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
116	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v		1.00000
117	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
118	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
119	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
120	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
121	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
122	C2v	C2v	C2v	C2v	C2v	C2v	-	C2v	C2v		1.00000
123	C2v	C2v	C2v	C2v	C2v	C2v	-	C2v	C2v		1.00000
124	C2v	C2v	C2v	C2v	C2v	C2v	-	C2v	C2v		1.00000
125	C3v	C3v	-	-	C3v	C3v	C3v	C3v	C3v		1.00000
126	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
127	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
128	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
129	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
130	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
131	C2v	C2v	C2v	C2v	C2v	C2v	-	C2v	C2v		1.00000
132	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
133	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
134	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
135	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
136	C4v	C4v	C4v	C4v	C4v	C4v	C4v	C4v	C4v		1.00000
137	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
138	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
139	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
140	Oh	Oh	Oh	Oh	Oh	Oh	Oh	Oh	Oh		1.00000
141	C2v	C2v	C2v	C2v	C2v	C2v	-	C2v	C2v		1.00000
142	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
143	C3v	C3v	-	-	C3v	-	C3v	C3v	C3v		1.00000
144	Cs	Cs	Cs	Cs	Cs	Cs	-	Cs	Cs		1.00000
145	C5v	C5v	C5v	C5v	C5v	-	-	C5v	C5v		1.00000
146	D5h	D5h	D5h	D5h	D5h	D5h	-	D5h	D5h		1.00000
147	Cs	Cs	Cs	Cs	Cs	Cs	-	Cs	Cs		1.00000
148	C2v	C2v	C2v	C2v	C2v	C2v	-	C2v	C2v		1.00000
149	Cs	Cs	Cs	Cs	Cs	Cs	-	Cs	Cs		1.00000
150	C2v	C2v	C2v	C2v	C2v	C2v	-	C2v	C2v		1.00000

Table S36: Dataset: gupta/Pb, part 2.

name	SYVA		SymMol		libmsym			SOFI		diff	dot	product
	0.001	0.05	0.001	0.05	'default'	'medium'	'loose'	0.001	0.05			
112	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
113	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
114	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
115	C2	C2	C2	C2	C2	C2	-	C2	C2			1.00000
116	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
117	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
118	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
119	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
120	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v			1.00000
121	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
122	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
123	C2v	C2v	C2v	C2v	C2v	C2v	-	C2v	C2v			1.00000
124	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
125	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000

Table S38: Dataset: gupta/Zn, part 2.

name	SYVA		SymMol		libmsym			SOFI		diff	dot product
	0.001	0.05	0.001	0.05	'default'	'medium'	'loose'	0.001	0.05		
104i	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
105	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
106	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
107	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
108	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
109	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
110	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
111	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
112	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
113	Cs	Cs	Cs	Cs	Cs	-	Cs	Cs	Cs		1.00000
114	Cs	Cs	Cs	Cs	Cs	Cs	-	Cs	Cs		1.00000
115	C5v	C5v	C5v	C5v	C5v	-	-	C5v	C5v		1.00000
116	C5v	C5v	C5v	C5v	C5v	-	-	C5v	C5v		1.00000
117	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
118	Cs	Cs	Cs	Cs	Cs	Cs	-	Cs	Cs		1.00000
119	Cs	Cs	Cs	Cs	Cs	Cs	-	Cs	Cs		1.00000
120	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
121	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
122	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
123	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
124	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
125	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
126	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
127	C2v	C2v	C2v	C2v	C2v	C2v	-	C2v	C2v		1.00000
128	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
129	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
130	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
131	C2v	C2v	C2v	C2v	C2v	C2v	-	C2v	C2v		1.00000
132	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
133	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
134	C3v	C3v	-	-	C3v	C3v	-	C3v	C3v		1.00000
135	Ih	Ih	Ci	Ci	Ih	Ih	Ih	Ih	Ih	!	0.91300
136	C5v	C5v	C5v	C5	C5v	C5v	-	C5v	C5v	!	1.00000
137	C2v	C2v	C2v	C2v	C2v	C2v	-	C2v	C2v		1.00000
138	C3v	C3v	-	-	C3v	-	-	C3v	C3v		1.00000
139	C2v	C2v	C2v	C2v	C2v	C2v	-	C2v	C2v		1.00000
140	Cs	Cs	Cs	Cs	Cs	Cs	-	Cs	Cs		1.00000
141	C5v	C5v	C5v	C5v	C5v	-	-	C5v	C5v		1.00000
142	Cs	Cs	Cs	Cs	Cs	Cs	-	Cs	Cs		1.00000
143	C2v	C2v	C2v	C2v	C2v	C2v	-	C2v	C2v		1.00000
144	C3v	C3v	-	-	C3v	C3v	-	C3v	C3v		1.00000
145	C2v	C2v	C2v	C2v	C2v	C2v	-	C2v	C2v		1.00000
146	C5v	C5v	C5v	C5	C5v	-	-	C5v	C5v	!	1.00000
147	Ih	Ih	Ci	Ci	Ih	Ih	Ih	Ih	Ih	!	0.99392
148	Cs	Cs	Cs	Cs	Cs	Cs	-	Cs	Cs		1.00000
149	Cs	Cs	Cs	Cs	Cs	Cs	-	Cs	Cs		1.00000
150	C3v	C3v	-	C3	C3v	-	-	C3v	C3v	!	1.00000

Table S40: Dataset: LJ, part 2.

name	SYVA		SymMol		'default'	libmsym			SOFI		diff	dot	product
	0.001	0.05	0.001	0.05		'medium'	'loose'	0.001	0.05				
35A	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000	
35B	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v		1.00000	
35C	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1		1.00000	
35D	C2v	C2v	C2v	C2v	C2v	C2v	C2v	-	C2v	C2v		1.00000	
35E	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000	
35F	C3v	C3v	-	-	C3v	C3v	C3v	C3v	C3v	C3v		1.00000	
36A	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000	
36B	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000	
36C	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000	
36D	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000	
36E	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000	
36F	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000	
36G	Cs	Cs	Cs	Cs	Cs	Cs	Cs	-	Cs	Cs		1.00000	
37A	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000	
37B	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000	
37C	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000	
37D	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000	
37E	C3v	C3v	-	-	C3v	C3v	C3v	C3v	C3v	C3v		1.00000	
38A	D2	D2	D2	D2	D2	D2	D2	D2	D2	D2	!	1.00000	
38B	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000	
38C	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000	
38D	Oh	Oh	Oh	Oh	Oh	Oh	Oh	Oh	Oh	Oh		0.98693	
38E	C5v	C5v	C5	C5	C5v	C5v	-	C5v	C5v	C5v	!	1.00000	
39A	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000	
39B	C5v	C5v	C5	C5	C5v	C5v	-	C5v	C5v	C5v	!	1.00000	
39C	C4v	C4v	C4	C4	C4v	C4v	-	C4v	C4v	C4v	!	1.00000	
40A	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000	
40B	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000	
40C	D4h	D4h	D4h	D4h	D4h	D4h	D4h	D4h	D4h	D4h	!	1.00000	
41A	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000	
41B	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000	
41C	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000	
41D	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000	
41E	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000	
41F	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000	
42A	C3v	C3v	-	-	C3v	C3v	C3v	C3v	C3v	C3v		1.00000	
42B	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000	
42C	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000	
42D	Cs	Cs	Cs	Cs	Cs	Cs	-	Cs	Cs	Cs		1.00000	
42E	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000	
43A	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000	
43B	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000	
43C	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000	
43D	C2v	C2v	C2v	C2v	C2v	C2v	Cs	C2v	C2v	C2v	!	1.00000	
43E	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000	
44A	C5v	C5v	C5	C5	C5v	C5v	C5v	C5v	C5v	C5v	!	1.00000	
44B	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000	
44C	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000	
44D	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000	
44E	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000	
45A	Ih	Ih	Ih	Ih	Ih	Ih	Ih	Ih	Ih	Ih	!	0.98813	
45B	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000	
45Bprime	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000	
45C	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000	
45D	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000	
45E	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000	
45F	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000	
46A	Cs	Cs	Cs	Cs	Cs	Cs	-	Cs	Cs	Cs		1.00000	
46B	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000	
46C	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000	
46D	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000	
46E	Cs	Cs	Cs	Cs	Cs	Cs	-	Cs	Cs	Cs		1.00000	
46Eprime	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000	
47A	C3	C3	C3	C3	C3	C3	-	C3	C3	C3		1.00000	
47B	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000	
47C	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000	
47D	Cs	Cs	Cs	Cs	Cs	Cs	-	Cs	Cs	Cs		1.00000	
48A	C2v	C2v	C2v	C2v	C2v	C2v	-	C2v	C2v	C2v		1.00000	
48B	Cs	Cs	Cs	Cs	Cs	Cs	-	Cs	Cs	Cs		1.00000	
48Bprime	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000	
48C	C2v	C2v	C2v	C2v	C2v	C2v	-	C2v	C2v	C2v		1.00000	
48D	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000	
49A	Cs	Cs	Cs	Cs	Cs	-	-	Cs	Cs	Cs		1.00000	
49B	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v		1.00000	
49C	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000	
49D	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v		1.00000	
50A	C5v	C5v	C5v	C5v	C5v	C5v	C5v	C5v	C5v	C5v		1.00000	
50B	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000	
50C	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000	
50D	D3h	D3h	D3h	D3h	D3h	D3h	D3h	D3h	D3h	D3h		1.00000	
51A	C2	C2	C2	C2	C2	C2	-	C2	C2	C2		1.00000	
51B	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000	
51Bprime	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000	
51C	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000	
51D	Cs	Cs	Cs	Cs	Cs	Cs	-	Cs	Cs	Cs		1.00000	
51E	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000	
52A	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000	
52B	C3v	C3v	-	-	C3v	C3v	C3v	C3v	C3v	C3v		1.00000	
52C	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000	
52D	C2v	C2v	C2v	C2v	C2v	C2v	-	C2v	C2v	C2v		1.00000	
52E	C2v	C2v	C2v	C2v	C2v	C2v	-	C2v	C2v	C2v		1.00000	
53A	D6d	D6d	C6	C6	D6d	D6d	-	D6d	D6d	D6d	!	1.00000	
53B	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000	
53C	Cs	Cs	Cs	Cs	Cs	Cs	-	Cs	Cs	Cs		1.00000	
54A	Cs	Cs	Cs	Cs	Cs	-	Cs	Cs	Cs	Cs		1.00000	
54B	C5v	C5v	C5	C5	C5v	C5v	C5v	C5v	C5v	C5v	!	1.00000	
54C	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000	
54D	C2v	C2v	C2v	C2v	C2v	C2v	-	C2v	C2v	C2v		1.00000	
55A	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000	
55B	Ih	Ih	Ih	Ih	Ih	Ih	Ih	Ih	Ih	Ih	!	0.94555	
55C	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000	
55D	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000	
56A	Cs	Cs	Cs	Cs	Cs	-	Cs	Cs	Cs	Cs		1.00000	
56B	C3v	C3v	-	-	C3v	-	C3v	C3v	C3v	C3v		1.00000	
56C	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000	
56D	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000	
56E	D3h	D3h	D3h	D3h	D3h	D3h	D3h	D3h	D3h	D3h		1.00000	
57A	D3h	D3h	D3h	D3h	D3h	D3h	D3h	D3h	D3h	D3h	!	1.00000	
57B	Cs	Cs	Cs	Cs	Cs	-	Cs	Cs	Cs	Cs		1.00000	
57C	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000	

