ToBaCCo 3.0 Manual

Authors

Ryther Anderson
Yamil Colón
Diego Gomez-Gualdrón[1]

[1] Please adress correspondance Diego Gómez-Gualdrón at dgomezgualdron@mines.edu

Table of contents

1 Background and Motivation

ToBaCCo stands for Topologically Based Crystal Constructor. It takes as input molecular building blocks and topological blueprints and outputs porous crystals constructed from the molecular building blocks with underlying topologies according to the topological blueprints as crystallographic information (.cif) files. These crystals can then be used in molecular simulation or as an aid to materials characterization. The ToBaCCo crystal construction procedure allows for the rapid computational "synthesis" of thousands of porous crystals, as well as the targeted synthesis of a single crystal or class of crystals. Our goal is for ToBaCCo to be an efficient and valuable tool for material discovery and characterization.

2 Installation and Dependencies

The ToBaCCo code can be cloned or downloaded from https://github.com/tobacco-mofs. ToBaCCo is written in Python 2.7, and requires the networkx, scipy, and numpy modules.
We recommend building a Python 2.7 environment specifically for ToBaCCo using Anaconda (see Anaconda website). Once the code is cloned or downloaded and the requirements are installed ToBaCCo is ready to run (no compilation steps are required).

3 Basic Usage

Essentially, ToBaCCo reads in a topological blueprint (see section 4), finds compatible node building blocks (see section 5) from a defined set, and constructs all possible crystal structures with the topology defined by the blueprint resulting from different combinations of the compatible node building blocks and members of a defined set of edge building blocks (see section 5). This is accomplished by scaling the blueprint (resulting in a geometrical representation of the topology defined by node coordinates) to fit the geometry of each node/edge combination, then translating/rotating the node and edge building blocks into the scaled coordinates. Thus, when presented with a large number of compatible nodes/edges many crystals can be built from a single topology. Topologies are read from the "templates" directory included with the ToBaCCo code. If multiple templates are present, crystal construction will be attempted for each template (in a for loop). Node and edge building blocks are read from the "nodes" and "edges" directory respectively. Each node and edge building block present in their respective directories will be considered for each topology. If no compatible nodes are found for a certain topology no crystal will be constructed for that topology. Compatibility is assessed by comparing the number of connections node building blocks can make (with edge building blocks), the number of incident edges for each type of node in the topological blueprint.

For specific details on crystallographic topologies and the ToBaCCo algorithm see

Colon, Y. J., Gomez-Gualdron, D. & Snurr, R. Q. Topologically-Guided, Automated Construction of MOFs and their Evaluation for Energy-Related Applications. *Cryst. Growth Des.* (2017). doi:10.1021/acs.cgd.7b00848

Anderson, R. & Gómez-Gualdrón, D. A. Increasing topological diversity during computational“synthesis” of porous crystals: how and why. *CrystEngComm* **21,** 1653–1665 (2019).

Once the desired templates and node/edge building blocks are in their respective directories ToBaCCo can be run in a terminal window thus:

```
python tobacco.py
```

(windows users can run the same command in an Anaconda terminal, which is included with the Anaconda distribution of Python). Constructed crystals will be output in the “output_cifs” directory included with ToBaCCo.

4 Topological Blueprints

ToBaCCo reads topological blueprints as CIFs (i.e. .cif files). The CIF format is commonly used to represent the geometry of molecular crystals, where each atom in the crystal is given a label (usually corresponding to its atomic symbol, possibly together with a numerical index), and a 3-dimensional coordinate. Optionally, each bond in the crystal can also be defined in a CIF as list of atoms pairs (each pair defining a bond between its members). Frequently, CIFs include only the atoms/bonds needed to generate all the atoms/bonds in the crystallographic unit cell once the space group symmetry operations (also provided in the CIF) are included. In the topological blueprint CIFs read by ToBaCCo, each node is represented by an element, with 3D coordinate, and edges are represented by bonds. Note that coordinates of *every* node are included (removing the need to include space group information).

