Codessa Manual: Fragments in Codessa

Introduction

A "Fragment" in Codessa is simply a named group of atoms. The number, type and connectivity (optionally including bond type) of atoms in a fragment can be required to be identical for each instance of a fragment in a molecule, such as in a functional group like a carboxyl group or phenyl group. The number, type and connectivity (optionally including bond type) of atoms in a fragment can also be allowed to be somewhat different or very different for each instance of a fragment in a molecule, such as in substituents of varying types. A simple example is a series of phenol derivatives, which has at least three useful common-substructure fragments that could be defined: the hydroxyl group of atoms or the phenyl group of atoms (including some or all hydrogen atoms) or the entire phenol group of atoms (excluding at least one hydrogen atom). The complement of a common-substructure fragment could also be used as a fragment, e.g., in the series of phenol derivatives, the atoms and bonds that are not part of a phenol fragment could be used as a "non-common-substructure" fragment. Usually, but not necessarily, a fragment is fully connected, i.e., each atom of a fragment is connected to at least one other atom in the fragment.

Fragments are defined by users in Codessa based on the molecules in the project of interest, but once defined, *instances* of a fragment in any given molecule can automatically determined by Codessa 3. Instances of a fragment in a given molecule can also defined explicitly by the user (in the case where the automatic determination for a given molecule is not what the user wants, for example).

Many descriptors that can be defined and calculated for a whole molecule can be expressed in terms of atomic and / or bond contributions and thus values for such descriptors can also be calculated for a fragment in a molecule. For example, "Maximum ESP Atomic Charge", "Average Bond Length for a C-O Bond" and "Volume", are descriptors that can have well-defined and potentially physically useful values for a fragment. Some descriptors, such as "HOMO-LUMO energy", do not make sense for fragments because they cannot be well-expressed in terms of atomic or bond contributions. Some descriptors - particularly those which have a constant value for molecules - make more sense for fragments than for molecules, e.g., "Net Charge". A special descriptor for any fragment is named "Number of Fragments<<...>>", which is a count of the number of instances of a Fragment in a molecule (... is a placeholder for the name of the fragment). Like other count descriptors (e.g, "Number of Atoms", "Number of Cl Atoms", "Number of Bonds", "Number of O-H Bonds", etc.), the "Number of Fragments<<...>>" descriptor always has a valid value, defaulting to 0.

Fragments can be useful in QSAR in multiple ways, most obviously when a set of molecules possess a common (which could mean identical or similar depending on the fragment definition) substructure, the idea being that values for some descriptors or descriptor combinations of the common substructure may correlate better with a property of interest than whole-molecule descriptor or descriptor combinations do. This is, of course, especially true for local or regional properties. Other ways fragments can be useful in QSAR are when a set of molecules has a varying substituent at a particular common position, the idea being that values for descriptor of descriptor combinations of the variable substituent may correlate better with a property of interest.

Sometimes it may be useful to consider the complement of a Fragment (a "Not Fragment"), the idea being, for example, that values for a descriptor or for descriptor combinations of the parts of the molecules that are *not* part of a common substructure may correlate relatively well with a property of interest.

 Revision Date: July 8, 2013

For Codessa, Semichem devised its own line notation (SCLN) for defining fragments that is pretty powerful and flexible and that can be used in conjunction with all Semichem software (including Codessa and Agui) both to automatically identify instances of a specified fragment in any molecule and to automatically generate an SCLN string for a specified group of atoms.

Semichem decided to develop SCLN instead of using an existing line notation, such as SMILES or InCHI, so that they could have more flexibility to do what was wanted and needed to do with fragments, without having to worry about adhering to any existing - and limiting - formats. The SCLN contains some unique features not found elsewhere, while some features found elsewhere are not found in SCLN.

Every Codessa project contains a "Fragment Table" that is initially empty. This is simply a table whose rows are Fragments and whose columns are Fragment characteristics like Fragment Names and Fragment Strings (a Fragment String is its definition in terms of SCLN) and rules (average, maximum, minimum, sum, etc) for calculating descriptor values for a fragment when multiple instances of the fragment are present in a molecule (e.g., multiple methylol groups if the fragment is a methylol group).

Fragments are defined by the user, and users will typically want to define their own Fragments for a given project and / or series of molecules. Once defined, Codessa can automatically identify instances of fragments in any molecule and calculate fragment descriptor values, just as it can automatically calculate molecule descriptor values.

For example, if one has a series of compounds with carboxylic acid groups that one wants to study, then one may want to add a Carboxyl Fragment to the Fragment Table with the following fields:

- Name = Carboxyl
- Symbol = COOH
- Fragment String = C(O,O(H),^C)
- Fragment Instances = At Least 1
- Multi-Instance Method = Average

The Fragment String in this case is C(O,O(H),^C), i.e, carboxyl group bonded to a carbon. Note that atoms whose symbols are preceded by ^ are technically not part of the fragment (i.e., are not used for the purposes of descriptor calculations) but are only used to help define the desired "environment" of the actual atoms of the fragment. Thus, the Carboxyl fragment as defined by C(O,O(H),^C) occurs in acetic acid but not formic acid.

A fragment can be added to the "Fragments" table either by using the "New" button in the "Fragments" table or visually by selecting appropriate atoms in a 3D window containing the fragment and then right-clicking and selecting "Make New Fragment String (...) From Selected Atoms".

Descriptor values for fragments are calculated just as descriptor values for molecules are. Descriptor values for fragments can be visualized in tables and used for generating and applying correlations and other purposes just as descriptor values for molecules can.

Fragment Strings (SCLN)

For a small number of molecules, it would be sufficient for the user to explicitly identify and list the atoms and bonds making up the fragment instances for each molecule of a set. For any given fragment, this can always be done by clicking the "Fragments" button of the "Structures" dialog and editing the appropriate row of the "Fragment Atoms" column in the "Fragments of Structure..." dialog. It can also always be done by selecting the appropriate atoms of the fragment in a 3D window showing the structure and then selecting "Assign Selected Atoms as Fragment" from the context menu obtained by right-clicking in the 3D window. For a large number of molecules, assigning fragment atoms molecule-by-molecule becomes tedious and error prone, and so a way of automatically identifying the fragment atoms in any molecule is needed. That is the primary purpose of the SCLN for fragments, i.e., the purpose of Fragment Strings.

A Fragment String is simply a series of characters that represents a group of atoms, how these atom are connected together (possibly including bond type) and possibly what their environment should be. For example, the Fragment String C(O,O(H),^C) represents a carboxyl group (the Fragment) bonded to a carbon atom (the environment).

Characters that can Appear in Fragment Strings, and Their Purpose

Characters	Purpose
a b c d e f g h i j k l m n o p q r s t u v w x y z	Used for element symbols of atoms. For example, a fragment string for an amide group bonded to a carbon atom via its carbonyl carbon would be: C(O,N(H,H),^C)
0 1 2 3 4 5 6 7 8 9	Used for atom labels. Atom labels are necessary for closing rings but are otherwise optional.

	For example, a fragment string for an oxirane group bonded to anything via one of its carbon atoms would be: C1(O(C(C1,H,H)),H,?)
*	Wildcard symbol that matches any element, with specified bonding to other atoms in the following parentheses. By default, * wildcard atoms are part of the fragment (as opposed to the environment). For example, a fragment string for an ethyl group bonded to any atom would be: C(H,H,H,C(H,H,*))
~	"Not" element. Must be followed by an element symbol. For example, ~H matches any atom other than hydrogen. For example, a fragment string for a methylene group would be: C(H,H,~H,~H)
$	Optional atom. Must be followed by an element symbol. Matches an atom of the specified element or no atom. For example, C(:H,:H,:H,:$N) matches CH3 and CH3N but not CH3O. C(:H,:H,:H,$?) matches CH3 and CH3N and CH3O (and also CH3F, etc).
?	Wildcard symbol that matches any element, with unspecified bonding to other atoms. ? wildcard atoms are never part of fragment. For example, a fragment string for any methyl group would be: C(H,H,H,?)
(	Begins bonding list for preceding atom. If an atom symbol is not followed by a parenthesis then any bonding (including none) for the atom is matched.
,	Separates items in a bonding list.
)	Ends bonding list for atom preceding the corresponding (
.	Used along with : to define bond types. See following table.
:	Used along with . to define bond types. See following table.
^	Indicates an environmental atom, not part of actual fragment, only used to help define the context of the fragment. For example, a fragment string for an amide group bonded to an environmental methylene group via its carbonyl carbon would be: C(O,N(H,H),^C(^H,^H,?))
!	"Not" Fragment. Complement of the fragment defined by the string following the ! character. Means everything in the molecule that is not part of the fragment defined by the characters following the ! character. If present, ! must be the first character of the fragment string. For example, !C1(O(C(C1,H,H)),H,?) matches any atom in the molecule that is not part of an oxirane group.
[	Begin metadata. Currently unused.
]	End metadata. Currently unused.

Bond Types in Fragment Strings

Bond Type SubString	Meaning
Nothing, i.e., no colons or periods	Any Bond Type
.	Weak Bond Type
:	Single Bond Type

:. (or .:)	Aromatic Bond Type
::	Double Bond Type
:::	Triple Bond Type

Bond type characters (. and :) must *precede* element symbol (e.g., C) or wildcard symbol (* or ?) or optional atom symbol ($) characters. Basically a colon (:) represents a "one-bond" and a period (.) represents a "half-bond", so that an "aromatic" or "resonant" bond type (one and a half bonds) would be represented by :.

Note that a fragment string for a fragment is typically not unique, i.e., except for trivial fragments, there is usually more than one way to specify a fragment using SCLN. Fragment strings are just a means to an end.

Note that atom numbering is needed for closing rings and for meta-data but is otherwise optional.

Note that atoms preceded by ^ are used to define the environment for the fragment but are not actually part of the fragment to be used for calculating descriptors.

Note that fragments defined using SCLN must be fully connected, i.e., all atoms of a fragment defined using SCLN must be bonded to at least one other atom of the fragment.

Examples of SCLN Using Bond Types:

Fragment Description	Fragment String
Free Carbon atom	C()
Any Carbon atom (free or bonded to anything)	C
Any Carbon atom with 4 single bonds to anything	C(:?,:?,:?,:?)
Any Carbon atom with 2 single bonds to anything and	C(:?,:?,::?)

one double bond to anything	
Any Carbon atom with 2 single bonds to anything and one double bond to an environmental Carbon atom	C(:?,:?,^::C)
Methyl group single bonded to any other atom	C(:H,:H,:H,:?)
Methyl group bonded to any other atom	C(:H,:H,:H,?)
Methyl group optionally bonded to any other atom	C(:H,:H,:H,$?)
Methyl group optionally bonded to a Carbon atom	C(:H,:H,:H,$C)
Methyl group bonded to any terminal atom	C(:H,:H,:H,*())
Methyl group bonded to any other atom that is bonded to one other atom of any element	C(:H,:H,:H,*(?))
Methylol group, -CH2OH	C(:H,:H,:O(:H),:?)
Methylol group, where methyl group is fragment and OH is environment	C(:H,:H,^:O(:H),:?)
Ethene group single bonded to anything	C(:H,:H,::C(:H,:?))
Acetylene group single bonded to anything	C(:H,:::C(?))
Ethyl group single bonded to anything	C(:H,:H,:H,:C(:H,:H,:?))
Methoxy group single bonded to anything	O(C(:H,:H,:H),:?)
Oxirane group single bonded to anything	C1(:H,:?,:C(:H,:H,:O2(:C1)),:O2)
Phenyl group with 1 arbitary substituent	C1(:.C(:.C(:.C(:.C(:.C2(:.C1,:H),:?),:H),:H),:H),:.C2,:H)
Phenyl group with 2 arbitary ortho substituents	C1(:.C(:.C(:.C(:.C(:.C2(:.C1,:H),:?),:?),:H),:H),:.C2,:H)
Phenyl group with 2 arbitary meta substituents	C1(:.C(:.C(:.C(:.C(:.C2(:.C1,:H),:?),:H),:?),:H),:.C2,:H)
Phenyl group with 2 arbitary para substituents	C1(:.C(:.C(:.C(:.C(:.C2(:.C1,:H),:?),:H),:H),:?),:.C2,:H)
para-Fluoro Phenyl group	C1(:.C(:.C(:.C(:.C(:.C2(:.C1,:H),:H),:F),:H),:H),:.C2,:?)
para-Fluoro Phenyl group with any bond types for ring bonds	C1(C(C(C(C(C2(C1,:H),:H),:F),:H),:H),C2,:?)
Oxirane group in an Oxa-Spiro-pentane environment	C2(:C(:O3(:C2),:H,:?),:O3,:^C(:^C1(:^H,:^H,:C2),:^H,:^H),:^C1)

Most of the above examples use bond types to specify the fragment. Bond type is often ambiguous and in some cases somewhat arbitrary, especially when defined via densities or density matrices, and so *it is often preferable and sufficient to specify a Fragment without using bond types*, at least partially.

