MedChemica

CREATING A STEP CHANGE IN MEDICINAL CHEMISTRY

SARkush®: Automated Markush-like Structure Generation for SAR Communication

Dr Lauren Reid
lauren.reid@medchemica.com

Twitter @MedChemica

www.medchemica.com

Oct 2022

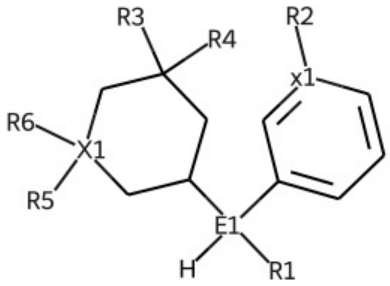
SARkush®

Structure Activity Relationship

Markush

MedChemica
CREATING A STEP CHANGE IN MEDICINAL CHEMISTRY

A structural representation designed to communicate SAR



Variable atoms/ groups:
- x = aromatic atom
- X = aliphatic ring atom
- E = linker atom
- R = side chain

compound id	X1	E1	R1	…	data
1	C	N	None	…	measurement1
2	O	C	[H]	…	measurement2
…	…	…	…	…	…

Importance of Markush Structures

MedChemica
CREATING A STEP CHANGE IN MEDICINAL CHEMISTRY

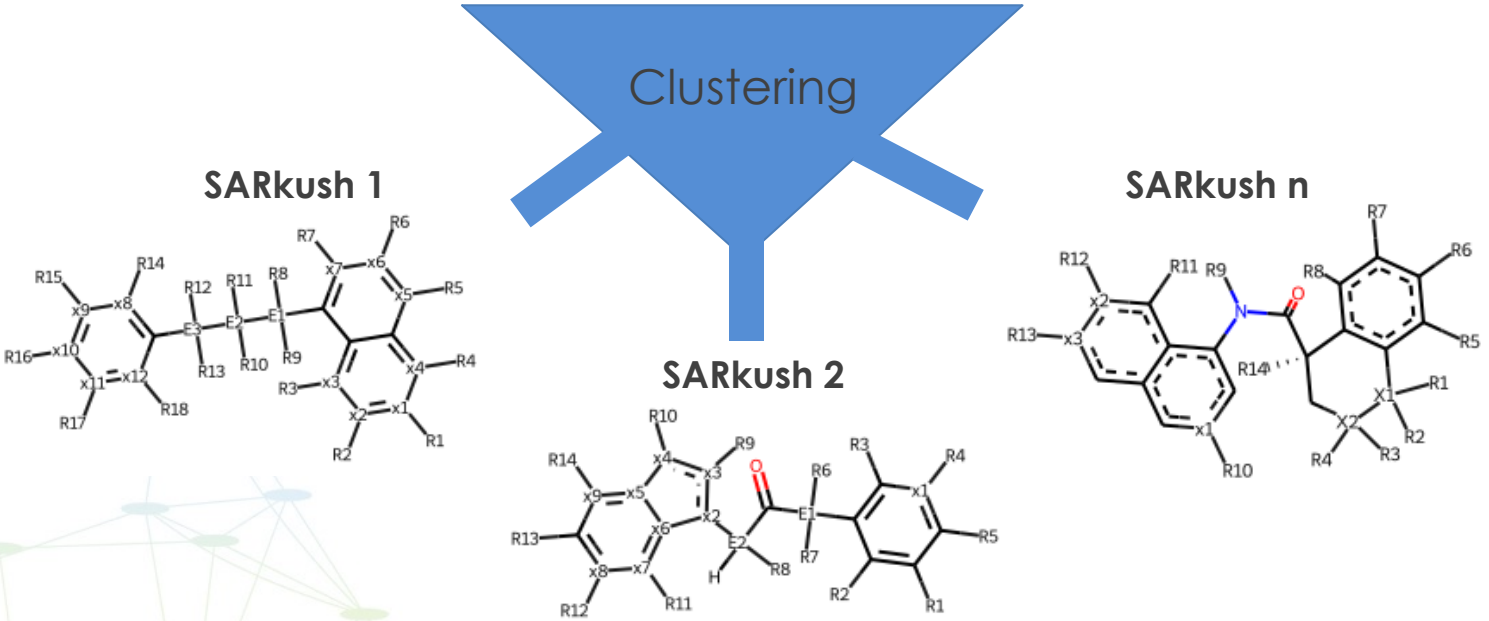
- SAR exploration is essential in drug discovery
- Projects can explore several chemical series and generate thousands of data points
- Markush structures and R group tables are an established and effective format for SAR communication
- Markush structures can provide input to Free-Wilson or QSAR approaches
- Challenges:
  - Manual production is time-consuming
  - R-group decomposition algorithms require the user to define the core
  - Chemical space explored in projects is ever-evolving

SARkush Tool Brief

MedChemica
CREATING A STEP CHANGE IN MEDICINAL CHEMISTRY

- **Automatically** cluster compounds into Markush-like (**SARkush**) structures
- Remove the need for users to **separate** compounds into **chemical series** and **cores**
- Generate SARkush structure **depictions** and compound **decomposition tables**

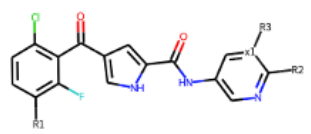
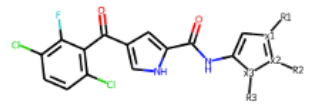
1000s of compounds in



A look at the output…

MedChemica
CREATING A STEP CHANGE IN MEDICINAL CHEMISTRY

- One excel spreadsheet with SARkush structural depictions:

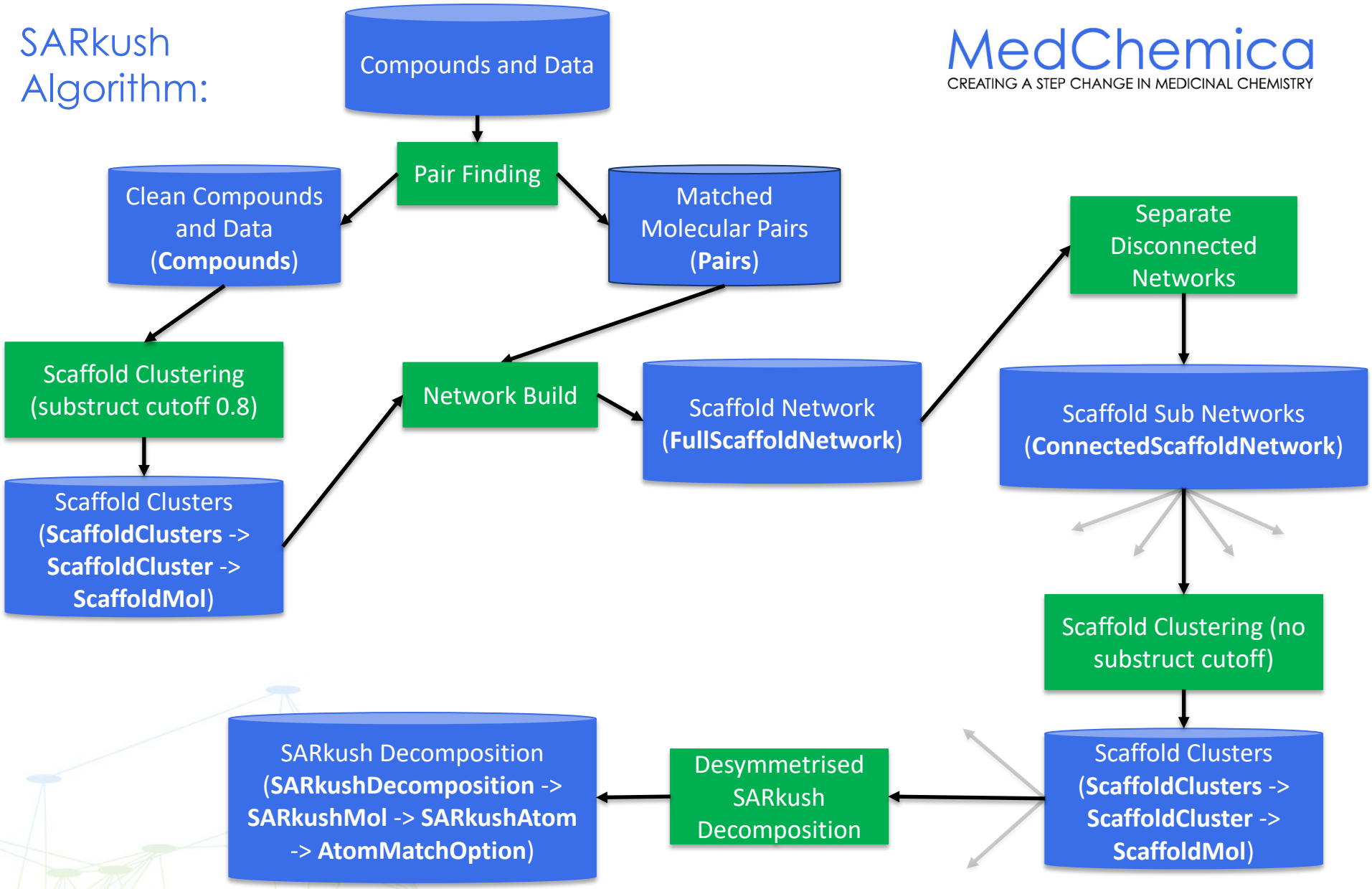
	A	B	C	D	E	F	G	H	I	J	K
1	network_id	sarkush_id	sarkush	no_of_compounds	no_of_compounds_cumulative	percentage_of_compounds	measurement_min	measurement_max	measurement_median	measurement_mean	measurement_mean_std_dev
2	1	1		17	17	65.38461538	5.455931956	8.301029996	7.853871964	7.310996944	0.965096097
3	1	2		9	26	34.61538462	5.908333042	8.15490196	7.102372909	7.057882674	0.801157503

- A txt file per SARkush structure with compound decomposition:

	A	C	D	E	F	G	H	I	J	K	L	M	N
1	compound_name	SARkush_smiles	x1	R1	R2	R3	qualifier	measurement	CLogP	PSA	RMM	HBA	HBD
2	33j	O=C(Nc1cnc([R2])[x1]([R3])c1)c1cc(C(=O)c2c(Cl)ccc([R1])c2F)c[nH]1	c	[R1]Cl	[R2]C([H])([H])C1([H])C([H])([H])C([H])([H])N(C([H])([H])[H])C([H])([H])C1([H])[H]	[R3][H]	=	8.22184875	4.52	78.09	489.37	6	2
3	33i	O=C(Nc1cnc([R2])[x1]([R3])c1)c1cc(C(=O)c2c(Cl)ccc([R1])c2F)c[nH]1	c	[R1]Cl	[R2]C([H])([H])N1C([H])([H])C([H])([H])N(C([H])([H])[H])C([H])([H])C1([H])[H]	[R3][H]	=	8.301029996	3.09	81.33	490.36	7	2
4	33h	O=C(Nc1cnc([R2])[x1]([R3])c1)c1cc(C(=O)c2c(Cl)ccc([R1])c2F)c[nH]1	c	[R1]Cl	[R2]C([H])([H])N1C([H])([H])C([H])([H])N([H])C([H])([H])C1([H])[H]	[R3][H]	=	8.301029996	2.71	90.12	476.33	7	3
5	33g	O=C(Nc1cnc([R2])[x1]([R3])c1)c1cc(C(=O)c2c(Cl)ccc([R1])c2F)c[nH]1	c	[R1]Cl	[R2]N(C([H])([H])[H])C1([H])C([H])([H])C([H])([H])N(C([H])([H])[H])C([H])([H])C1([H])[H]	[R3][H]	=	7.602059991	4.6	81.33	504.38	7	2
6	33f	O=C(Nc1cnc([R2])[x1]([R3])c1)c1cc(C(=O)c2c(Cl)ccc([R1])c2F)c[nH]1	c	[R1]Cl	[R2]N1C([H])([H])C([H])([H])N([H])C([H])([H])C1([H])[H]	[R3][H]	=	7.886056648	3.09	90.12	462.3	7	3
7	33k	O=C(Nc1cnc([R2])[x1]([R3])c1)c1cc(C(=O)c2c(Cl)ccc([R1])c2F)c[nH]1	c	[R1]Cl	[R2]OC1([H])C([H])([H])C([H])([H])N(C([H])([H])[H])C([H])([H])C1([H])[H]	[R3][H]	=	7.853871964	4.34	87.32	491.34	7	2
8	32m	O=C(Nc1cnc([R2])[x1]([R3])c1)c1cc(C(=O)c2c(Cl)ccc([R1])c2F)c[nH]1	n	[R1]Cl	[R2]C([H])([H])N1C([H])([H])C([H])([H])N(C([H])([H])[H])C([H])([H])C1([H])[H]		=	8.15490196	2.48	94.22	491.35	8	2
9	32l	O=C(Nc1cnc([R2])[x1]([R3])c1)c1cc(C(=O)c2c(Cl)ccc([R1])c2F)c[nH]1	n	[R1]Cl	[R2]C([H])([H])N1C([H])([H])C([H])([H])N([H])C([H])([H])C1([H])[H]		=	8.096910013	2.09	103.01	477.32	8	3
10	32i	O=C(Nc1cnc([R2])[x1]([R3])c1)c1cc(C(=O)c2c(Cl)ccc([R1])c2F)c[nH]1	n	[R1]Cl	[R2]N([H])C1([H])C([H])([H])C([H])([H])N(C([H])([H])[H])C([H])([H])C1([H])[H]		=	7.853871964	3.35	103.01	491.35	8	3
11	32g	O=C(Nc1cnc([R2])[x1]([R3])c1)c1cc(C(=O)c2c(Cl)ccc([R1])c2F)c[nH]1	n	[R1]Cl	[R2]N1C([H])([H])C([H])([H])N(C([H])([H])[H])C([H])([H])C1([H])[H]		=	7.431798276	2.61	94.22	477.32	8	2
12	32k	O=C(Nc1cnc([R2])[x1]([R3])c1)c1cc(C(=O)c2c(Cl)ccc([R1])c2F)c[nH]1	n	[R1]Cl	[R2]N1C([H])([H])C([H])([H])N([H])C([H])([H])C1([H])[H]		=	7.886056648	2.22	103.01	463.29	8	3
13	32b	O=C(Nc1cnc([R2])[x1]([R3])c1)c1cc(C(=O)c2c(Cl)ccc([R1])c2F)c[nH]1	n	[R1][H]	[R2]C([H])([H])[H]		=	5.920818754	2.56	87.74	358.75	6	2
14	32d	O=C(Nc1cnc([R2])[x1]([R3])c1)c1cc(C(=O)c2c(Cl)ccc([R1])c2F)c[nH]1	n	[R1][H]	[R2]N(C([H])([H])[H])C([H])([H])[H]		=	5.455931956	2.56	90.98	387.8	7	2
15	32f	O=C(Nc1cnc([R2])[x1]([R3])c1)c1cc(C(=O)c2c(Cl)ccc([R1])c2F)c[nH]1	n	[R1][H]	[R2]N1C([H])([H])C([H])([H])N(C([H])([H])[H])C([H])([H])C1([H])[H]		=	7.142667504	1.99	94.22	442.87	8	2
16	32e	O=C(Nc1cnc([R2])[x1]([R3])c1)c1cc(C(=O)c2c(Cl)ccc([R1])c2F)c[nH]1	n	[R1][H]	[R2]N1C([H])([H])C([H])([H])OC([H])([H])C1([H])[H]		=	5.677780705	1.74	100.21	429.83	8	2
17	32c	O=C(Nc1cnc([R2])[x1]([R3])c1)c1cc(C(=O)c2c(Cl)ccc([R1])c2F)c[nH]1	n	[R1][H]	[R2]OC([H])([H])[H]		=	6.102372909	2.31	96.97	374.75	7	2
18	32a	O=C(Nc1cnc([R2])[x1]([R3])c1)c1cc(C(=O)c2c(Cl)ccc([R1])c2F)c[nH]1	n	[R1][H]	[R2][H]		=	6.397940009	2	87.74	344.73	6	2