In general, crystallographic topologies can have *N* types of nodes and *M* types of edges. ToBaCCo is capable of reading topologies with up to 30 types of nodes and any number of edges. By convention, the first five node types are labelled with the elements V, Er, Ti, Ce, S (covers most useful cases). The next 25 types can be labelled by any elements 1 through 28 (H through Ni), which have not already been used. An example template for the topology **pfm** is shown below.

```
data_pfm\(2)\(2)
_audit_creation_date              2018-12-01
_audit_creation_method            'Materials Studio'
_symmetry_space_group_name_H-M    'P1'
_symmetry_Int_Tables_number       1
_symmetry_cell_setting            triclinic
loop_
_symmetry_equiv_pos_as_xyz
  x,y,z
_cell_length_a                    14.1420
_cell_length_b                    14.1420
_cell_length_c                    28.2840
_cell_angle_alpha                 90.0000
_cell_angle_beta                  90.0000
_cell_angle_gamma                 90.0000
```

```
loop_
_atom_site_label
_atom_site_type_symbol
_atom_site_fract_x
_atom_site_fract_y
_atom_site_fract_z
_atom_site_U_iso_or_equiv
_atom_site_adp_type
_atom_site_occupancy
V1    V    0.00000   0.00000   0.00000   0.00000  Uiso   1.00
V2    V    0.50000   0.50000   0.50000   0.00000  Uiso   1.00
Er3   Er   0.00000   0.50000   0.25000   0.00000  Uiso   1.00
Er4   Er   0.50000   0.00000   0.25000   0.00000  Uiso   1.00
Er5   Er   0.00000   0.50000   0.75000   0.00000  Uiso   1.00
Er6   Er   0.50000   0.00000   0.75000   0.00000  Uiso   1.00
Ti7   Ti   0.00000   0.00000   0.50000   0.00000  Uiso   1.00
Ti8   Ti   0.50000   0.50000   0.00000   0.00000  Uiso   1.00
loop_
_geom_bond_atom_site_label_1
_geom_bond_atom_site_label_2
_geom_bond_distance
_geom_bond_site_symmetry_2
_ccdc_geom_bond_type
V1    Er3   10.000   .     S
V1    Er4   10.000   .     S
V1    Ti8   10.000   1_445 S
V1    Ti8   10.000   1_545 S
V1    Ti8   10.000   1_455 S
V1    Er5   10.000   1_554 S
V1    Er5   10.000   1_544 S
V1    Er6   10.000   1_454 S
V1    Er6   10.000   1_554 S
V1    Er3   10.000   1_545 S
V1    Er4   10.000   1_455 S
V2    Ti7   10.000   .     S
V2    Er3   10.000   .     S
V2    Er4   10.000   .     S
V2    Er5   10.000   .     S
V2    Er6   10.000   .     S
V2    Ti7   10.000   1_665 S
V2    Ti7   10.000   1_655 S
V2    Ti7   10.000   1_565 S
V2    Er4   10.000   1_565 S
V2    Er3   10.000   1_655 S
V2    Er6   10.000   1_565 S
V2    Er5   10.000   1_655 S
Er3   V1    10.000   1_565 S
```

```
Er3   V2    10.000   1_455 S
Er4   V1    10.000   1_655 S
Er4   V2    10.000   1_545 S
Er5   V1    10.000   1_556 S
Er5   V1    10.000   1_566 S
Er5   V2    10.000   1_455 S
Er6   V1    10.000   1_656 S
Er6   V1    10.000   1_556 S
Er6   V2    10.000   1_545 S
Ti7   V2    10.000   1_445 S
Ti7   V2    10.000   1_455 S
Ti7   V2    10.000   1_545 S
Ti8   V1    10.000   1_665 S
Ti8   V1    10.000   1_565 S
Ti8   V1    10.000   1_655 S
V1    Ti8   10.000   .     S
```

The **pfm** topology has three types of nodes (labelled, V, Er, and Ti). For the template to be compatible with ToBaCCo the unit cell parameters (a, b, c, α, β, γ) must be include and defined thus:

```
_cell_length_a                 14.1420
_cell_length_b                 14.1420
_cell_length_c                 28.2840
_cell_angle_alpha               90.0000
_cell_angle_beta                90.0000
_cell_angle_gamma               90.0000
```

In addition, in the section defining node coordinates, e.g.

```
V1    V     0.00000   0.00000   0.00000   0.00000  Uiso   1.00
V2    V     0.50000   0.50000   0.50000   0.00000  Uiso   1.00
Er3   Er    0.00000   0.50000   0.25000   0.00000  Uiso   1.00
Er4   Er    0.50000   0.00000   0.25000   0.00000  Uiso   1.00
Er5   Er    0.00000   0.50000   0.75000   0.00000  Uiso   1.00
Er6   Er    0.50000   0.00000   0.75000   0.00000  Uiso   1.00
Ti7   Ti    0.00000   0.00000   0.50000   0.00000  Uiso   1.00
Ti8   Ti    0.50000   0.50000   0.00000   0.00000  Uiso   1.00
```