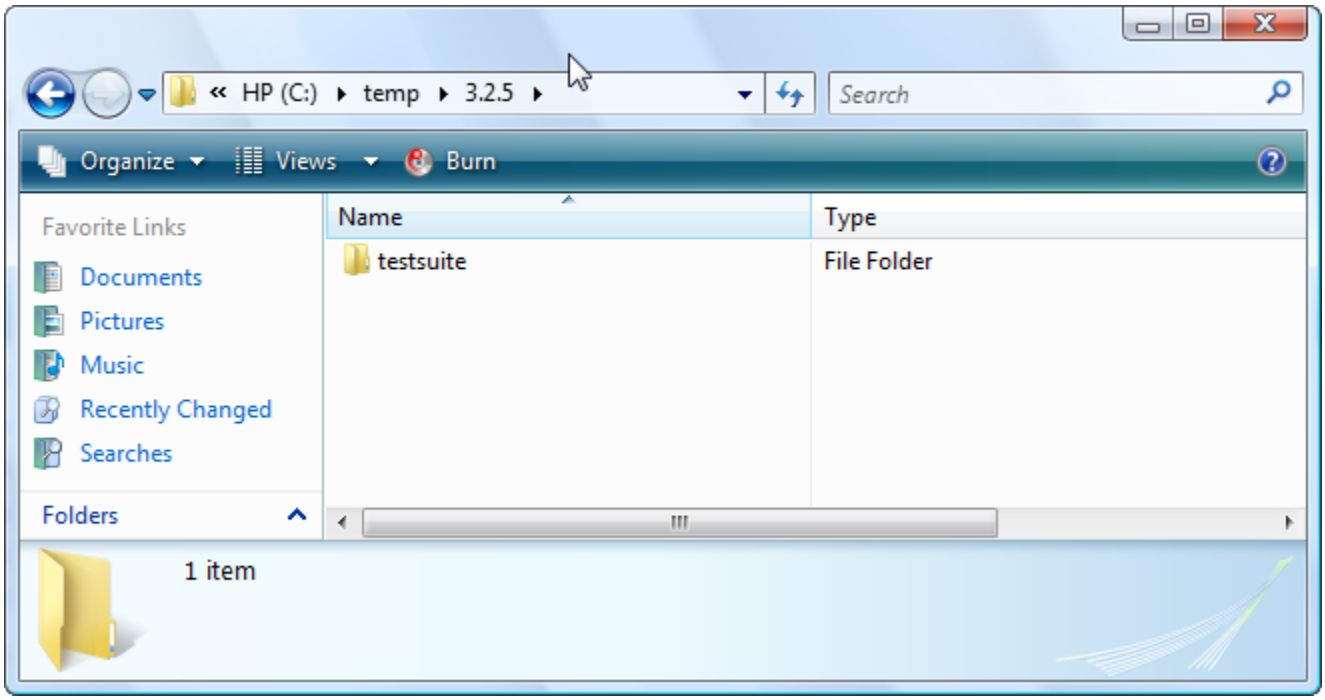
Examples of SCLN Without Bond Types

Fragment Description	Fragment String
Free Carbon atom	C()
Any Carbon atom, free or bonded to anything	C
Any Carbon atom with 4 bonds to anything	C(?,?,?,?)
Any Carbon atom with 2 bonds (of any type) to anything	C(?,?,::?)

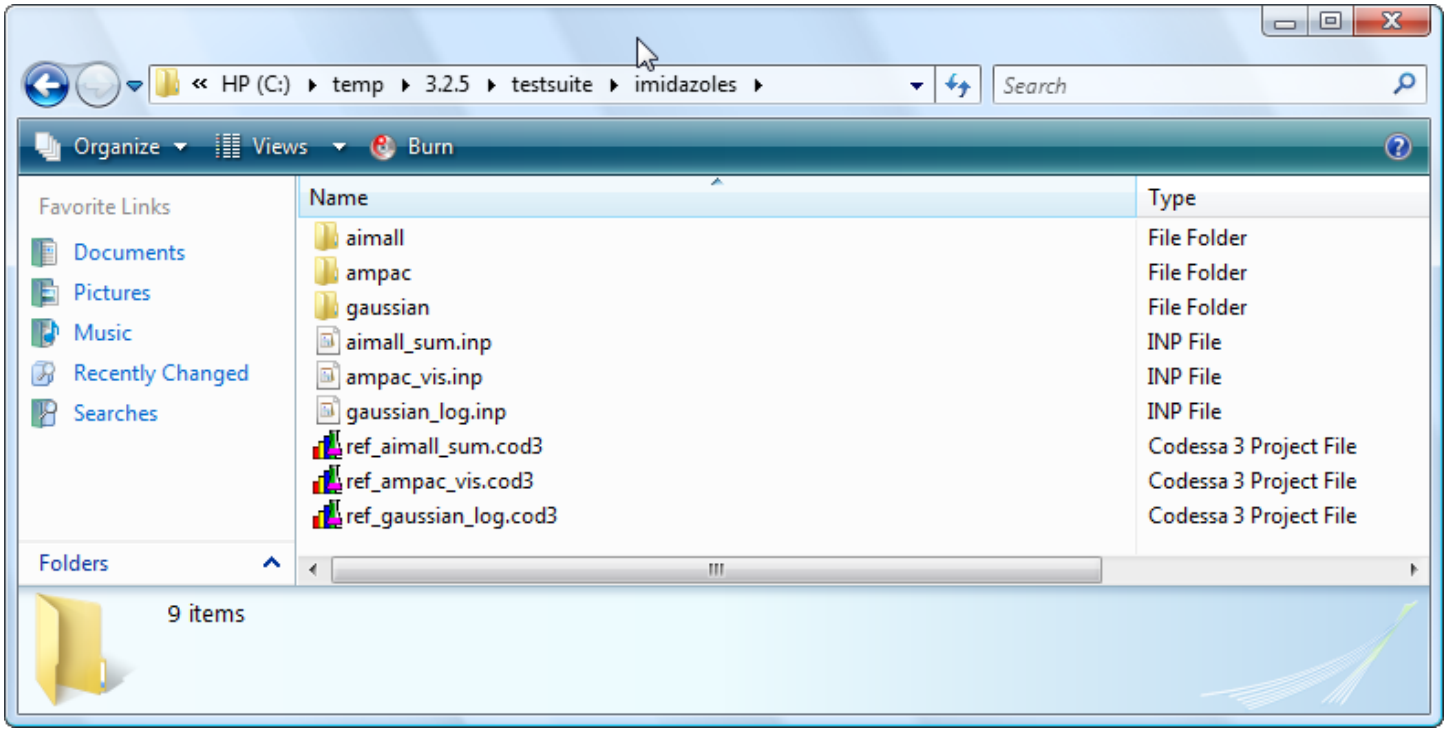
and one double bond to anything	
Any Carbon atom with 2 bonds (of any type) to any atom and one double bond to another (environmental) Carbon atom	C(?,?,^::C)
Methyl group bonded to any other atom	C(H,H,H,?)
Methyl group (optionally bonded to any atom	C(H,H,H,$?)
Methyl group optionally bonded to a Carbon atom	C(H,H,H,$C)
Methyl group bonded to any terminal atom	C(H,H,H,*())
Methyl group bonded to any atom that is also bonded to one other atom of any element	C(H,H,H,*(?))
Methylol Group, no bond types specified	C(H,H,O(H),?)
Methylol group where methyl is fragment and OH is environment	C(H,H,^O(H),?)
Ethene group bonded to anything	C(H,H,C(H,?))
Acetylene group bonded to anything	C(H,C(?))
Ethyl group bonded to anything	C(H,H,H,C(H,H,?))
Methoxy group bonded to anything	O(C(H,H,H),?)
Oxirane group bonded to anything	C1(H,?,C(H,H,O2(C1)),O2)
Phenyl group with 1 arbitary substituent	C1(C(C(C(C(C2(C1,H),?),H),H),H),C2,H)
Phenyl group with 2 arbitary ortho substituents	C1(C(C(C(C(C2(C1,H),?),?),H),H),C2,H)
Phenyl group with 2 arbitary meta substituents	C1(C(C(C(C(C2(C1,H),?),H),?),H),C2,H)
Phenyl group with 2 arbitary para substituents	C1(C(C(C(C(C2(C1,H),?),H),H),?),C2,H)
para-Fluoro Phenyl group	C1(C(C(C(C(C2(C1,H),H),F),H),H),C2,?)
para-Fluoro Phenyl group with any bond type for ring bonds and single bonds to ring hydrogens and arbitary para substituent	C1(C(C(C(C(C2(C1,:H),:H),:F),:H),:H),C2,:?)
Oxirane group in an oxa-spiro-pentane environment	C2(C(O3(C2),H,?),O3,^C(^C1(^H,^H,C2),^H,^H),^C1)

Example of Using Fragments

Step 1. Copy the testsuite folder from the Codessa-3.2 installation directory (usually C:\Program Files (x86)\Semichem, Inc\Codessa-3.2) to a folder for which you have write access (e.g., C:\Temp\3.2.5).

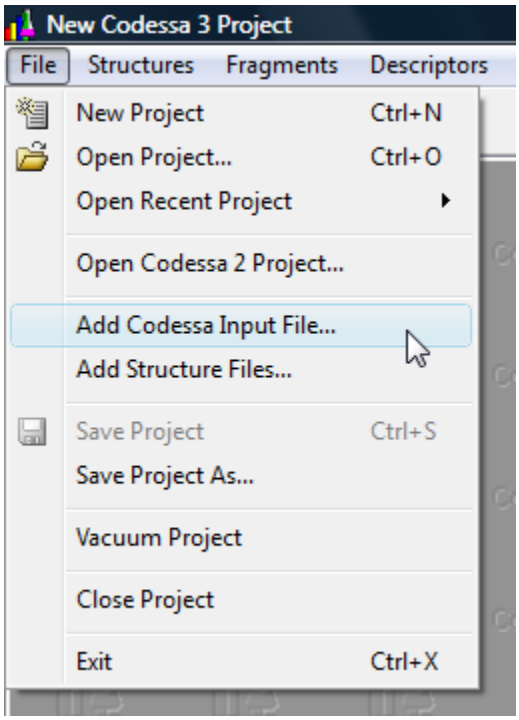


Step 2. Delete the ampac_vis.cod3 file from the Imidazole folder in the C:\Temp\3.2.3\testsuite folder.

