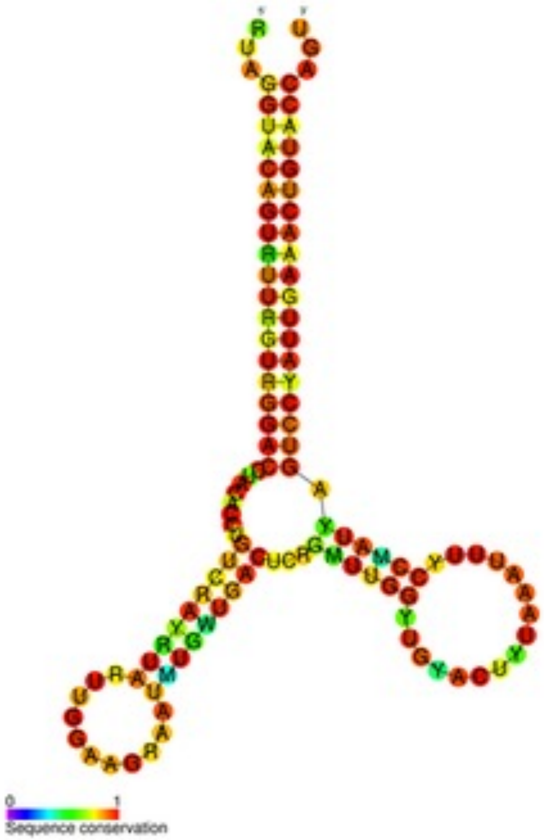
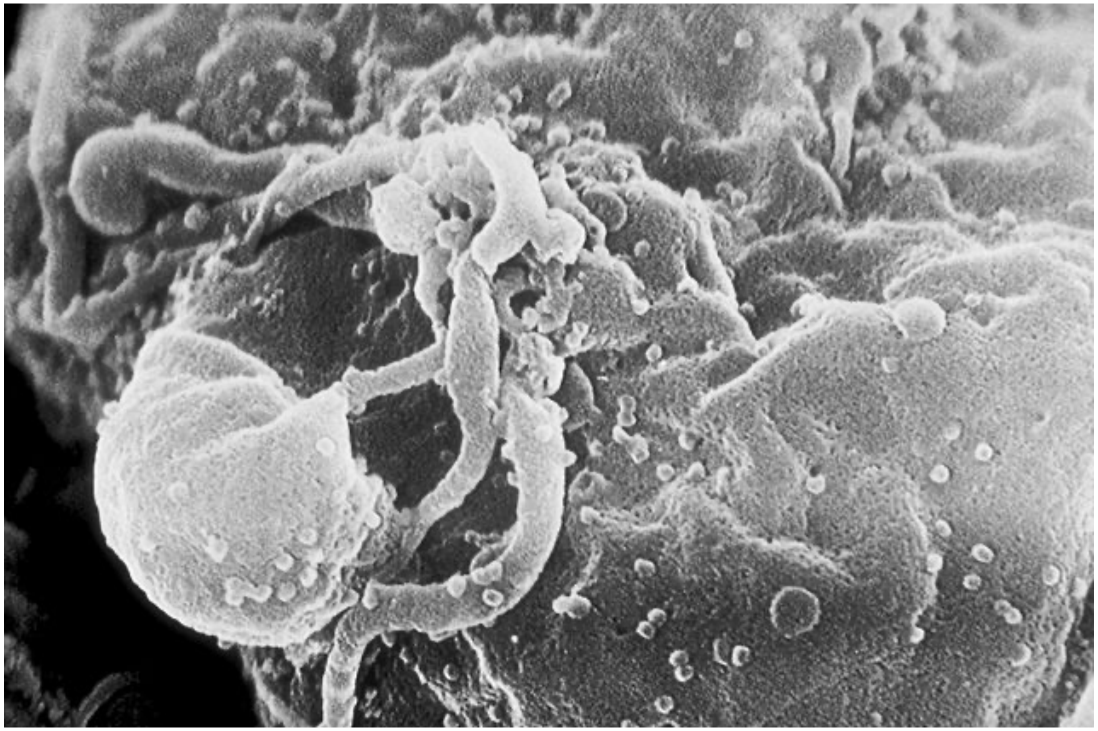
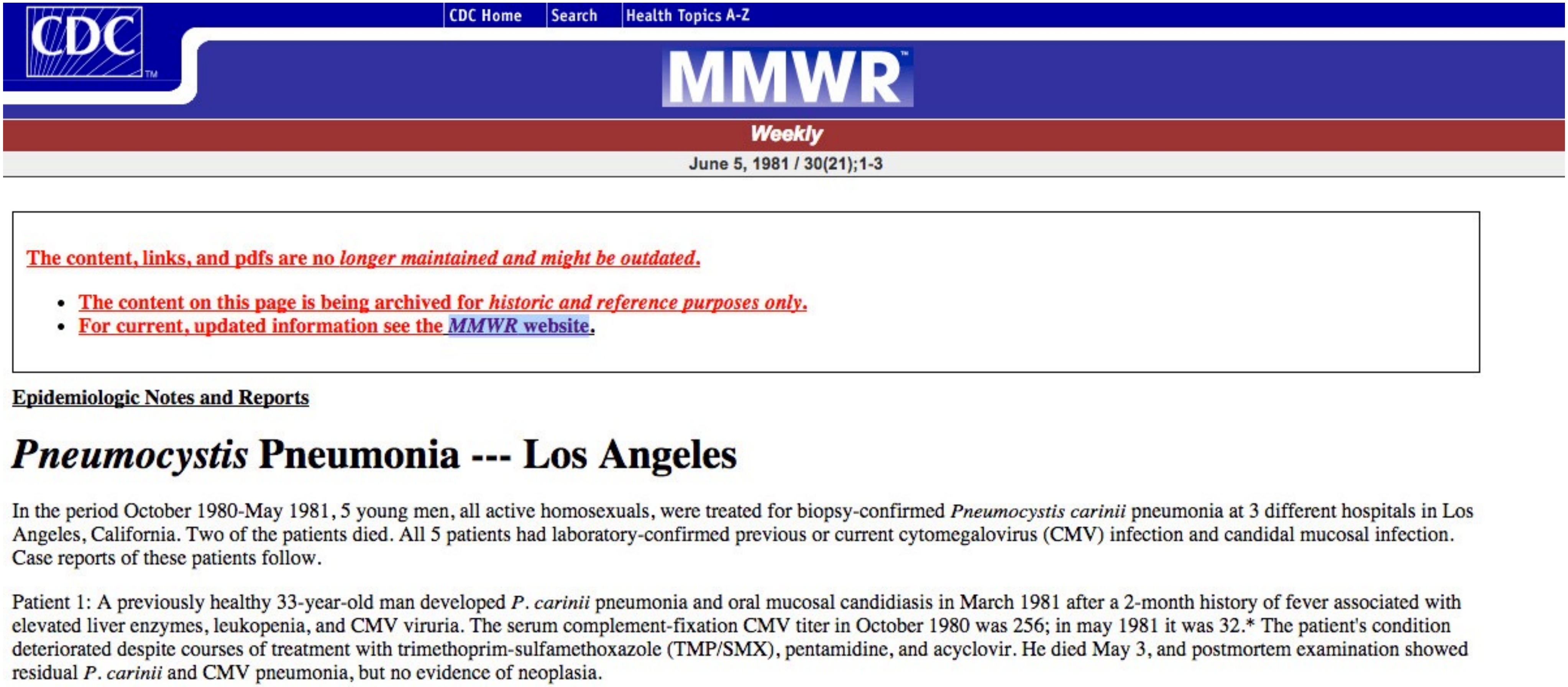
RNA secondary structure prediction