for **pfm**, the first five columns *must* correspond to the node type combined with a numerical index (making a unique label for each node in the template), node type, fractional x-coordinate, fractional y-coordinate, and fractional z-coordinate, respectively and in that order. Any additional columns will be ignored. Finally, the first four columns in the section defining the edges (i.e. "bonds" between nodes) *must* correspond to the first node defining the edge, second node defining the edge, the edge length in Å, and the edge symmetry label, respectively and in that order. Any additional columns will be ignored. The first five edges in the **pfm** topology, defined following this convention are reproduced below.

```
V1    Er3   10.000  .    S
V1    Er4   10.000  .    S
V1    Ti8   10.000  1_445 S
V1    Ti8   10.000  1_545 S
V1    Ti8   10.000  1_455 S
```

Any other section present in the template CIF will be ignored, although we recommend the “loop” sections defining the node coordinate and edge columns should be retained for clarity and to retain compatibility with future versions of ToBaCCo. For example:

```
loop_
_atom_site_label
_atom_site_type_symbol
_atom_site_fract_x
_atom_site_fract_y
_atom_site_fract_z
_atom_site_U_iso_or_equiv
_atom_site_adp_type
_atom_site_occupancy
```

5 Molecular Building Blocks

ToBaCCo reads molecular building blocks from CIFs. Unlike topological templates, molecular building blocks are comprised of actual atoms bonded together into molecules, and the coordinate and bond sections of the CIFs used to define building blocks for ToBaCCo reflect this. That is, atoms are labelled by their element, and the bonding section should contain the actual bonds present in the molecule. An example CIF for a node building block is shown below.

```
data_sym_3_on_2
_audit_creation_date              2018-12-11
_audit_creation_method           'tobacco_3.0'
_symmetry_space_group_name_H-M    'P1'
_symmetry_Int_Tables_number       1
_symmetry_cell_setting            triclinic
loop_
_symmetry_equiv_pos_as_xyz
  x,y,z
_cell_length_a                   20.0000
_cell_length_b                   20.0000
_cell_length_c                   20.0000
_cell_angle_alpha                90.0000
_cell_angle_beta                 90.0000
_cell_angle_gamma                90.0000
loop_
_atom_site_label
_atom_site_type_symbol
_atom_site_fract_x
```

```
_atom_site_fract_y
_atom_site_fract_z
_atom_site_U_iso_or_equiv
_atom_site_adp_type
_atom_site_occupancy
_atom_site_charge
X1      C  -0.0770000000   0.0000000000   0.0000000000  0.00000 Uiso  1.00       0.034261
C2      C  -0.0385000000  -0.0666800000   0.0000000000  0.00000 Uiso  1.00      -0.127236
X3      C   0.0385000000  -0.0666800000   0.0000000000  0.00000 Uiso  1.00       0.009264
C4      C   0.0770000000   0.0000000000  -0.0000000000  0.00000 Uiso  1.00      -0.112496
X5      C   0.0385000000   0.0666800000  -0.0000000000  0.00000 Uiso  1.00       0.03418
C6      C  -0.0385000000   0.0666800000  -0.0000000000  0.00000 Uiso  1.00      -0.106703
H7      H  -0.0670000000  -0.1160500000   0.0000000000  0.00000 Uiso  1.00       0.090128
H8      H   0.1340000000  -0.0000000000  -0.0000000000  0.00000 Uiso  1.00       0.065416
H9      H  -0.0670000000   0.1160500000  -0.0000000000  0.00000 Uiso  1.00       0.063889
loop_
_geom_bond_atom_site_label_1
_geom_bond_atom_site_label_2
_geom_bond_distance
_geom_bond_site_symmetry_2
_ccdc_geom_bond_type
X1      C2 1.540    .  A
X1      C6 1.540    .  A
C2      X3 1.540    .  A
C2      H7 1.140    .  S
X3      C4 1.540    .  A
C4      X5 1.540    .  A
C4      H8 1.140    .  S
X5      C6 1.540    .  A
C6      H9 1.140    .  S
```

For the building block CIF to be compatible with ToBaCCo the unit cell parameters (a, b, c, α, β, γ) must be include and defined, for example:

```
_cell_length_a                 10.0000
_cell_length_b                 10.0000
_cell_length_c                 10.0000
_cell_angle_alpha              90.0000
_cell_angle_beta               90.0000
_cell_angle_gamma              90.0000
```

