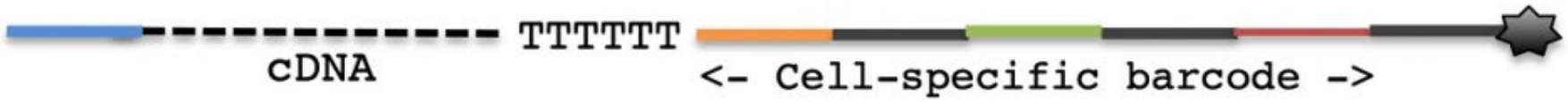
SPLiT-seq barcodes

Three rounds of barcoding (96 barcodes used for each round):

AACGTGAT	AACGTGAT	AACGTGAT
AAACATCG	AAACATCG	AAACATCG
ATGCCTAA	ATGCCTAA	ATGCCTAA
AGTGGTCA	AGTGGTCA	AGTGGTCA
...	...	...
ACATTGGC	ACATTGGC	ACATTGGC

442,368 x AGTGGTCAAGTGGTCAAAACATCG

Read 2: 94 nucleotides



BC(11-18,49-56,87-94)

AGTGGTCAAGTGGTCA**GGGGGGGG**
GGGGGGGGGGGGGGGGGGGGG

- oligo(dT)
- Random hexamers

Reads the lists of barcodes used in each round from .txt files

```
list1 = barcodes1.read().split("\n")
list2 = barcodes2.read().split("\n")
list3 = barcodes3.read().split("\n")
```

```
all_list = [list1, list2, list3]
```

Combinatoric iterators: `itertools.product(*all_list)`

Output the final barcode sequences to .txt file

Global Pairwise sequence alignment

```
SeqIO.parse("seq1.fasta", "fasta"))
SeqIO.parse("seq2.fasta", "fasta"))
```

	A	C	G	T
A	0	5	2	5
C	5	0	5	2
G	2	5	0	5
T	5	2	5	0

Gap cost: 5

```
for i in range(1, m+1):
    for j in range(1, n+1):
        Table[i][j] = min(
            Table[i][j - 1] + gap,
            Table[i - 1][j] + gap,
            Table[i - 1][j - 1] +
            cost[tags[str2[i - 1]], tags[str1[j - 1]]]
        )
```

Alignment graph:

```
        A  A  T  A  A  T
   [ 0  5 10 15 20 25 30]
A  [ 5  0  5 10 15 20 25]
A  [10  5  0  5 10 15 20]
G  [15 10  5  5  7 12 17]
G  [20 15 10 10  7  9 14]
C  [25 20 15 12 12 12 11]
A  [30 25 20 17 12 12 16]
```

Optimal cost:
16

Optimal alignment:
AATAAT-
AA-GGCA

Global Pairwise sequence alignment

```
SeqIO.parse("seq1.fasta", "fasta"))
SeqIO.parse("seq2.fasta", "fasta"))
```

	A	C	G	T
A	0	5	2	5
C	5	0	5	2
G	2	5	0	5
T	5	2	5	0

Gap cost: 5

```
for i in range(1, m+1):
    for j in range(1, n+1):
        Table[i][j] = min(
            Table[i][j - 1] + gap,
            Table[i - 1][j] + gap,
            Table[i - 1][j - 1] +
            cost[tags[str2[i - 1]], tags[str1[j - 1]]]
        )
```


Alignment graph:

```
        A  A  T  A  A  T
   [ 0  5 10 15 20 25 30]
A  [ 5  0  5 10 15 20 25]
A  [10  5  0  5 10 15 20]
G  [15 10  5  5  7 12 17]
G  [20 15 10 10  7  9 14]
C  [25 20 15 12 12 12 11]
A  [30 25 20 17 12 12 16]
```

Optimal cost:
16

Optimal alignment:
AATAAT-
AA-GGCA