"HIV-budding" by Photo Credit: C. GoldsmithContent Providers: CDC/ C. Goldsmith, P. Feorino, E. L. Palmer, W. R. McManus
"RF01418". Licensed under Public domain via Wikimedia Commons -
http://commons.wikimedia.org/wiki/File:RF01418.png#mediaviewer/File:RF01418.png

Acquired immune deficiency syndrome



MMWR

Weekly

June 5, 1981 / 30(21);1-3

The content, links, and pdfs are no *longer maintained and might be outdated.*

- The content on this page is being archived for *historic and reference purposes only.*
- For current, updated information see the *MMWR* website.

Epidemiologic Notes and Reports

Pneumocystis Pneumonia --- Los Angeles

In the period October 1980-May 1981, 5 young men, all active homosexuals, were treated for biopsy-confirmed *Pneumocystis carinii* pneumonia at 3 different hospitals in Los Angeles, California. Two of the patients died. All 5 patients had laboratory-confirmed previous or current cytomegalovirus (CMV) infection and candidal mucosal infection. Case reports of these patients follow.

Patient 1: A previously healthy 33-year-old man developed *P. carinii* pneumonia and oral mucosal candidiasis in March 1981 after a 2-month history of fever associated with elevated liver enzymes, leukopenia, and CMV viruria. The serum complement-fixation CMV titer in October 1980 was 256; in may 1981 it was 32.* The patient's condition deteriorated despite courses of treatment with trimethoprim-sulfamethoxazole (TMP/SMX), pentamidine, and acyclovir. He died May 3, and postmortem examination showed residual *P. carinii* and CMV pneumonia, but no evidence of neoplasia.

Morbidity and Mortality Weekly Report from CDC, June 5 1981.

http://www.cdc.gov/mmwr/preview/mmwrhtml/june_5.htm

Science 20 May 1983

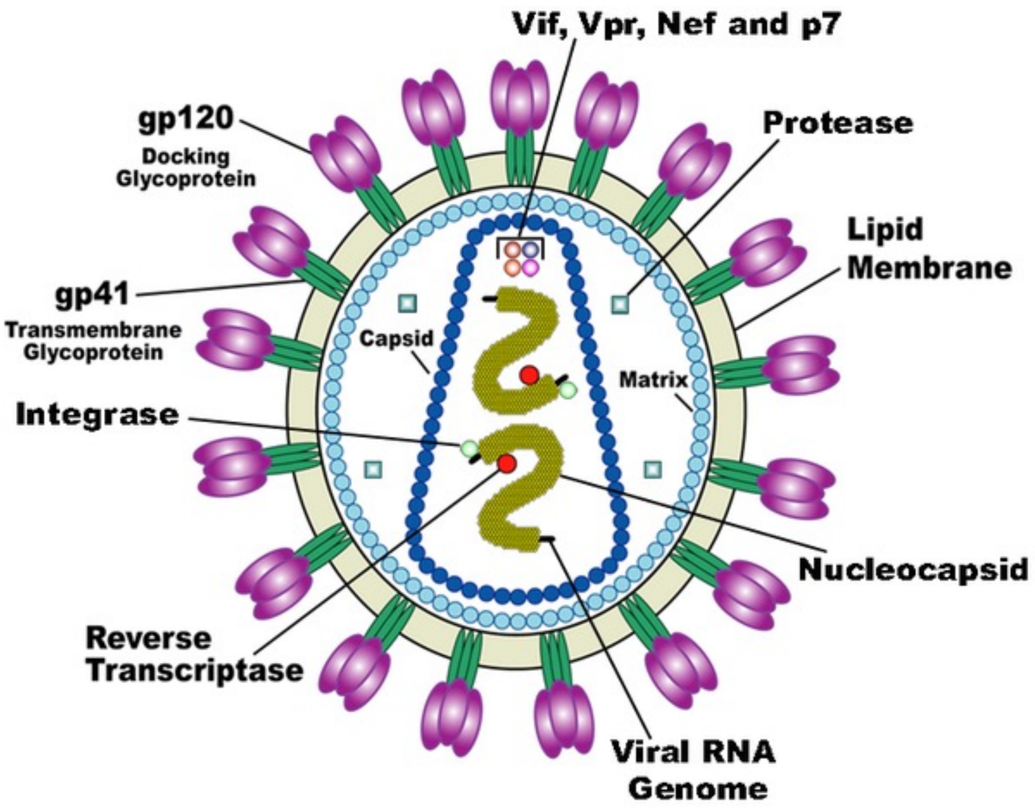
Isolation of human T-cell leukemia virus in acquired immune deficiency syndrome (AIDS)