Also, in the section defining atom coordinates, e.g.,

```
X1      C  -0.0770000000   0.0000000000   0.0000000000  0.00000 Uiso  1.00       0.034261
C2      C  -0.0385000000  -0.0666800000   0.0000000000  0.00000 Uiso  1.00      -0.127236
X3      C   0.0385000000  -0.0666800000   0.0000000000  0.00000 Uiso  1.00       0.009264
```

C4	C	0.0770000000	0.0000000000	-0.0000000000	0.00000	Uiso	1.00	-0.112496
X5	C	0.0385000000	0.0666800000	-0.0000000000	0.00000	Uiso	1.00	0.03418
C6	C	-0.0385000000	0.0666800000	-0.0000000000	0.00000	Uiso	1.00	-0.106703
H7	H	-0.0670000000	-0.1160500000	0.0000000000	0.00000	Uiso	1.00	0.090128
H8	H	0.1340000000	-0.0000000000	-0.0000000000	0.00000	Uiso	1.00	0.065416
H9	H	-0.0670000000	0.1160500000	-0.0000000000	0.00000	Uiso	1.00	0.063889

the first five columns *must* correspond to the element combined with a numerical index (making a unique label for each atom in the template), element, fractional x-coordinate, fractional y-coordinate, and fractional z-coordinate, respectively and in that order. If the user is employing the ToBaCCo MBBB charge assignment method (see section 6), the last column should give the charge of each atom. Atoms which are connection points are labelled as "X", but their correct element (in this case carbon) is given in the next column. Not that in the above example there are three connection site atoms, all carbon.

Finally, the first five columns in the section defining the bonds *must* correspond to the first atom defining the bond, second atom defining the bond, the bond length in Å, the bond symmetry label, and the bond type ("S" for single, "D" for double, "T" for triple, and "A" for aromatic). respectively and in that order. Any additional columns will be ignored. The bond section for the example node building block is shown below.

X1	C2	1.540	.	A
X1	C6	1.540	.	A
C2	X3	1.540	.	A
C2	H7	1.140	.	S
X3	C4	1.540	.	A
C4	X5	1.540	.	A
C4	H8	1.140	.	S
X5	C6	1.540	.	A
C6	H9	1.140	.	S

CIFs for edge building blocks follow the same format as those for node building blocks, although edge building blocks must always have only two connection site atoms.

Finally, the coordinates given in the building block CIFs are converted to cartesian coordinates. So, if a building block lies across a periodic boundary in the CIF, some atoms may be separated by the box distance when converted to cartesian coordinates. To avoid this ToBaCCo automatically moves all the building block coordinates into periodic images (if necessary) to ensure the correct geometry is preserved in cartesian coordinates. However, this does not work if the end-to-end length of the building block is longer than half the box length. Thus, building blocks should be placed in boxes larger than this.

6 Atomic Charges

ToBaCCo can build crystals with atomic charges according to the Molecular Building Block Based (MBBB) charge assignment method described in

Argueta, E. *et al.* Molecular Building Block-Based Electronic Charges for High-Throughput Screening of Metal–Organic Frameworks for Adsorption Applications. *J. Chem. Theory Comput.* **14,** 365–376 (2018).

In this method, charges assigned to the molecular building blocks (typically using a non-periodic DFT calculation) of a crystal are reproduced in the infinite crystal (as closely as possible). If the user specifies that charges are to be used (see section 7), charges are read from each node/edge building block and reassigned to the constructed CIF. Typically, this results in a small, but non-negligible net charge in the crystalline unit cell. This net charge is cancelled by slightly rescaling the charges of all non-metal atoms in the crystal to result in a zero net charge.

7 Configuration and Advanced Usage

The ToBaCCo distribution contains a file "configuration.py" which can be altered to change how ToBaCCo handles output, node and edge assignment, scaling templates, and special cases such as one atom nodes or node-to-node connections (i.e. connections without an edge building block). An example of this file is shown below.

```
IGNORE_ALL_ERRORS = False
PRINT = False
CONNECTION_SITE_BOND_LENGTH = 1.54
WRITE_CHECK_FILES = False
WRITE_CIF = True
ALL_NODE_COMBINATIONS = False
USER_SPECIFIED_NODE_ASSIGNMENT = False
COMBINATORIAL_EDGE_ASSIGNMENT = True
CHARGES = True
SYMMETRY_TOL = {2:0.10, 3:0.10, 4:0.35, 5:0.25, 6:0.30, 7:0.35, 8:0.40, 9:0.60, 10:0.60, 12:0.60}
BOND_TOL = 5.0
ORIENTATION_DEPENDENT_NODES = False
PLACE_EDGES_BETWEEN_CONNECTION_POINTS = True
RECORD_CALLBACK = False
OUTPUT_SCALING_DATA = True
FIX_UC = (0,0,0,0,0,0)
MIN_CELL_LENGTH = 5.0
OPT_METHOD = 'L-BFGS-B'
PRE_SCALE = 1.00
SCALING_ITERATIONS = 1
SINGLE_METAL_MOFS_ONLY = True
MOFS_ONLY = True
MERGE_CATENATED_NETS = True
RUN_PARALLEL = False
REMOVE_DUMMY_ATOMS = True
```