Table S42: Dataset: morse, part 2.

name	SYVA		SymMol		'default'	libmsym			SOFI		diff	dot	product
	0.001	0.05	0.001	0.05		'medium'	'loose'	0.001	0.05				
57D	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000	
58A	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000	
58B	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v		1.00000	
58C	D3h	D3h	D3h	D3h	D3h	D3h		D3h	D3h			1.00000	
58D	C1	C1	C1	C1	C1	C1		C1	C1			0.00000	
58E	C3v	C3v			C3v	C3v	C3v	C3v	C3v			1.00000	
59A	C3v	C3v			C3v	C3v		C3v	C3v			1.00000	
59B	C2v	C2v	C2v	C2v	C2v			C2v	C2v			1.00000	
59C	C1	C1	C1	C1	C1		C1	C1	C1			0.00000	
59D	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000	
59E	Td	Td	Td	Td	Td	Td	Td	Td	Td	Td		0.86271	
60A	Cs	Cs	Cs	Cs	Cs		Cs	Cs	Cs			1.00000	
60B	Cs	Cs	Cs	Cs	Cs		Cs	Cs	Cs			1.00000	
60C	C2v	C2v	C2v	C2v	C2v	C2v		C2v	C2v			1.00000	
60D	Cs	Cs	Cs	Cs	Cs		Cs	Cs	Cs			1.00000	
60E	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000	
60F	Cs	Cs	Cs	Cs	Cs		Cs	Cs	Cs			1.00000	
61A	Td	Td	Td	Td	Td	Td	Td	Td	Td	Td		0.80426	
61B	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000	
61Bprime	Cs	Cs	Cs	Cs	Cs		Cs	Cs	Cs			1.00000	
61C	C3v	C3v	C3v	C3v	C3v	C3v		C3v	C3v			1.00000	
62A	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000	
62B	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000	
62C	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000	
62D	C2v	C2v	C2v	C2v	C2v	C2v		C2v	C2v			1.00000	
62E	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000	
62F	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000	
62G	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000	
62H	C1	C1	C1	C1	C1			C1	C1			0.00000	
63A	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000	
63B	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000	
63C	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000	
63D	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000	
64A	D3d	D3d	S6	S6	D3d	D3d		D3d	D3d		!	1.00000	
64B	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000	
64C	C2v	C2v	C2v	C2v	C2v	C2v	C1	C2v	C2v		!	1.00000	
64D	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000	
65A	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000	
65B	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000	
65C	C2	C2	C2	C2	C2	C2	C2	C2	C2	C2		1.00000	
65D	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000	
65E	C2v	C2v	C2v	C2v	C2v	C2v		C2v	C2v			1.00000	
65F	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000	
65G	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000	
65H	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000	
66A	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000	
66B	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000	
66C	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000	
66D	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000	
66E	Cs	Cs	Cs	Cs	Cs	Cs		Cs	Cs			1.00000	
66F	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000	
66G	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000	
67A	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000	
67B	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000	
67C	C2	C2	C2	C2	C2	C2	C2	C2	C2	C2		1.00000	
67D	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000	
67E	C2v	C2v	C2v	C2v	C2v	C2v	Cs	C2v	C2v		!	1.00000	
67F	C3v	C3v			C3v	C3v	C3v	C3v	C3v	C3v		1.00000	
68A	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000	
68B	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000	
68C	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000	
68D	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000	
68Dprime	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v		1.00000	
68E	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000	
68F	Cs	Cs	Cs	Cs	Cs	Cs	C1	Cs	Cs		!	1.00000	
68G	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000	
68H	C3v	C3v			C3v	C3v	C3v	C3v	C3v			1.00000	
69A	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000	
69B	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000	
69C	C5v	C5v	C5	C5	C5v	C5v		C5v	C5v		!	1.00000	
69D	Cs	Cs	Cs	Cs	Cs	Cs		Cs	Cs			1.00000	
69E	Cs	Cs	Cs	Cs	Cs	Cs		Cs	Cs			1.00000	
69F	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000	
69G	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000	
70A	C2v	C2v	C2v	C2v	C2v		C2v	C2v	C2v			1.00000	
70B	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000	
70C	C5	C5	C5	C5	C5			C5	C5			1.00000	
70D	C5v	C5v	C5v	C5v	C5v	C5v		C5v	C5v			1.00000	
70E	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000	
70F	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000	
70G	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000	
71A	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000	
71B	C5	C5	C5	C5	C5	C5	C1	C5	C5		!	1.00000	
71C	C5v	C5v	C5v	C5v	C5v	C5v		C5v	C5v			1.00000	
71D	C2v	C2v	C2v	C2v	C2v	C2v		C2v	C2v			1.00000	
71E	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000	
72A	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000	
72B	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000	
72C	Cs	Cs	Cs	Cs	Cs	Cs		Cs	Cs			1.00000	
72D	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000	
72E	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000	
72F	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000	
73A	C3	C3	C3	C3	C3	C3	C3	C3	C3	C3		1.00000	
73B	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000	
73C	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000	
73D	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000	
73E	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000	
73F	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000	
74A	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v		1.00000	
74B	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000	
74C	C5v	C5v	C5	C5	C5v			C5v	C5v		!	1.00000	
74D	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000	
75A	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000	
75B	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000	
75C	D5h	D5h	D5h	C5	D5h			D5h	D5h		!	1.00000	
75D	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000	
76A	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000	
76B	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000	
76C	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000	
76D	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000	

Table S43: Dataset: morse, part 3.

name	SYVA		SymMol		libmsym			SOFI		diff	dot	product
	0.001	0.05	0.001	0.05	'default'	'medium'	'loose'	0.001	0.05			
76E	Cs	Cs	Cs	Cs	Cs	Cs	-	Cs	Cs			1.00000
76F	C2v	C2v	C2v	C2v	C2v	C2v	-	C2v	C2v			1.00000
76G	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
77A	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
77B	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
77C	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
77D	C2v	C2v	C2v	C2v	C2v	C2v	Cs	C2v	C2v	!		1.00000
77E	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
78A	C3	C3	C3	C3	C3	C3	-	C3	C3			1.00000
78B	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
78C	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
78D	Cs	Cs	Cs	Cs	Cs	Cs	-	Cs	Cs			1.00000
78Dprime	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
78E	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
78F	Cs	Cs	Cs	Cs	Cs	Cs	-	Cs	Cs			1.00000
78G	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
79A	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
79B	C2v	C2v	C2v	C2v	C2v	C2v	-	C2v	C2v			1.00000
79C	C2v	C2v	C2v	C2v	C2v	C2v	-	C2v	C2v			1.00000
79D	C2v	C2v	C2v	C2v	C2v	C2v	-	C2v	C2v			1.00000
79E	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
79F	D3h	D3h	D3h	-	D3h	D3h	D3h	D3h	D3h			1.00000
80A	C3	C3	C3	C3	C3	C3	C3	C3	C3			1.00000
80B	Cs	Cs	Cs	Cs	Cs	Cs	-	Cs	Cs			1.00000
80C	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
80D	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
80E	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
80F	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
80G	Cs	Cs	Cs	Cs	Cs	Cs	-	Cs	Cs			1.00000
147A	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
147B	Ih	Ih	Ci	Ci	Ih	Ih	Ih	Ih	Ih	!		0.97282
147C	Cs	Cs	Cs	Cs	Cs	Cs	-	Cs	Cs			1.00000
147D	Cs	Cs	Cs	Cs	Cs	Cs	C1	Cs	Cs	!		1.00000
147E	C3v	C3v	-	C3	C3v	C3v	C3v	C3v	C3v	!		1.00000