RC Gallo, PS Sarin, EP Gelmann, M Robert-Guroff, E Richardson, VS Kalyanaraman, D Mann, GD Sidhu, RE Stahl, S Zolla-Pazner, J Leibowitch, M Popovic

Isolation of a T-lymphotropic retrovirus from a patient at risk for acquired immune deficiency syndrome (AIDS)

F Barre-Sinoussi, JC Chermann, F Rey, MT Nugeyre, S Chamaret, J Gruest, C Dauguet, C Axler-Blin, F Vezinet-Brun, C Rouzioux, W Rozenbaum, L Montagnier

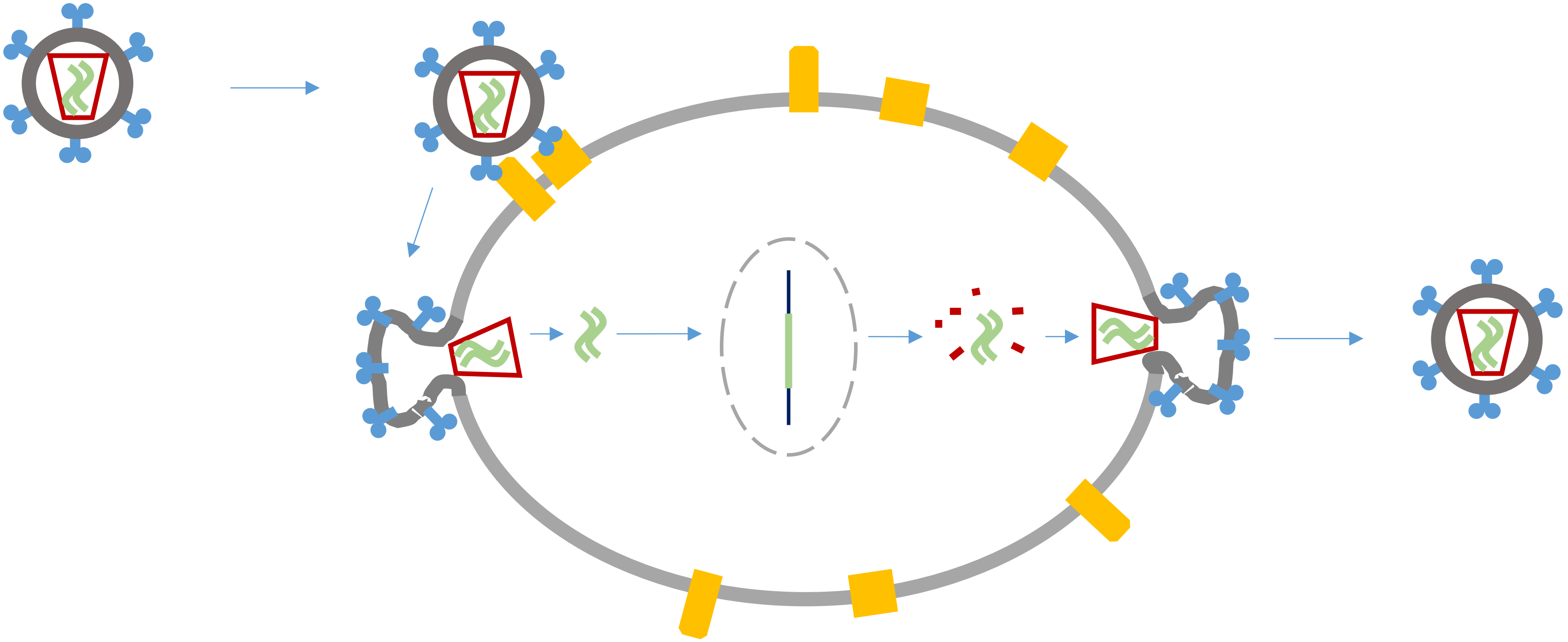
The HIV virus



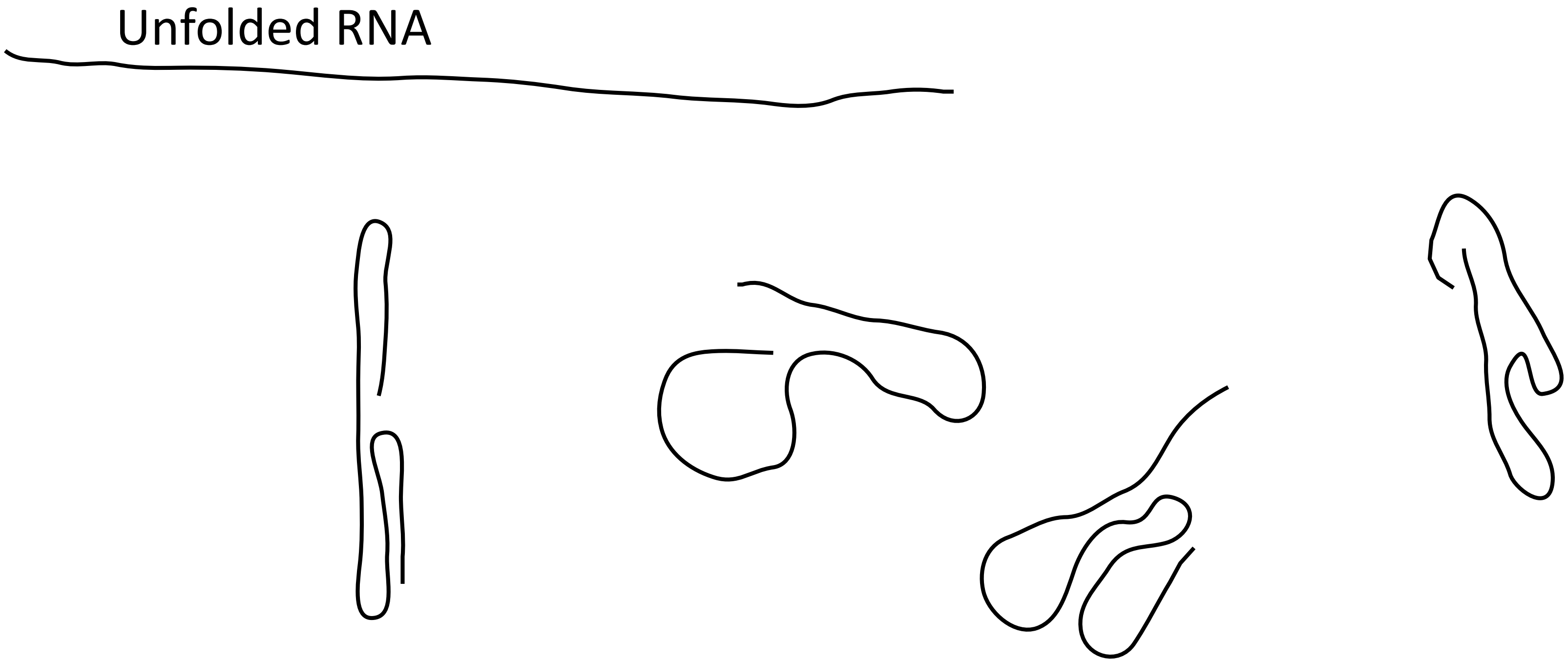
- RNA genome
- 9 genes
- The pol gene encodes reverse transcriptase (and more), a key protein in the cycle of HIV infection

