Jmol SQL for DSSR

Jmol supports a rich query language for exploring DSSR structure annotations as well as selecting atoms associated with specific DSSR characteristics.

-- The _M.dssr associative array

After loading a file from the PDB with the /dssr attribute:

```
LOAD  =1ehz/dssr
```

The associative array _M.dssr holds the DSSR information. This array has the following main keys and subkeys of interest to this discussion:

bulges
coaxStacks
coaxStacks.stems
hairpins
hbonds
helicies
helicies.pairs
iloops
isoCanonPairs
junctions
kissingLoops
kissingLoops.hairpins
multiplets
nonStack
nts
pairs
ssSegments
stacks
stems
stems.pairs

Each key or subkey is itself an array of associative arrays. For example,

```
LOAD =1ehz/dssr
PRINT _M.dssr.stems.length
```

4

```
PRINT _M.dssr.stems[1].pairs.length
```

7

PRINT _M.dssr.stems[1].pairs[1]

```
{
 "DSSR" : "cW-W"
 "LW" : "cWW"
 "Saenger" : "19-XIX"
 "bp" : "G-C"
 "index" : 1
 "name" : "WC"
 "nt1" : "|1|A|G|1||||"
 "nt2" : "|1|A|C|72||||"
}
```

-- Using .select()

Jmol SQL can target these values in order to extract specific array elements using the .select() function:

PRINT _M.dssr.pairs.select("where name='Imino'")

```
 {
      "DSSR" : "cW-W"
      "LW" : "cWW"
      "Saenger" : "08-VIII"
      "bp" : "g-A"
      "index" : 21
      "name" : "Imino"
      "nt1" : "|1|A|M2G|26||||"
      "nt2" : "|1|A|A|44||||"
 }
```

PRINT _M.dssr.pairs.select("where name != 'WC'").count

```
14
```

**PRINT _M.dssr.coaxStacks[1].stems.select("where strand1='GACAC' or
strand2='GACAC'")[1].pairs.count**

```
5
```

-- Involving Jmol variables

The expressions in these selections can use any Jmol math expression where the variables are the keys in the associative arrays being targeted. As such, they can contain references to other Jmol variables:

x = "GACAC"
PRINT _M.dssr.coaxStacks[1].stems.select("where strand1=x or strand2=x")[1].pairs.count

5

-- Unit IDs and atom selection

Atom selection in Jmol is based on identifying *unit ids.*
[ref: http://rna.bgsu.edu/main/rna-3d-hub-help/unit-ids]
Any data containing unit ids can be passed to the SELECT command, whether directly:

SELECT "|1|A|A|44||||"

-- Indirect atom selection

Atom selection can also be done indirectly, using @{...} notation with a math expression. In this case, we can use the _M.dssr array:

SELECT @{_M.dssr.pairs.select("where name != 'WC'")}

Note that the unit id is a string value, while _M.dssr.pairs is not. This is not an issue; Jmol will convert the array to its string value and find all unit ids present in that string.

Atom selection using within(dssr, ...)

In addition, atom selection from DSSR can be done using the within() atom selection method. This is the most efficient way to select atoms using DSSR information, and it, too, can use Jmol SQL. In this case, the notation is simplified. (The ON keyword is optional and simply adds visual highlighting to the selection).

SELECT ON within(dssr, "nts")
SELECT ON within(dssr, "nts[WHERE v0>17 and v1 <-39]")

Either with or without the WHERE option, within(dssr, ...) can select just one of the array elements returned by the selection. This is accomplished by appending *.n*, where *n* is an integer starting with 1. (Note that 0 requests the last array element in Jmol, not the first.)

SELECT ON within(dssr, "nts.2")
SELECT ON within(dssr, "nts[WHERE v0>1 and v1 <0].2")