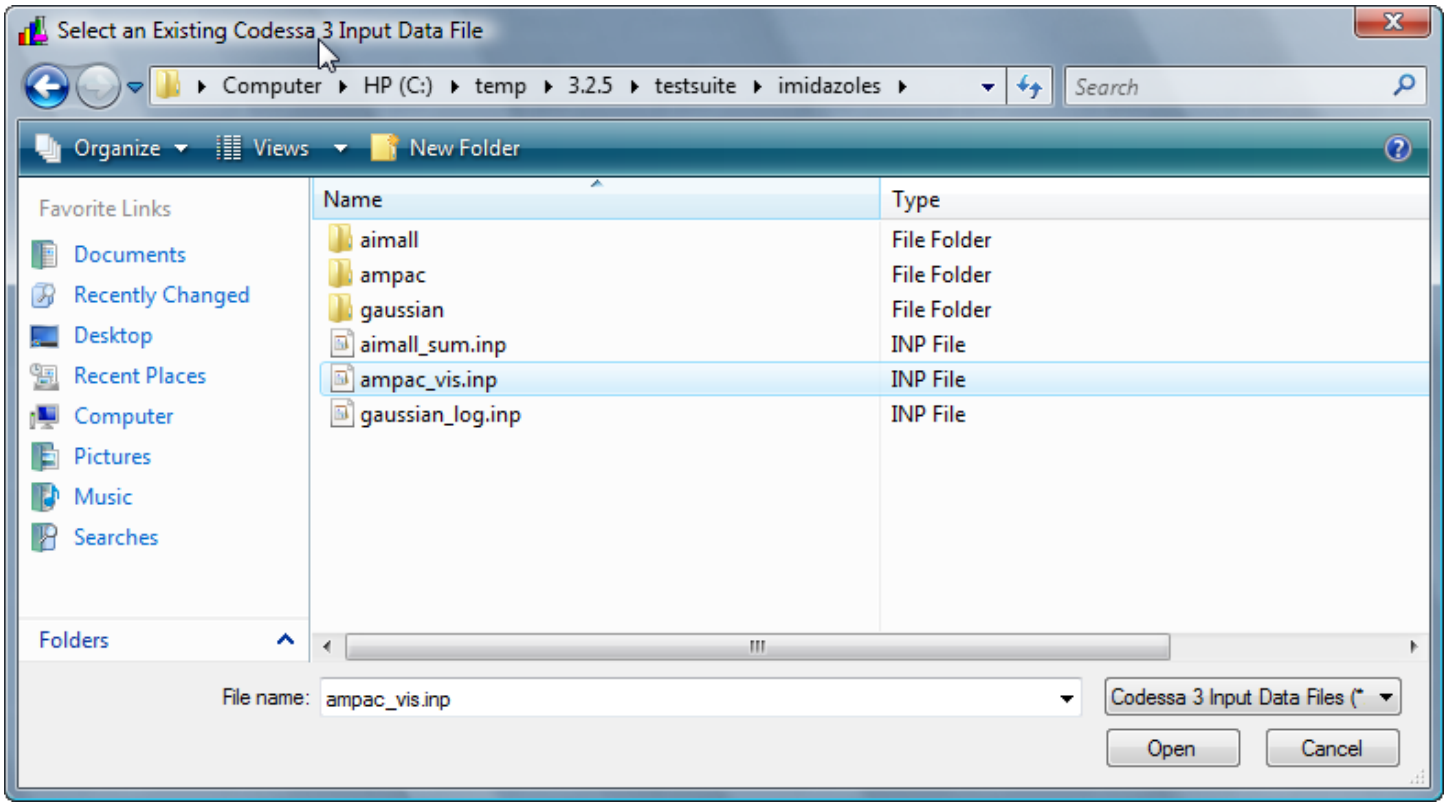


Step 3. Launch Codessa-3.2.5 from the desktop or quicklaunch area.

Step 4. Select "Add Codessa Input File..." from the File menu.

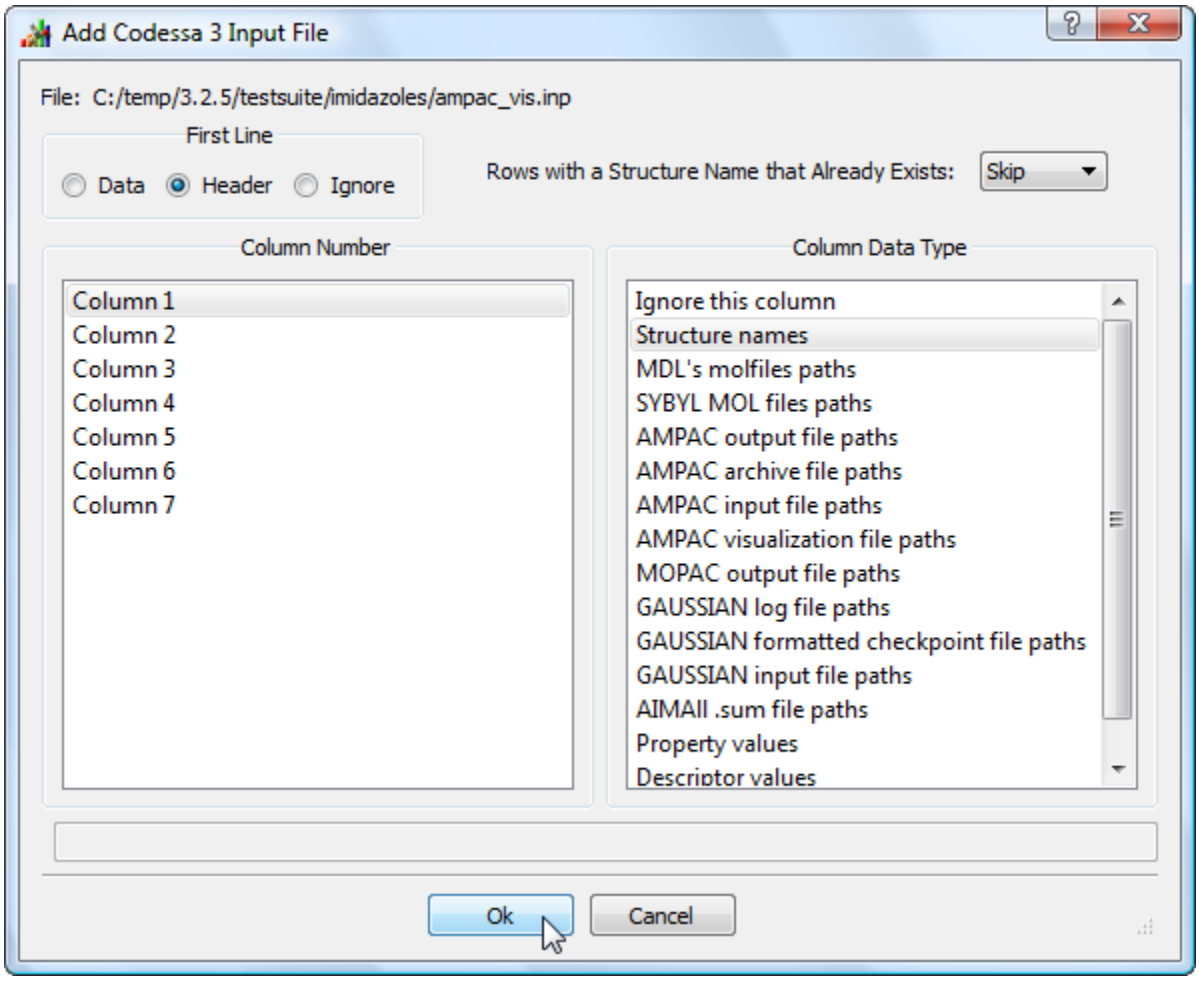


Step 5. Open the ampac_vis.inp file

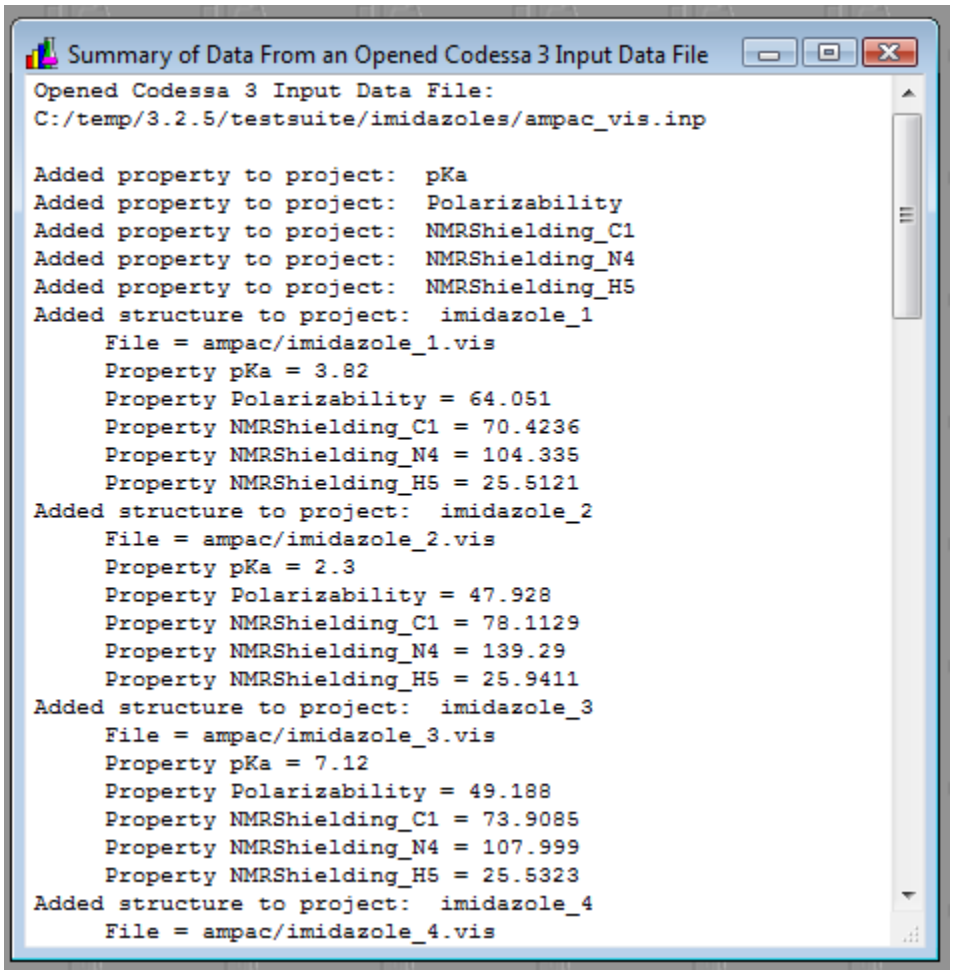


Step 6. In the "Column Types of Codessa 3 Input Data File" dialog, verify that each of the columns is correct and then select the "Ok" button. The Column Types should be:

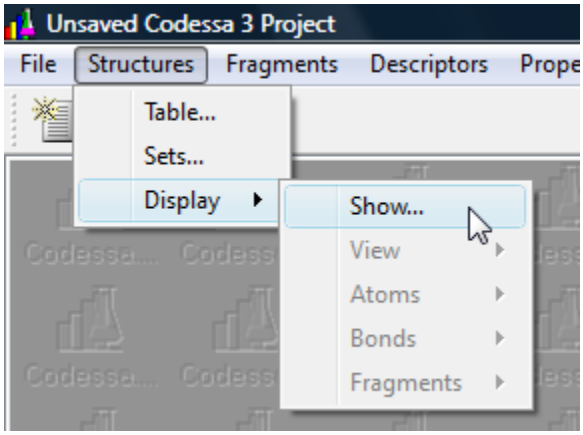
- Column 1 = Structure Names
- Column 2 = Ampac Visualization File Paths
- Column 3 = pKa Property values
- Column 4 = Polarizability Property values
- Column 5 = NMRShielding_C1 Property values
- Column 6 = NMRShielding_N4 Property values
- Column 7 = NMRShielding_H5 Property values



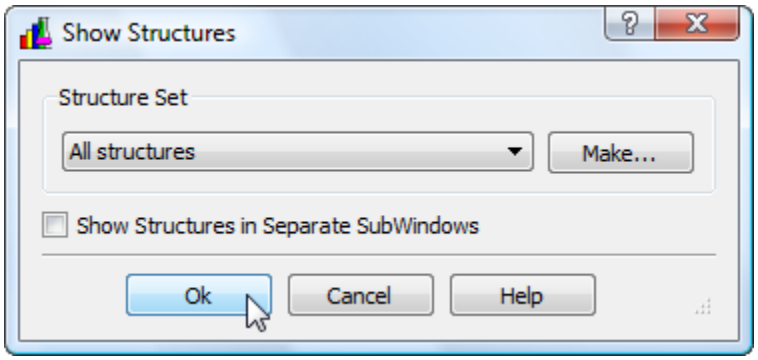
Step 7. Inspect the contents of the "Summary of Data From an Opened Codessa 3 Input File", which should indicate that 15 Imidazole structures were loaded from corresponding Ampac .vis files, along with 5 property values for each structure. Minimize or close the summary window when you are done inspecting it.