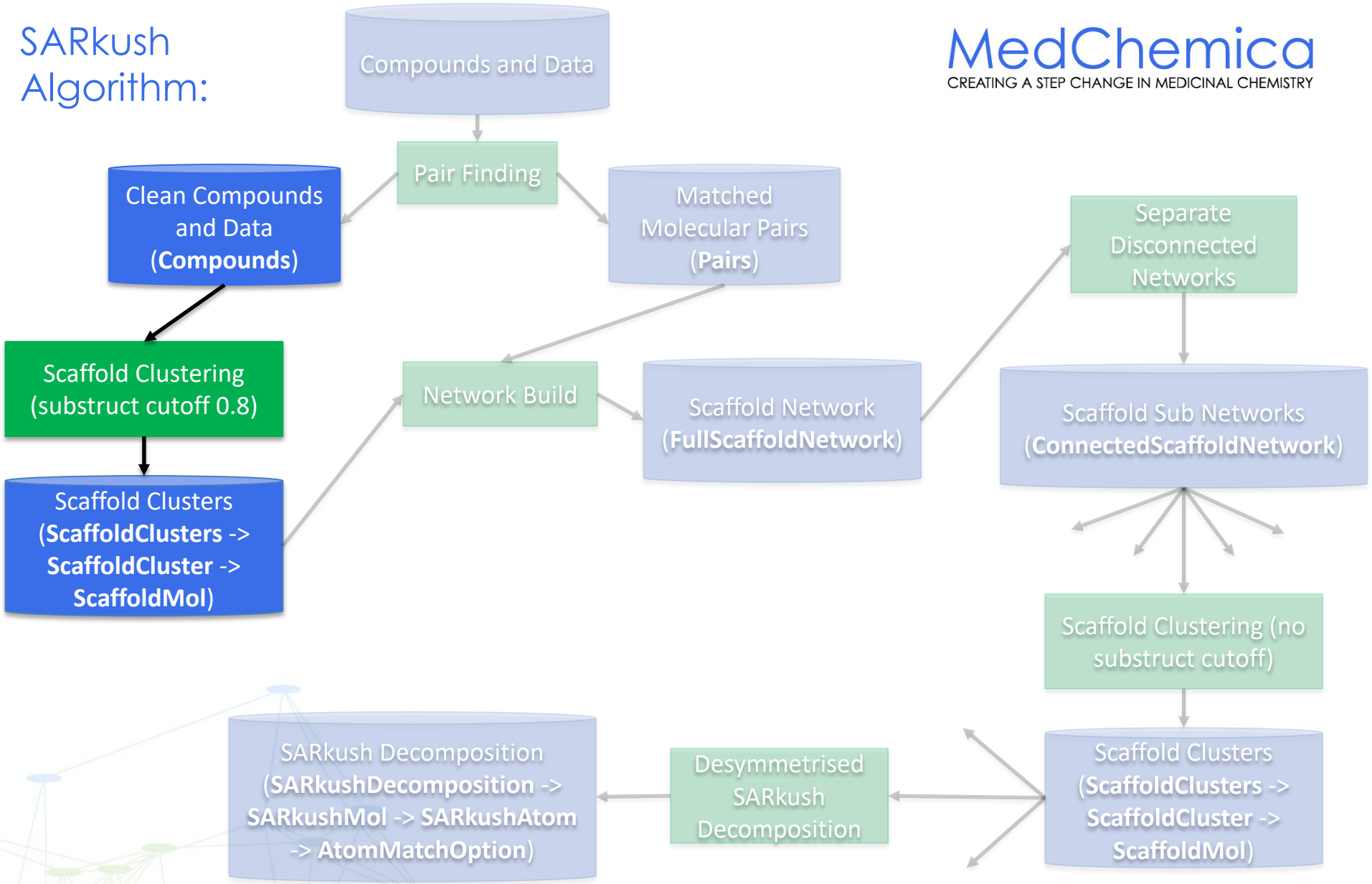
SARkush Algorithm:

MedChemica
CREATING A STEP CHANGE IN MEDICINAL CHEMISTRY



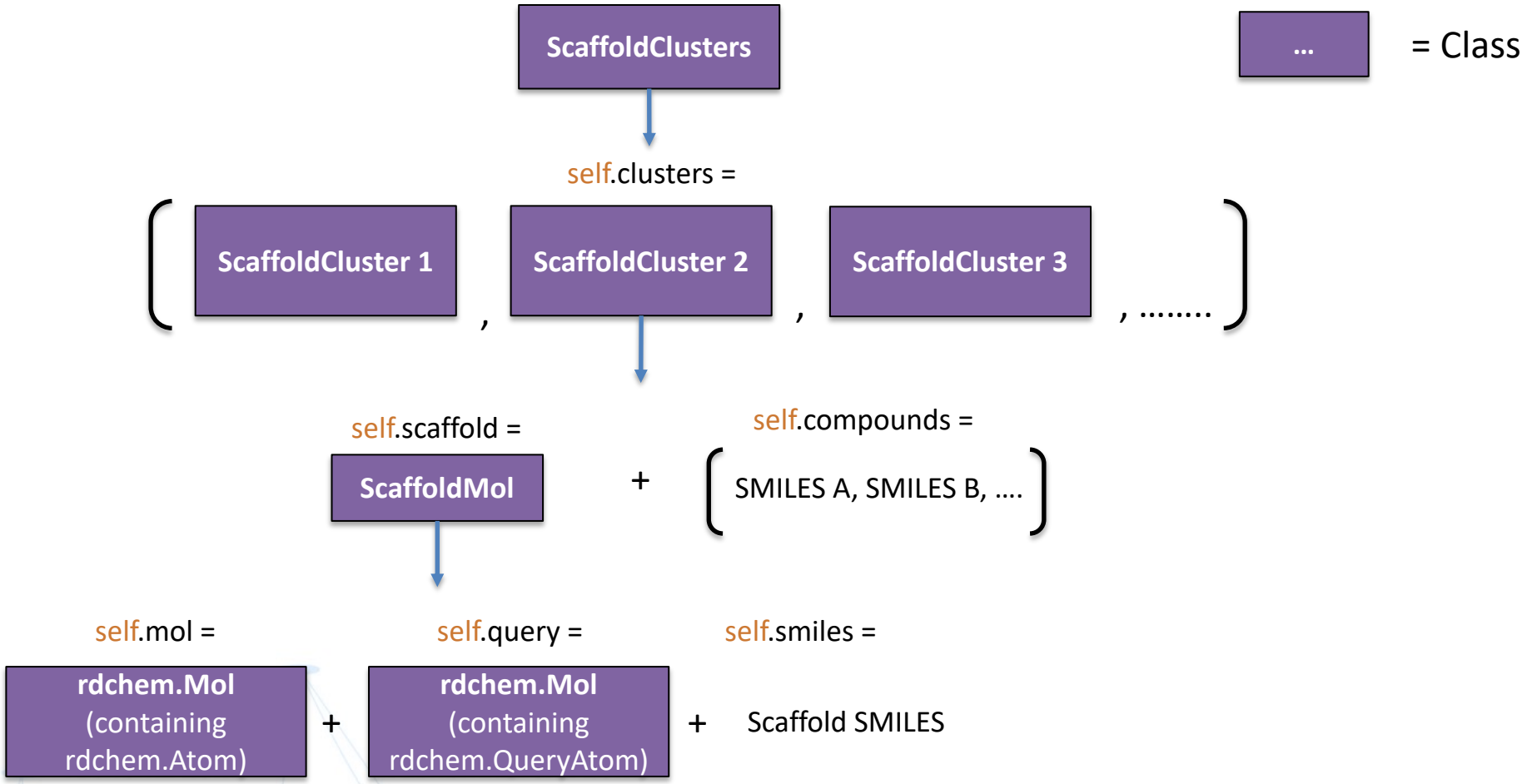
SARkush Algorithm:

MedChemica
CREATING A STEP CHANGE IN MEDICINAL CHEMISTRY



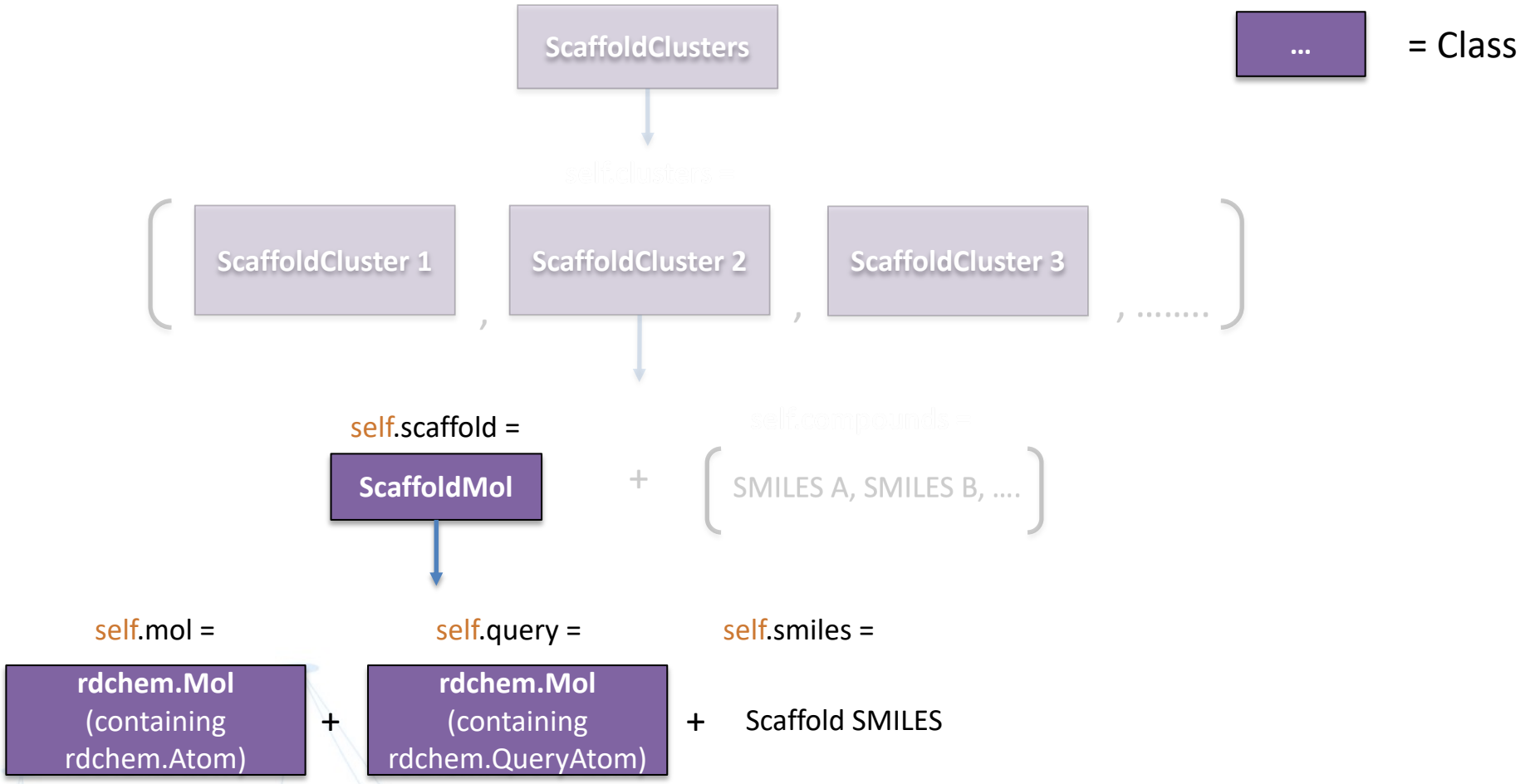
Classes used in Scaffold Clustering

MedChemica
CREATING A STEP CHANGE IN MEDICINAL CHEMISTRY



Classes used in Scaffold Clustering

MedChemica
CREATING A STEP CHANGE IN MEDICINAL CHEMISTRY



Scaffold SMILES

MedChemica
CREATING A STEP CHANGE IN MEDICINAL CHEMISTRY

- Generic * atoms with isotope labels

```
self._isotope_labels = {
                        "aliphatic_chiral":      1,
                        "carbonyl_heteroatom":   2,
                        "aliphatic_racemic":     3,
                        "E_bond_atom":           4,
                        "Z_bond_atom":           5,
                        "aromatic":              6 ,}
```