"HIV Virion-en" by US National Institute of Health (redrawn by Carl Henderson)
http://en.wikipedia.org/wiki/HIV#mediaviewer/File:HIV_Virion-en.png

The HIV replication cycle and the pol gene

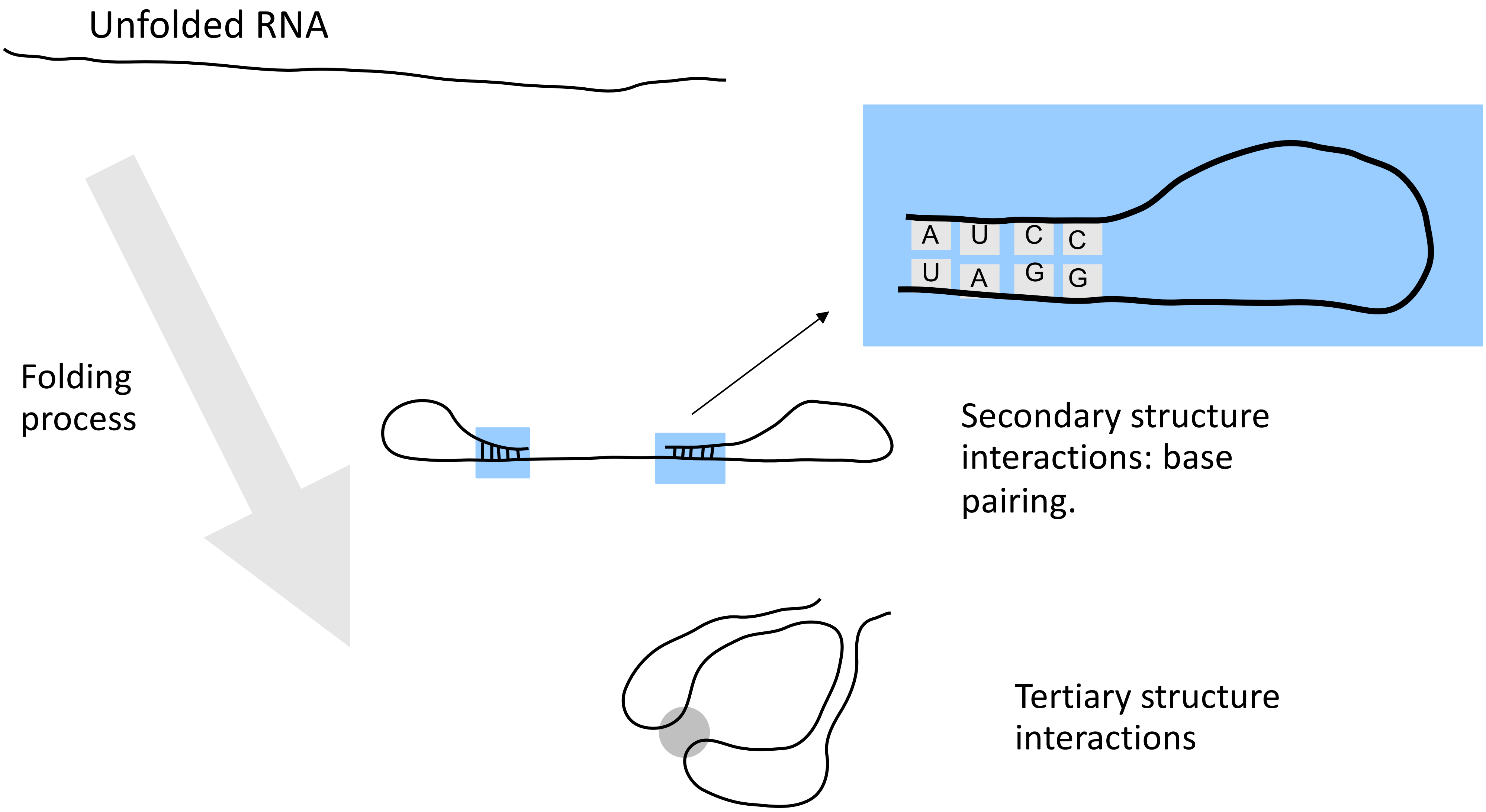


RNA folding problem



Unfolded RNA

Hierarchy in RNA folding



Unfolded RNA

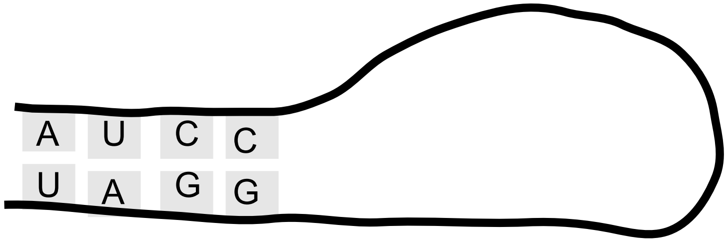
Folding process

Secondary structure interactions: base pairing.