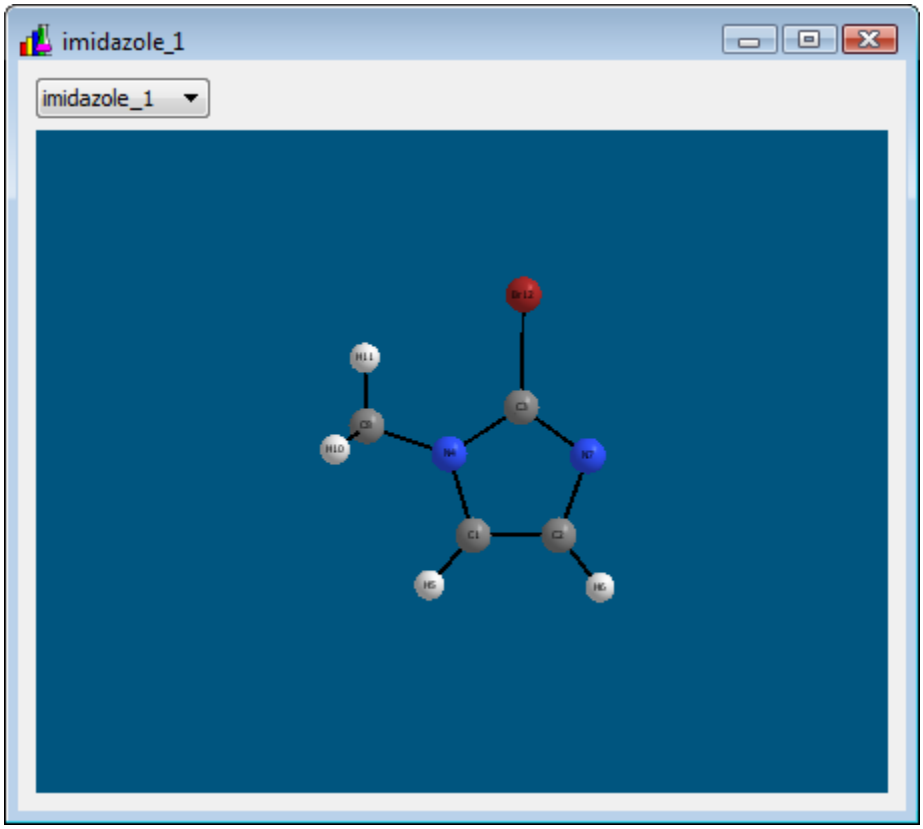
Step 8. Select the "Structures->Display->Show" menu item.



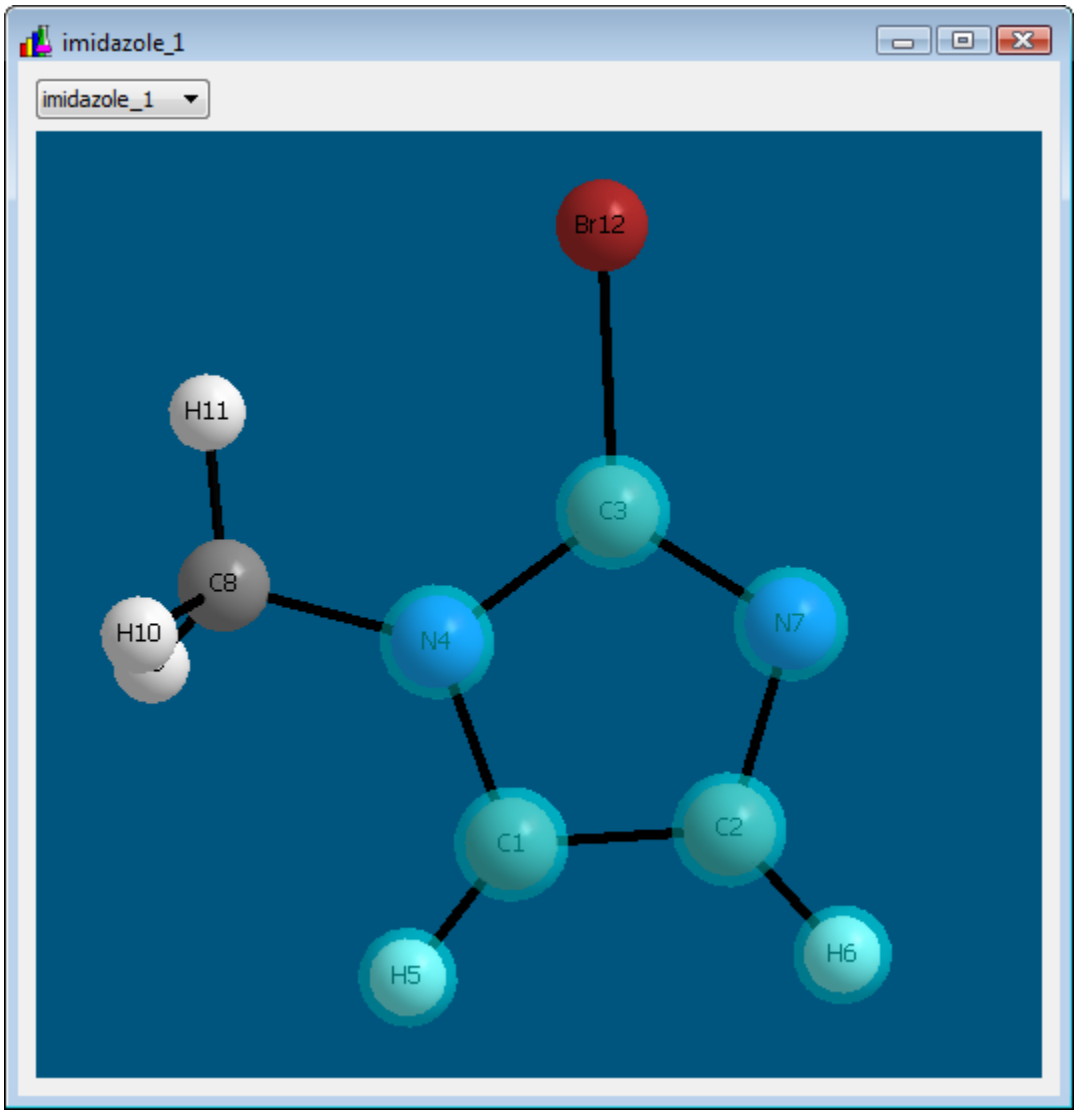
Step 9. In the "Show Structures" dialog, select the "Ok" button.



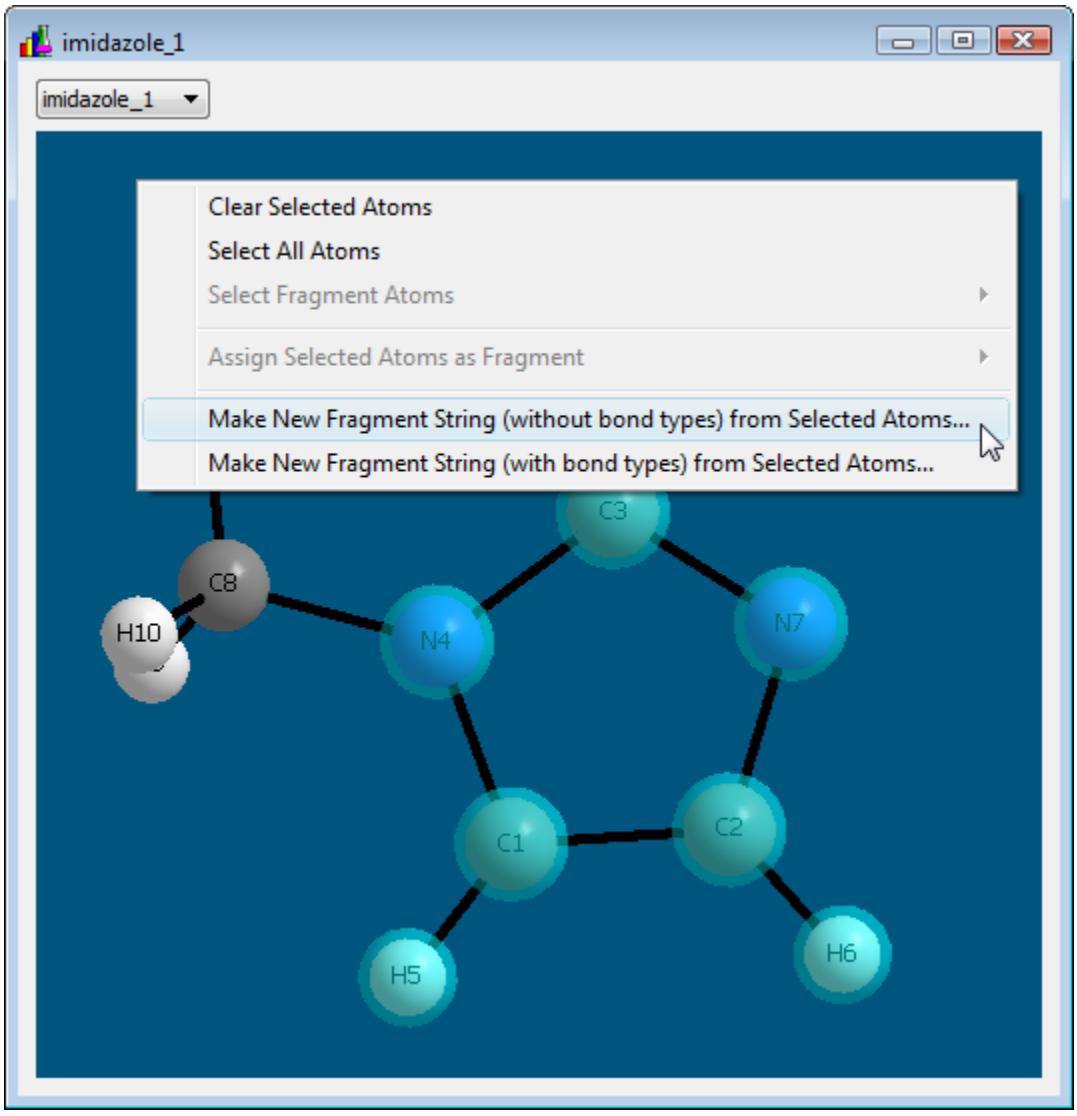
Step 10. In the 3D window, use the spin-box to view the 15 different structures. For the currently visible structure, use the mouse within the molecule portion of the 3D window to rotate, translate and zoom the molecule. Moving the mouse pointer left and right with the left mouse-button held down will rotate the molecule about the screen y-axis. Moving the mouse pointer up and down with the left mouse button held down will rotate the molecule about the screen x-axis. Moving the mouse pointer left and right with the left mouse-button held down and the shift-key pressed will translate the molecule about the along the screen x-axis. Moving the mouse pointer up and down with the left mouse-button held down and the shift-key pressed will translate the molecule about the along the screen y-axis. Using the mouse scroll-wheel or the F5 and F6 keyboard keys will zoom the structure in and out. When you are done playing with the structures, switch back to the "Imidazole_1" structure.