Each element of this file and their possible assignments will be explained in turn in the following sub-sections.

7.1 PRINT [True | False]

If True, additional output will be printed about the cycle-cocyle space of each template. This is included primarily for debugging purposes and does not affect input or the output structures. Generally it is best to have Print = False so one doesn't drown in information.

7.2 CONNECTION_SITE_BOND_LENGTH [float]

This defines the distance between node-edge connection sites in Å, e.g. the bond length between node connection site atoms and bond connection site atoms. The resulting crystal will connection site bond lengths as close as possible to this value. Generally, this is set between 1.5 and 1.7 Å (since it typically corresponds to organic bonds involving carbon). This distance will be fixed by geometry-optimization according to a force-field, but it is good practice to make it close to the physical value.

7.3 WRITE_CHECK_FILES [True | False]

If True, several additional CIFs are written to the "check_cifs" directory for each template/constructed crystal. CIFs of the template node geometry before and after scaling will be written, as well as a CIF for each crystal that does not include any bonds between atoms. This option is included primarily for debugging purposes, and should generally be swiched off.

7.4 WRITE_CIF [True | False]

If True, a CIF for each constructed crystal will be written as a CIF. This option is included to be used in future versions of ToBaCCo which will output crystals in additional formats.

7.5 USER_SPECIFIED_NODE_ASSIGNMENT [True | False]

If True, the user can specify exactly the molecular building block assigned to each node type. This specification can be made either with a text file (recommended) or by standard input (keyboard prompt). A standard input prompt will only be made if the text file input cannot be found in the ToBaCCo directory. The node specification text file is called "vertex_assignment.txt", a template for which is shown below.

```
V  N1.cif
Er N2.cif
Ti N3.cif
Ce N4.cif
S  N5.cif
```

...

Each line gives a CIF (second column) to use for each node type (see section 4). Up to 30 node types can be assigned, above only five are shown. For each assignment a CIF of the same name *must* be present in the "nodes" directory. If any node building block is not compatible with its assigned node, no crystal will be output. Different nodes can be assigned the same CIF. This option should be set to True when the user wants to target a specific crystal or class of crystals.

7.6 COMBINATORIAL_EDGE_ASSIGNMENT [True | False]

If True, crystals of all possible edge assignments will be constructed. For example, if a template has two edge types and two edge building blocks are provided, four unique crystals can be constructed. If False, edges will be assigned edge building blocks in alphanumeric order according to the name of their CIF. Setting this option to False can make targeted crystal synthesis more efficient for topologies with many edge types.

7.7 CHARGES [True | False]

If True, atomic charges provided in the molecular building block CIFs (see section 5) will be remapped to the constructed crystals (see section 6). Any net charge resulting from this mapping (should be minimal, if the building block charges were assigned correctly), will be removed by rescaling the charges of non-metal atoms by a small amount.

7.8 SYMMETRY_TOL [dictionary of floats]

The default value of SYMMETRY_TOL is {3:0.2, 4:0.2, 5:0.2, 6:0.2, 7:0.5, 8:0.5, 9:0.5, 10:0.5, 12:0.5, 24:1.0}. This defines the root-mean squared deviation allowed between node building block unit connection site vectors and the unit edge vectors *when comparing their directions only* (i.e. unit vectors are compared so magnitude is not considered). This is defined on a coordination number basis. That is, for the above default values, nodes with coordination number 4 will not be assigned building blocks whose geometry deviates from the *unscaled* node geometry by more than 0.2 Å. In general, higher coordination number nodes should be allowed more deviation. Note that the default values are strict and should be relaxed depending on the needs of the user.