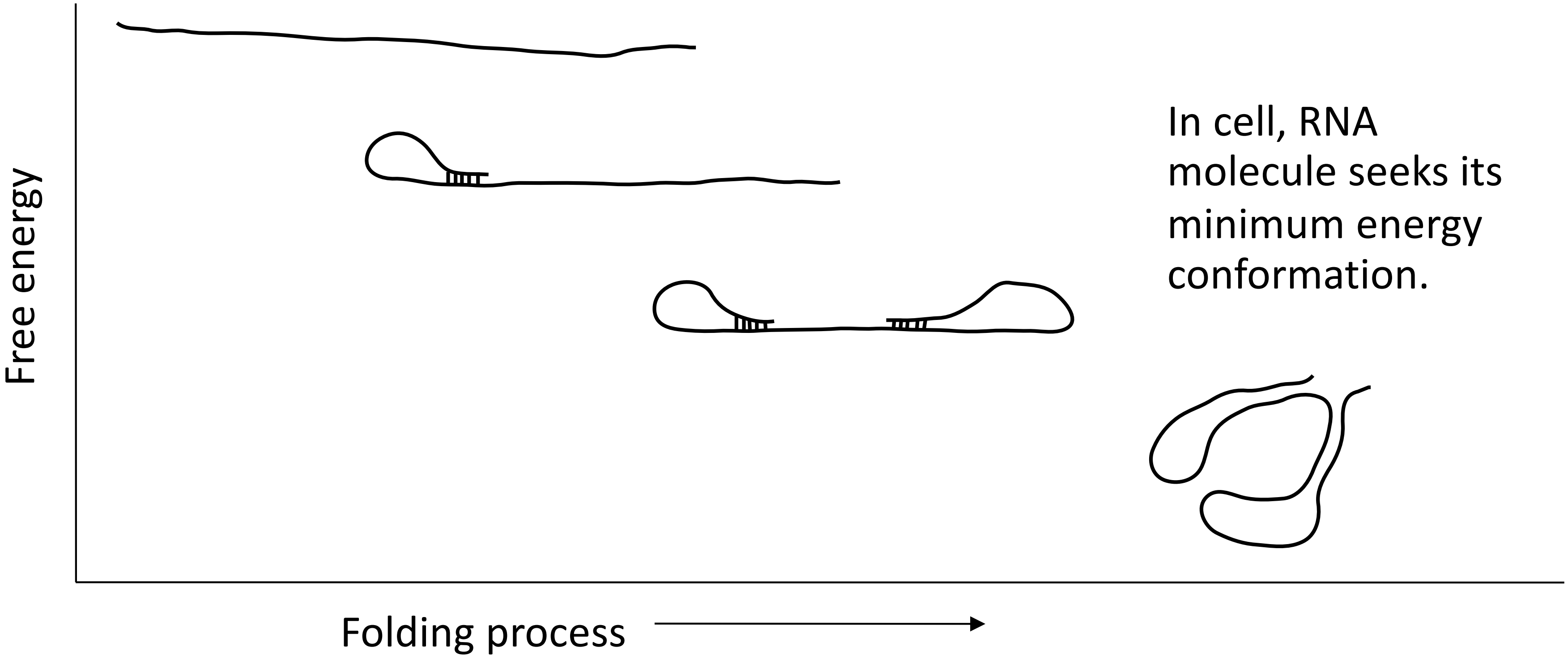
Tertiary structure interactions

Final folded RNA

Energy and folding

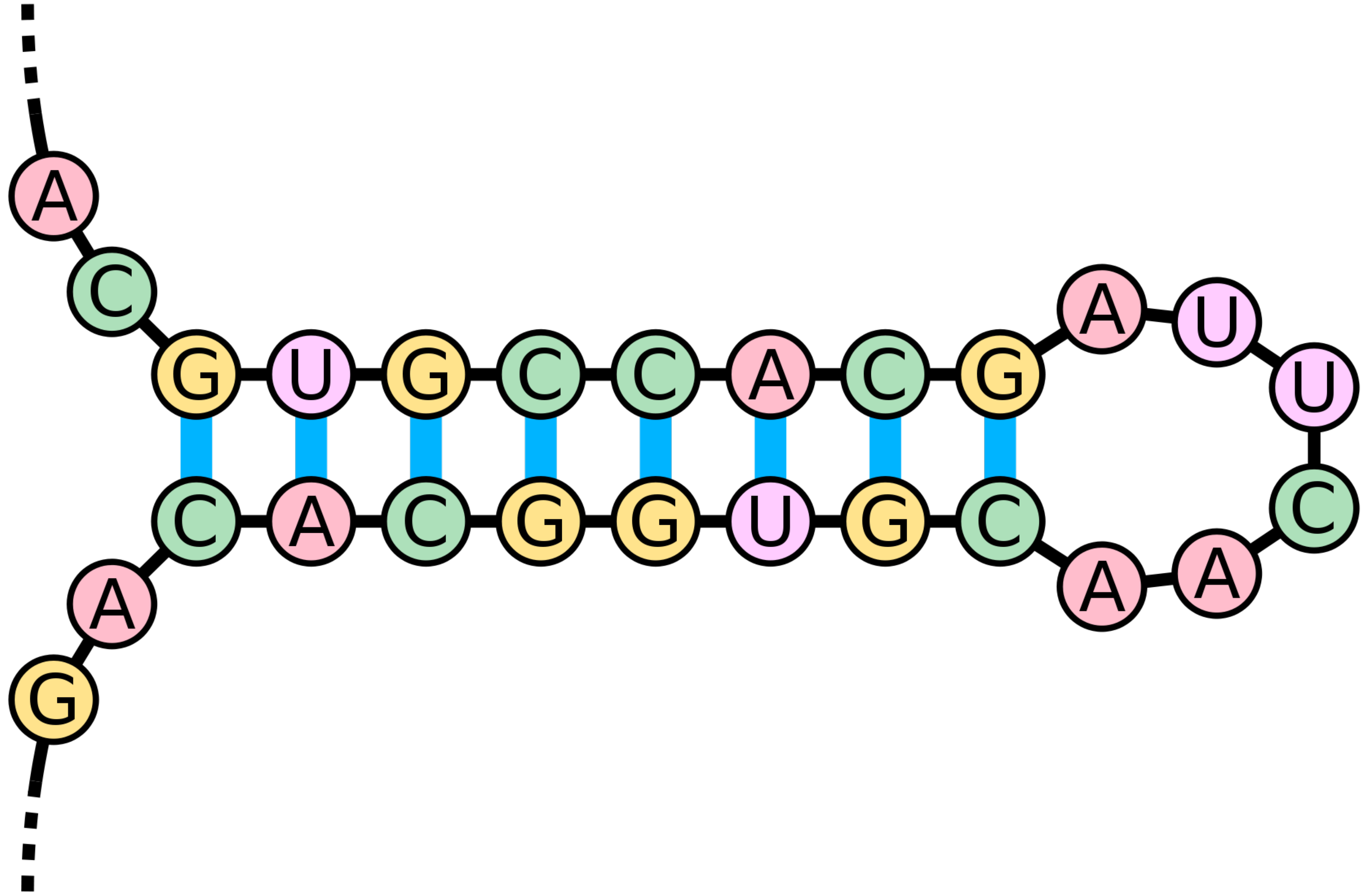


Interactions stabilize molecule, putting it in a lower energy state.



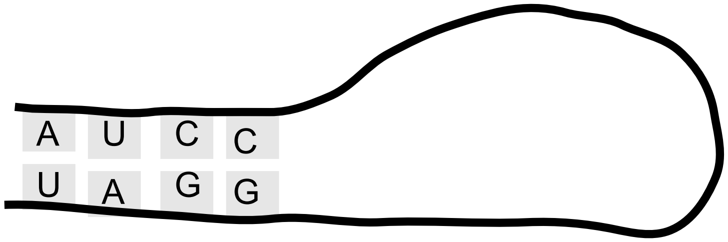
In cell, RNA molecule seeks its minimum energy conformation.

Hairpin loops and other structures



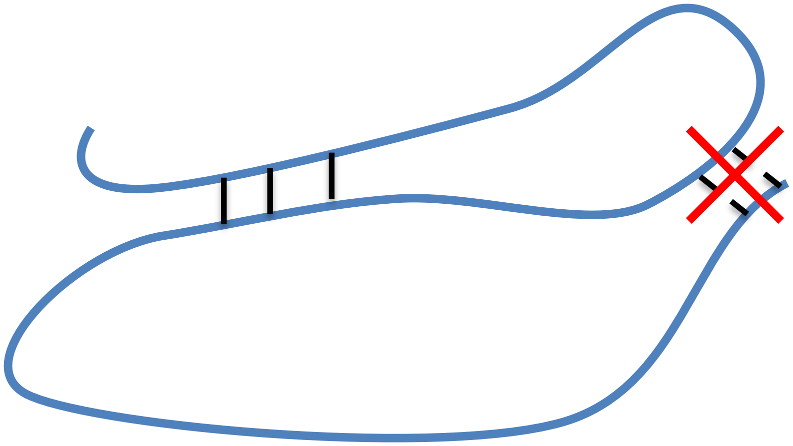
A simple use-it-or-lose-it RNA folding algorithm