Table S44: Dataset: morse, part 4.

name	SYVA		SymMol		libmsym			SOFI		diff	dot	product
	0.001	0.05	0.001	0.05	'default'	'medium'	'loose'	0.001	0.05			
10	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v			1.00000
11	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v			1.00000
12	C5v	C5v	C5v	C5v	C5v	C5v	C5v	C5v	C5v			1.00000
13	Ih	Ih	Ih	Ih	Ih	Ih	Ih	Ih	Ih			1.00000
14	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v			1.00000
15	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v			1.00000
16	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
17	C2	C2	C2	C2	C2	C2	C2	C2	C2			1.00000
18	C5v	C5v	C5	C5	C5v	C5v	C5v	C5v	C5v	!		1.00000
19	D5h	D5h	C5	C5	D5h	D5h	D5h	D5h	D5h	!		1.00000
20	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v			1.00000
21	C2	C2	C2	C2	C2	C2	C2	C2	C2			1.00000
22	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
23	D3h	D3h	D3h	D3h	D3h	D3h	D3h	D3h	D3h			1.00000
24	D3	D3	D3	D3	D3	D3	D3	D3	D3			1.00000
25	C3	C3	C3	C3	C3	C3	C3	C3	C3			1.00000
26	Td	Td	Td	Td	Td	Td	Td	Td	Td			0.87031
27	C1	C1	-	C1	C1	C1	C1	C1	C1			0.00000
28	T	T	T	T	T	T	T	T	T			0.97859
29	C3	C3	C3	C3	C3	C3	C3	C3	C3			1.00000
30	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
31	C2	C2	C2	C2	C2	-	C2	C2	C2			1.00000
32	D3	D3	D3	D3	D3	C3	-	D3	D3	!		1.00000
33	C2	C2	C2	C2	C2	C2	C2	C2	C2			1.00000
34	T	T	T	T	T	T	T	T	T			0.92770
35	D3	D3	D3	D3	D3	D3	-	D3	D3			1.00000
36	C2	C2	C2	C2	C2	C2	C2	C2	C2			1.00000
37	C2	C2	C2	C2	C2	C2	C2	C2	C2			1.00000
38	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
39	C5	C5	C5	C5	C5	-	C5	C5	C5			1.00000
40	C1	Cs	C1	Cs	Cs	Cs	-	C1	Cs	!M		1.00000
41	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
42	D2	D2	D2	D2	D2	D2	D2	D2	D2			1.00000
43	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
44	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
45	D3	D3	D3	D3	D3	D3	-	D3	D3			1.00000
46	C2	C2	C2	C2	C2	C2	C2	C2	C2			1.00000
47	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
48	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
49	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v			1.00000
50	Cs	Cs	Cs	Cs	Cs	Cs	-	Cs	Cs			1.00000
51	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v			1.00000
52	C3v	C3v	-	-	C3v	C3v	C3v	C3v	C3v			1.00000
53	C2v	C2v	C2v	C2v	C2v	C2v	-	C2v	C2v			1.00000
54	C5v	C5v	C5	C5	C5v	C5v	-	C5v	C5v	!		1.00000
55	Ih	Ih	Ih	Ih	Ih	Ih	Ih	Ih	Ih			0.94694
56	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
57	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v			1.00000
58	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v			1.00000
59	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v			1.00000
60	D3	D3	D3	D3	D3	D3	-	D3	D3			1.00000
61	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v			1.00000
62	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
63	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v			1.00000
64	D3	D3	D3	D3	D3	C3	C1	D3	D3	!		1.00000
65	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
66	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
67	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
68	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
69	C5	C5	C5	C5	C5	-	C1	C5	C5	!		1.00000
70	C5	C5	C5	C5	C5	C5	-	C5	C5			1.00000
71	C5	C5	C5	C5	C1	-	-	C5	C5	!		1.00000
72	C2	C2	C2	C2	C2	C2	C2	C2	C2			1.00000
73	Cs	Cs	Cs	Cs	Cs	Cs	-	Cs	Cs			1.00000
74	D3d	D3d	D3d	D3d	D3d	-	-	D3d	D3d			1.00000
75	C2	C2	C2	C2	C2	-	C2	C2	C2			1.00000
76	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
77	C2	C2	C2	C2	C2	C2	C1	C2	C2			1.00000
78	C1	C2	C1	C1	C1	C2	-	C1	C2	!M		1.00000
79	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
80	D3	D3	D3	D3	D3	D3	-	D3	D3			1.00000
81	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
82	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
83	C1	D3	C1	D3	C1	D3	-	C1	D3	!M		1.00000
84	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
85	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
86	C3	C3	C3	C3	C3	C3	C3	C3	C3			1.00000
87	C2v	C2v	C2v	C2v	C2v	C2v	-	C2v	C2v			1.00000
88	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
89	C1	C3	C1	C3	C1	C1	C3	C1	C3	!M		0.99991
90	C1	C1	-	C1	C1	C1	C1	C1	C1			0.00000
91	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
92	C1	T	C1	T	C1	T	T	C1	T	!M		0.98969
93	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
94	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
95	C3	C3	C3	C3	C3	C3	C3	C3	C3			1.00000
96	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
97	C1	C2	C1	C1	C1	-	-	C1	C2	!M		0.99996
98	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
99	C1	C2	C1	C1	C1	C2	-	C1	C2	!M		0.99994
100	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
101	D3	D3	D3	D3	D3	C3	-	D3	D3	!		1.00000
102	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
103	C2	C2	C2	C2	C2	C2	-	C2	C2			1.00000
104	C1	C2	C1	C1	C1	C1	C2	C1	C2	!M		0.99999
105	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
106	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
107	C1	C2	C1	C2	C2	C2	-	C1	C2	!M		1.00000
108	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
109	C1	Cs	C1	C1	C1	C1	C1	C1	C1	!		0.00000
110	C2	-	C2	C2	C2	C2	C2	C2	C2			1.00000
111	C1	C5	C1	C5	C5	C5	-	C1	C5	!M		1.00000
112	C1	Cs	C1	C1	C1	C1	C1	C1	C1	!		0.00000
113	C1	Cs	C1	C1	C1	C1	C1	C1	C1	!		0.00000
114	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
115	Cs	Cs	Cs	Cs	Cs	Cs	-	Cs	Cs			1.00000
116	D5h	D5h	D5h	D5h	D5h	D5h	-	D5h	D5h			1.00000
117	C2v	-	C2v	C2v	C2v	C2v	Cs	C2v	C2v	!		1.00000
118	C1	-	C1	C1	C1	C1	C1	C1	C1			0.00000

Table S45: Dataset: murrel-mottram/Na, part 1.