- E.g

[2*]=[3*]([3*][6*]1:[6*]:[6*]:[6*]:[6*]2:[6*]:[6*]:[6*]:[6*]:[6*]:1:2)[3*]1[3*][3*][3*][6*]2:[6*]:[6*]:[6*]:[6*]:[6*]:21

- Contains aromatic/ aliphatic and chiral/ racemic information
- Canonicalisation means identical scaffolds can be found with string matching

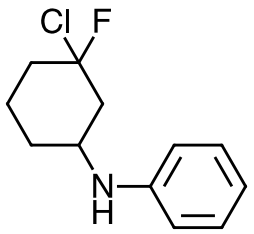
Generic Atom Scaffolds

MedChemica
CREATING A STEP CHANGE IN MEDICINAL CHEMISTRY

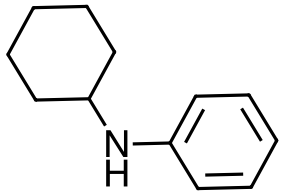
ScaffoldMol .smiles_to_scaffold_mol(smiles)

c1ccc2c(c1)cncc2NC(=O)C3CCOc4c3cc(cc4)Cl

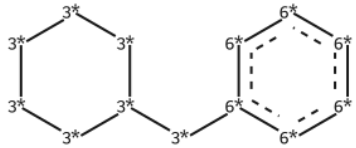
self._create_editable_mol()



self._cut_off_ring_and_branch_substituents()



self._set_atoms_to_generic()



self.mol =

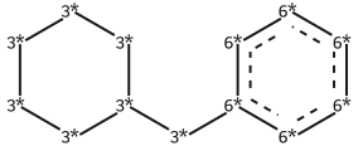
self.get_scaffold_smiles()

self.smiles =

[3*]1[3*][3*][3*]([3*][6*]2:[6*]:[6*]:[6*]:[6*]:[6*]:2)[3*][3*]1

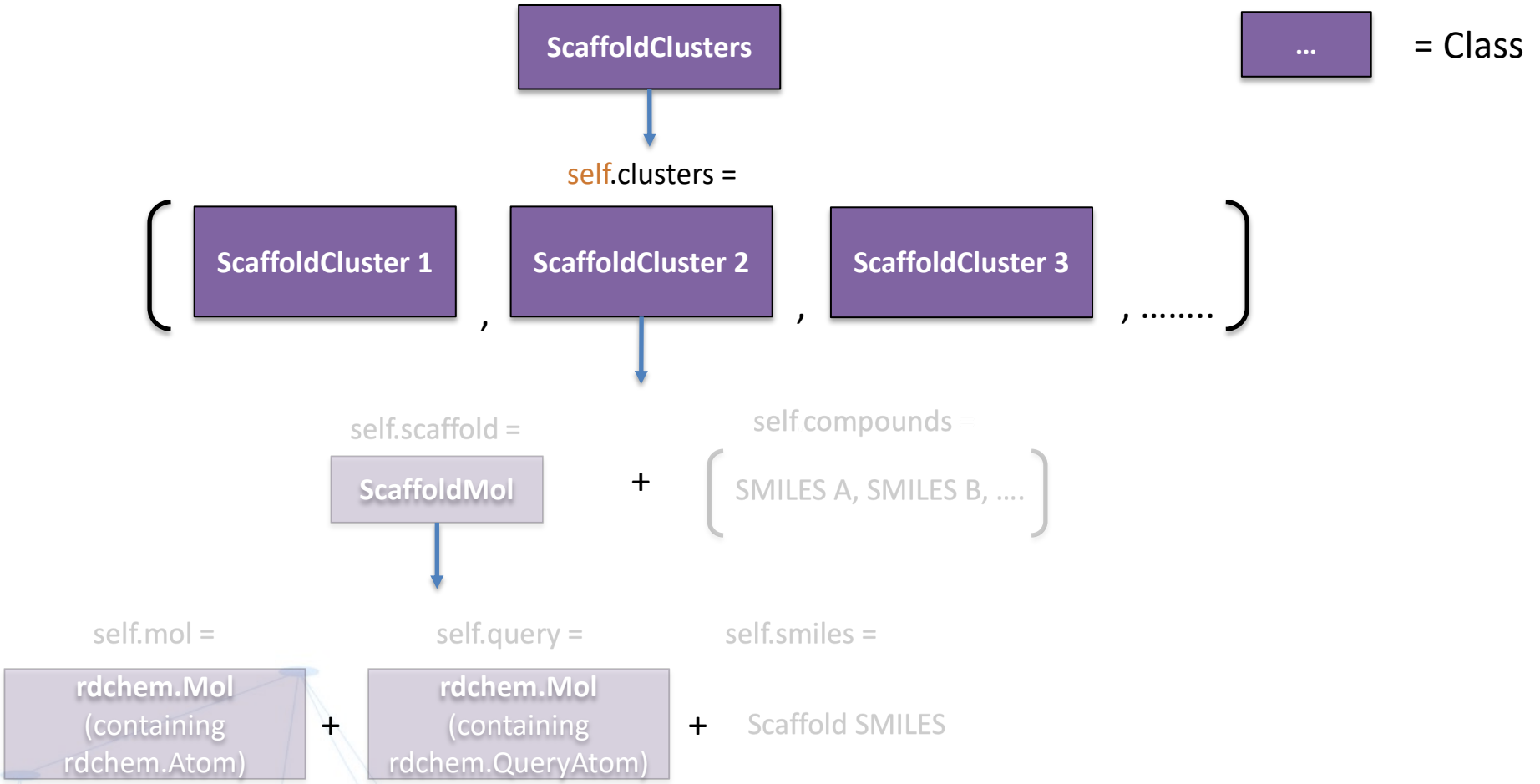
self._create_scaffold_query()

self.query =



Classes used in Scaffold Clustering

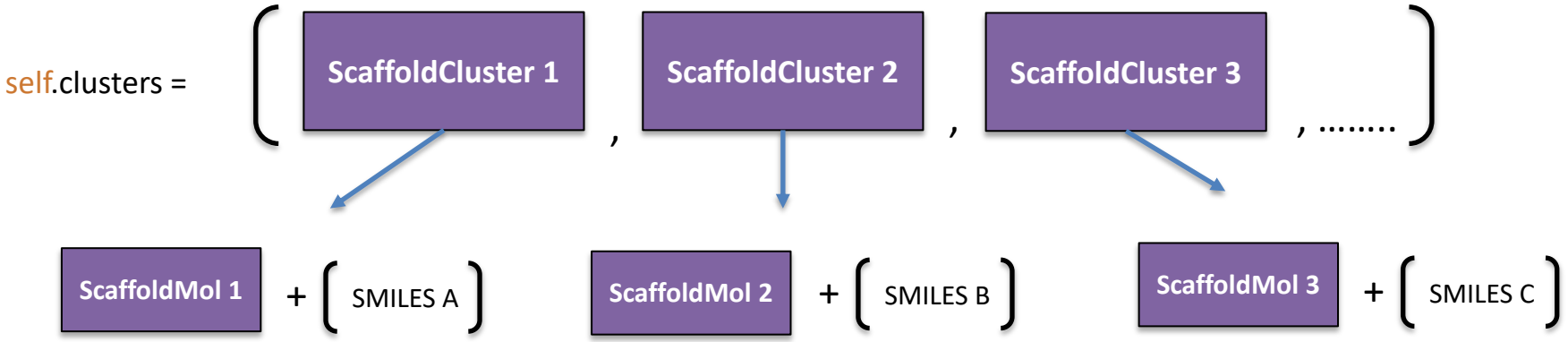
MedChemica
CREATING A STEP CHANGE IN MEDICINAL CHEMISTRY



Generic Atom Scaffold Clustering

MedChemica
CREATING A STEP CHANGE IN MEDICINAL CHEMISTRY

ScaffoldClusters .seed_single_compound_clusters(["SMILES A", "SMILES B", "SMILES C"])



Generic Atom Scaffold Clustering

MedChemica
CREATING A STEP CHANGE IN MEDICINAL CHEMISTRY

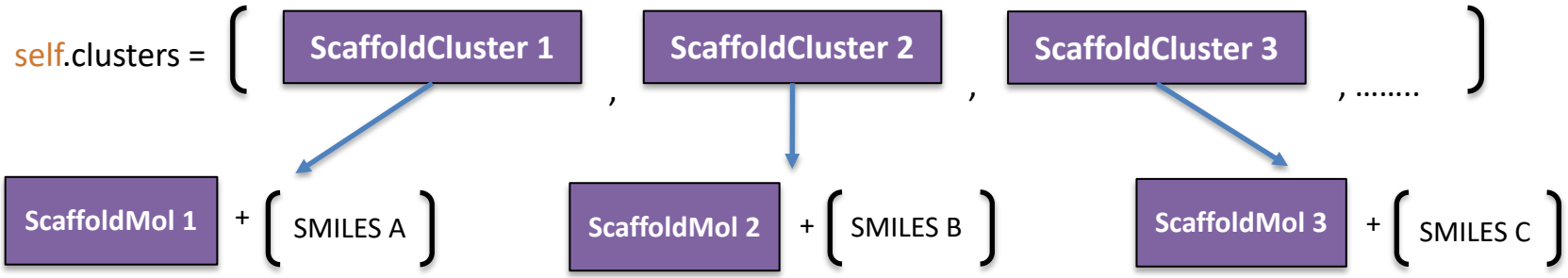
ScaffoldClusters

```
.scaffold_clustering(self, substruct_match_cutoff=0.8):

    self._merge_clusters_on_string_match()

    self._merge_clusters_on_substructure_match(cutoff=substruct_match_cutoff)

    self._order_clusters_by_num_compounds()
```

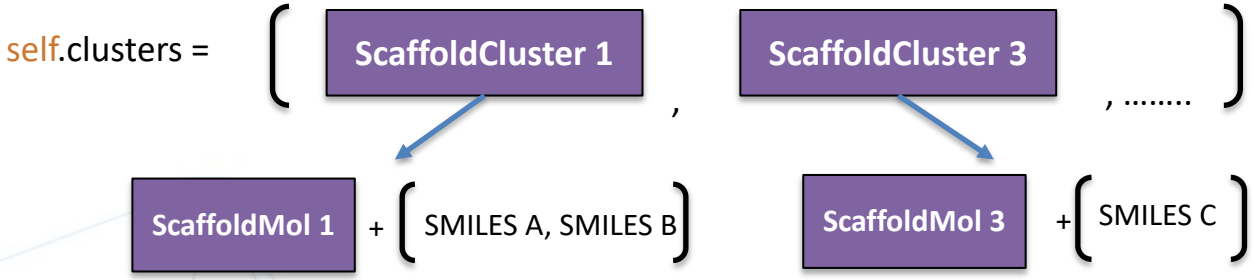
Generic Atom Scaffold Clustering

MedChemica
CREATING A STEP CHANGE IN MEDICINAL CHEMISTRY

ScaffoldClusters._merge_clusters_on_string_match()



TRANSFORMS TO



IF:

ScaffoldMol 1.get_scaffold_smiles() == **ScaffoldMol 2**.get_scaffold_smiles() != **ScaffoldMol 3**.get_scaffold_smiles()

Generic Atom Scaffold Clustering

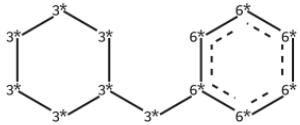
MedChemica
CREATING A STEP CHANGE IN MEDICINAL CHEMISTRY

ScaffoldClusters._merge_clusters_on_substructure_match(cutoff=0.8)

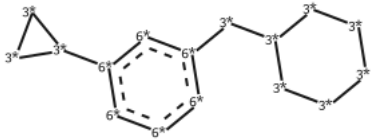
self.clusters = (**ScaffoldCluster 1** , **ScaffoldCluster 3** , ...)