- Simple scoring system
  - 1 for each base pair
  - 0 for unpaired bases



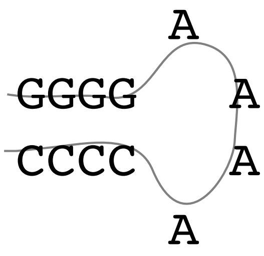
Score: 4

Assume no pseudoknots

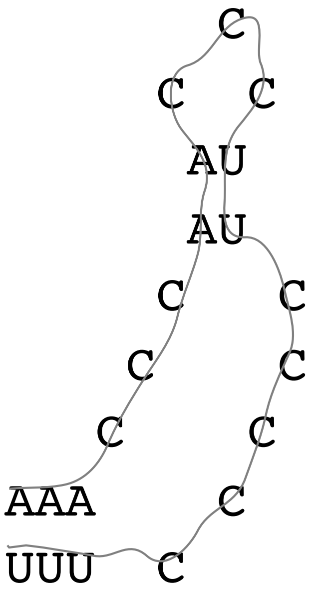


A simple use-it-or-lose-it RNA folding algorithm

```
>>> fold('GGGGAAAACCCC')
4
```



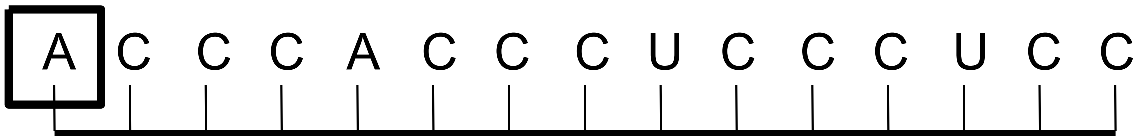
```
>>> fold('AAACCCAACCCUUCCCCUUU')
5
```



```
>>> fold('A')
0
```

Use-it-or-lose-it!