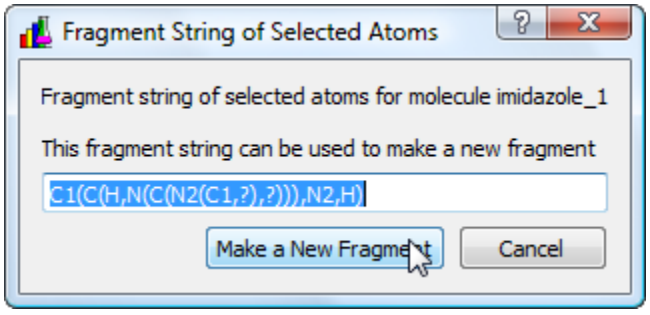
Step 11. For the "Imidazole_1" structure, click on the C1, C2, C3, N4, H5, H6 and N7 atoms.



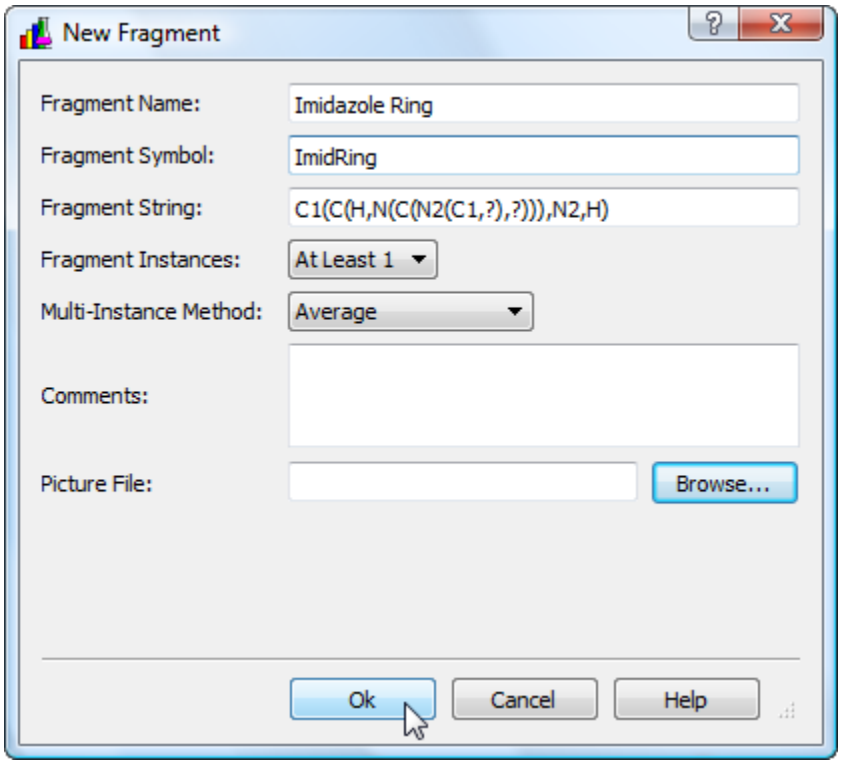
Step 12. Right-click in the 3D window to show a context menu. Select the "Make New Fragment String (without bond types) from Selected Atoms...".



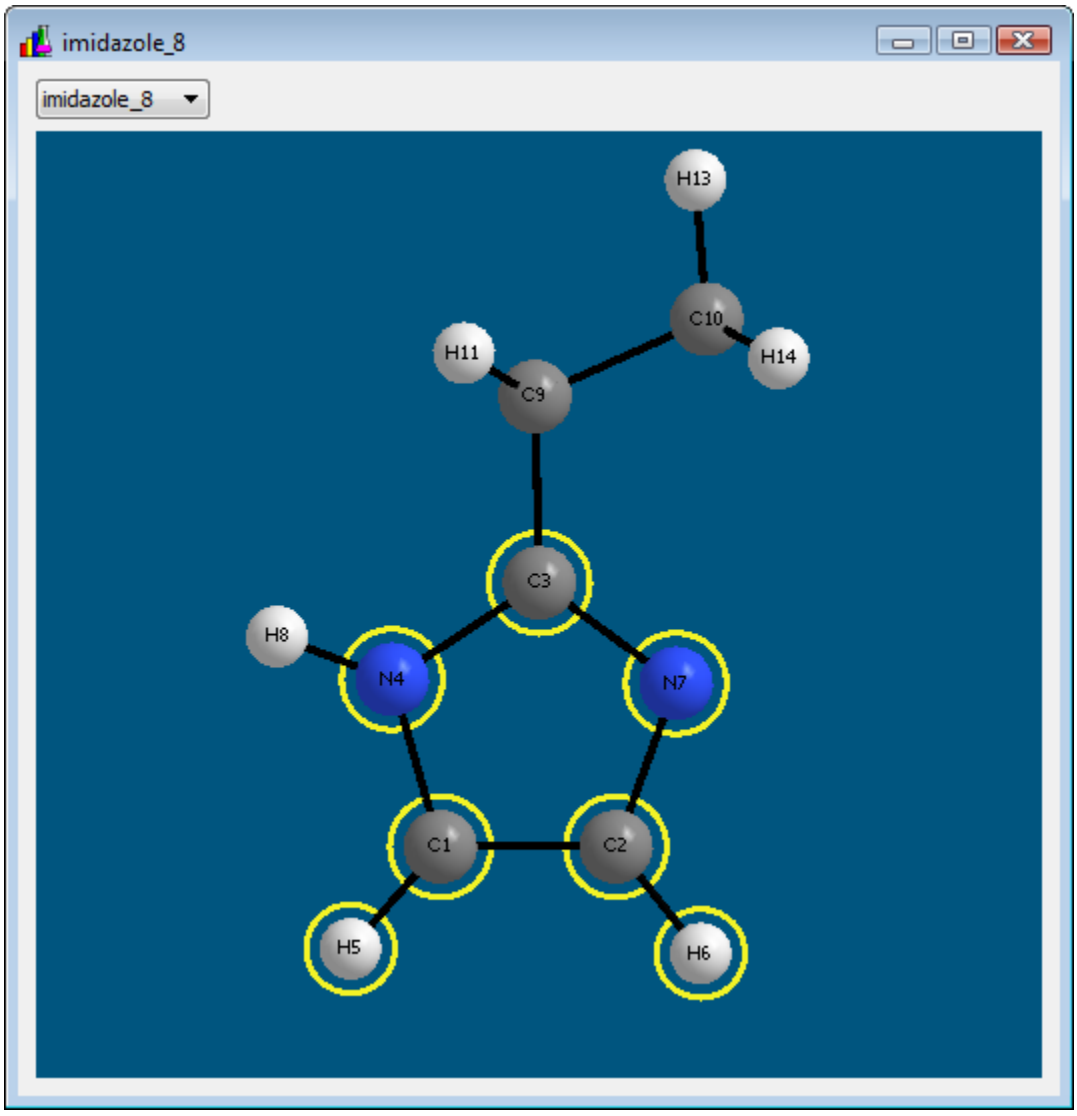
Step 13. In the "Fragment String of Selected Atoms" dialog, click the "Make New Fragment" button.



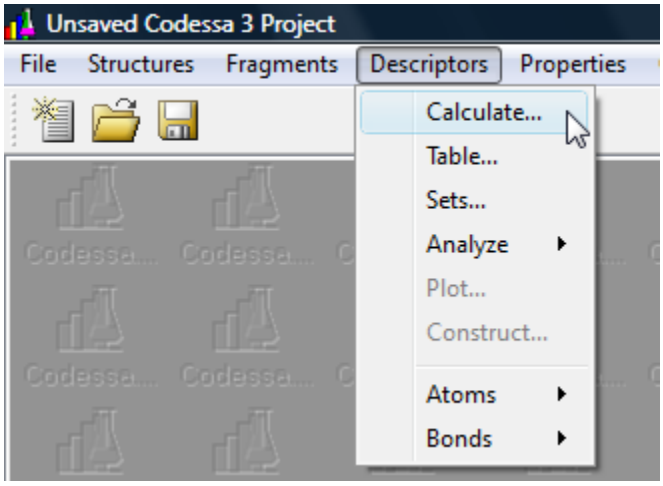
Step 14. In the "New Fragment" dialog, enter "Imidazole Ring" (without the quotes) for the Fragment Name and "ImRing" (without the quotes) for the Fragment Symbol, then click the "Ok" button.



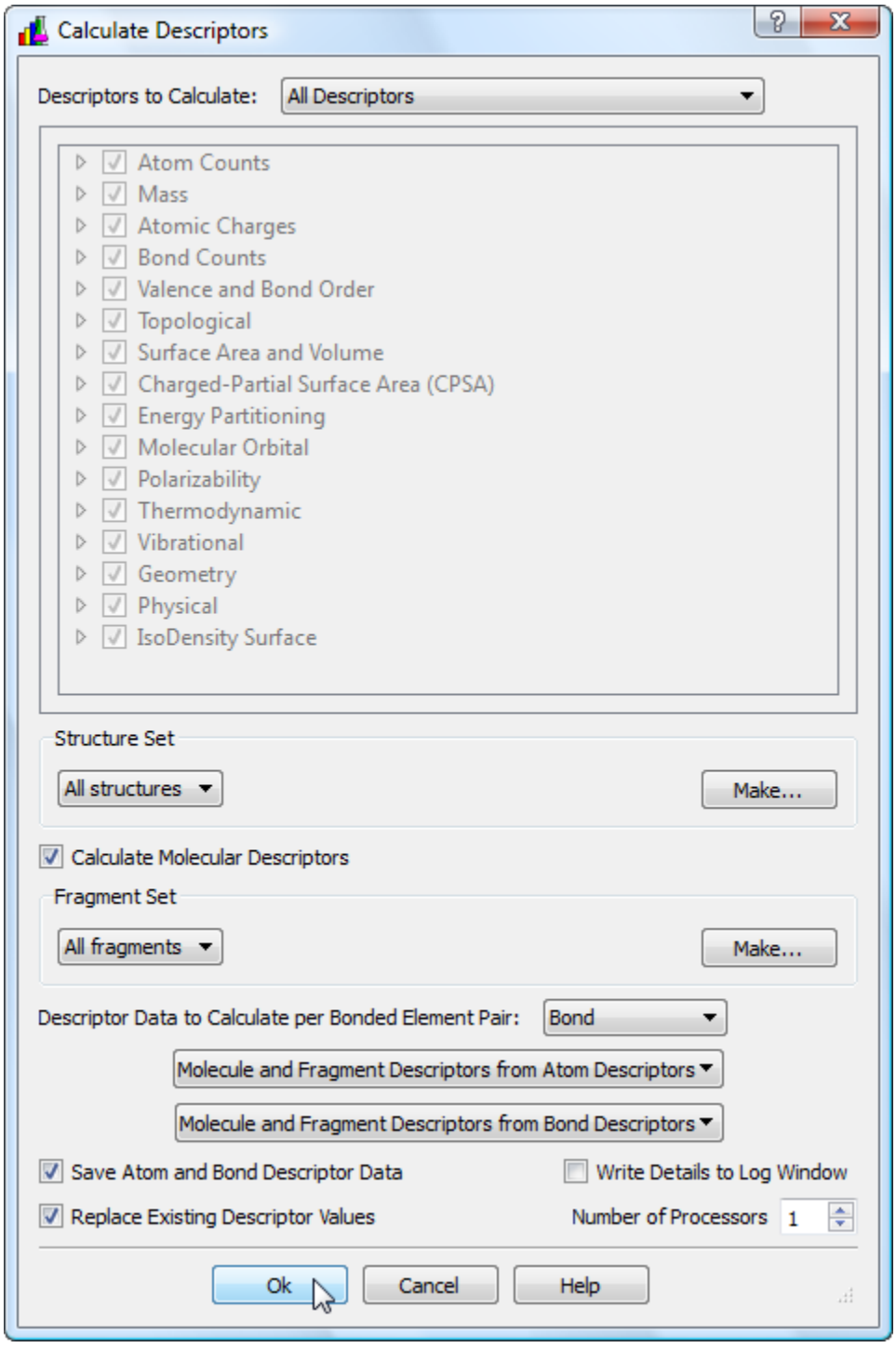
Step 15. Go back to the 3D window of structures and flip through each of the 15 structures and note that the atoms of the just defined "Imidazole Ring" fragment are highlighted (yellow rings) for each structure. When you are done inspecting the structures and their Imidazole Ring fragments, minimize or close the 3D window.