ScaffoldCluster 1 → **ScaffoldMol 1** + (SMILES A, SMILES B)

ScaffoldCluster 3 → **ScaffoldMol 3** + (SMILES C)



TRANSFORMS TO



self.clusters = (**ScaffoldCluster 1**)

ScaffoldCluster 1 → **ScaffoldMol 1** + (SMILES A, SMILES B, SMILES C)

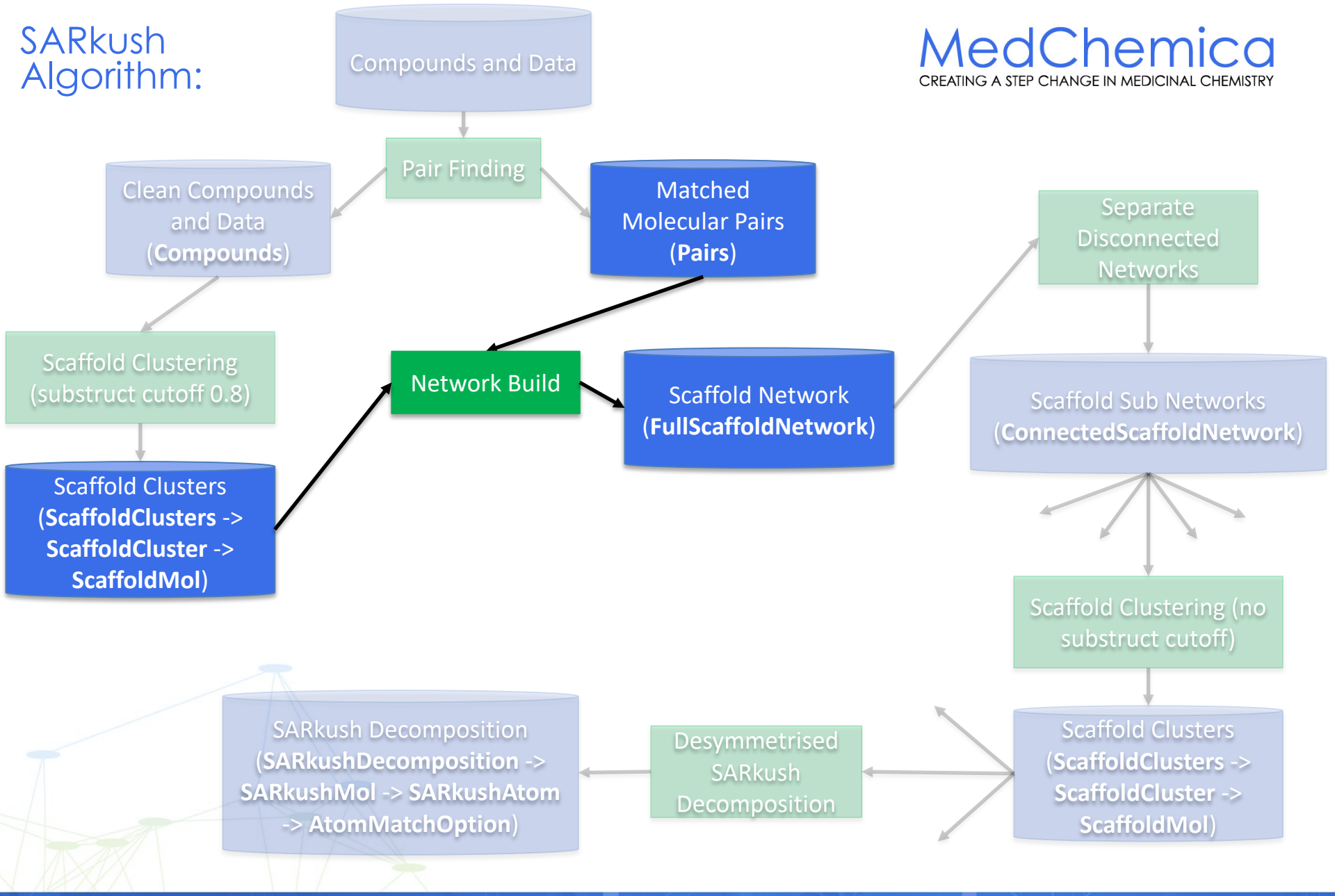
IF

ScaffoldMol 3.substructure_match_with_other_scaffold(**ScaffoldMol 1**) **AND**

ScaffoldMol 1.num_atoms() >= **ScaffoldMol 3**.num_atoms() * cutoff

SARkush Algorithm:

MedChemica
CREATING A STEP CHANGE IN MEDICINAL CHEMISTRY



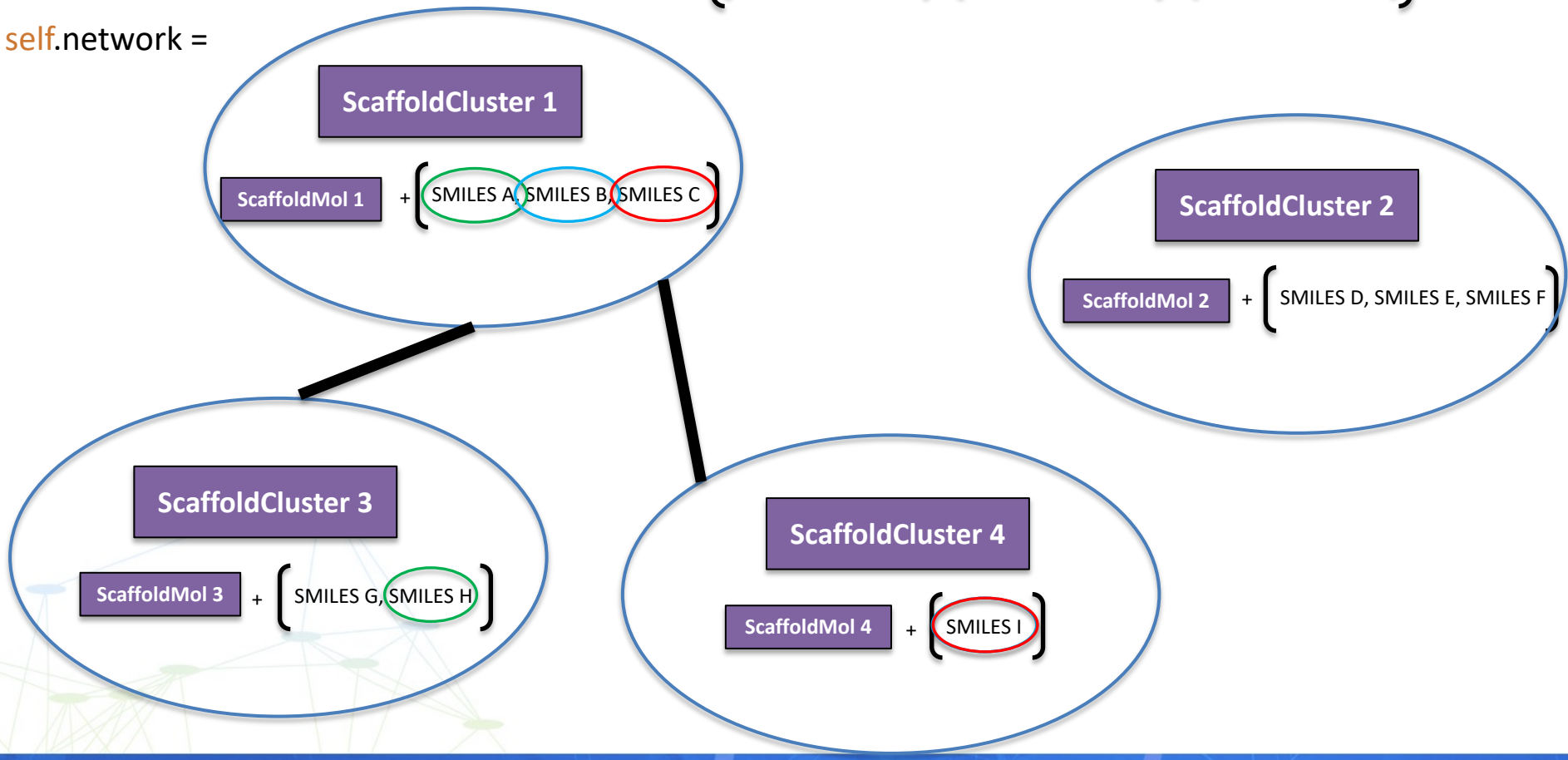
Scaffold Network

MedChemica
CREATING A STEP CHANGE IN MEDICINAL CHEMISTRY

FullScaffoldNetwork .build_full_network(**ScaffoldClusters** , **Pairs**)

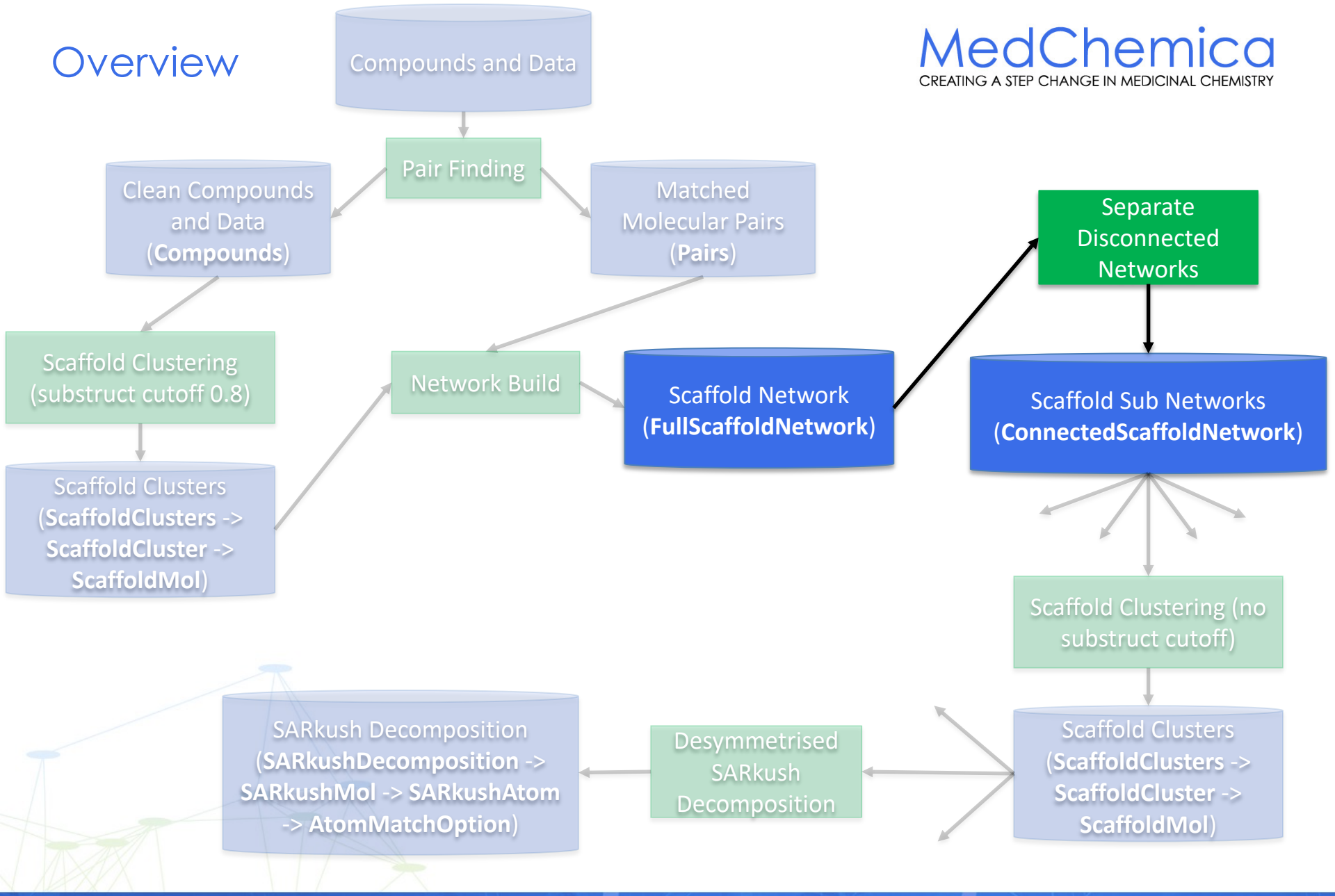
{ { SMILES A, SMILES H }, { SMILES B, SMILES I }, { SMILES C, SMILES I } }