7.9 BOND_TOL [float]

This defines the distance criteria in Å used to form connection site bonds, in two atoms labeled "X" (i.e. connection sites) are within this distance a bond will be added between them. Note that this tolerance can be quite large and still result in accurate structures as the structure is scanned for incorrect connection site bonding before the final CIF is written. A values of between 3 and 10 Å is usually sufficient but should always greater than CONNECTION_SITE_BOND_LENGTH. 10 Å is a reasonable default which works with most building blocks.

7.10 ORIENTATION_DEPENDENT_NODES [True | False]

This flag is included for future versions where nodes can be placed with a specified orientation. This should be False for most users but setting it to True will likely produce the same results for most nodes.

7.11 PLACE_EDGES_BETWEEN_CONNECTION_POINTS [True | False]
If True, edge building blocks are adjusted in the final crystal to have their center of mass between the node connection sites they are connected to, rather than the node center of mass. In cases where an imperfect scaling solution is achieved for the template, this produces more realistic structures. We recommend that this option is always set to True, as for perfect or near perfect scaling is achieved the resulting crystal is negligibly changed, but the structure is significantly improved for poor scaling results.

7.12 RECORD_CALLBACK [True | False]

If True, callback data from the scaling optimizer will be recorded. The shows how the objective function decreases over time. In addition, an animated .xyz file showing the scaling process will be written (which one can watch in VMD), and the changes in the unit cell parameters during scaling are also written as an .mp4 file. This option was mainly included for use by our group when presenting about ToBaCCo but could be useful to other users as well.

7.13 MIN_CELL_LENGTH float

Sets the minimum length for the *a*, *b*, *c* cell vectors. Using 0.0 or 1.0 works well, sometimes larger values should be set to prevent cell collapse.

7.14 OPT_METHOD string

Currently this flag should be 'L-BFGS-B' or 'differential_evolution'. This sets the optimization algorithm to use. If the value is 'L-BFGS-B' local minimization will be used according to the scipy L-BFGS-B algorithm. If the value is 'differential_evolution' global minimization will be used according to the scipy differential evolution algorithm.

7.15 FIX_UC list or tuple of 0/1s of length 6.

This flag can be used to fix some or all unit cell parameters during scaling. Each entry in the list corresponds to a unit cell parameter in the order $a, b, c, \alpha, \beta, \gamma$. For example, if the value given is (1,0,0,0,0,0) the a parameter will be frozen during scaling. This is useful when using 2D templates where one dimension should be neglected during scaling. It is also useful for certain materials characterization usages, e.g. if one wants to match an experimental PXRD pattern where only unit cell parameters and/or node positions are known.

7.16 PRE_SCALE float

The unscaled unit cell parameters are multiplied by this value before scaling, this is for materials characterization usage, and can be used to change the net before scaling (which is mainly useful if the unit cell is fixed).

7.17 SCALING ITERATIONS integer

The ToBaCCo scaling algorithm can be run in a series of iterations, where during each iteration the unit cell parameters and vertex positions are optimized, then the parameters and positions are perturbed a small amount (followed by another optimization/perturbation, etc.). Larger values of this parameter will increase the time made to build each MOF but may also improve the symmetry/organization of the final structure, we recommend between 3 and 5 iterations in general, but this is a parameter which should be explored. However, good results can usually be obtained with only 1 iteration. Setting this parameter to 0 means *no scaling will be performed*.

7.18 SINGLE_METAL_MOFS_ONLY [True | False]

If True, ToBaCCo will only output MOFs that have nodes with exactly one metal element. This is useful for building large databases where covalent organic frameworks and multi-metal MOFs are not wanted.

7.19 MOFS_ONLY [True | False]

If True, ToBaCCo will only output MOFs (i.e. metal containing structures). This is useful for building large databases where covalent organic frameworks are not wanted. If SINGLE_METAL_MOFS_ONLY is False and this flag is True, multi-metal MOFs can be output.

7.20 MERGE_CATENATED_NETS [True | False]

ToBaCCo automatically detects a catenated net and will make a CIF corresponding to each network that makes up the catenated system. To merge these CIFs into a single CIF, set this flag to True.

7.21 RUN_PARALLEL [True | False]

If True, ToBaCCo will run parallel across the set of templates, i.e. each template will be sent to a different processor. Note that many error messages are disabled in this mode, so use with caution. I would not recommend using the for a typical machine with 4 cores. However, if you have access to an HPC platform, and can run on 10s to 100s of processors, this is a big speed up for database creation.

