Raw Data:

Position: 1 2 3 4 5 6
Reference: A C T A G - - C

Sequences/Reads/each base in a read
Read 1: A C T A G A - C (Insertion with respect to reference @ pos 5)
Read 2: A C G A C - - T
Read 3: A C T A G A T C (2 Insertions with respect to reference @ pos 5)

Quality scores for each base
Position: 1 2 3 4 5 6
Read 1: 30,25,10,15,20,5,,25
Read 2: 30, 21,19,25,30,,,18
Read 3: 15,20,25,30,35,15,18,27

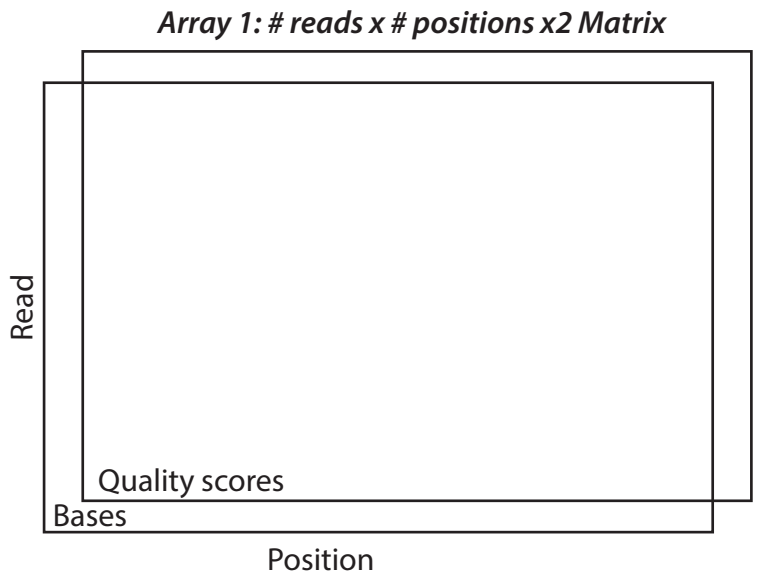
Representation 1: ***Dataset of 1 Matrix table, and 1 insertion table/array***

Create two data arrays:
* One array to store all of the letters and quality scores of the reads, ***but leave out insertions***
* Second array will keep track of location of insertions letters and quality scores

Benefits:
* Keep insertions as a seperate table accounts for the fact that it will be very sparse (very few reads and positions will have insertions)
* Quality scores are assocaited with each read in Array1

Cons:
* Will want to easily associate insertion locations with Array 1. Not sure if these types of merges would be effective/well designed for large datastructures (i.e. in pandas, they dont perform too well with large sizes)
* ***Quality score is optional,*** so if no quality scores are provided, then would be doubling the size of the matrix with empty data

Array 1: # reads x # positions x2 Matrix



Array 2: (total insertion events x 5)

Read	Position	Insertion	Letter	Quality
1	5	1	A	5
3	5	1	A	15
3	5	2	T	18

(MultiIndex)