name	SYVA		SymMol		libmsym			SOFI		diff	dot	product
	0.001	0.05	0.001	0.05	'default'	'medium'	'loose'	0.001	0.05			
119	C1	-	C1	C1	C1	C1	-	C1	Cs	!M	0.99992	
120	C1	C2v	C1	C2	C1	-	C1	C1	C2	!M	1.00000	
121	C1	-	C1	C1	C1	C1	-	C1	C1		0.00000	
122	C1	-	C1	C1	C1	C1	C1	C1	D2	!M	0.99998	
123	C1	-	C1	C1	C1	C1	-	C1	C1		0.00000	
124	C1	-	C1	C1	C1	C1	C1	C1	C1		0.00000	
125	C1	-	C1	C1	C1	C1	C1	C1	C1		0.00000	
126	C1	-	C1	C1	C1	C1	C1	C1	C1		0.00000	
127	C2	-	C2	C2	C2	C2	-	C2	C2		1.00000	
128	C1	-	C1	C1	C1	C1	C1	C1	C1		0.00000	
129	C2	-	C2	C2	C2	C2	-	C2	C2		1.00000	
130	C1	-	C1	C1	C1	C1	C1	C1	C1		0.00000	
131	C2	-	C2	C2	C2	C2	-	C2	C2		1.00000	
132	C1	-	C1	Cs	Cs	Cs	-	C1	Cs	!M	1.00000	
133	C1	-	C1	Cs	C1	C1	C1	C1	C1		0.00000	
134	Ih	Ih	Ih	Ih	Ih	Ih	Ih	Ih	Ih		0.94606	
135	Ih	Ih	Ih	Ih	Ih	Ih	Ih	Ih	Ih		0.91157	
136	C5v	-	C5v	C5	C5v	C5v	-	C5v	C5v	!	1.00000	
137	C2v	-	C2v	C2v	C2v	C2v	-	C2v	C2v	!	1.00000	
138	C3v	-	C3v	C3v	C3v	C3v	-	C3v	C3v		1.00000	
139	C2v	-	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000	
140	Cs	-	Cs	Cs	Cs	Cs	-	Cs	Cs		1.00000	
141	C5v	-	C5v	C5v	C5v	C5v	-	C5v	C5v		1.00000	
142	C1	-	C1	Cs	Cs	-	-	Cs	Cs	!	1.00000	
143	C2v	-	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000	
144	C3v	-	C3v	C3v	C3v	C3v	-	C3v	C3v		1.00000	
145	C2v	-	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000	
146	C5v	-	C5v	C5v	C5v	C5v	-	C5v	C5v		1.00000	
147	Ih	-	Ih	Ih	Ih	Ih	Ih	Ih	Ih		0.95272	
148	Cs	-	Cs	Cs	Cs	-	-	Cs	Cs		1.00000	
149	C1	-	C1	Cs	-	-	-	C1	Cs	M	1.00000	
150	C3v	-	C3v	-	C3v	-	-	C3v	C3v		1.00000	
151	C1	-	C1	C1	C1	-	Cs	C1	Cs	!M	0.99839	
152	C1	-	C1	Cs	Cs	-	Cs	C1	Cs	!M	1.00000	
153	C1	-	C1	-	C3v	C3v	C3v	C1	C3v	!M	1.00000	
154	C1	-	C1	Cs	Cs	-	Cs	C1	Cs	!M	1.00000	
155	Cs	-	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000	
156	C1	-	C1	C1	C1	C1	Cs	C1	Cs	!M	0.99810	
157	C1	-	C1	C2	C2v	C2v	-	C1	C2v	!M	1.00000	
158	C1	-	C1	C1	Cs	-	-	C1	Cs	!M	1.00000	
159	C2v	-	C2v	C2v	C2v	C2v	C1	C2v	C2v	!	1.00000	
160	C1	-	C1	C1	C1	C1	C1	C1	C1		0.00000	
161	C1	-	C1	C1	C1	C1	C1	C1	C1		0.00000	
162	C1	-	C1	C1	C1	C1	C1	C1	C1		0.00000	
163	Cs	-	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000	
164	C1	-	C1	C1	C1	C1	C1	C1	C1		0.00000	
165	C1	-	C1	C1	C1	C1	C1	C1	C1		0.00000	
166	C1	-	C1	C1	C1	C1	C1	C1	C1		0.00000	
167	C1	-	C1	C1	C1	C1	C1	C1	C1		0.00000	
168	C1	-	C1	C1	C1	C1	C1	C1	C1		0.00000	
169	C1	-	C1	C1	C1	C1	C1	C1	C1		0.00000	
170	C1	-	C1	C1	C1	C1	C1	C1	C1		0.00000	
171	C5	-	C5	C5	C5	C5	-	C5	C5		1.00000	
172	C5	-	C5	C5	C1	C5	-	C5	C5	!	1.00000	
173	C5	-	C5	C5	C5	C5	-	C5	C5		1.00000	
174	C1	-	C1	C1	C1	C1	C1	C1	C1		0.00000	
175	C1	-	C1	C1	C1	-	-	C1	C1		0.00000	
176	C5	-	C5	C5	-	-	C1	C5	C5	!	1.00000	
177	C5	-	C5	C5	C5	-	-	C5	C5		1.00000	
178	C5	-	C5	C5	-	-	C1	C5	C5	!	1.00000	
179	C1	-	C1	C1	C1	C1	-	C1	C1		0.00000	
180	C1	-	C1	C2h	C1	Ci	-	C1	C2h	!M	1.00000	
181	C1	-	C1	C1	C1	C1	-	C1	Cs	!M	0.99999	
182	C1	-	C1	C2	C2	C1	-	C1	C2	!M	1.00000	
183	C1	-	C1	C1	C1	-	-	C1	Cs	!M	0.99997	
184	D3d	-	S6	S6	D3d	D3	-	D3d	D3d	!	1.00000	
185	C1	-	C1	C1	C1	C1	C1	C1	C1		0.00000	
186	C1	-	C1	C1	C1	C1	-	C1	C1		0.00000	
187	C1	-	C1	C1	C1	C1	C1	C1	C1		0.00000	
188	C1	-	C1	C1	C1	C1	C1	C1	C1		0.00000	
189	C1	-	C1	C1	C1	C1	C1	C1	C1		0.00000	
190	C1	-	C1	C1	C1	C1	C1	C1	C1		0.00000	
191	C1	-	C1	C3	C1	-	C1	C1	C3	!M	0.99999	
192	C1	-	C1	C1	C1	C1	C1	C1	C1		0.00000	
193	C1	-	C1	C1	C1	C1	C1	C1	C1		0.00000	
194	D3	-	C1	D3	D3	-	-	D3	D3	!	1.00000	
195	C1	-	C1	C1	C1	C1	-	C1	C1		0.00000	
196	C1	-	C1	C1	C1	C1	-	C1	C1		0.00000	
197	C1	-	C1	C1	C1	C1	C1	C1	C1		0.00000	
198	C1	-	C1	C1	C1	C1	C1	C1	C2	!M	0.99997	
199	C1	-	C1	C1	C1	C1	-	C1	C1		0.00000	
200	C1	-	C1	C1	C1	C1	-	C1	C2	!M	0.99034	
201	C1	-	C1	C1	C1	C1	C1	C1	C1		0.00000	
202	C1	-	C1	C1	C1	C1	C1	C1	C1		0.00000	
203	C3	-	C1	C3	C3	C3	C3	C3	C3	!	1.00000	
204	C1	-	C1	C1	C1	C1	C1	C1	C1		0.00000	
205	C1	-	C1	C1	C1	C1	C1	C1	C1		0.00000	
206	C3	-	C1	C3	C1	-	C3	C3	C3	!	1.00000	
207	C1	-	C1	C3	C3	C1	C3	C1	C3	!M	1.00000	
208	C1	-	C1	C1	C1	C1	C1	C1	C1		0.00000	
209	C1	-	C1	C1	C1	C1	C1	C1	C1		0.00000	
210	C1	-	C1	C1	C1	C1	C1	C1	C2	!M	0.99998	
211	C1	-	C1	C1	C1	C1	C1	C1	C1		0.00000	
212	C1	-	C1	C1	C1	C1	-	C1	C1		0.00000	
213	C3	-	C3	C3	C3	C3	-	C3	C3		1.00000	
214	C1	-	C1	C1	C1	C1	C1	C1	C1		0.00000	
215	C1	-	C1	C1	C1	C1	C1	C1	C1		0.00000	
216	C1	-	C1	T	T	T	T	C1	T	!M	0.95174	
217	C1	-	C1	C1	C1	C1	C1	C1	C1		0.00000	
218	C1	-	C1	C1	C1	C1	C1	C1	C1		0.00000	
219	C1	-	C1	C1	-	C1	C1	C1	C3	!M	1.00000	
220	C1	-	C1	C1	C1	C1	C1	C1	C1		0.00000	
221	C1	-	C1	C1	C1	C1	C1	C1	C1		0.00000	
222	C1	-	C1	C3	C3	C3	-	C1	C3	!M	1.00000	
223	C3	-	C3	C3	C3	-	C1	C3	C3	!	1.00000	
224	C1	-	C1	C1	C1	C1	C1	C1	C1		0.00000	
225	C1	-	C1	C1	C1	C1	-	C1	C1		0.00000	
226	C2	-	C1	C2	C2	-	-	C2	C2	!	1.00000	
227	C1	-	C1	C1	C1	-	C1	C1	C1		0.00000	
228	C2	-	C2	C2	C1	-	-	C2	C2	!	1.00000	

Table S46: Dataset: murrel-mottram/Na, part 2.