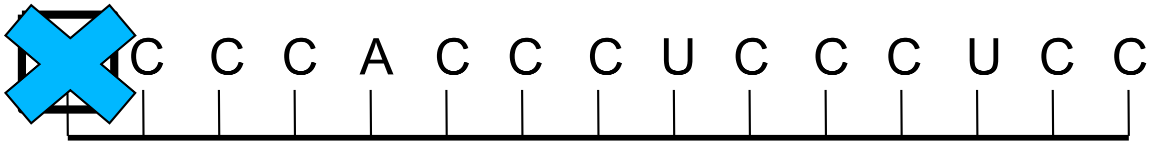
`RNA = ACCCACCCUCCCUCC`



A C C C A C C C U C C C U C C

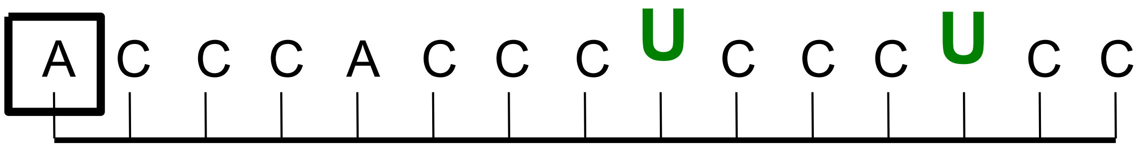
Use-it-or-lose-it!

`RNA = ACCCACCCUCCCUCC`



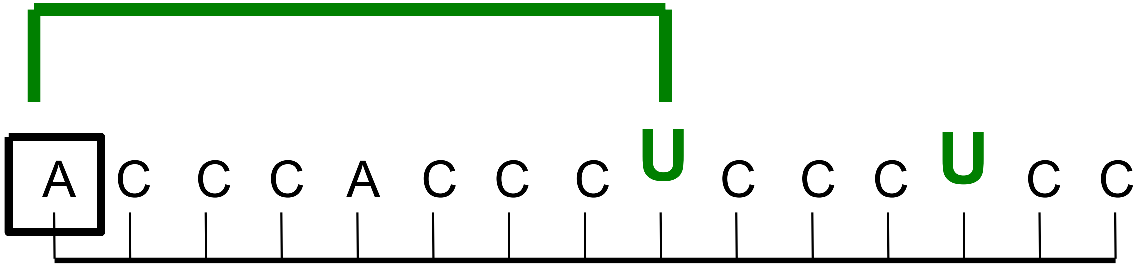
Use-it-or-lose-it!

`RNA = ACCCACCCUCCCUCC`



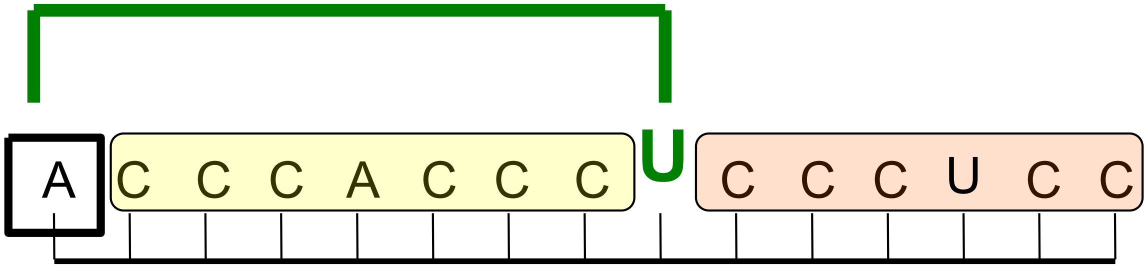
Use-it-or-lose-it!

`RNA = ACCCACCCUCCCUCC`



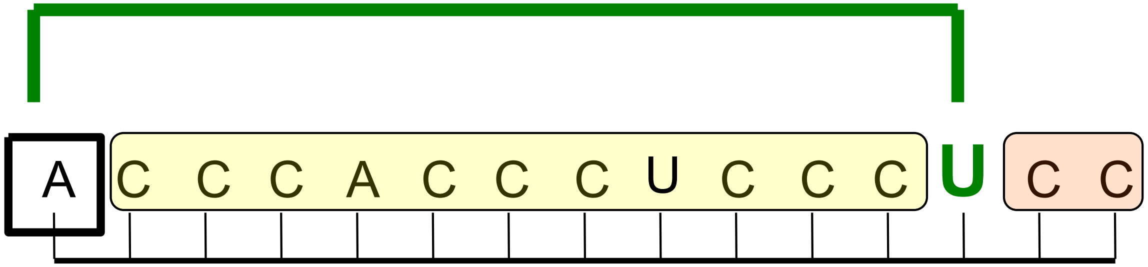
Use-it-or-lose-it!

`RNA = ACCCACCCUCCCUCC`



Use-it-or-lose-it!

`RNA = ACCCACCCUCCCUCC`



```python
def is_complement(base1,base2):
    """Returns boolean indicating if 2 RNA bases are
    complementary."""

    if base1 == "A" and base2 == "U":
        return True
    elif base1 == "U" and base2 == "A":
        return True
    elif base1 == "C" and base2 == "G":
        return True
    elif base1 == "G" and base2 == "C":
        return True
    elif base1 == "G" and base2 == "U":
        return True
    elif base1 == "U" and base2 == "G":
        return True
    else:
        return False
```