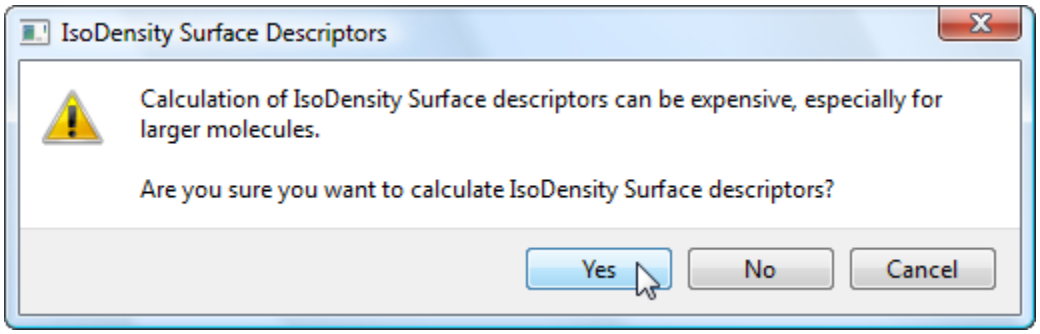
Step 16. Select the "Descriptors->Calculate..." menu item.



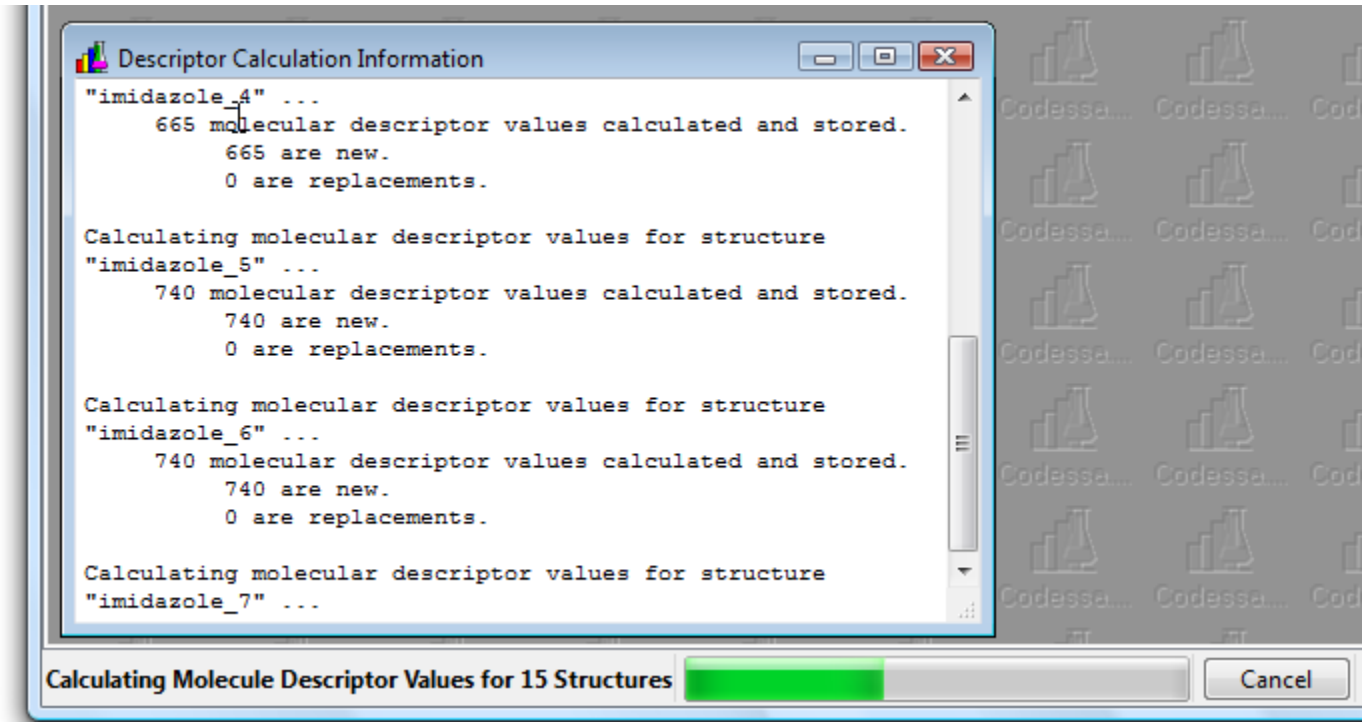
Step 17. In the "Calculate Descriptors" dialog, select "All fragments" for the Fragment Set and then click the "Ok" button to begin calculation of molecule descriptor values and fragment descriptor values for all 15 structures.



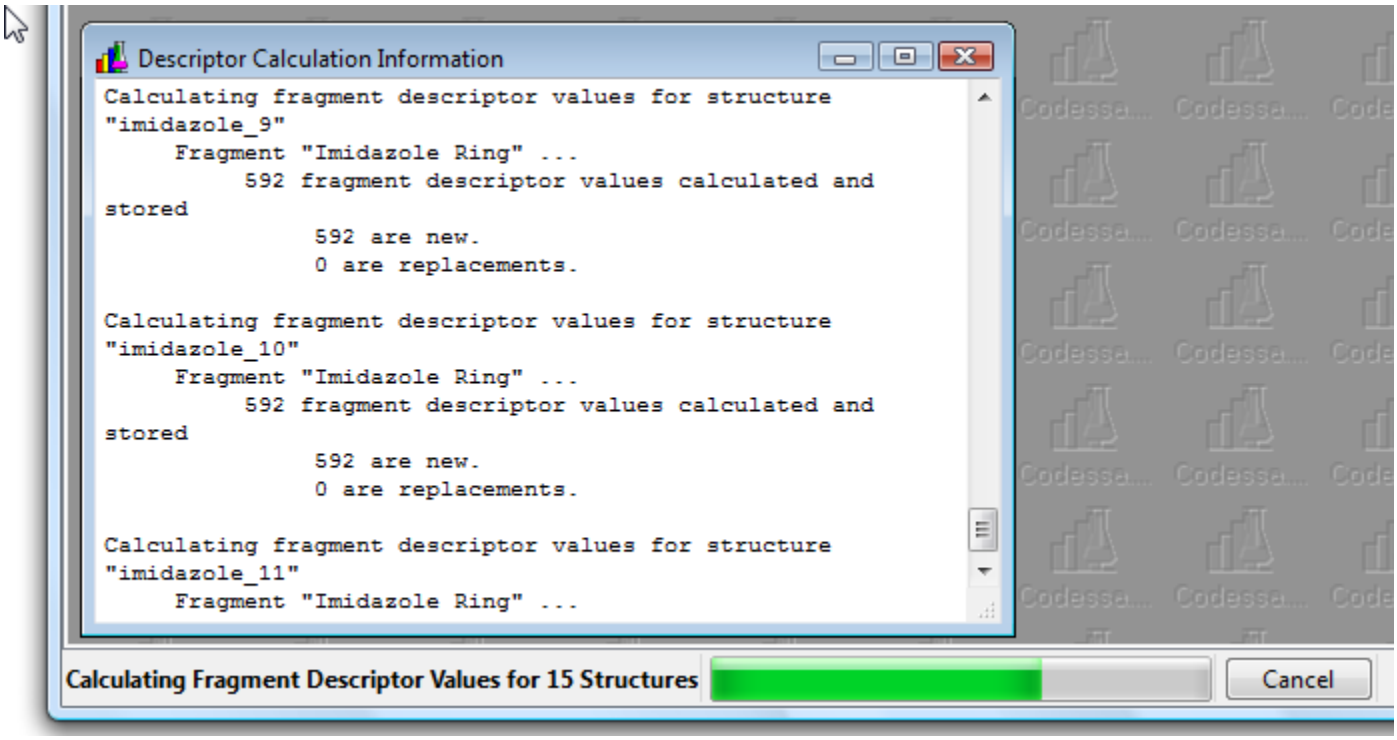
Step 18. In the "Isodensity Surface Descriptors" dialog, click the "Yes" button.



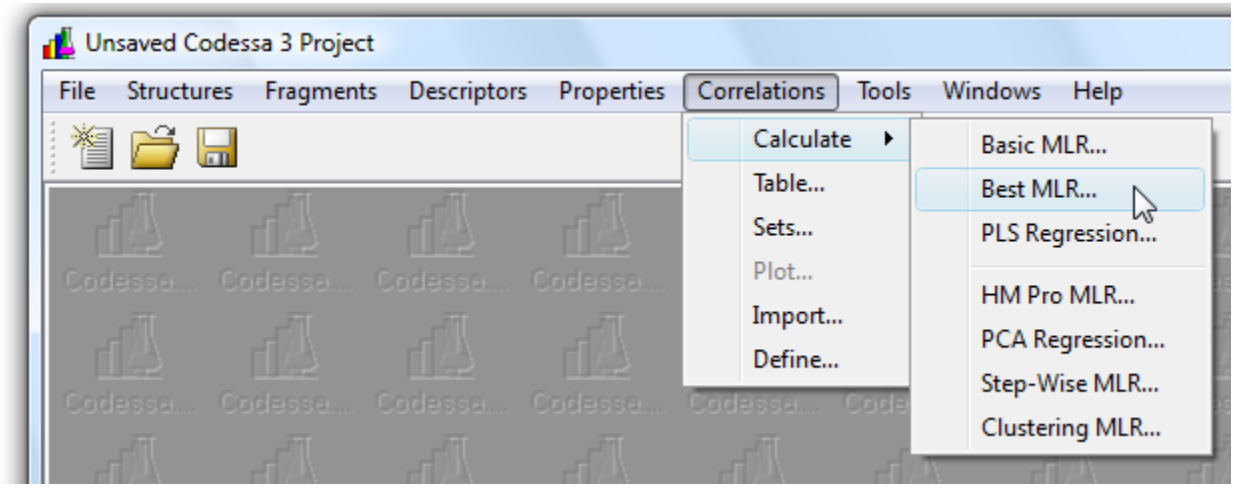
Step 19. Wait for the molecule descriptor values and fragment descriptor values to be calculated for all 15 structures. This will take a few minutes. The progress of the descriptor value calculations is shown in the "Descriptor Calculation Information" window and in the main window progress bar.