self.network =



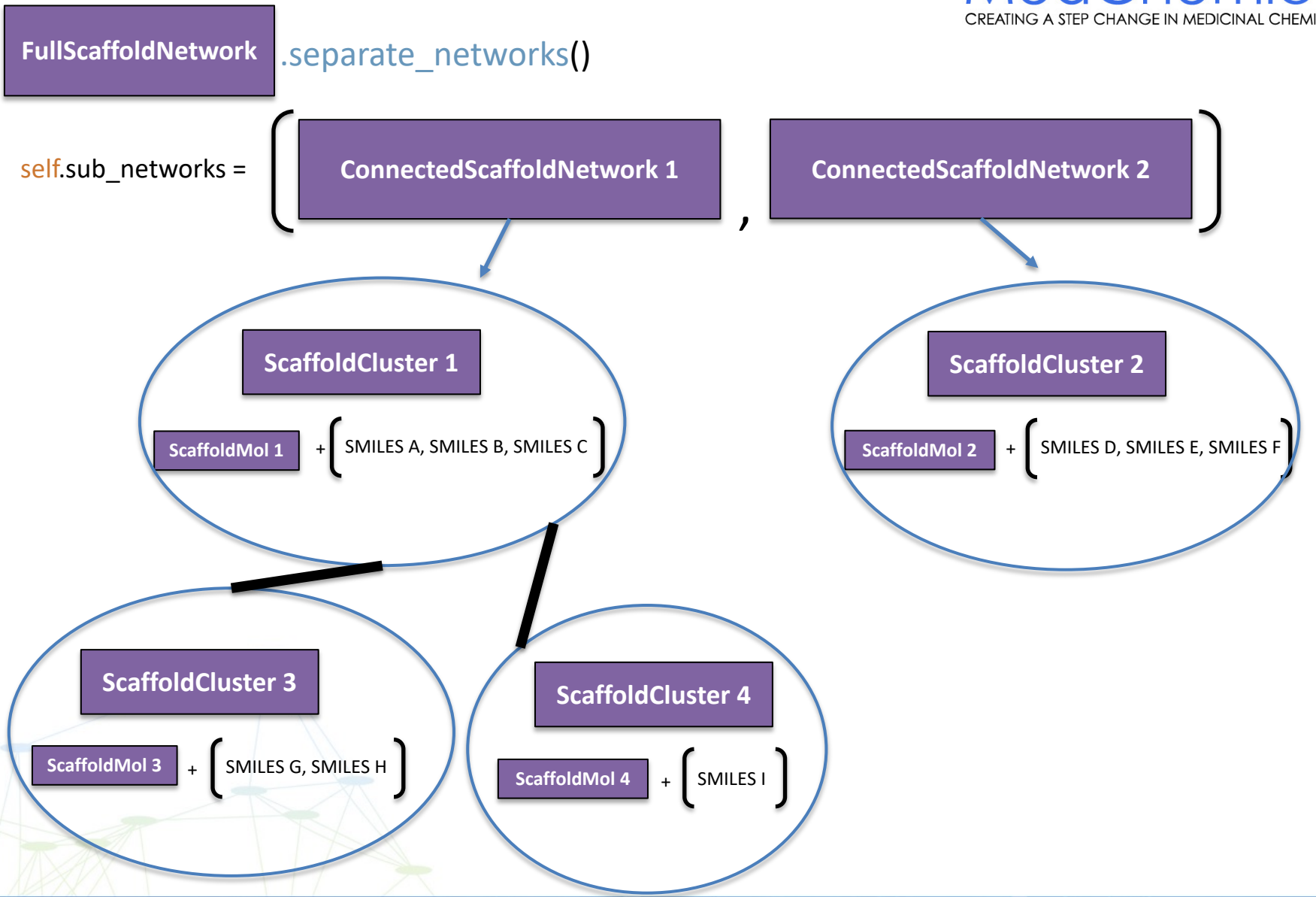
Overview

MedChemica
CREATING A STEP CHANGE IN MEDICINAL CHEMISTRY



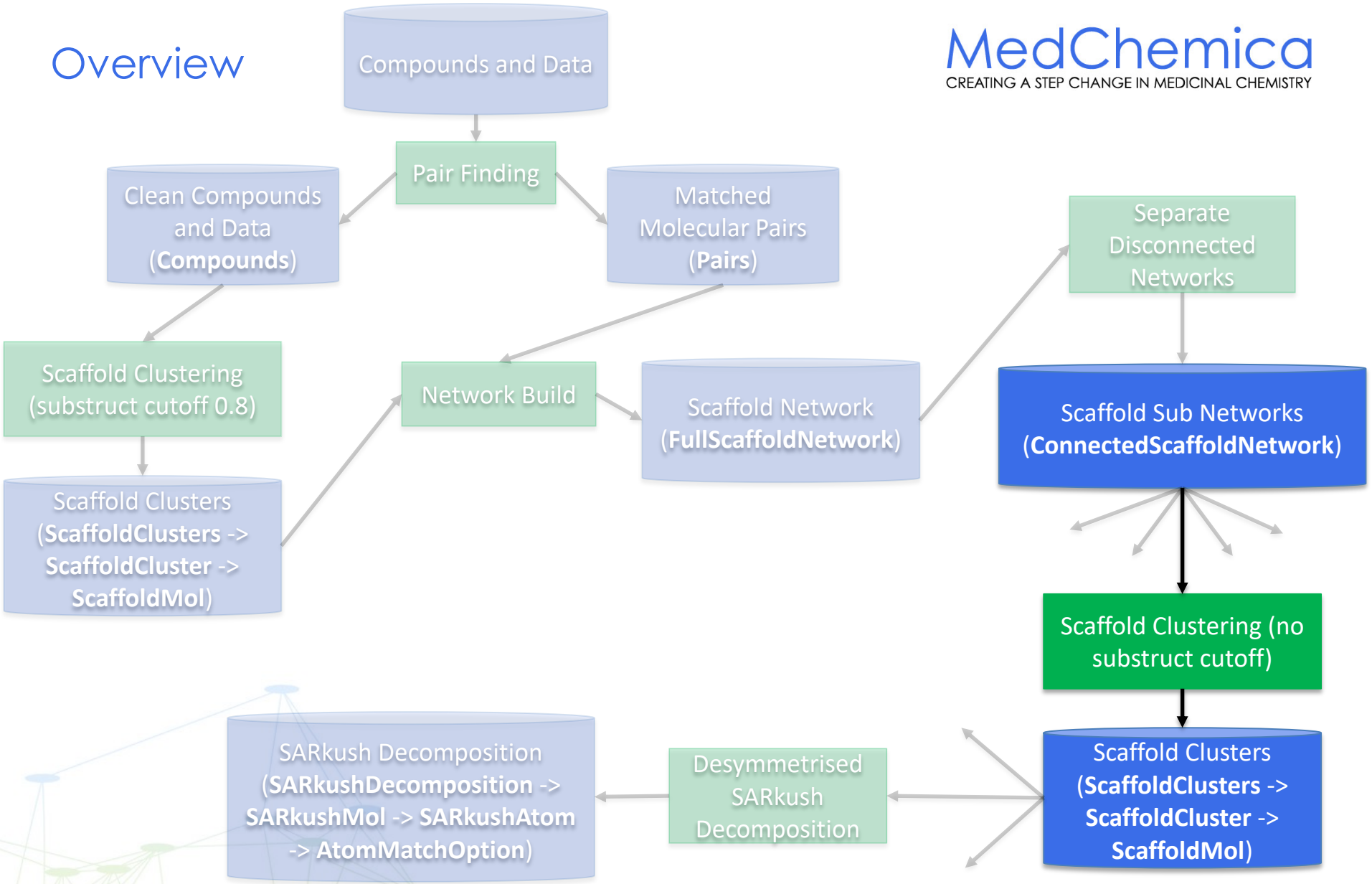
Scaffold Network

MedChemica
CREATING A STEP CHANGE IN MEDICINAL CHEMISTRY



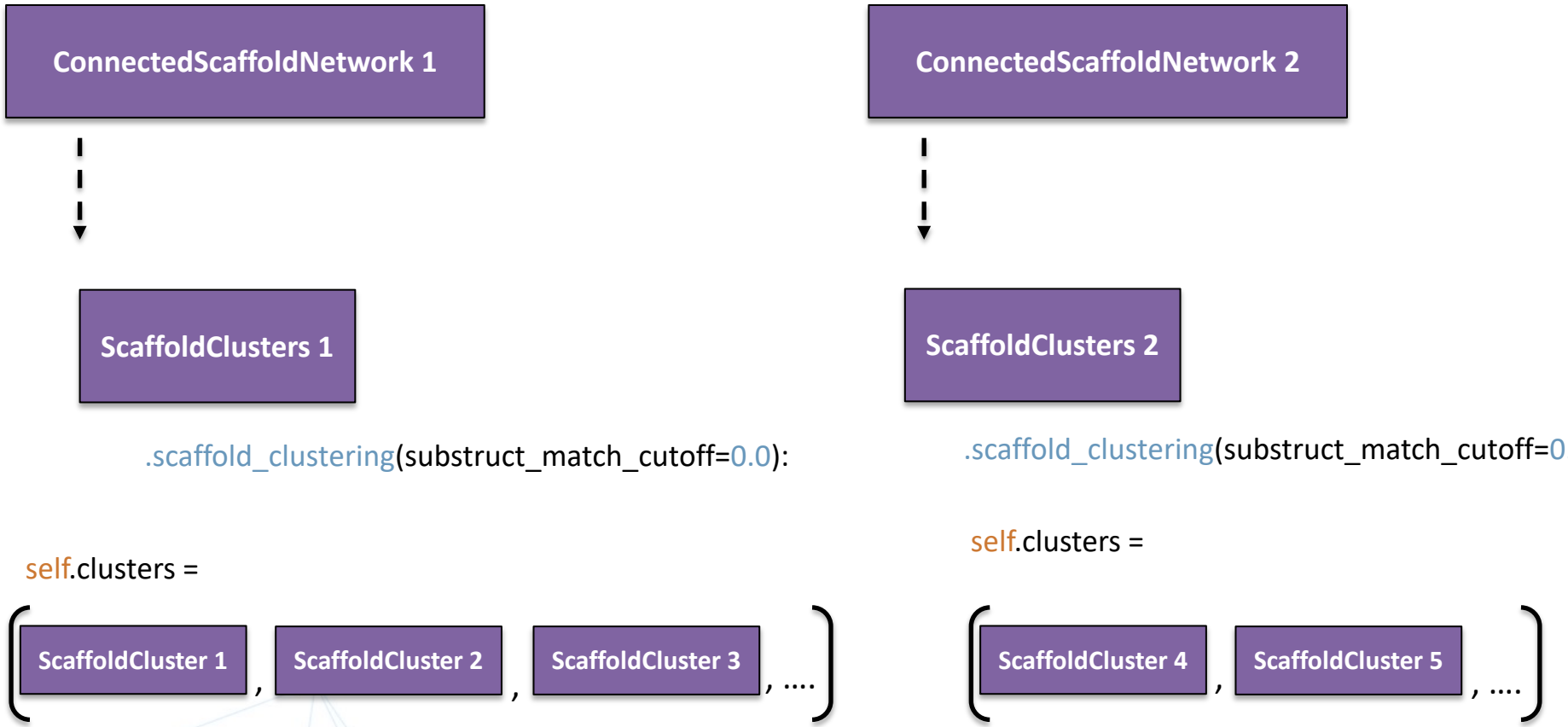
Overview

MedChemica
CREATING A STEP CHANGE IN MEDICINAL CHEMISTRY



Generic Atom Scaffold Clustering

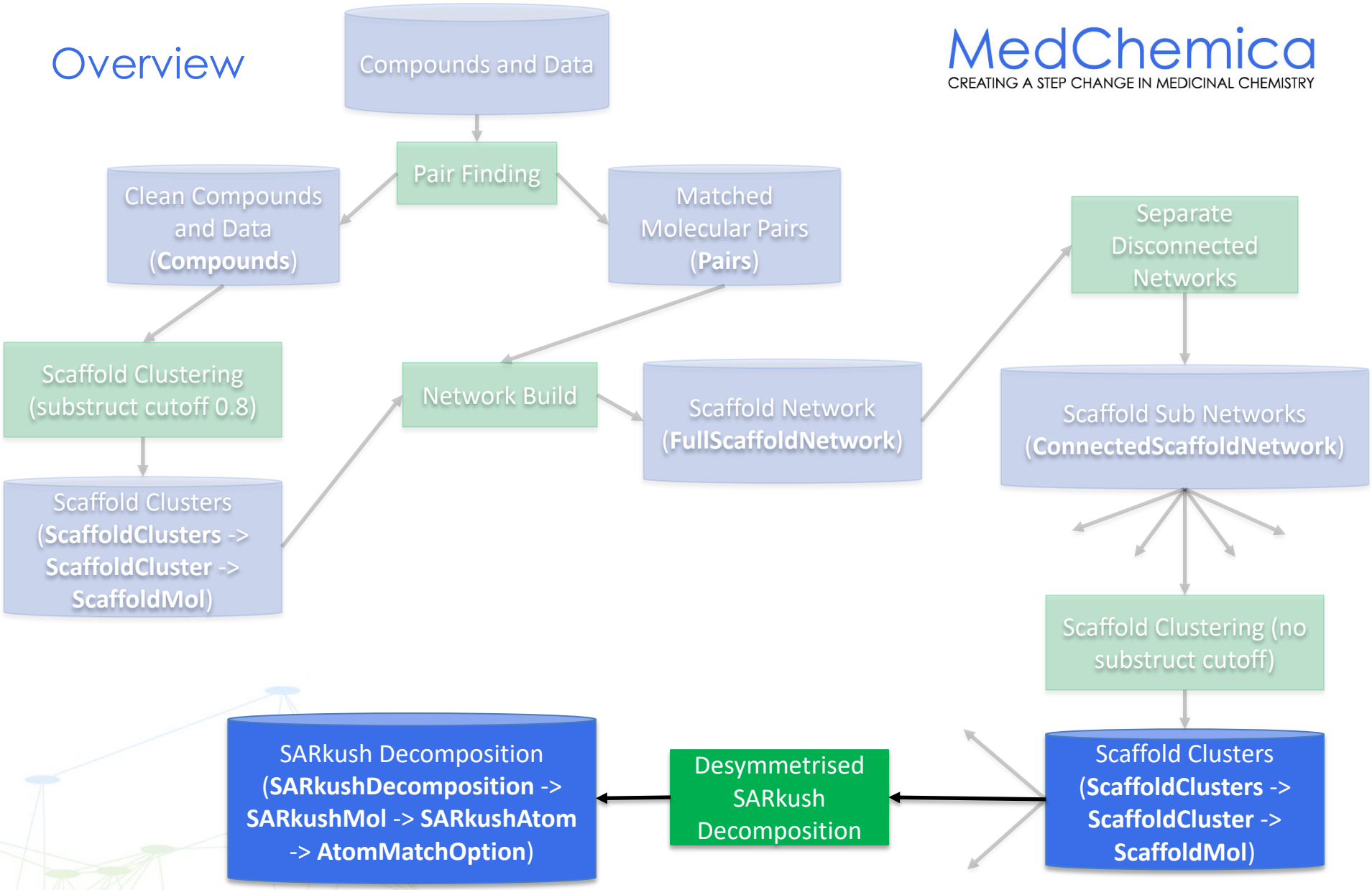
MedChemica
CREATING A STEP CHANGE IN MEDICINAL CHEMISTRY



- A ScaffoldClusters object is created from each ConnectedScaffoldNetwork, and scaffold clustering is performed with no substructure match cutoff
- We're left with multiple ScaffoldCluster objects, originating from one of multiple ConnectedScaffoldNetworks

Overview

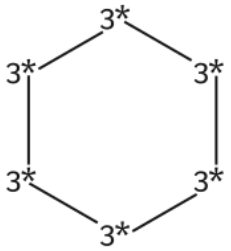
MedChemica
CREATING A STEP CHANGE IN MEDICINAL CHEMISTRY



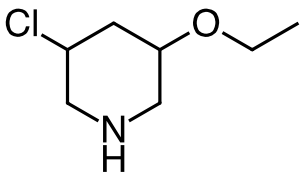
What do we mean by "desymmetrised decomposition"?