Worksheet

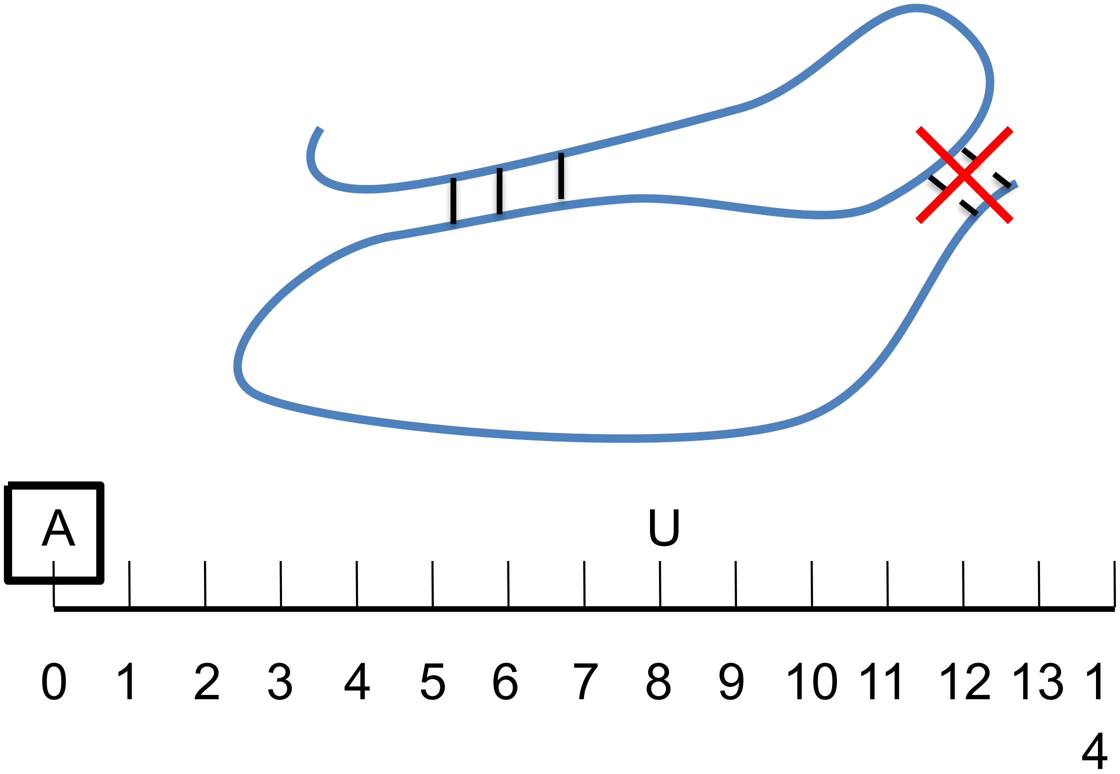
Q

```
def fold(RNA):
    """Takes an RNA sequence as input and returns
    the maximum number of base pairs it can form with itself."""

    if len(RNA) < 2:
        return 0
    else:
```

Worksheet

```
def fold(RNA):
    """Takes an RNA sequence as input and returns
    the maximum number of base pairs it can form with itself."""

    if len(RNA) < 2:
        return 0
    else:
        best_so_far = fold(RNA[1:])   # lose it case
        for i in range(1,len(RNA)):   # use it cases
            if is_complement(RNA[0],RNA[i]):
                score = 1 + fold(RNA[1:i]) + fold(RNA[(i+1):])
                if score >= best_so_far:
                    best_so_far = score
        return best_so_far
```

Importance of the no-pseudoknots assumption



```
score = 1 + fold(RNA[1:7]) + fold(RNA[9:])
```

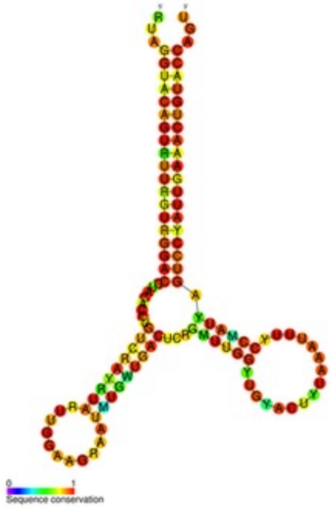
This week's homework

- Write memoized version of fold
- Find regions of strong secondary structure in HIV pol gene
- Care package version to give pairing interactions
- Drawing secondary structure interactions

"Care-package" -ization

```
>>> shortestPath("A", "E", FiveCities, FiveDists)
10
```

```
>>> sp2("A", "E", FiveCities, FiveDists, {})
[10, ['A', 'C', 'D', 'E']]
```



Looking for regions with significant secondary structure

Sliding window

Returning pairing interactions

```
>>> MNV1 = "UGCGCCCUCUGUGCGCAACACAGAGAAACGCA" # MNV1 murine
norovirus

>>> get_struct(MNV1, {})
(11, [(0, 31), (1, 30), (2, 29), (3, 28), (4, 24), (7, 23),
(8, 22), (9, 21), (10, 20), (11, 19), (12, 18)])
```

Exercise: the numbers game (aka pegs)

```
>>> num_game([9, 10, 9])
18

>>> num_game([10, 3, 5, 15, 4])
25
```

```
def num_game(num_list):
    """Chooses non-adjacent numbers from num_list
    such that their sum is maximized. Returns sum."""
    if num_list == []:
        return 0
    else:
        use_it = num_list[0] + num_game(num_list[2:])
        lose_it = num_game(num_list[1:])
        return max(use_it, lose_it)
```

```
def show_num_game(num_list):
    """Chooses non-adjacent numbers from numList such that their sum
    is maximized. Returns list with some as first element, and a list
    with the winning combination as the second element."""
```

```
>>> show_num_game([9, 10, 9])
[18, [9, 9]]

>>> show_num_game([10, 3, 5, 15, 4])
[25, [10, 15]]
```

```python
def show_num_game(num_list):
    if num_list == []:
        return [0,[]]
    else:
        lose_it = show_num_game(num_list[1:])
        use_it = show_num_game(num_list[2:])
        use_it = [num_list[0] + use_it[0], [num_list[0]] + use_it[1]]
        if use_it[0] > lose_it[0]:
            return use_it
        else:
            return lose_it
```

```
>>> show_num_game([9, 10, 9])
[18, [9, 9]]

>>> show_num_game([10, 3, 5, 15, 4])
[25, [10, 15]]
```

`get_struct` – one subtlety

```
>>> get_struct("AAGGGGUU", {})
(2, [(0, 7), (1, 6)])
```

```
>>> get_struct("AGGGGU", {})
(1, [(0, 5)])
```

Demo

What’s coming up next?

- Object-oriented programs (OOPs)