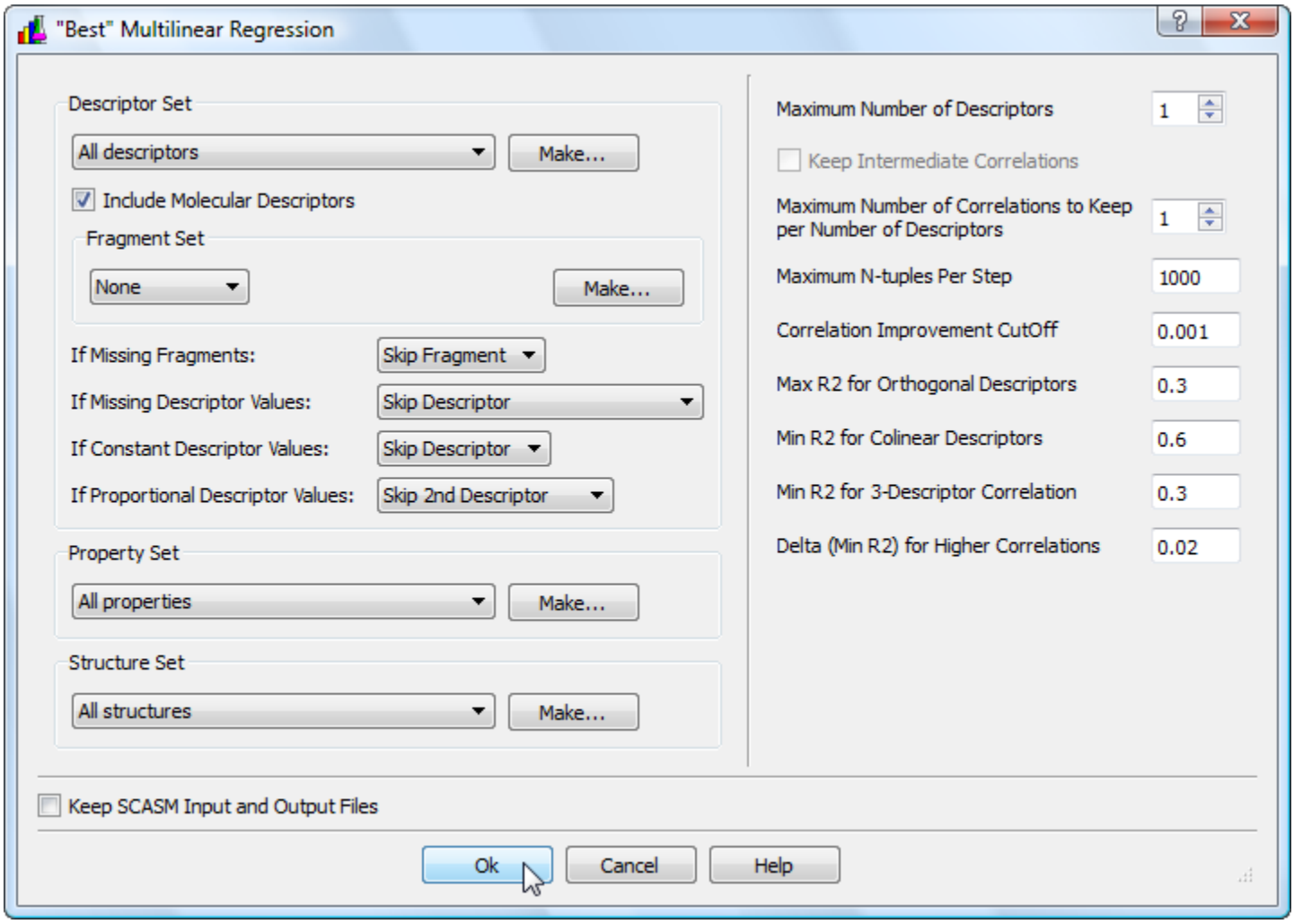
 Revision Date: July 8, 2013



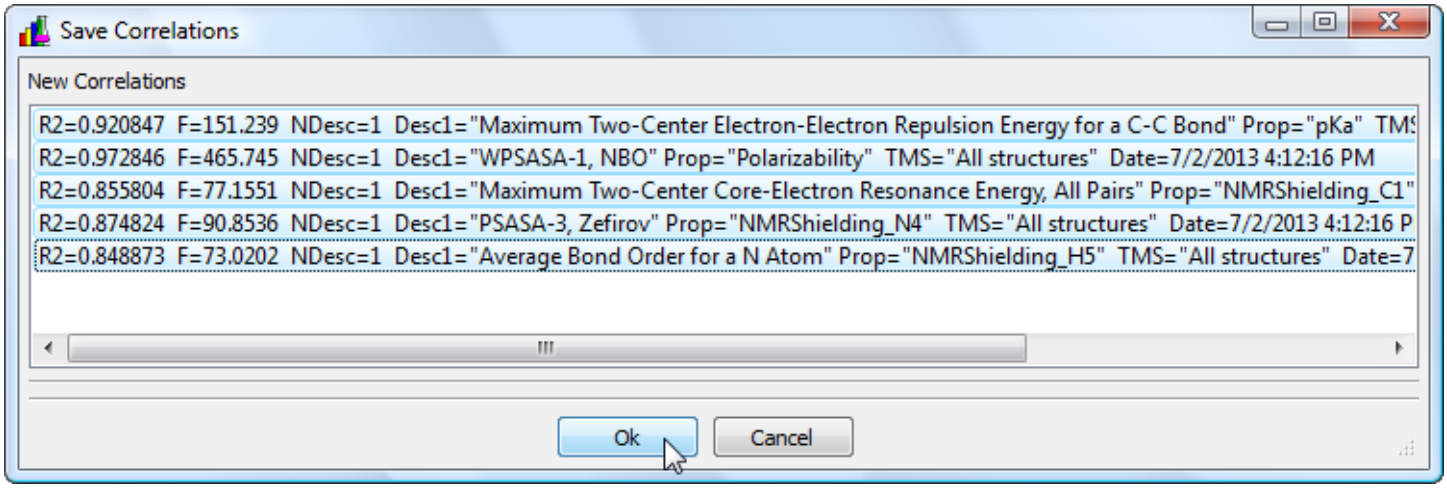
Step 20. Once the molecule and fragment descriptor values are all calculated, select the "Correlations->Calculate->Best MLR" menu item.



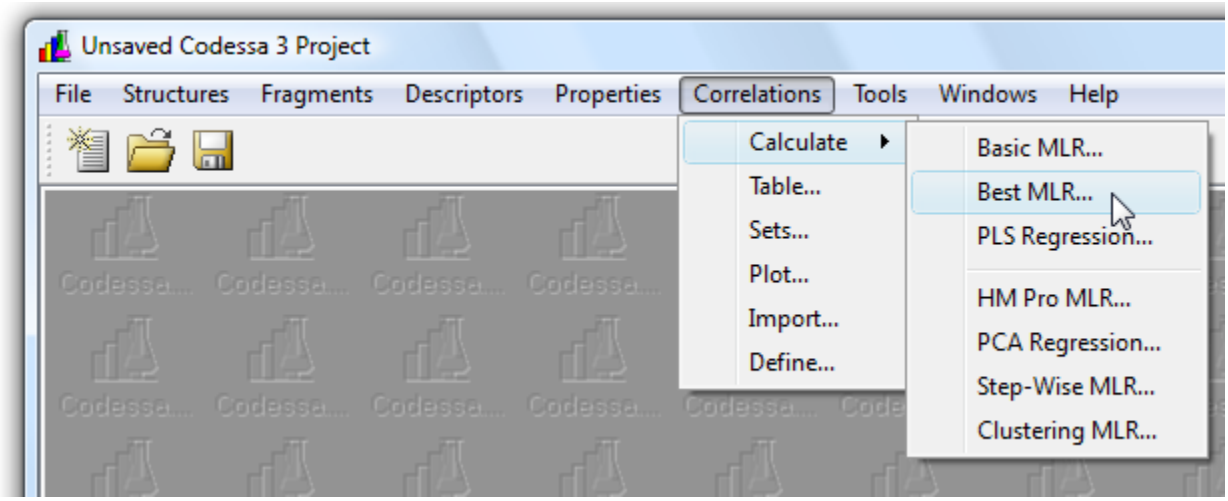
Step 21. In the ""Best" Multilinear Regression" dialog, accept the default settings. "Include Molecule Descriptors" should be checked and the Fragment Set should be "None". Then click "Ok" to begin the calculation to find, for each of the 5 properties, the descriptor whose molecular values correlate best with the property values for the set of all 15 structures.



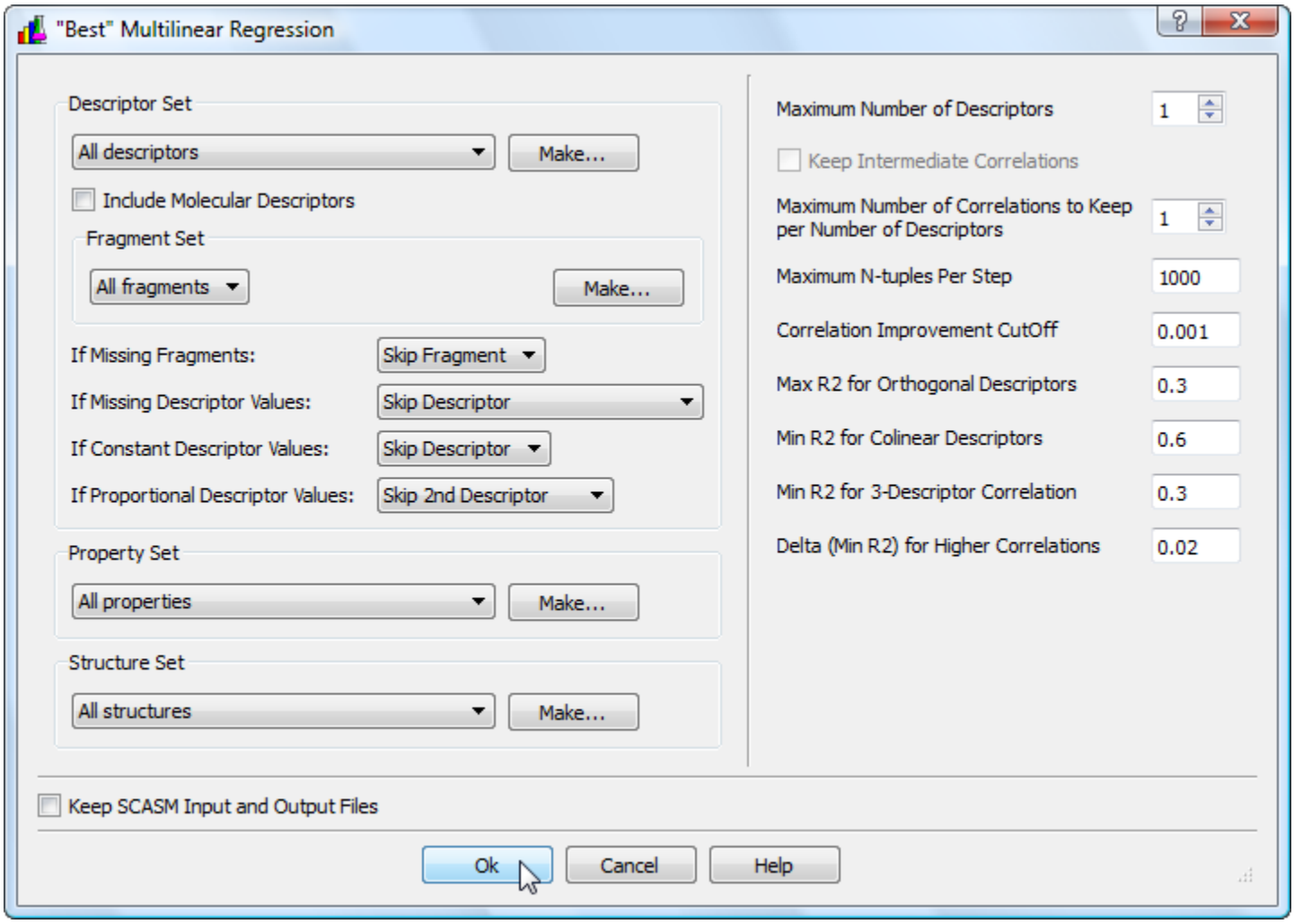
Step 22. When the calculation of the best correlation is completed for each of the 5 properties, a "Save Correlations" dialog will appear. Select all 5 correlations and then click Ok.



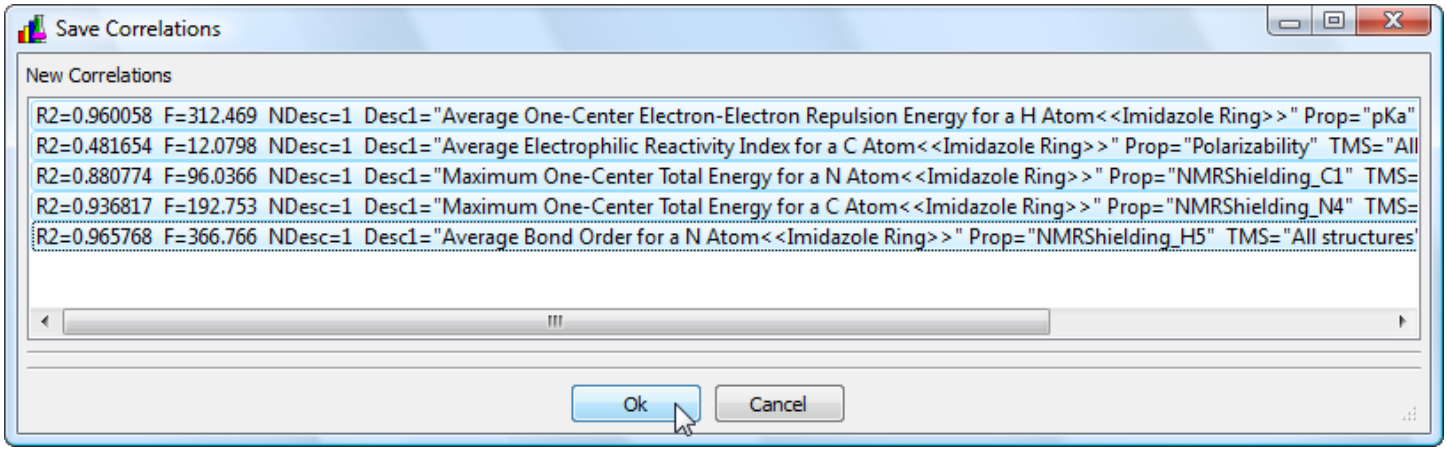
Step 23. Select the "Correlations->Calculate->Best MLR" menu item again.