7.22 REMOVE_DUMMY_ATOMS [True | False]

This will remove dummy atoms (element ‘Fr’) from building blocks. Dummy atoms can be used to produce MOFs with single atom nodes or node to node connections. In ToBaCCo, one atom nodes are described by the node atom and several dummy atoms which define connection sites, for example:

```
data_1AN
_audit_creation_date              2018-08-15
_audit_creation_method            'Materials Studio'
```

```
_symmetry_space_group_name_H-M    'P1'
_symmetry_Int_Tables_number       1
_symmetry_cell_setting            triclinic
loop_
_symmetry_equiv_pos_as_xyz
  x,y,z
_cell_length_a                    10.0000
_cell_length_b                    10.0000
_cell_length_c                    10.0000
_cell_angle_alpha                 90.0000
_cell_angle_beta                  90.0000
_cell_angle_gamma                 90.0000
loop_
_atom_site_label
_atom_site_type_symbol
_atom_site_fract_x
_atom_site_fract_y
_atom_site_fract_z
_atom_site_U_iso_or_equiv
_atom_site_adp_type
_atom_site_occupancy
_atom_site_charge
C1    C    0.27679   0.15821   0.30141   0.00000  Uiso   1.00   0.00 -0.02191
X2    Fr   0.23879   0.05439   0.32922   0.00000  Uiso   1.00   0.00 0
X3    Fr   0.23879   0.18603   0.19759   0.00000  Uiso   1.00   0.00 0
X4    Fr   0.23879   0.23421   0.37741   0.00000  Uiso   1.00   0.00 0
X5    Fr   0.39079   0.15821   0.30141   0.00000  Uiso   1.00   0.00 0
loop_
_geom_bond_atom_site_label_1
_geom_bond_atom_site_label_2
_geom_bond_distance
_geom_bond_site_symmetry_2
_ccdc_geom_bond_type
C1    X2    1.140   .    S
C1    X3    1.140   .    S
C1    X4    1.140   .    S
C1    X5    1.140   .    S
```

is the CIF for a one atom node consisting of carbon, which can form four connections (dummy atoms are labelled “Fr”). The second element of the tuple defines how many bonds one atom nodes of a particular element can form. Thus, in the above example, carbon nodes can form four bonds, as can zinc nodes. In general, the first element in the tuple should always be True, since it does not affect output if one atom nodes are not included.

8 Standard Output

By default, ToBaCCo prints some useful information, additional information can be output using the PRINT, TRACE_BOND_MAKING, and RECORD_CALLBACK flags discussed in the previous section. Below is the default output (i.e. with all the above flags set to False) for a constructing a MOF according to the **acs** topology:

```
================================================================================================
template : acs.cif
================================================================================================

Number of vertices =  2
Number of edges =  6

*********************************************************************
RMSD of the compatible node BBs with assigned vertices:
*********************************************************************

vertex V (6 connected)
    6c_Cr_1.cif deviation = 0.09012 (within tolerance)
    6c_Cu_1.cif deviation = 0.42265 (outside tolerance)
    6c_Zr_1.cif deviation = 0.49096 (outside tolerance)
    6c_Zn_1.cif deviation = 0.42265 (outside tolerance)
* 1 compatible building blocks out of 4 available for node V *

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
vertex assignment :  ['6c_Cr_1.cif']
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
edge assignment :  ('1B_2OH.cif',)
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

scaling unit cell and vertex positions...

*******************************************
The scaled unit cell parameters are :
*******************************************
a    : 16.04626
b    : 16.04604
c    : 15.78555
alpha: 90.0002
beta : 90.00024
gamma: 119.99985
```

```
computing X-X bonds...

*******************************************
Bond formation :
*******************************************
distance search tolerance is 15.213 Angstroms
there were 12 X-X bonds formed
bond check passed

*******************************************
Charge information :
*******************************************
old net charge                : 0.60271
new net charge (after rescaling): 0.0
rescaling magnitude           : 0.00548

writing cif...

Normal termination of Tobacco_3.0 after
--- 1.70928192139 seconds ---
```

First, the template name is printed with the number of edges and vertices (users should check the number of edges and vertices is correct)

```
===============================================================================================
template : acs.cif
===============================================================================================

Number of vertices =  2
Number of edges =  6
```