name	SYVA		SymMol		libmsym			SOFI		diff	dot	product
	0.001	0.05	0.001	0.05	'default'	'medium'	'loose'	0.001	0.05			
229	C1	-	C1	C1	C1	C1	C1	C1	C1		0.0000	
230	C1	-	C1	C1	C1	C1	-	C1	C2	!M	0.99991	
231	C1	-	C1	C1	C1	C1	-	C1	C1		0.0000	
232	C1	-	C1	D3	D3	-	-	C1	D3	!M	1.0000	
233	C1	-	C1	C1	C1	C1	C1	C1	C1		0.0000	
234	C1	-	C1	C1	C1	C1	C1	C1	C1		0.0000	
235	C1	-	C1	C1	C1	C1	C1	C1	C1		0.0000	
236	C2	-	C2	C2	C2	C1	C1	C1	C2	!	1.0000	
237	C1	-	C1	C1	C1	C1	C1	C1	C1		0.0000	
238	C1	-	C1	C1	C1	C1	C1	C1	C1		0.0000	
239	C1	-	C1	C1	C1	C1	C1	C1	C1		0.0000	
240	C1	-	C1	C1	C1	C1	C1	C1	C1		0.0000	
241	C1	-	C1	C1	C1	C1	C1	C1	C1		0.0000	
242	C1	-	C1	C2	C2	C1	-	C1	C2	!M	1.0000	
243	C1	-	C1	C1	C1	C1	C1	C1	C1		0.0000	
244	C1	-	C1	C1	C1	C1	-	C1	C2	!M	1.0000	
245	C1	-	C1	C1	C1	C1	C1	C1	C1		0.0000	
246	D5	-	C5	D5	-	-	-	D5	D5	!	1.0000	
247	C5	-	C5	C5	C5	-	-	C5	C5		1.0000	
248	C5	-	C5	C5	C1	-	-	C5	D5	!M	1.0000	
249	C1	-	C1	C1	C1	C1	C1	C1	C1		0.0000	
250	C1	-	C1	C1	C1	C1	C1	C1	C1		0.0000	
251	-	-	C1	C1	C1	C1	C1	C1	C1		0.0000	
252	-	-	C1	C1	C1	C1	C1	C1	C1		0.0000	
253	-	-	C5	C5	C5	-	-	C5	C5		1.0000	
254	-	-	C1	C1	C1	C1	C1	C1	C1		0.0000	
255	-	-	C1	C1	C1	C1	C1	C1	C1		0.0000	
256	-	-	C1	C1	C1	C1	C1	C1	C1		0.0000	
257	-	-	C1	C1	C1	C1	C1	C1	C1		0.0000	
258	-	-	C5	D5	C1	-	-	C5	D5	!M	1.0000	
259	-	-	C1	C2	C1	C1	C1	C1	C2	!M	1.0000	
260	-	-	C1	C1	C1	C1	C1	C1	C1		0.0000	
261	-	-	C1	C1	C1	C1	C1	C1	C1		0.0000	
262	-	-	C1	C1	C1	C1	C1	C1	C1		0.0000	
263	-	-	C1	C1	C1	C1	C1	C1	C1		0.0000	
264	-	-	C1	C1	C1	C1	C1	C1	C1		0.0000	
265	-	-	C1	C1	C1	C1	C1	C1	C1		0.0000	
266	-	-	C1	C1	C1	C1	C1	C1	C1		0.0000	
267	-	-	C1	C1	C1	C1	C1	C1	C1		0.0000	
268	-	-	C1	C1	C1	C1	C1	C1	C1		0.0000	
269	-	-	C1	C1	C1	C1	C1	C1	C1		0.0000	
270	-	-	C1	C1	C1	C1	C1	C1	C1		0.0000	
271	-	-	C1	C1	C1	C1	C1	C1	C1		0.0000	
272	-	-	C2	C2	C2	C2	-	C2	C2		1.0000	
273	-	-	C1	C1	C1	C1	C1	C1	C1		0.0000	
274	-	-	C1	C1	C1	C1	C1	C1	C1		0.0000	
275	-	-	C1	C1	C1	C1	C1	C1	C1		0.0000	
276	-	-	C1	C1	C1	C1	C1	C1	C1		0.0000	
277	-	-	C1	C1	C1	C1	C1	C1	C1		0.0000	
278	-	-	C1	C1	C1	C1	C1	C1	C1		0.0000	
279	-	-	C1	C1	C1	C1	C1	C1	C1		0.0000	
280	-	-	C1	C1	C1	C1	C1	C1	C1		0.0000	
281	-	-	C1	C1	C1	C1	C1	C1	C1		0.0000	
282	-	-	Cs	Cs	Cs	C1	Cs	Cs	Cs	!	1.0000	
283	-	-	C1	C1	C1	C1	C1	C1	C1		0.0000	
284	-	-	-	-	C3v	C3v	C3v	C3v	C3v		1.0000	
285	-	-	-	-	C3v	C3v	C3v	C3v	C3v		1.0000	
286	-	-	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.0000	
287	-	-	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.0000	
288	-	-	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.0000	
289	-	-	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.0000	
290	-	-	C5v	C5v	C5v	C5v	-	C5v	C5v		1.0000	
291	-	-	C5v	C5v	C5v	C5v	-	C5v	C5v		1.0000	
292	-	-	C5v	C5v	C5v	C5v	-	C5v	C5v		1.0000	
293	-	-	C1	Cs	Cs	Cs	-	Cs	Cs	!	1.0000	
294	-	-	C1	C1	C1	C1	C1	C1	Cs	!M	0.98891	
295	-	-	C1	Cs	Cs	Cs	Cs	Cs	Cs	!	1.0000	
296	-	-	Ih	Ih	Ih	Ih	Ih	Ih	Ih		0.97463	
297	-	-	Ih	Ih	Ih	Ih	Ih	Ih	Ih		0.97429	
298	-	-	C5v	C5	C5v	-	-	C5v	C5v	!	1.0000	
299	-	-	C2v	C2v	C2v	C2v	-	C2v	C2v		1.0000	
300	-	-	-	-	C3v	-	-	C3v	C3v		1.0000	
301	-	-	C2v	C2v	C2v	C2v	-	C2v	C2v		1.0000	
302	-	-	Cs	Cs	Cs	-	-	Cs	Cs		1.0000	
303	-	-	C2	C2	C2	C2	C2	C2	C2		1.0000	
304	-	-	C1	C1	Cs	-	-	Cs	Cs	!	1.0000	
305	-	-	C2v	C2v	C2v	-	-	C2v	C2v		1.0000	
306	-	-	-	-	C3v	C3v	-	C3v	C3v		1.0000	
307	-	-	C2v	C2v	C2v	-	-	C2v	C2v		1.0000	
308	-	-	C5v	C5	C5v	-	-	C5v	C5v	!	1.0000	
309	-	-	Ih	Ih	Ih	Ih	Ih	Ih	Ih		0.99314	
310	-	-	Cs	Cs	Cs	-	-	Cs	Cs		1.0000	
311	-	-	Cs	Cs	Cs	-	-	Cs	Cs		1.0000	
312	-	-	-	-	C3v	C3v	-	C3v	C3v		1.0000	
313	-	-	C1	Cs	-	-	-	Cs	Cs	!	1.0000	
314	-	-	Cs	Cs	-	-	-	Cs	Cs		1.0000	
315	-	-	-	-	C3v	-	-	C3v	C3v		1.0000	
316	-	-	-	-	C3v	C3v	-	C3v	C3v		1.0000	
317	-	-	Cs	Cs	Cs	-	-	Cs	Cs		1.0000	
318	-	-	C1	Cs	C1	Cs	Cs	C1	Cs	!M	1.0000	
319	-	-	C1	C1	C1	Cs	C3v	Cs	C3v	!M	1.0000	
320	-	-	C1	Cs	Cs	Cs	Cs	Cs	Cs	!	1.0000	
321	-	-	C1	Cs	Cs	Cs	Cs	C1	Cs	!M	1.0000	
322	-	-	C1	C1	C1	-	-	C1	C3v	!M	1.0000	
323	-	-	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.0000	
324	-	-	C1	C1	C1	Cs	Cs	C1	Cs	!M	0.99996	
325	-	-	C1	C3v	C1	Cs	Cs	Cs	C3v	!M	1.0000	
326	-	-	C1	C1	C1	Cs	-	C1	Cs	!M	1.0000	
327	-	-	C1	C2v	C2v	C2v	-	Cs	C2v	!M	1.0000	
328	-	-	C1	C1	C1	C1	-	C1	C1		0.0000	
329	-	-	C1	C1	C1	C1	C1	C1	C1		0.0000	
330	-	-	C1	C1	C1	C1	C1	C1	C1		0.0000	
331	-	-	C1	C1	C1	C1	C1	C1	C1		0.0000	
332	-	-	C1	C1	C1	C1	C1	C1	C1		0.0000	
333	-	-	C1	C1	C1	C1	C1	C1	C1		0.0000	
334	-	-	C1	C1	C1	C1	C1	C1	C1		0.0000	
335	-	-	C1	C1	C1	C1	C1	C1	C1		0.0000	
336	-	-	C1	C1	C1	C1	C1	C1	C1		0.0000	
337	-	-	C1	C1	C1	C1	C1	C1	C1		0.0000	
338	-	-	C1	C1	C1	C1	C1	C1	C1		0.0000	

Table S47: Dataset: murrel-mottram/Na, part 3.

name	SYVA		SymMol		libmsym			SOFI		diff	dot product
	0.001	0.05	0.001	0.05	'default'	'medium'	'loose'	0.001	0.05		
339	-	-	C1	C1	C1	C1	C1	C1	C1		0.00000
340	-	-	C1	C1	C1	C1	C1	C1	C1		0.00000
341	-	-	C1	C1	C1	C1	C1	C1	C1		0.00000
342	-	-	C1	C1	C1	C1	C1	C1	C1		0.00000
343	-	-	C1	C1	C1	C1	C1	C1	C1		0.00000
344	-	-	C1	C1	C1	C1	C1	C1	C1		0.00000
345	-	-	C1	C1	C1	C1	C1	C1	C1		0.00000
346	-	-	C1	C1	C1	C1	C1	C1	C1		0.00000
347	-	-	C5	C5	C5	-	-	C5	C5		1.00000
348	-	-	C5	C5	-	-	-	C5	C5		1.00000
349	-	-	C1	C1	C1	C1	C1	C1	C1		0.00000
350	-	-	C1	C1	C1	C1	C1	C1	C1		0.00000
351	-	-	C1	C1	C1	C1	-	C1	C1		0.00000
352	-	-	C1	C1	C1	C1	C1	C1	C1		0.00000
353	-	-	C5	C5	C1	-	-	C5	C5	!	1.00000
354	-	-	C1	C1	C1	C1	C1	C1	C1		0.00000
355	-	-	C1	C1	C1	C1	C1	C1	C1		0.00000
356	-	-	C1	C1	C1	C1	C1	C1	C1		0.00000
357	-	-	S6	S6	D3d	D3	-	D3d	D3d	!	1.00000
358	-	-	C1	C1	C1	-	-	C1	Cs	!M	0.99998
359	-	-	C1	Cs	C1	-	-	C1	Cs	!M	1.00000
360	-	-	C1	C1	C1	-	-	C1	C1		0.00000
361	-	-	C1	C1	C1	-	C1	C1	C1		0.00000
362	-	-	C1	C1	C1	C1	-	C1	Cs	!M	0.99988
363	-	-	C1	C1	C1	C1	-	C1	C1		0.00000
364	-	-	C1	Cs	C1	-	C1	Cs	Cs	!	1.00000
365	-	-	C1	C2	C1	C1	-	C1	C2	!M	1.00000
366	-	-	C1	C1	C1	-	-	C1	Cs	!M	0.85440
367	-	-	C1	C1	C1	C1	-	C1	C1		0.00000
368	-	-	C1	Cs	C1	-	-	C1	Cs	!M	1.00000
369	-	-	C1	C3v	-	-	-	C1	C3v	M	1.00000
370	-	-	C1	C1	C1	C1	C1	C1	C1		0.00000
371	-	-	C1	C1	C1	C1	C1	C1	C1		0.00000
372	-	-	C1	C1	C1	C1	C1	C1	C1		0.00000
373	-	-	C1	C1	C1	C1	C1	C1	C1		0.00000
374	-	-	C1	C1	C1	C1	C1	C1	C1		0.00000
375	-	-	C1	C1	C1	C1	C1	C1	C1		0.00000
376	-	-	C1	C1	C1	C1	C1	C1	C1		0.00000
377	-	-	C1	C1	C1	C1	C1	C1	C1		0.00000
378	-	-	C1	C1	C1	C1	C1	C1	C1		0.00000
379	-	-	C1	C1	C1	C1	C1	C1	C1		0.00000
380	-	-	C1	C1	C1	C1	C1	C1	C1		0.00000