Step 24. In the ""Best" Multilinear Regression" dialog, uncheck the"Include Molecule Descriptors" box and change the Fragment Set to "All fragments". Then click "Ok" to begin the calculation to find, for each of the 5 properties, the descriptor whose fragment values correlate best with the property values for all 15 structures.

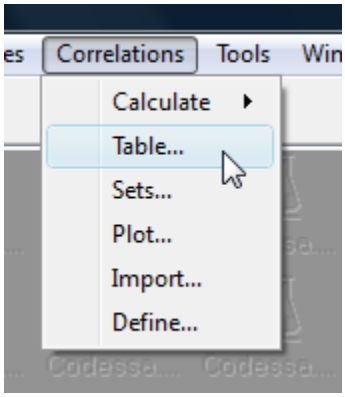


Step 25. When the calculation of the best correlation is completed for each of the 5 properties, a "Save Correlations" dialog will appear. Select all 5 correlations and then click Ok.



 Revision Date: July 8, 2013

Step 26. At this point we have 10 correlations, the first 5 involving molecule descriptor values and the last 5 involving fragment descriptor values, where the Fragment is the Imidazole Ring. It is time to compare these correlations. Select "Correlations->Table".



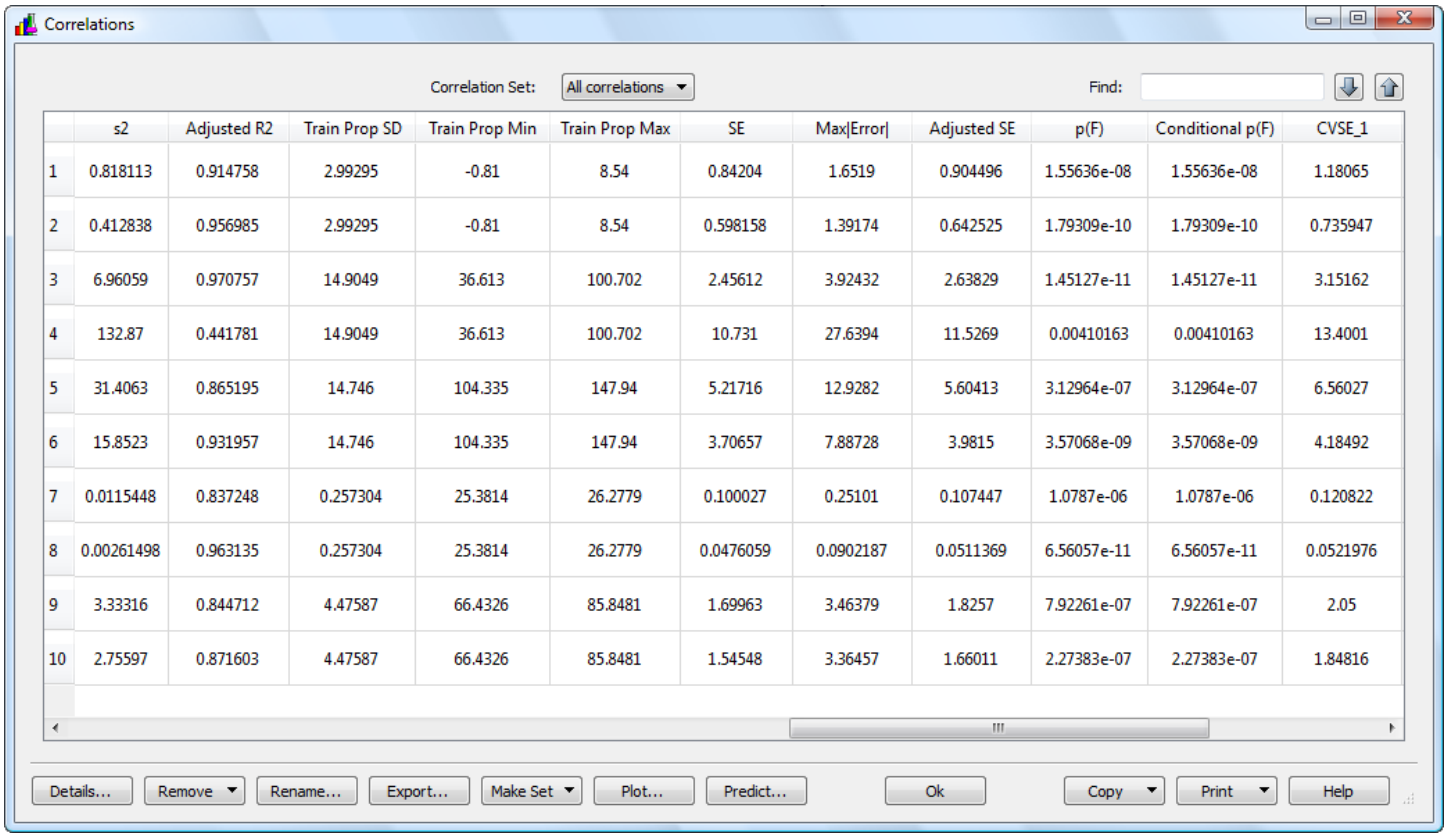
Step 27. In the "Correlations" table, click twice on the "Property" column header to sort the correlations by Property name in descending (alphabetically in this case) order. Note that properties that are expected to have "localized" descriptions from the Imidazole Ring (pKa for the Imidazole Ring nitrogen and the NMR shielding of nuclei in the Imidazole Ring) have better correlations with the Imidazole Ring fragment descriptor values than with the whole molecule descriptor values. The Polarizability property is an extensive property and thus it correlates much better with the best molecule descriptor value (Volume) than with the best fragment descriptor value.

Correlations

Correlation Set: All correlations Find:

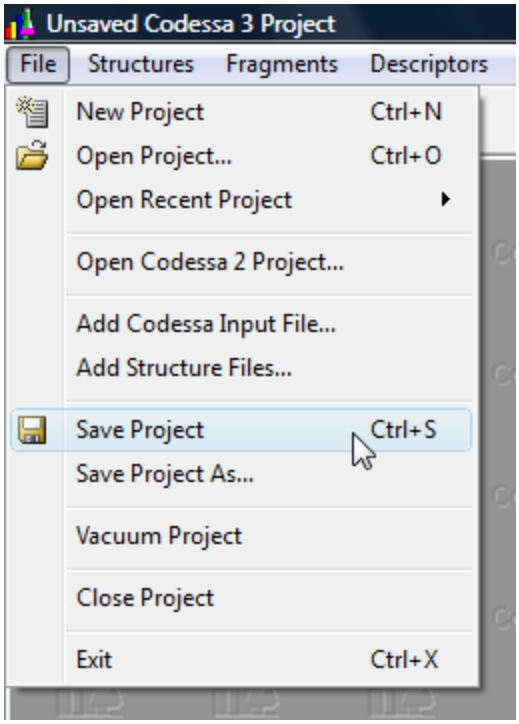
	DBID	Name	Method	Property	NDesc	Descriptor Names	R2	CVR2_1	F
1	0	R2=0.920847 F=151.239 NDe...	Best MLR	pKa	1	Maximum Two-Center Electron-Electron Repulsion Energy for a C-C Bond	0.920847	0.844389	151.239
2	5	R2=0.960058 F=312.469 NDe...	Best MLR	pKa	1	Average One-Center Electron-Electron Repulsion Energy for a H Atom<<Imidazole Ring>>	0.960058	0.939536	312.469
3	1	R2=0.972846 F=465.745 NDe...	Best MLR	Polarizability	1	WPSASA-1, NBO	0.972846	0.95529	465.745
4	6	R2=0.481654 F=12.0798 NDe...	Best MLR	Polarizability	1	Average Electrophilic Reactivity Index for a C Atom<<Imidazole Ring>>	0.481654	0.191732	12.0798
5	3	R2=0.874824 F=90.8536 NDe...	Best MLR	NMRShielding_N4	1	PSASA-3, Zefirov	0.874824	0.802077	90.8536
6	8	R2=0.936817 F=192.753 NDe...	Best MLR	NMRShielding_N4	1	Maximum One-Center Total Energy for a C Atom<<Imidazole Ring>>	0.936817	0.919457	192.753
7	4	R2=0.848873 F=73.0202 NDe...	Best MLR	NMRShielding_H5	1	Average Bond Order for a N Atom	0.848873	0.779507	73.0202
8	9	R2=0.965768 F=366.766 NDe...	Best MLR	NMRShielding_H5	1	Average Bond Order for a N Atom<<Imidazole Ring>>	0.965768	0.958846	366.766
9	2	R2=0.855804 F=77.1551 NDe...	Best MLR	NMRShielding_C1	1	Maximum Two-Center Core-Electron Resonance Energy, All Pairs	0.855804	0.790226	77.1551
10	7	R2=0.880774 F=96.0366 NDe...	Best MLR	NMRShielding_C1	1	Maximum One-Center Total Energy for a N Atom<<Imidazole Ring>>	0.880774	0.8295	96.0366

Details... Remove Rename... Export... Make Set Plot... Predict... Ok Copy Print Help



	s2	Adjusted R2	Train Prop SD	Train Prop Min	Train Prop Max	SE	Max|Error|	Adjusted SE	p(F)	Conditional p(F)	CVSE_1
1	0.818113	0.914758	2.99295	-0.81	8.54	0.84204	1.6519	0.904496	1.55636e-08	1.55636e-08	1.18065
2	0.412838	0.956985	2.99295	-0.81	8.54	0.598158	1.39174	0.642525	1.79309e-10	1.79309e-10	0.735947
3	6.96059	0.970757	14.9049	36.613	100.702	2.45612	3.92432	2.63829	1.45127e-11	1.45127e-11	3.15162
4	132.87	0.441781	14.9049	36.613	100.702	10.731	27.6394	11.5269	0.00410163	0.00410163	13.4001
5	31.4063	0.865195	14.746	104.335	147.94	5.21716	12.9282	5.60413	3.12964e-07	3.12964e-07	6.56027
6	15.8523	0.931957	14.746	104.335	147.94	3.70657	7.88728	3.9815	3.57068e-09	3.57068e-09	4.18492
7	0.0115448	0.837248	0.257304	25.3814	26.2779	0.100027	0.25101	0.107447	1.0787e-06	1.0787e-06	0.120822
8	0.00261498	0.963135	0.257304	25.3814	26.2779	0.0476059	0.0902187	0.0511369	6.56057e-11	6.56057e-11	0.0521976
9	3.33316	0.844712	4.47587	66.4326	85.8481	1.69963	3.46379	1.8257	7.92261e-07	7.92261e-07	2.05
10	2.75597	0.871603	4.47587	66.4326	85.8481	1.54548	3.36457	1.66011	2.27383e-07	2.27383e-07	1.84816

Step 28. Select the "File->Save Project" menu item to save the project to a file.



Step 29. In the "Specify a Codessa 3 Project File" file save dialog, specify a name for the project, e.g., "ampac_vis.cod3", and click the "Save" button.