MedChemica
CREATING A STEP CHANGE IN MEDICINAL CHEMISTRY

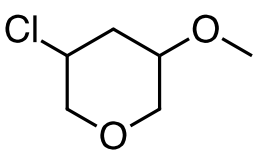
Scaffold



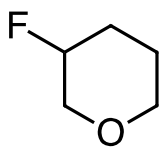
Compounds



1

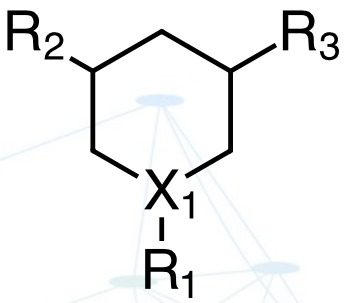


2



3

SARkush Decomposition:

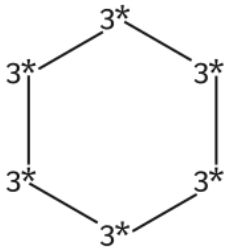


compound	X1	R1	R2	R3
1	N	[R1][H]	[R2]Cl	[R3]OCC
2	O	None	[R2]Cl	[R3]OC
3	O	None	[R2]F	[R3][H]

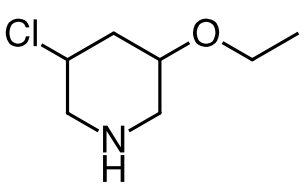
What do we mean by "desymmetrised decomposition"?

MedChemica
CREATING A STEP CHANGE IN MEDICINAL CHEMISTRY

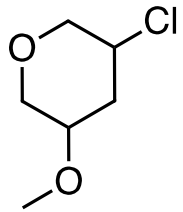
Scaffold



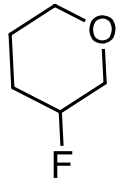
Compounds



1

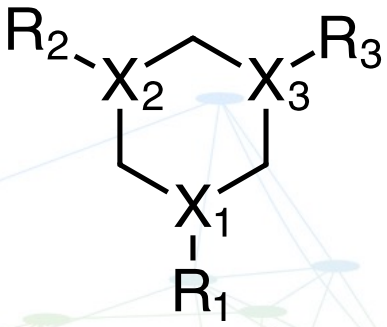


2



3

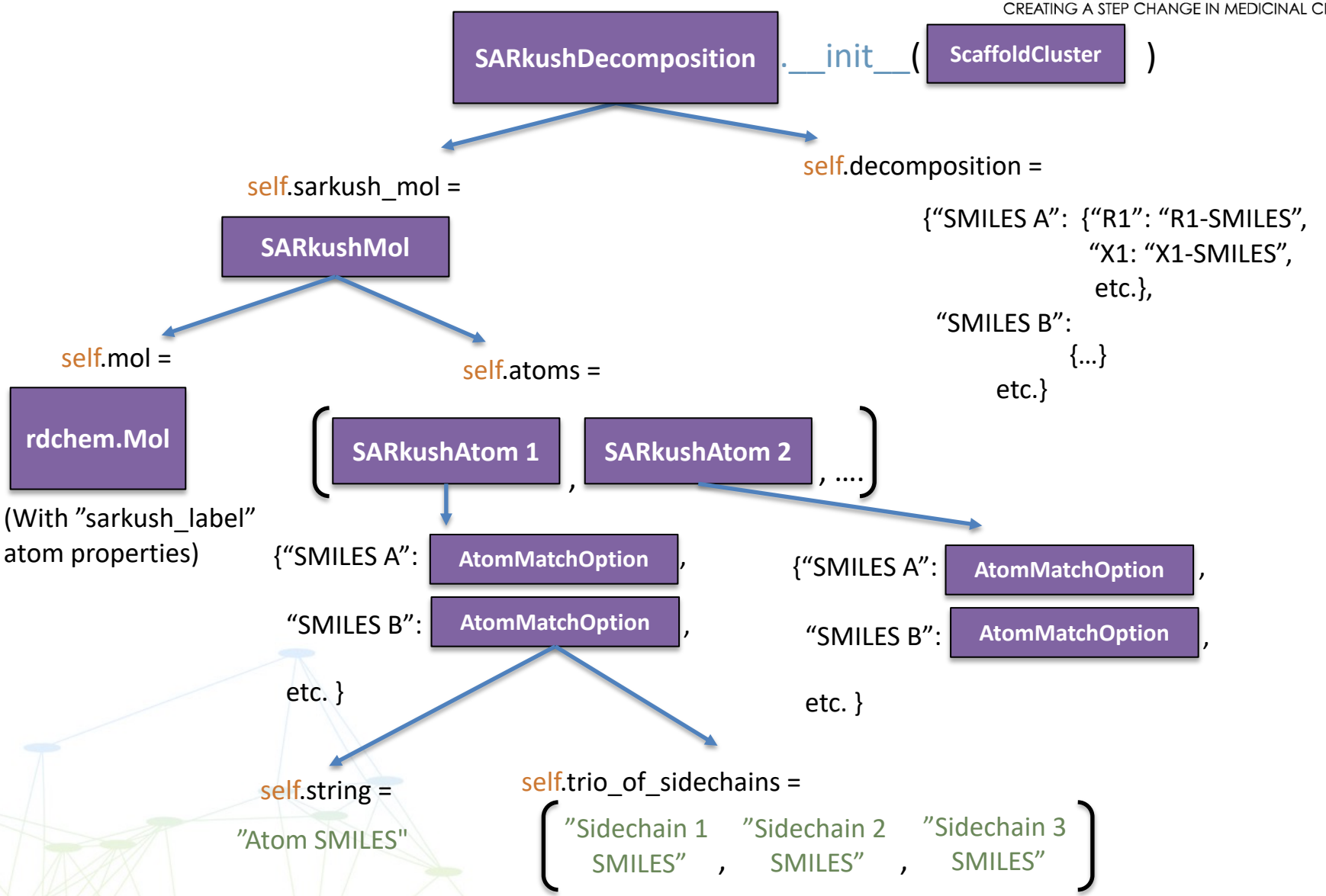
SARkush Decomposition:



compound	X1	X2	X3	R1	R2	R3
1	N	C	C	[R1]H	[R2]Cl	[R3]OCC
2	C	O	C	[R1]OC	None	[R3]Cl
3	C	C	O	[R1]F	[R2][H]	None

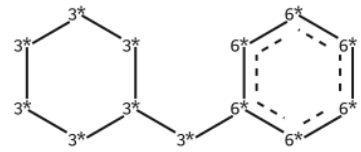
Classes used in SARkush Decomposition

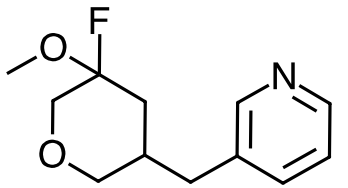
MedChemica
CREATING A STEP CHANGE IN MEDICINAL CHEMISTRY

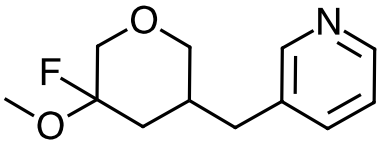


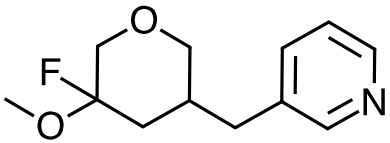
AtomMatchOption

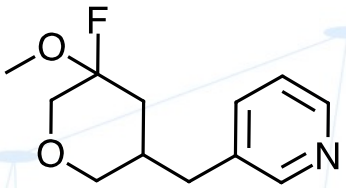
MedChemica
CREATING A STEP CHANGE IN MEDICINAL CHEMISTRY











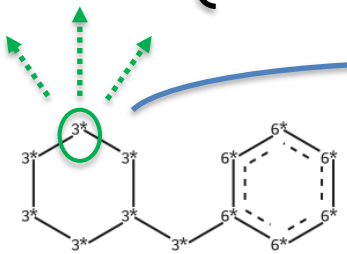
AtomMatchOption

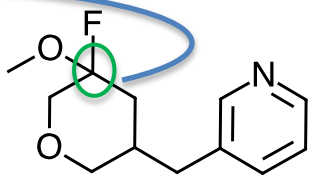
self.string =

"C"

self.trio_of_sidechains =

("*OC", "*F", None)





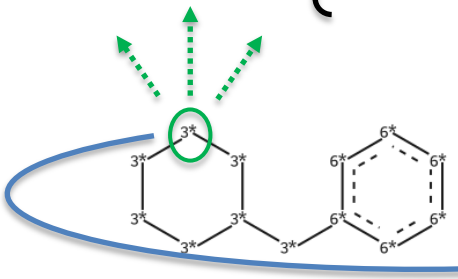
AtomMatchOption

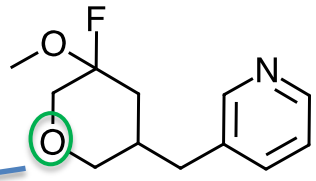
self.string =

"O"

self.trio_of_sidechains =

(None, None, None)





SARkush Decomposition

MedChemica
CREATING A STEP CHANGE IN MEDICINAL CHEMISTRY

SARkushDecomposition.__init__(**ScaffoldCluster** , report_groups_implicitly=False)

self._perform_decomposition(**ScaffoldCluster** .compounds, **ScaffoldCluster** .scaffold)

self._desymmetrised_atom_assignment(compounds, **ScaffoldMol**)

self._atom_assignments_to_sarkush_decomposition()

SARkushDecomposition

MedChemica
CREATING A STEP CHANGE IN MEDICINAL CHEMISTRY

self._desymmetrised_atom_assignment(compounds, **ScaffoldMol**)

self.sarkush_mol = **SARkushMol**.__init__ (**ScaffoldMol**)

- **Iterate over compounds in order of descending number of sidechains:**
  - **For first compound:**

    match_atom_ids = **ScaffoldMol**.get_substructure_match_for_mol(**rdchem.Mol**)

    mol_match = [**AtomMatchOption** for atom_id in match_atom_ids]

    self. **SARkushMol** .add_mol_match(mol_match, compound_SMILES)

  - **For following compounds:**

    mol_match_options = **ScaffoldMol** .get_substructure_matches_for_mol(**rdchem.Mol**)

    - **Iterate over mol_match_options:**

      mol_match = [**AtomMatchOption** for atom_id in match_atom_ids]

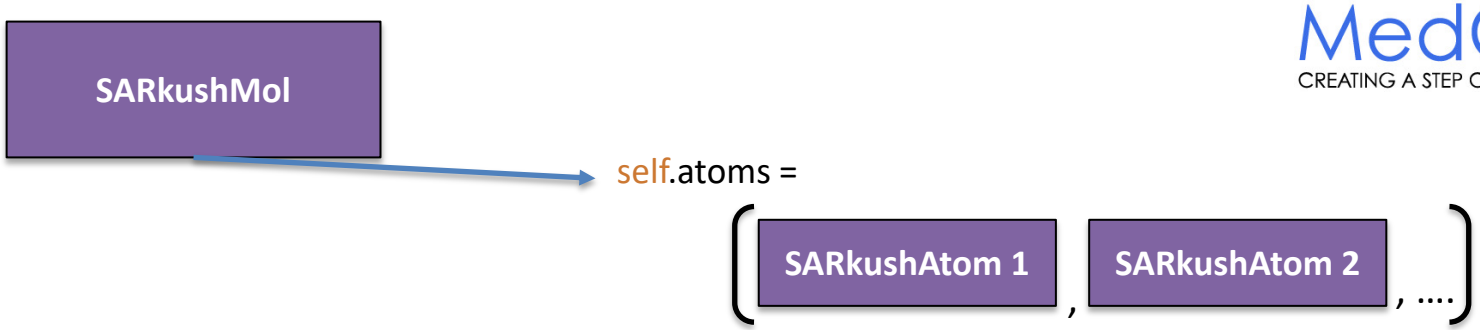
      match_score = self. **SARkushMol** .score_new_mol_match_option(mol_match)

    - **For highest scoring mol_match:**

      self. **SARkushMol** .add_mol_match(best_mol_match, compound_SMILES)

Finds ANY atom match between Scaffold and Mol

Finds ALL atom matches between Scaffold and Mol



self.score_new_mol_match_option(AtomMatchOption 1 , AtomMatchOption 2 , ...)

