The VirtualHelix Class

_3pBase   _5pBase

b._3pBase == a._5pBase == None

a._3pBase==b && b._5pBase==a

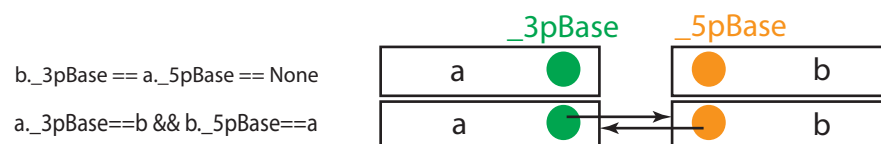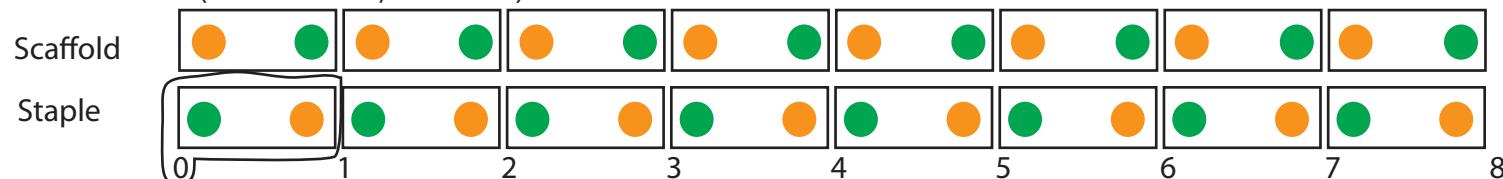| a | ● | | ● | b |
| a | ● → | | ← ● | b |

Follow along in cadnano2/model/virtualhelix_tutorial.py or run caDNAno using `python main.py -i` to get a debug console in caDNAno proper.

```
from model.enum import StrandType
vh = virtualHelix(numBases=8, idnum=0)
```

Scaffold

Staple

0   1   2   3   4   5   6   7   8

Bases are associated with the lesser
of the two possible indices, and can be
fetched (which you should never do unless
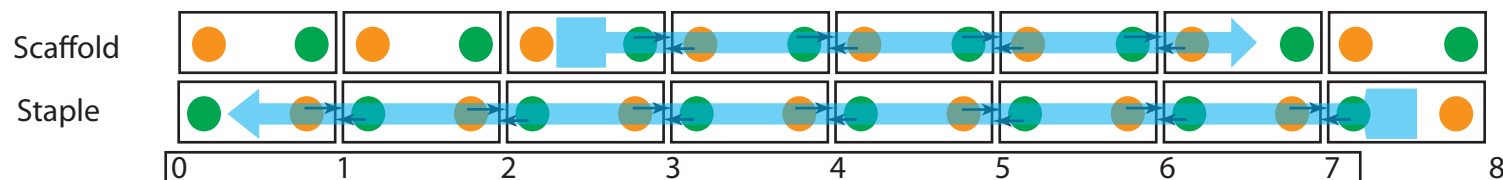self is a VirtualHelix) like
vh.strand(StrandType.Staple)[0]

Note that the signly linked list sometimes
runs against the indexes in the
vh.strand(strandType) array.

Modification

```
vh.connectStrand(StrandType.Scaffold, 2, 6)
vh.connectStrand(StrandType.Staple, 0, 7)
```
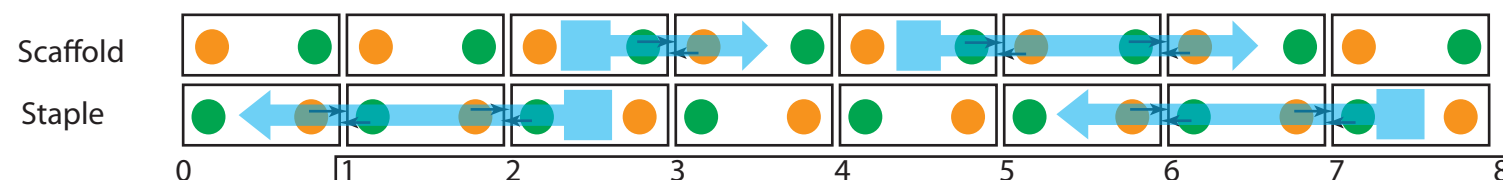
Inside caDNAno you should not have to do anything
in addition to calling these methods to get changes to propagate
to the graphical representations you can see and edit.