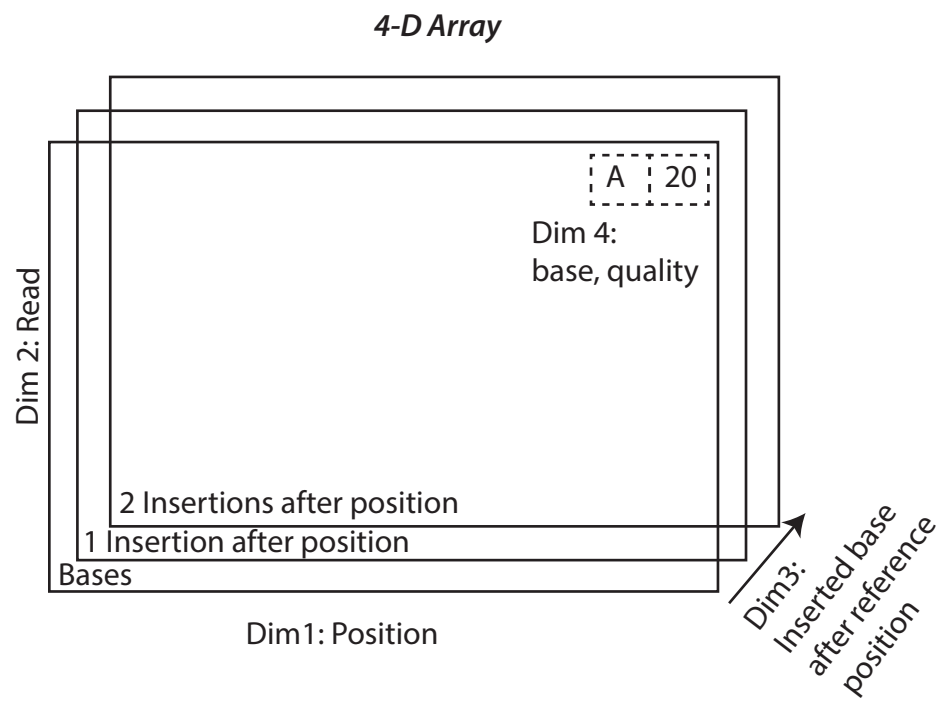
Representation 2: ***4-D Array***

Represent data as as single 4-D matrix
* Dim 1 = Coordinates represented by positions
* Dim 2 = Coordinates represented by reads
* Dim 3 = Coordinates represented by the maximum # of insertions observed within the entire set of
* Dim 4 = Coordinates = (base/letter, quality score)

Benefits:
* Every aspect of aligned sequence are linked by reference position and read. So very easy to visualize results and perform operations.

Cons:
* Dimensions 3 and 4 can be very sparse and will create unnecessarily large data structures to represent the data. For example:
  * Dont include quality scores, then size of matrix doubles with zeros in second coordinate of Dim 4
  * Have a million reads and only one of those reads has 10 insertions, then have created 10 extra matrices of zeros to represent a single read only

4-D Array



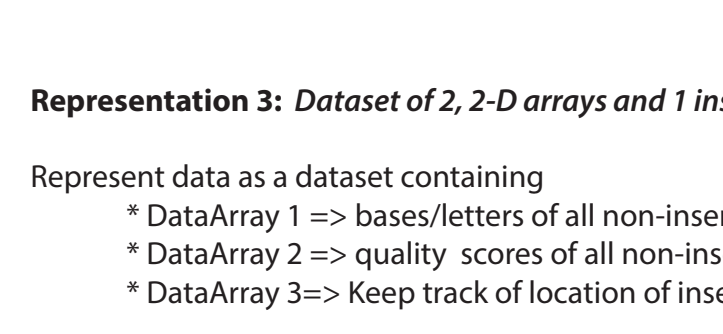
Representation 3: ***Dataset of 2, 2-D arrays and 1 insertion table/array***

Represent data as a dataset containing
* DataArray 1 => bases/letters of all non-inserted positions
* DataArray 2 => quality scores of all non-inserted positions
* DataArray 3=> Keep track of location of insertions letters and quality scores

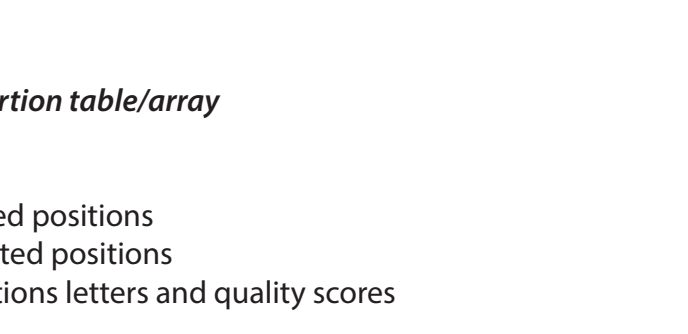
Benefits:
* Accounts for sparsity in DataArray2 and 3 (will not store data if user does not provide quality or insertions)

Cons:
* Harder for user to link together:
  * Letters
  * Quality scores
  * Insertions

Array 1: # reads x # positions



Array 2: # reads x # positions
IF PROVIDED (0x0 if empty)



Array 3: (total insertion events x 5)

Read	Position	Insertion	Letter	Quality
1	5	1	A	5
3	5	1	A	15
3	5	2	T	18

(MultiIndex)