- **Iterate over SARkushAtoms self.atoms :**

  atom_score = SARkushAtom i .score_new_atom_match_option(AtomMatchOption i)

- **Return cumulative atom score**

SARkushAtom

{"SMILES A": **AtomMatchOption A**,
"SMILES B": **AtomMatchOption B**,
etc. }

```
self.possible_sidechain_matches =
    [ [0, 1, 2], [0, 2, 1],
      [1, 0, 2], [1, 2, 0],
      [2, 0, 1], [2, 1, 0] ]
```

String comparisons using own scoring function

self.score_new_atom_match_option(**AtomMatchOption C**)

- **Iterate over compound AtomMatchOptions in self :**

  cumulative_atom_score += **AtomMatchOption C** .atom_similarity(**AtomMatchOption**)

- **Iterate over self.possible_sidechain_matches:**
  - **Iterate over sidechain ids in sidechain match option:**

    sidechain_match_score += **AtomMatchOption C** .sidechain_similarity(new_sidechain_id, **AtomMatchOption**, sidechain_id)

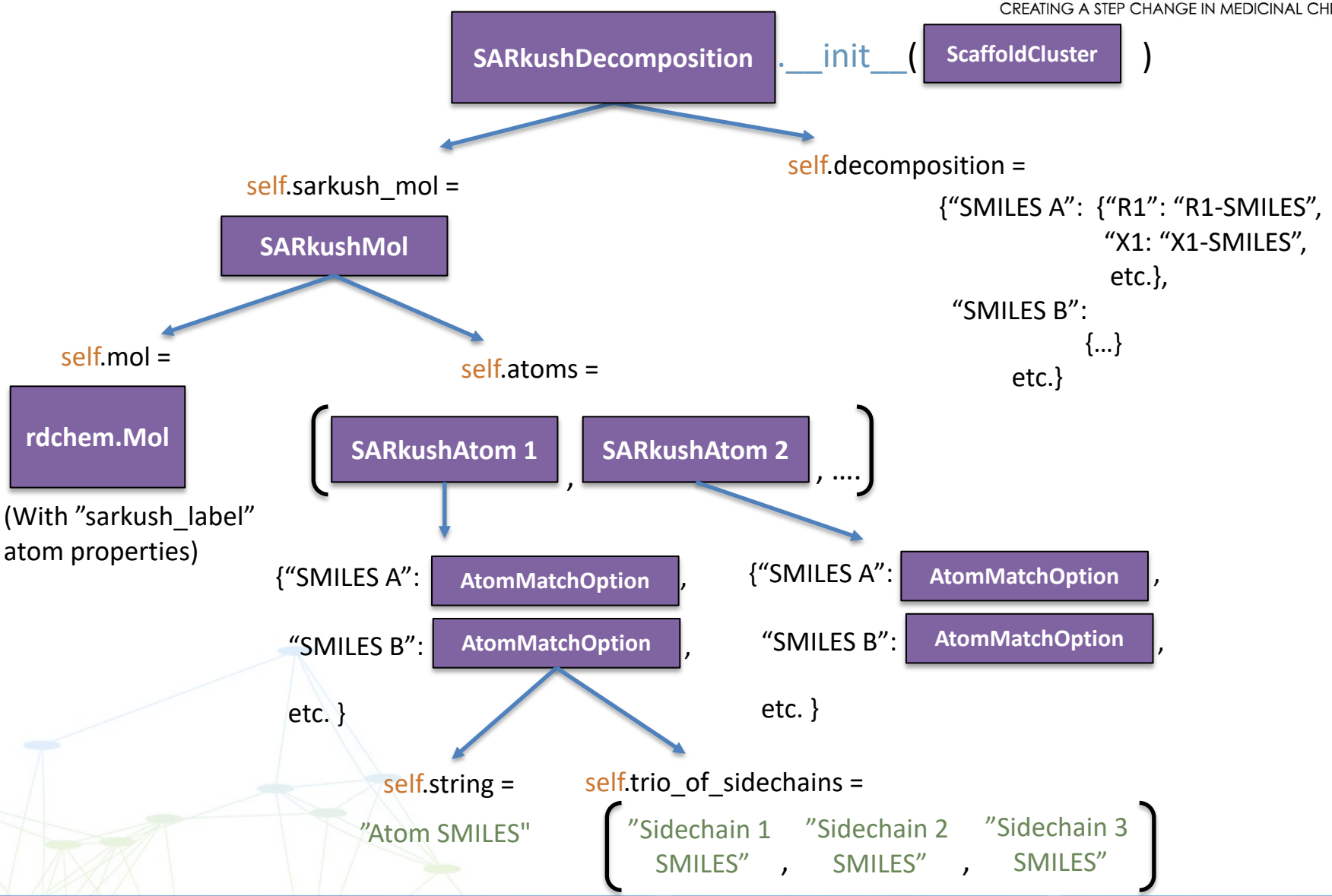
    Uses Morgan Fingerprints

  - **Store best scoring sidechain match option**

- **Return sum of cumulative atom score and cumulative best sidechain match score**

Classes used in SARkush Decomposition

MedChemica
CREATING A STEP CHANGE IN MEDICINAL CHEMISTRY



SARkush Decomposition

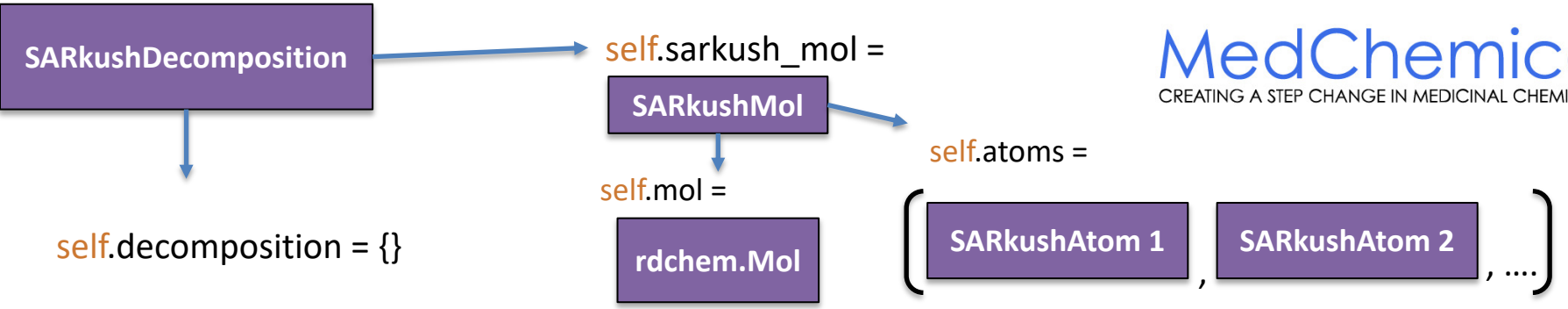
MedChemica
CREATING A STEP CHANGE IN MEDICINAL CHEMISTRY

SARkushDecomposition.__init__(**ScaffoldCluster** , report_groups_implicitly=False)

self._perform_decomposition(**ScaffoldCluster** .compounds, **ScaffoldCluster** .scaffold)

self._desymmetrised_atom_assignment(compounds, **ScaffoldMol**)

self._atom_assignments_to_sarkush_decomposition()



MedChemica
CREATING A STEP CHANGE IN MEDICINAL CHEMISTRY

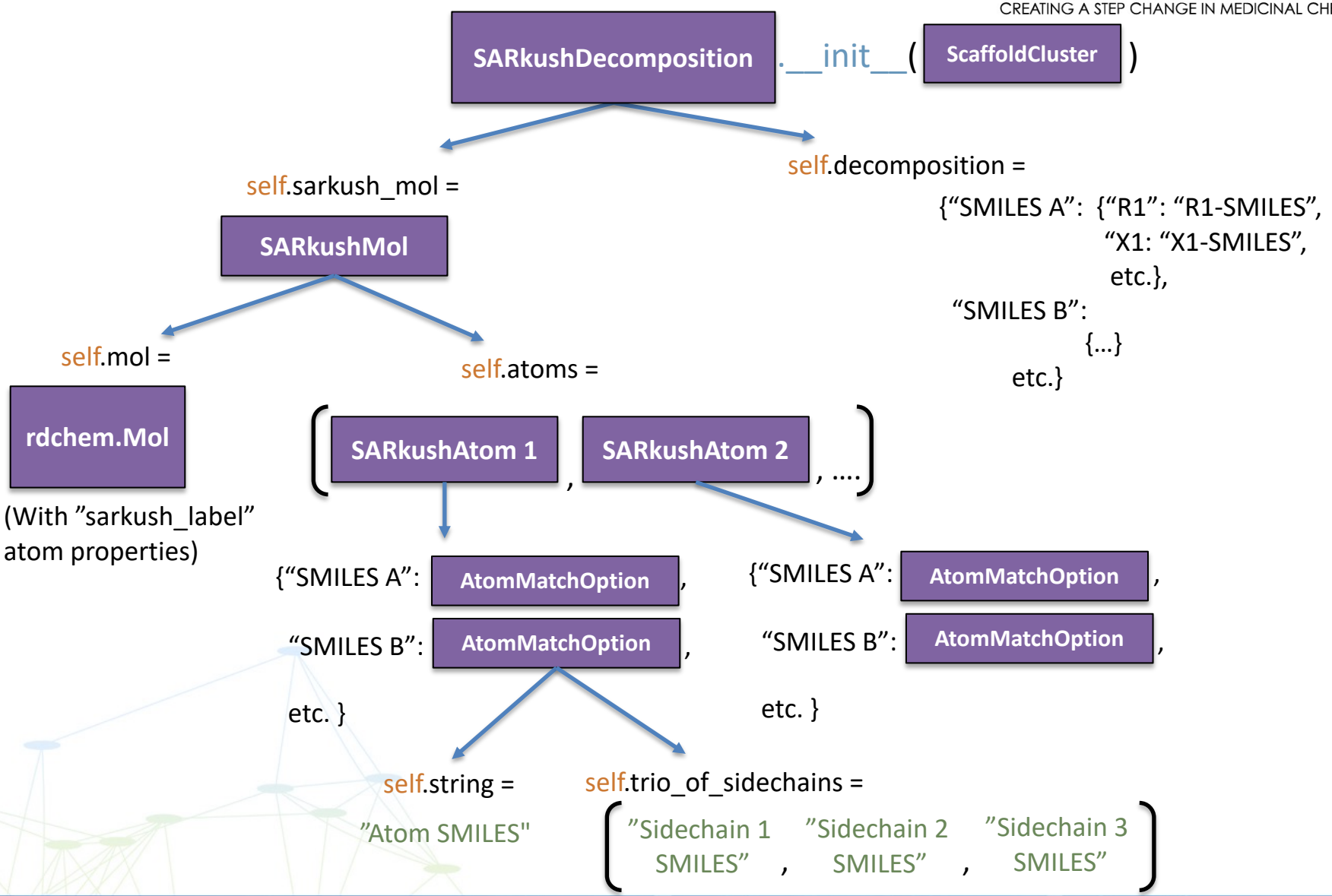
self._atom_assignments_to_sarkush_decomposition()

- **Iterate over SARkushAtoms in self. SARkushMol .atoms:**

```
if SARkushAtom.is_constant():
    self.SARkushMol.replace_variable_atom_with_constant_atom(sarkush_atom_id, SARkushAtom.return_atom_i_string(0))
    self._add_sidechains_to_sarkush_mol(sarkush_atom_id, SARkushAtom)

else:
    sarkush_label = self.SARkushMol.set_atom_sarkush_label(sarkush_atom_id)
    self._add_sarkush_atom_to_decomposition_dict(sarkush_label, SARkushAtom)
    self._add_sidechains_to_sarkush_mol(sarkush_atom_id, SARkushAtom)
```

Classes used in SARkush Decomposition

MedChemica
CREATING A STEP CHANGE IN MEDICINAL CHEMISTRY



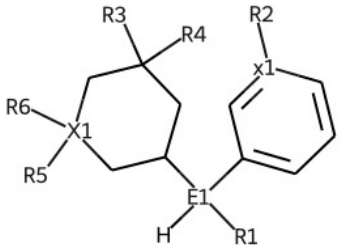
Example

MedChemica
CREATING A STEP CHANGE IN MEDICINAL CHEMISTRY

SARkushDecomposition.__init__(**ScaffoldCluster**)

self.sarkush_mol =

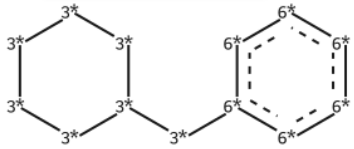
SARkushMol



[H][E1](c1ccc[x1]([R2])c1)(C1CC([R3])([R4])C[X1]([R5])([R6])C1)[R1]