Scaffold

Staple

0   1   2   3   4   5   6   7   8
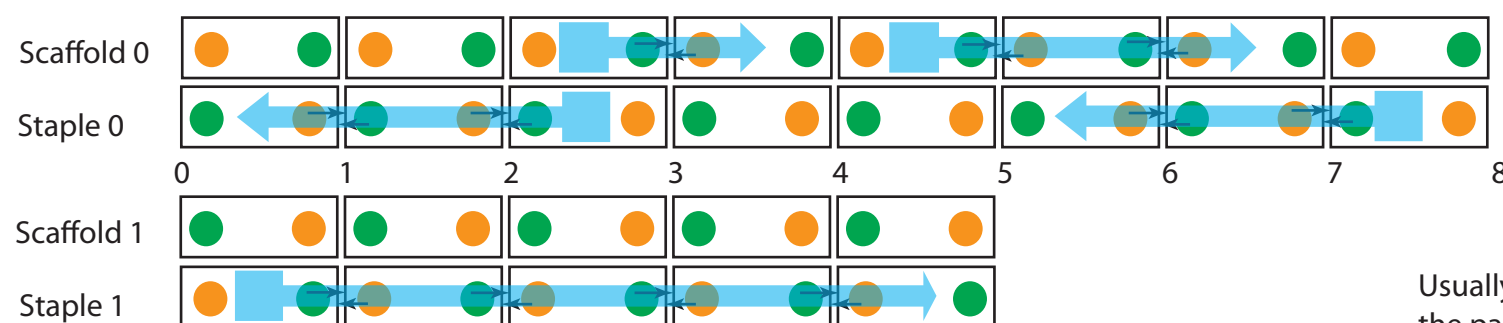
```
vh.clearStrand(StrandType.Scaffold, 3.5, 4.5)
vh.clearStrand(StrandType.Staple, 2.5, 5.5)
```

Incidentally, vh.clearStrand(StrandType.Staple(3, 5) would
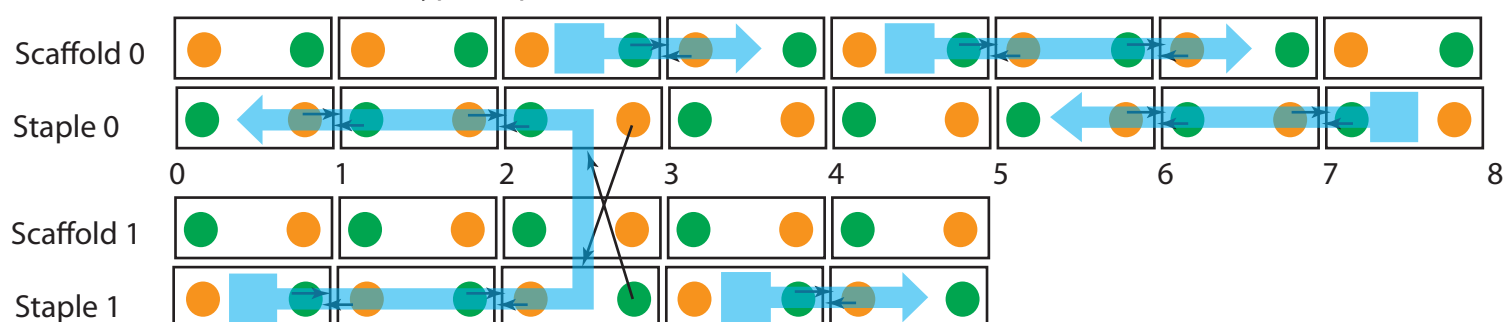have done the same thing since it's a doubly linked list.

Scaffold

Staple

0   1   2   3   4   5   6   7   8

```
vh1 = VirtualHelix(numBases=5, idnum=1)
vh1.connectStrand(StrandType.Staple, 0, 4)
```

Scaffold 0

Staple 0

0   1   2   3   4   5   6   7   8

Scaffold 1

Staple 1

```
vh.installXoverFrom3To5(StrandType.Staple, 2, vh1, 2)
```

Scaffold 0

Staple 0

0   1   2   3   4   5   6   7   8

Scaffold 1

Staple 1

```
vh.undoStack().undo()
vh.undoStack().undo()
```

```
vh.undoStack().undo()
vh.undoStack().undo()
```

```
vh1.undoStack().undo()
```

Usually vh.undoStack() gets
the parent document's undo
stack, but in test.py there is no
document so each vh will have a
different undo stack.

```
vh.undoStack().undo()
```