Next, the root mean squared deviation between coordination-compatible nodes/vertices is printed (for each vertex, with the candidate CIF names printed as well). If the deviation is outside the tolerance defined by SYMMETRY_TOL that will be indicated. Finally, for each vertex, the number of combatable building blocks out of the total number of candidates will be printed.

```
*****************************************************************
RMSD of the compatible node BBs with assigned vertices:
*****************************************************************

vertex V (6 connected)
    6c_Cr_1.cif deviation = 0.09012 (within tolerance)
```

```
    6c_Cu_1.cif deviation = 0.42265 (outside tolerance)
    6c_Zr_1.cif deviation = 0.49096 (outside tolerance)
    6c_Zn_1.cif deviation = 0.42265 (outside tolerance)
* 1 compatible building blocks out of 4 available for node V *
```

Next, the current vertex and edge assignments are printed. These indicate the start of each new MOF made according to each template (for this template only one MOF is made, so only one vertex and edge assignment is presented).

```
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
vertex assignment :  ['6c_Cr_1.cif']
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++


++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
edge assignment :  ('1B_2OH.cif',)
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

Next, ToBaCCo will indicate it is scaling the unit cell parameters and vertex positions (this can take some time, depending on the size of the template). Once scaled, the resulting unit cell parameters are printed.

```
scaling unit cell and vertex positions...


*******************************************
The scaled unit cell parameters are :
*******************************************
a    : 16.04626
b    : 16.04604
c    : 15.78555
alpha: 90.0002
beta : 90.00024
gamma: 119.99985
```

Next, ToBaCCo will form X-X bonds. Again, this can take some time depending on the size of the template. Once the bonds are formed a report about the bond formation process will be printed, showing the bond search tolerance (the maximum distance between connection sites considered in the bond search, this is set to the maximum edge length unless EXPANSIVE_BOND_SEARCH is set to True), the number of bonds formed, and whether the bond check was passed (ToBaCCo checks that the number of bonds formed is correct for each template). If the bond check is not passed that will be clearly indicated.

```
computing X-X bonds...
```

```
*********************************************
Bond formation :
*********************************************
distance search tolerance is 15.213 Angstroms
there were 12 X-X bonds formed
bond check passed
```

Finally, information about charges will be printed, indicating if the constructed MOF has a net charge (usually a small one), and the new net charge after rescaling the charges of the non-metal atoms by the indicated amount. Once this is complete, a cif will be written and the next MOF will be built, or ToBaCCo will terminate (as here).

```
*********************************************
Charge information :
*********************************************
old net charge                : 0.60271
new net charge (after rescaling): 0.0
rescaling magnitude           : 0.00548

writing cif...

Normal termination of Tobacco_3.0 after
--- 1.70928192139 seconds ---
```

9 CIF Naming Convention

The cif naming convention used by ToBaCCo is this:

topology_v1-cifname1_v2-cifname2_ … _ vN-cifnameN_edge1_edge2_ … _ edgeN.cif,

where topology is the RCSR topology code, cifnameN is the name of the building block assigned to vertex N (vN), and edgeN is the name of the Nth edge building block used. Every vertex building block used is listed, even if the same building block is used for multiple vertices, only the used edge building blocks are listed (so if the same edge is used for all three edge types in a certain template only the name of that edge will be listed). This is to keep the final cif name (relatively) short, while still making sure that unique cifs of the same composition are not overwritten during large scale production.

9 Known Issues

9.1 The "Accordion Template"

Some templates will "accordion" during scaling, meaning collapse in one dimension. This happens if one dimension can be collapsed while still maintaining good agreement between the template vertices and the nodular building blocks, while also not changing the edge lengths. In such cases, collapsing one unit cell dimension may results in a slightly lower value for the objective function. ToBaCCo can detect when this happens and will not write a cif should this unfortunate event transpire. The **lvtb** topology is particularly prone to accordion (using the typical configurations described in Section 7), should the user wish to observe this process in action. We recommend freezing the collapsing unit cell dimension using the FIX_UC flag (this dimension will be scaled statically as well as possible, and then frozen during the scaling minimization procedure), or using a larger value for SCALING_CONVERGENCE_TOLERANCE. Generally, the latter is recommended, since we hypothesize that the accordion behavior occurs when the collapse lowers the objective function by relatively small amounts (e.g. for the **lvtb** topology, the collapse can be prevented by using a convergence tolerance value on 1e-3, which is small enough to produce good results, but large enough to prevent the collapse).

9.2 When using large building blocks, the connection site bond distances are too short.

This can happen when using especially large linker or node building blocks. Usually this problem can be fixed by placing the building block in a larger box (a box length that is more than twice the maximum dimension of the building blocks should always be used). The problem is that certain COM to connection site distances will not be calculated correctly if the box is too small. We have found that placing building blocks in boxes ~3 times the maximum dimension of the building block is a safe choice.