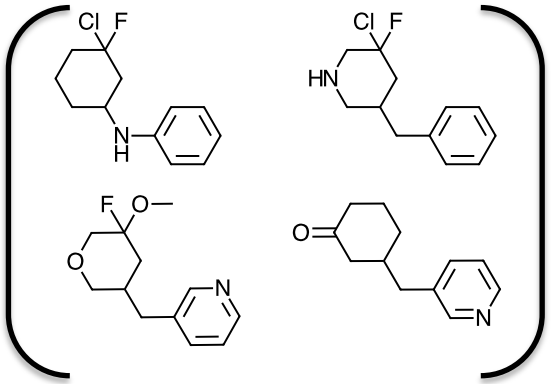
self.scaffold =

ScaffoldMol

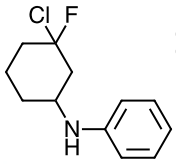


[3*]1[3*][3*][3*]([3*][6*]2:[6*]:[6*]:[6*]:[6*]:[6*]:2)[3*][3*]1

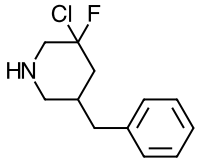
self.compounds =



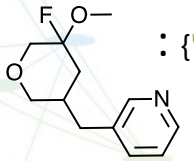
self.decomposition =



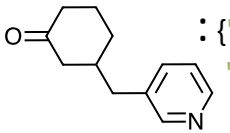
: {'E1': 'N', 'R1': None, 'x1': 'c',
'R2': '*[H]', 'R3': '*F', 'R4': '*Cl',
'X1': 'C', 'R5': '*[H]', 'R6': '*[H]'}



: {'E1': 'C', 'R1': '*[H]', 'x1': 'c',
'R2': '*[H]', 'R3': '*F', 'R4': '*Cl',
'X1': 'N', 'R5': '*[H]', 'R6': None},



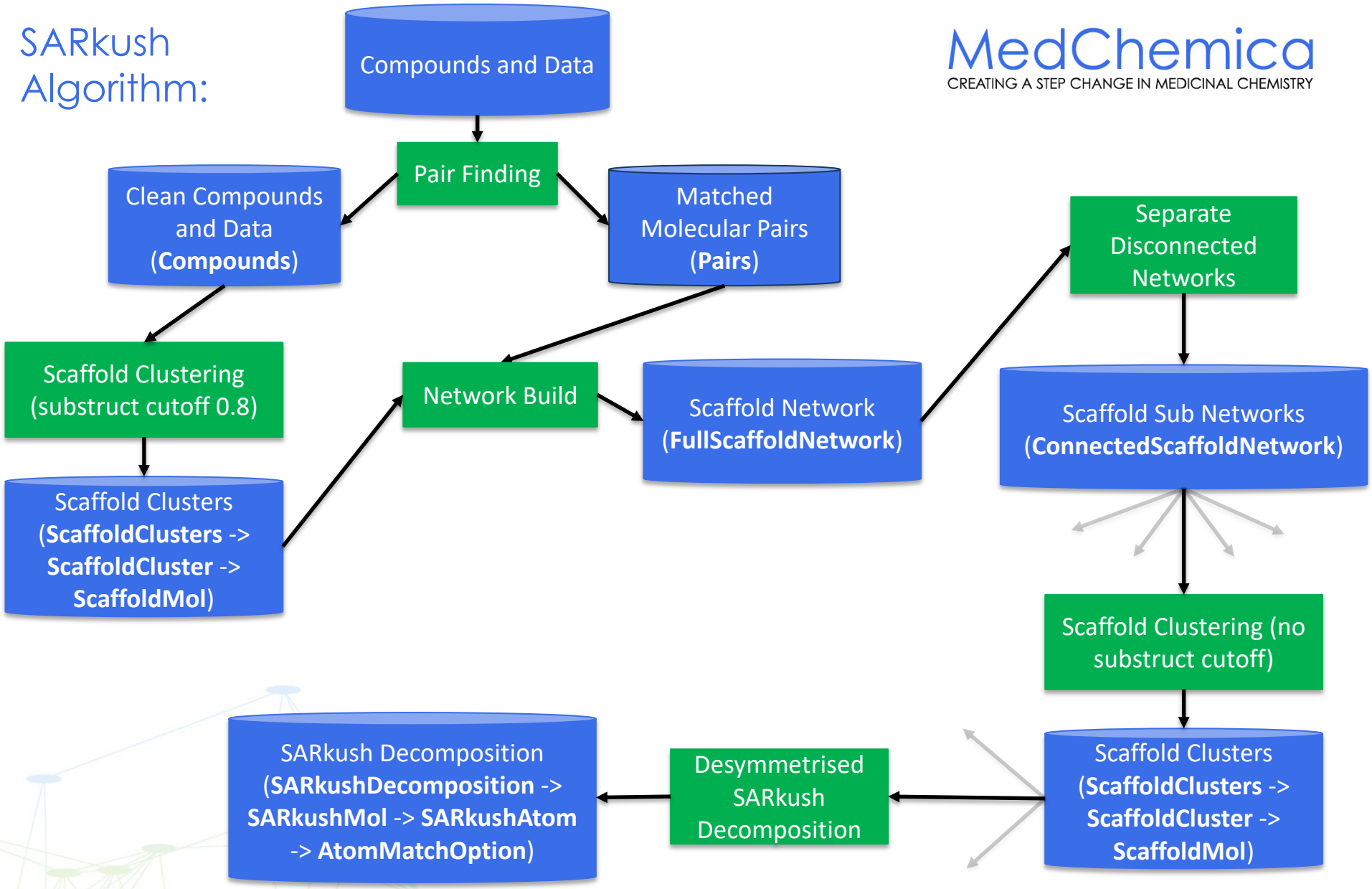
: {'E1': 'C', 'R1': '*[H]', 'x1': 'n',
'R2': None, 'R3': '*F', 'R4': '*OC([H])([H])[H]',
'X1': 'O', 'R5': None, 'R6': None}



: {'E1': 'C', 'R1': '*[H]', 'x1': 'n',
'R2': None, 'R3': '*=O', 'R4': None,
'X1': 'C', 'R5': '*[H]', 'R6': '*[H]'}

SARkush Algorithm:

MedChemica
CREATING A STEP CHANGE IN MEDICINAL CHEMISTRY



Thank you!

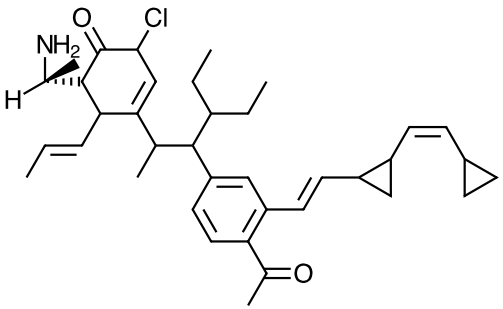
Cheminformatics Expectations:

Cheminformatics Reality:

Complicated Test Molecule

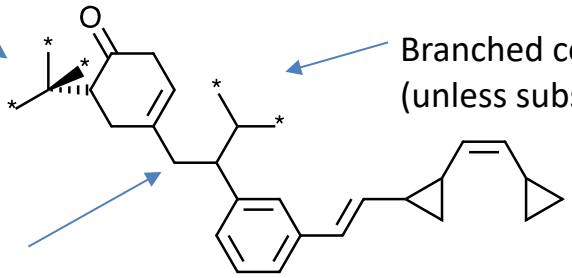
MedChemica
CREATING A STEP CHANGE IN MEDICINAL CHEMISTRY

ScaffoldMol



self._cut_off_ring_and_branch_substituents()

Chiral centres are retained using dummy atoms



Branched centres are retained using dummy atoms (unless substituents are terminal atoms)

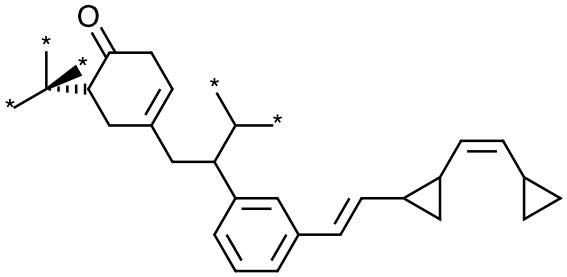
If a branched centre leads to a terminal atom, the terminal atom is removed

Ring substituents are removed

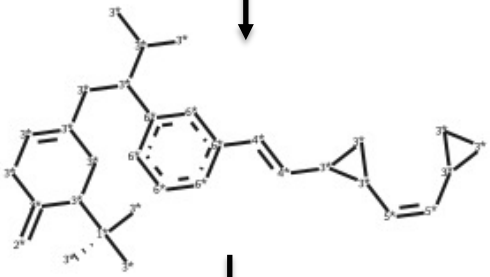
Complicated Test Molecule

MedChemica
CREATING A STEP CHANGE IN MEDICINAL CHEMISTRY

ScaffoldMol



self._set_atoms_to_generic()



self.mol

self.get_scaffold_smiles()

self.smiles

Isotope 2 = carbonyl or thiocarbonyl heteroatom

Isotope 4 = atom belonging to E double bond

Isotope 6 = aromatic atom

```
[2*]=[3*]1[3*][3*]=[3*]([3*][3*]([3*]([3*])[3*])[6*]2:[6*]:[6*]:[6*]:[6*](/[4*]=[4*]/
[3*]3[3*][3*]3/[5*]=[5*]\[3*]3[3*][3*]3):[6*]:2)[3*][3*]1[1*]([3*])([3*])[3*]
```

Isotope 5 = atom belonging to Z double bond

Isotope 1 = aliphatic chiral atom

- the "@" is missing from the SMILES because the asymmetry has been removed, meaning rdkit no longer recognises the chirality
- using an isotope label to encode the presence of chirality allows us to separate these scaffolds from the racemic ones

Isotope 3 = aliphatic racemic atom

AtomMatchOption

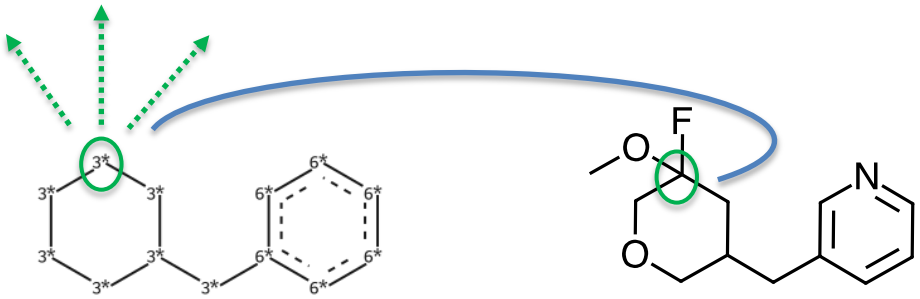
```
self._carbon = ["C", "[13C]", "c"]
self._halogens = ["Cl", "F", "Br", "I"]
self._heteroatoms = ["O", "S", "N", "P", "Si", "Se", "o", "s", "n", "p",
                     "si", "se", "[n+]", "[S+]", "[N+]", "[s+]", "B", "b"]
self._hydrogen = "[H]"


self._exact_atom_match_score = 1.0
self._same_atom_different_impl_groups = 0.5
self._no_match_score = 0.0
self._both_heteroatoms_score = 0.5
self._both_halogen_score = 0.5
self._one_heteroatom_one_halogen = 0.33
```

self.atom_similarity(**AtomMatchOption**)

- **Compares strings and scores based on above definitions**

Desymmetrisation also important for sidechains

MedChemica
CREATING A STEP CHANGE IN MEDICINAL CHEMISTRY



AtomMatchOption

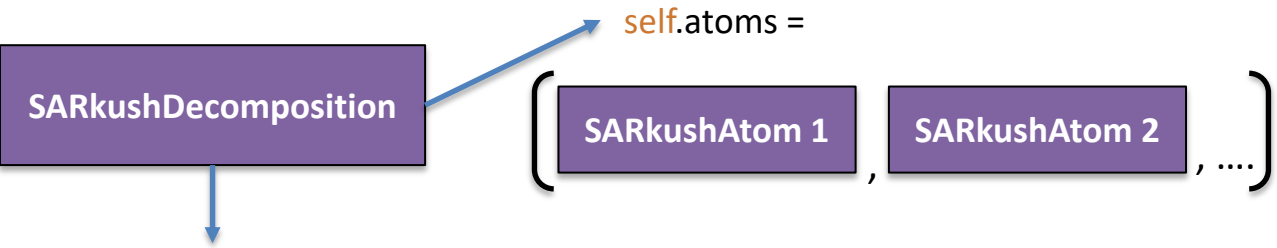
self.string =

"C"

self.trio_of_sidechains = 1 of:

("*OC", "*F", None) ("*F", "*OC", None) (None, "*OC", "*F")

("*OC", "*F", None) ("*F", None, "*OC") (None, "*OC", "*F")



MedChemica
CREATING A STEP CHANGE IN MEDICINAL CHEMISTRY

self.sarkush_mol = SARkushMol

self._add_sidechains_to_sarkush_mol(sarkush_atom_id, SARkushAtom)

- **Iterate over sidechain ids:**

```
if SARkushAtom.sidechain_i_is_constant(sidechain_id):
        constant_sidechain = SARkushAtom.return_sidechain_i(0, sidechain_id)
        self.SARkushMol.add_sidechain(sarkush_atom_id, constant_sidechain)
else:
        sarkush_label = self.SARkushMol.add_R_group(sarkush_atom_id)
        self._add_R_group_to_decomposition_dict(sarkush_label, SARkushAtom, sidechain_position=sidechain_id)
```