Table S48: Dataset: murrel-mottram/Na, part 4.

name	SYVA		SymMol		libmsym			SOFI		diff	dot product
	0.001	0.05	0.001	0.05	'default'	'medium'	'loose'	0.001	0.05		
3	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v		1.00000
4	C4v	C4v	C4v	C4v	C4v	C4v	C4v	C4v	C4v		1.00000
5	C1	C1	C1	C1	C1	C1	-	C1	C1		0.00000
6	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
7	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
8	C4v	C4v	C4v	C4v	C4v	C4v	C4v	C4v	C4v		1.00000
9	C3h	C3h	C3h	C3h	C3h	C3h	-	C3h	C3h		1.00000
10	C2	C2	C2	C2	C2	C2	C2	C2	C2		1.00000
11	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
12	C4v	C4v	C4v	C4v	C4v	C4v	C4v	C4v	C4v		1.00000
13	Oh	Oh	Oh	Oh	Oh	Oh	Oh	Oh	Oh		0.94097
14	Cs	-	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
15	Cs	-	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
16	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
17	C4v	-	C4v	C4v	C4v	C4v	-	C4v	C4v		1.00000
18	C1	-	C1	C1	C1	C1	C1	C1	C1		0.00000
19	C2	-	C2	C2	C2	C2	C2	C2	C2		1.00000
20	Cs	-	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
21	C4v	-	C4v	C4v	C4v	C4v	C4v	C4v	C4v		1.00000
22	D4h	-	C4h	C4h	D4h	D4h	-	D4h	D4h	!	1.00000
23	C1	-	C1	C1	C1	C1	C1	C1	C1		0.00000
24	D3	-	D3	D3	D3	D3	D3	D3	D3		1.00000
25	Cs	-	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
26	C4v	-	C4	C4	C4v	C4v	-	C4v	C4v	!	1.00000
27	C2v	-	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
28	Cs	-	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
29	C1	-	C1	C1	C1	C1	C1	C1	C1		0.00000
30	Cs	-	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
31	D4h	-	C4h	C4h	D4h	D4h	-	D4h	D4h	!	1.00000
32	Cs	-	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
33	C1	-	C1	C1	C1	C1	C1	C1	C1		0.00000
34	Cs	-	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
35	Cs	-	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000

Table S49: Dataset: NaCl_N/welch, part 1.

name	SYVA		SymMol		libmsym			SOFI		diff	dot	product
	0.001	0.05	0.001	0.05	'default'	'medium'	'loose'	0.001	0.05			
3	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v			1.00000
4	C4v	C4v	C4v	C4v	C4v	C4v	C4v	C4v	C4v			1.00000
5	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v			1.00000
6	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
7	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v			1.00000
8	C4v	C4v	C4v	C4v	C4v	C4v	C4v	C4v	C4v			1.00000
9	C3h	C3h	C3h	C3h	C3h	C3h	-	C3h	C3h			1.00000
10	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
11	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
12	C4v	C4v	C4v	C4v	C4v	C4v	C4v	C4v	C4v			1.00000
13	Oh	Oh	Oh	Oh	Oh	Oh	Oh	Oh	Oh			0.94424
14	Cs	-	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
15	C1	Cs	C1	C1	C1	C1	C1	C1	C1	!		0.00000
16	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v			1.00000
17	C4v	C4v	C4v	C4v	C4v	C4v	C4v	C4v	C4v			1.00000
18	C1	-	C1	C1	C1	C1	C1	C1	C1			0.00000
19	Cs	-	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
20	Cs	-	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
21	C4v	-	C4v	C4v	C4v	C4v	C4v	C4v	C4v			1.00000
22	D4h	-	C4h	C4h	D4h	D4h	D4h	D4h	D4h	!		1.00000
23	Cs	-	Cs	Cs	Cs	Cs	-	Cs	Cs			1.00000
24	C1	-	C1	C1	C1	C1	Cs	Cs	Cs			1.00000
25	Cs	-	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
26	C4v	-	C4	C4	C4v	C4v	-	C4v	C4v	!		1.00000
27	C2v	-	C2v	C2v	C2v	C2v	C2v	C2v	C2v			1.00000
28	Cs	-	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
29	Cs	-	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
30	Cs	-	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
31	D4h	-	C4h	C4h	D4h	D4h	-	D4h	D4h	!		1.00000
32	Cs	-	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
33	C1	-	C1	C1	C1	C1	C1	C1	C1			0.00000
34	Cs	-	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
35	Cs	-	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000

Table S50: Dataset: NaCl_N/CBM, part 1.

name	SYVA		SymMol		libmsym			SOFI		diff	dot	product
	0.001	0.05	0.001	0.05	'default'	'medium'	'loose'	0.001	0.05			
17A	C2	C2	C2	C2	C2	C2	C2	C2	C2			1.00000
17B	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
17C	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v			1.00000
27A	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v			1.00000
27B	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
28A	Cs	Cs	Cs	Cs	Cs	Cs	-	Cs	Cs			1.00000
28B	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
30A	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v			1.00000
30B	C2v	C2v	C2v	C2v	C2v	C2v	-	C2v	C2v			1.00000
31A	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
31B	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
31C	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
32A	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v			1.00000
32B	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v			1.00000
33A	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
33B	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
34A	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v			1.00000
34B	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
35A	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
35B	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
36A	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
36B	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
37A	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
37B	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
38A	Oh	Oh	Oh	Oh	Oh	Oh	Oh	Oh	Oh			0.99980
38B	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
44A	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
44B	C2	C2	C2	C2	C2	C2	C2	C2	C2			1.00000
57A	Cs	Cs	Cs	Cs	Cs	-	Cs	Cs	Cs			1.00000
57B	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
66A	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
66B	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
67A	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
67B	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
72A	Cs	Cs	Cs	Cs	Cs	Cs	-	Cs	Cs			1.00000
72B	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
75A	Cs	D5h	C1	C5	C5v	-	-	C1	D5h	!M		1.00000
75B	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
76A	Cs	Cs	Cs	Cs	Cs	Cs	-	Cs	Cs			1.00000
76B	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
76C	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
77A	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v			1.00000
77B	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
77C	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
78A	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000
78B	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
82A	C1	C1	C1	C1	C1	-	C1	C1	C1			0.00000
82B	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
83A	C2v	C2v	C2v	C2v	C2v	C2v	-	C2v	C2v			1.00000
83B	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
84A	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
84B	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
86A	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
86B	C1	C1	C1	C1	C1	C1	C1	C1	C1			0.00000
98A	Td	Td	Td	Td	Td	Td	Td	Td	Td			0.82308
98B	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs			1.00000

Table S51: Dataset: QLJ, part 1.

name	SYVA		SymMol		'default'	libmsym		SOFI		diff	dot product
	0.001	0.05	0.001	0.05		'medium'	'loose'	0.001	0.05		
3	D3h	D3h	D3h	D3h	D3h	D3h	D3h	D3h	D3h		1.00000
4	Td	Td	Td	Td	Td	Td	Td	Td	Td		0.99850
5	D3h	D3h	D3h	D3h	D3h	D3h	D3h	D3h	D3h		1.00000
6	Oh	Oh	Oh	Oh	Oh	Oh	Oh	Oh	Oh		0.93406
7	D5h	D5h	D5h	D5h	D5h	D5h	D5h	D5h	D5h		1.00000
8	D2d	D2d	D2d	D2d	D2d	D2d	D2d	D2d	D2d		1.00000
9	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
10	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v		1.00000
11	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
12	C5v	C5v	C5v	C5v	C5v	C5v	C5v	C5v	C5v		1.00000
13	Ih	Ih	Ci	Ci	Ih	Ih	Ih	Ih	Ih	!	0.98801
14	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v		1.00000
15	D6d	-	C6	C6	D6d	D6d	D6d	D6d	D6d	!	1.00000
16	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
17	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	!	1.00000
18	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
19	D5h	D5h	C5	C5	D5h	D5h	D5h	D5h	D5h	!	1.00000
20	C1	C1	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
21	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
22	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
23	C2	C2	C2	C2	C2	C2	C2	C2	C2		1.00000
24	C2	C2	C2	C2	C2	C2	C1	C2	C2	!	1.00000
25	C3	C3	C3	C3	C3	C3	C3	C3	C3		1.00000
26	D3h	D3h	D3h	D3h	D3h	D3h	D3h	D3h	D3h		1.00000
27	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
28	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
29	C2	C2	C2	C2	C2	C2	C2	C2	C2		1.00000
30	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
31	C3	C3	C3	C3	C3	C3	C3	C3	C3		1.00000
32	D2d	D2d	D2d	D2d	D2d	D2d	D2d	D2d	D2d		1.00000
33	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
34	C2	C2	C2	C2	C2	C2	C2	C2	C2		1.00000
35	D3	D3	D3	D3	D3	D3	-	D3	D3		1.00000
36	C2v	C2v	C2v	C2v	C2v	C2v	-	C2v	C2v		1.00000
37	C3v	C3v	-	-	C3v	-	C3v	C3v	C3v		1.00000
38	Oh	Oh	Oh	Oh	Oh	Oh	-	Oh	Oh		1.00000
39	C4v	C4v	C4	C4	C4v	C4v	-	C4v	C4v	!	1.00000
40	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
41	Cs	Cs	Cs	Cs	Cs	-	Cs	Cs	Cs		1.00000
42	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
43	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
44	C2	C2	C2	C2	C2	C2	C2	C2	C2		1.00000
45	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
46	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
47	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
48	C2v	C2v	C2v	C2v	C2v	C2v	-	C2v	C2v		1.00000
49	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v		1.00000
50	D3h	D3h	D3h	D3h	D3h	D3h	D3h	D3h	D3h		1.00000
51	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
52	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
53	C2v	C2v	C2v	C2v	C2v	C2v	-	C2v	C2v		1.00000
54	C5v	C5v	C5	C5	C5v	C5v	-	C5v	C5v	!	1.00000
55	Ih	Ih	Ci	Ci	Ih	Ih	Ih	Ih	Ih	!	0.98488
56	Cs	Cs	Cs	Cs	Cs	Cs	-	Cs	Cs		1.00000
57	Cs	Cs	Cs	Cs	Cs	Cs	-	Cs	Cs		1.00000
58	D3h	D3h	D3h	D3h	D3h	D3h	-	D3h	D3h		1.00000
59	Td	Td	Td	Td	Td	Td	Td	Td	Td		0.86070
60	C2v	C2v	C2v	C2v	C2v	C2v	-	C2v	C2v		1.00000
61	C3v	C3v	C3v	C3v	C3v	C3v	-	C3v	C3v		1.00000
62	Cs	Cs	Cs	Cs	Cs	Cs	-	Cs	Cs		1.00000
63	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
64	C2v	C2v	C2v	C2v	C2v	C2v	-	C2v	C2v		1.00000
65	C2v	C2v	C2v	C2v	C2v	C2v	-	C2v	C2v		1.00000
66	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
67	C2v	C2v	C2v	C2v	C2v	C2v	Cs	C2v	C2v	!	1.00000
68	C3v	C3v	-	-	C3v	-	C3v	C3v	C3v		1.00000
69	C2v	C2v	C2v	C2v	C2v	C2v	Cs	C2v	C2v	!	1.00000
70	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
71	C2v	C2v	C2v	C2v	C2v	C2v	Cs	C2v	C2v	!	1.00000
72	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
73	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
74	C5v	C5v	C5	C5	C5v	-	-	C5v	C5v	!	1.00000
75	D5h	D5h	D5h	C5	D5h	D5h	-	D5h	D5h	!	1.00000
76	C2v	C2v	C2v	C2v	C2v	C2v	-	C2v	C2v		1.00000
77	C2v	C2v	C2v	C2v	C2v	C2v	Cs	C2v	C2v	!	1.00000
78	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
79	D3h	D3h	D3h	-	D3h	D3h	D3h	D3h	D3h		1.00000
80	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000

Table S52: Dataset: sutton_9_6, part 1.

name	SYVA		SymMol		'default'	libmsym			SOFI		diff	dot product
	0.001	0.05	0.001	0.05		'medium'	'loose'	0.001	0.05			
3	D3h	D3h	D3h	D3h	D3h	D3h	D3h	D3h	D3h	D3h		1.00000
4	Td	Td	Td	Td	Td	Td	Td	Td	Td	Td		0.98731
5	D3h	D3h	D3h	D3h	D3h	D3h	D3h	D3h	D3h	D3h		1.00000
6	Oh	Oh	Oh	Oh	Oh	Oh	Oh	Oh	Oh	Oh		0.88855
7	D5h	D5h	D5h	D5h	D5h	D5h	D5h	D5h	D5h	D5h		1.00000
8	D2d	D2d	D2d	D2d	D2d	D2d	D2d	D2d	D2d	D2d		1.00000
9	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
10	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v		1.00000
11	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
12	C2	C2	C2	C2	C2	C2	C2	C2	C2	C2		1.00000
13	Ih	Ih	C1	C1	Ih	Ih	Ih	Ih	Ih	Ih	!	0.99522
14	C6v	C6v	C6v	C6v	C6v	C6v	C6v	C6v	C6v	C6v		1.00000
15	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
16	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
17	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
18	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
19	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
20	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
21	C6v	C6v	C6	C6	C6v	C6v	C6v	C6v	C6v	C6v	!	1.00000
22	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
23	C2	C2	C2	C2	C2	C2	C2	C2	C2	C2		1.00000
24	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
25	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
26	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
27	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
28	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
29	C2	C2	C2	C2	C2	C2	C2	C2	C2	C2		1.00000
30	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v		1.00000
31	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
32	C3	C3	C3	C3	C3	C3	C3	C3	C3	C3		1.00000
33	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
34	C3	C3	C3	C3	C3	C3	C3	C3	C3	C3		1.00000
35	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
36	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
37	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
38	Oh	Oh	Oh	Oh	Oh	Oh	Oh	Oh	Oh	Oh		1.00000
39	C4v	C4v	C4	C4	C4v	C4v	C4v	C4v	C4v	C4v	!	1.00000
40	D4h	D4h	D4h	D4h	D4h	D4h	D4h	D4h	D4h	D4h	!	1.00000
41	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
42	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
43	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
44	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
45	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
46	C3	C3	C3	C3	C3	C3	C3	C3	C3	C3		1.00000
47	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
48	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
49	D5h	D5h	C5	C5	D5h	D5h	D5h	D5h	D5h	D5h	!	1.00000
50	D3h	D3h	D3h	D3h	D3h	D3h	D3h	D3h	D3h	D3h		1.00000
51	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
52	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
53	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v		1.00000
54	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
55	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
56	D2h	D2h	D2h	D2h	D2h	D2h	D2h	D2h	D2h	D2h		1.00000
57	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
58	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
59	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
60	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
61	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v		1.00000
62	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
63	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
64	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
65	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
66	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
67	C2v	C2v	C2v	C2v	C2v	C2v	Cs	C2v	C2v	C2v	!	1.00000
68	C3v	C3v	-	-	C3v	C3v	C3v	C3v	C3v	C3v		1.00000
69	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
70	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
71	C2v	C2v	C2v	C2v	C2v	C2v	Cs	C2v	C2v	C2v	!	1.00000
72	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
73	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
74	C5v	C5v	C5	C5	C5v	-	-	C5v	C5v	C5v	!	1.00000
75	D5h	D5h	D5h	D5h	D5h	-	-	D5h	D5h	D5h		1.00000
76	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
77	C2v	C2v	C2v	C2v	C2v	C2v	Cs	C2v	C2v	C2v	!	1.00000
78	Cs	Cs	Cs	Cs	Cs	Cs	C1	Cs	Cs	Cs	!	1.00000
79	Oh	Oh	Oh	Oh	Oh	Oh	Oh	Oh	Oh	Oh		0.98778
80	C4v	C4v	C4v	C4v	C4v	C4v	C4v	C4v	C4v	C4v		1.00000

Table S53: Dataset: sutton_10_8, part 1.

name	SYVA		SymMol		'default'	libmsym		SOFI		diff	dot product
	0.001	0.05	0.001	0.05		'medium'	'loose'	0.001	0.05		
3	D3h	D3h	D3h	D3h	D3h	D3h	D3h	D3h	D3h		1.00000
4	Td	Td	Td	Td	Td	Td	Td	Td	Td		0.99681
5	D3h	D3h	D3h	D3h	D3h	D3h	D3h	D3h	D3h		1.00000
6	Oh	Oh	Oh	Oh	Oh	Oh	Oh	Oh	Oh		1.00000
7	D5h	D5h	D5h	D5h	D5h	D5h	D5h	D5h	D5h		1.00000
8	D2d	D2d	D2d	D2d	D2d	D2d	D2d	D2d	D2d		1.00000
9	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
10	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v		1.00000
11	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
12	C5v	C5v	C5v	C5v	C5v	C5v	C5v	C5v	C5v		1.00000
13	Ih	-	Ci	Ci	Ih	Ih	Ih	Ih	Ih	!	0.94451
14	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v		1.00000
15	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
16	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
17	C2	C2	C2	C2	C2	C2	C2	C2	C2		1.00000
18	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
19	D5h	D5h	C5	C5	D5h	D5h	D5h	D5h	D5h	!	1.00000
20	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
21	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
22	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
23	D3h	D3h	D3h	D3h	D3h	D3h	-	D3h	D3h		1.00000
24	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
25	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v		1.00000
26	D3h	D3h	D3h	D3h	D3h	D3h	D3h	D3h	D3h		1.00000
27	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
28	C3v	C3v	-	-	C3v	C3v	C3v	C3v	C3v		1.00000
29	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
30	C2v	C2v	C2v	C2v	C2v	C2v	Cs	C2v	C2v	!	1.00000
31	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
32	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
33	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
34	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
35	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
36	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
37	C3v	C3v	-	-	C3v	C3v	C3v	C3v	C3v		1.00000
38	Oh	Oh	Oh	Oh	Oh	Oh	Oh	Oh	Oh		1.00000
39	C4v	C4v	C4	C4	C4v	C4v	-	C4v	C4v	!	1.00000
40	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
41	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
42	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
43	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
44	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
45	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
46	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
47	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
48	C2v	C2v	C2v	C2v	C2v	C2v	-	C2v	C2v		1.00000
49	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v		1.00000
50	D3h	D3h	D3h	D3h	D3h	D3h	D3h	D3h	D3h		1.00000
51	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
52	C3v	C3v	-	-	C3v	C3v	C3v	C3v	C3v		1.00000
53	C2v	C2v	C2v	C2v	C2v	C2v	-	C2v	C2v		1.00000
54	C5v	C5v	C5	C5	C5v	C5v	-	C5v	C5v	!	1.00000
55	Ih	Ih	Ci	Ci	Ih	Ih	Ih	Ih	Ih	!	0.97111
56	C3v	C3v	-	-	C3v	-	-	C3v	C3v		1.00000
57	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
58	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v	C3v		1.00000
59	Td	Td	Td	Td	Td	Td	Td	Td	Td		0.96339
60	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
61	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
62	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v	C2v		1.00000
63	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
64	C2v	C2v	C2v	C2v	C2v	C2v	-	C2v	C2v		1.00000
65	C2v	C2v	C2v	C2v	C2v	C2v	-	C2v	C2v		1.00000
66	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
67	C2v	C2v	C2v	C2v	C2v	C2v	Cs	C2v	C2v	!	1.00000
68	C3v	C3v	-	-	C3v	C3v	C3v	C3v	C3v		1.00000
69	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
70	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs	Cs		1.00000
71	C2v	C2v	C2v	C2v	C2v	C2v	Cs	C2v	C2v	!	1.00000
72	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
73	C2v	C2v	C2v	C2v	C2v	C2v	Cs	C2v	C2v	!	1.00000
74	C5v	C5v	C5	C5	C5v	-	-	C5v	C5v	!	1.00000
75	D5h	D5h	D5h	C5	D5h	D5h	-	D5h	D5h	!	1.00000
76	Cs	Cs	Cs	Cs	Cs	Cs	-	Cs	Cs		1.00000
77	C2v	C2v	C2v	C2v	C2v	C2v	Cs	C2v	C2v	!	1.00000
78	C1	C1	C1	C1	C1	C1	C1	C1	C1		0.00000
79	D3h	D3h	D3h	-	D3h	D3h	D3h	D3h	D3h		1.00000
80	Cs	Cs	Cs	Cs	Cs	Cs	-	Cs	Cs		1.00000

Table S54: Dataset: sutton_12_6, part 1.

References

- [1] M. Gunde, N. Salles, L. Grisanti, L. MARTIN-SAMOS, and A. Hemeryck, “SOFI: Finding point group symmetries in atomic clusters as finding the set of degenerate solutions in a shape-matching problem,” Jul. 2024. [Online]. Available: <https://doi.org/10.5281/zenodo.12773914>
- [2] L. Gyevi-Nagy and G. Tasi, “Syva: A program to analyze symmetry of molecules based on vector algebra,” *Computer Physics Communications*, vol. 215, pp. 156–164, 2017. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0010465517300310>
- [3] T. Pilati and A. Forni, “SYMMOL: a program to find the maximum symmetry group in an atom cluster, given a prefixed tolerance,” *Journal of Applied Crystallography*, vol. 31, no. 3, pp. 503–504, Jun 1998. [Online]. Available: <https://doi.org/10.1107/S0021889898002180>
- [4] M. Johansson and V. Veryazov, “Automatic procedure for generating symmetry adapted wavefunctions,” *Journal of Cheminformatics*, vol. 9, no. 1, p. 8, Feb 2017. [Online]. Available: <https://doi.org/10.1186/s13321-017-0193-3>
- [5] D. J. Wales, J. P. K. Doye, A. Dullweber, M. P. Hodges, F. Y. N. F. Calvo, J. Hernández-Rojas, and T. F. Middleton, “The cambridge cluster database.” [Online]. Available: <https://www-wales.ch.cam.ac.uk/CCD.html>
- [6] D. J. Wales and J. P. K. Doye, “Global optimization by basin-hopping and the lowest energy structures of lennard-jones clusters containing up to 110 atoms,” *The Journal of Physical Chemistry A*, vol. 101, no. 28, pp. 5111–5116, 1997. [Online]. Available: <https://doi.org/10.1021/jp970984n>
- [7] F. Calvo, J. P. K. Doye, and D. J. Wales, “Quantum partition functions from classical distributions: Application to rare-gas clusters,” *Journal of Chemical Physics*, vol. 114, pp. 7312–7329, 2001. [Online]. Available: <https://doi.org/10.1063/1.1359768>
- [8] J. P. K. Doye and D. J. Wales, “Global minima for transition metal clusters described by suttonchen potentials,” *New J. Chem.*, vol. 22, pp. 733–744, 1998. [Online]. Available: <http://dx.doi.org/10.1039/A709249K>
- [9] J. P. K. Doye, “A model metal potential exhibiting polytetrahedral clusters,” *The Journal of Chemical Physics*, vol. 119, no. 2, pp. 1136–1147, 2003. [Online]. Available: <https://doi.org/10.1063/1.1574797>
- [10] J. P. K. Doye and S. C. Hendy, “On the structure of small lead clusters,” *The European Physical Journal D - Atomic, Molecular, Optical and Plasma Physics*, vol. 22, no. 1, pp. 99–107, Jan 2003. [Online]. Available: <https://doi.org/10.1140/epjd/e2002-00232-x>
- [11] J. P. K. Doye, “Identifying structural patterns in disordered metal clusters,” *Phys. Rev. B*, vol. 68, p. 195418, Nov 2003. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevB.68.195418>
- [12] E. G. Noya, J. P. K. Doye, D. J. Wales, and A. Aguado, “Geometric magic numbers of sodium clusters: Interpretation of the melting behaviour,” *The European Physical Journal D*, vol. 43, no. 1, pp. 57–60, Jul 2007. [Online]. Available: <https://doi.org/10.1140/epjd/e2007-00092-x>
- [13] J. P. Doye, “Lead clusters: Different potentials, different structures,” *Computational Materials Science*, vol. 35, no. 3, pp. 227–231, 2006, proceedings of the 4th International Conference on the Theory of Atomic and Molecular Clusters (TAMC-IV). [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0927025605001278>
- [14] J. Elliott, Y. Shibuta, and D. Wales, “Global minima of transition metal clusters described by finnisinclair potentials: A comparison with semi-empirical molecular orbital theory,” *Philosophical Magazine*, vol. 89, no. 34-36, pp. 3311–3332, 2009. [Online]. Available: <https://doi.org/10.1080/14786430903270668>
- [15] J. P. Doye and D. J. Wales, “The structure of (c60)_n clusters,” *Chemical Physics Letters*, vol. 262, no. 1, pp. 167–174, 1996. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/0009261496010391>
- [16] J. P. K. Doye, D. J. Wales, W. Branz, and F. Calvo, “Modeling the structure of clusters of c₆₀ molecules,” *Phys. Rev. B*, vol. 64, p. 235409, Nov 2001. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevB.64.235409>
- [17] D. J. Wales and J. P. K. Doye, *Theoretical Predictions of Structure and Thermodynamics in the Large Cluster Regime*. Dordrecht: Springer Netherlands, 1996, pp. 241–279. [Online]. Available: https://doi.org/10.1007/978-94-009-0211-4_9

- [18] J. P. K. Doye and D. J. Wales, "The effect of the range of the potential on the structure and stability of simple liquids: from clusters to bulk, from sodium to," *Journal of Physics B: Atomic, Molecular and Optical Physics*, vol. 29, no. 21, p. 4859, nov 1996. [Online]. Available: <https://dx.doi.org/10.1088/0953-4075/29/21/002>
- [19] J. P. K. Doye and D. J. Wales, "Structural consequences of the range of the interatomic potential a menagerie of clusters," *J. Chem. Soc., Faraday Trans.*, vol. 93, pp. 4233–4243, 1997. [Online]. Available: <http://dx.doi.org/10.1039/A706221D>
- [20] J. P. K. Doye, D. J. Wales, and S. I. Simdyankin, "Global optimization and the energy landscapes of dzugutov clusters," *Faraday Discuss.*, vol. 118, pp. 159–170, 2001. [Online]. Available: <http://dx.doi.org/10.1039/B008881L>
- [21] J. P. K. Doye and D. J. Wales, "Polytetrahedral clusters," *Phys. Rev. Lett.*, vol. 86, pp. 5719–5722, Jun 2001. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevLett.86.5719>
- [22] J. P. K. Doye and D. J. Wales, "Structural transitions and global minima of sodium chloride clusters," *Phys. Rev. B*, vol. 59, pp. 2292–2300, Jan 1999. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevB.59.2292>
- [23] J. P. K. Doye and L. Meyer, "Mapping the magic numbers in binary lennard-jones clusters," *Phys. Rev. Lett.*, vol. 95, p. 063401, Aug 2005. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevLett.95.063401>
- [24] M. D. Hanwell, D. E. Curtis, D. C. Lonie, T. Vandermeersch, E. Zurek, and G. R. Hutchison, "Avogadro: an advanced semantic chemical editor, visualization, and analysis platform," *Journal of Cheminformatics*, vol. 4, no. 1, p. 17, Aug 2012. [Online]. Available: <https://doi.org/10.1186/1758-2946-4-17>