RNA Report

January 24, 2020

1 Introduction

The Nussinov algorithm is a dynamic programming algorithm that finds the maximum number of base pairs for a given sequence. This algorithm works by finding the maximum number of base pairs in a given substructure and then expanding that substructure or merging two substructures to give the total base pair count. This can be summarized by the following where $E(i, j)$ is the maximum number of base pairs for a sub-sequence:

$$E(i,j) = \begin{cases} E(i, j-1) \\ E(i+1, j) \\ E(i+1, j-1) + s(i,j) \\ max_{i<k<j-1}\{E(i,k) + E(k+1, j)\} \end{cases}$$

The Nussinov algorithm tries to maximize the base pair count since a larger number of base pairs implies a more stable secondary structure. While this is true to an extent, this is an overly simplified energy model that does not take into account base pair stacking, loop energies, multiloops and pseudoknots. Therefore more advanced energy folding models (e.g. loop-based energy models) will provide more accurate secondary structures.

2 Materials and Methods

The accuracy of the RNA secondary structure can be improved if the structure of a subsequence is known. The task here is to extend the Nussinov algorithm to incorporate a single known substructure (in the form of dot bracket notation). The constraint consists of a string with characters "(", ")", "x" and ".". Positions with the characters "(" and ")" must be base paired together, positions with "x" cannot be base paired at all and positions with "." have no constraints.

In order to implement the constraint, one simply has to check the position of the constraint string. If the position has a base pair character, add the appropriate score to the corresponding matrix cell. If the position has a character that cannot be base paired, do not add the score to the cell or bifurcate. If the position does not have any constraints, fill the matrix as usual. The traceback procedure will build the structure according to this modified matrix, therefore the implementation of the traceback does not need to be change. The following code accomplishes this:

```
def score_matrix(seq, con, loop_size, weighted):
    '''Returns the filled nussinov matrix for a given sequence
    Adapted from Stefan E Seemann and Giulia I. Corsi'''
    l = len(seq)
```

```
    m = init_matrix(seq) # initialzie matrix
    pairs = []

    if con == None: #if no constraint create a string of size l
        con = "." * l

    for diag in range(loop_size+1, l): # the diagonal of the matrix to loop over
        for i in range(0, l-diag): # the entry on the diagonal to fill
            j = i + diag
            if con[i] == "(" and con[j] == ")":
                m[i, j] = m[i+1, j-1] + get_score(i, j, seq, weighted)
            elif con[i] == "x" or con[j] == "x": #check for unpaired sites and force skip
                m[i, j] = max(m[i, j-1], m[i+1, j])
            elif con[i] == "." and con[j] == ".":
                 m[i, j] = max_score(m, i, j, loop_size, weighted)
    return m
```

The Nussinov algorithm only scores according to total base pair count. However, this is not representative of the lowest possible free energy as different types of base pairs have different free energies, e.g. GC base pairs have a larger contribution to stability then AU pairs. An energy aware version is used here to more accurately reflect the energy contribution of each type of base pair. This is done using the following code:

```
def get_score(i, j, seq):
    '''Given a position i and j, returns either a score'''
    score = 0
    pair = seq[i] + seq[j]
    scores = {'AU': 2, 'UA': 2, 'GU': 1, 'UG': 1, 'GC': 3, 'CG': 3}
    #get score
    if pair in scores:
        score = scores[pair]
    return score
```

3 Results

The following results were obtained after running our implementation of the Nussniov algorithm and calculating the base pair distance between the two sequences:

```
====== Unconstrained sequence ======
Seq: GGGGGUAUAGCUCAGGGGUAGAGCAUUUGACUGCAGAUCAAGAGGUCCCUGGUUCAAAUCCAGGUGCCCCCU
DBS: ((((((((.(((((....).)))))))...)).(((((((....))))(((((.(...).)))))))))).
Ind: 012345678901234567890123456789012345678901234567890123456789012345678901
Score: 67
====== Constrained sequence ======
Seq: GGGGGUAUAGCUCAGGGGUAGAGCAUUUGACUGCAGAUCAAGAGGUCCCUGGUUCAAAUCCAGGUGCCCCCU
DBS: (((((((((.(.(...)))))((.(((((((.......)))))))).))(((((.(...).)))))))))).
Ind: 012345678901234567890123456789012345678901234567890123456789012345678901
Score: 62
Base Pair Distance: 32
```

The following figures show the sketches of the predicted structures[2] and the results of running the sequence on the RNAfold webserver:

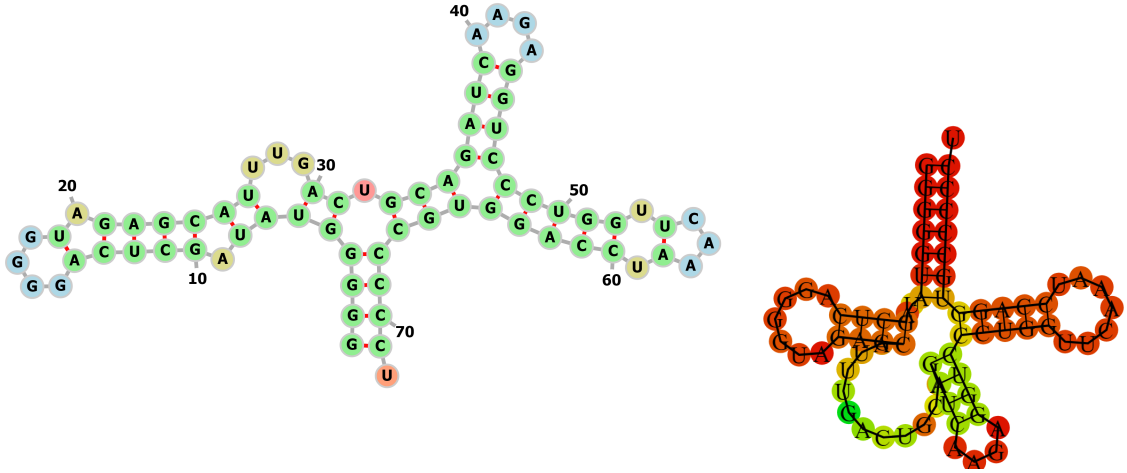

Figure 1: Sketch of the unconstrained prediction (left) vs the RNAfold server prediction (right) with a minimum free energy of -26.80 kcal/mol.

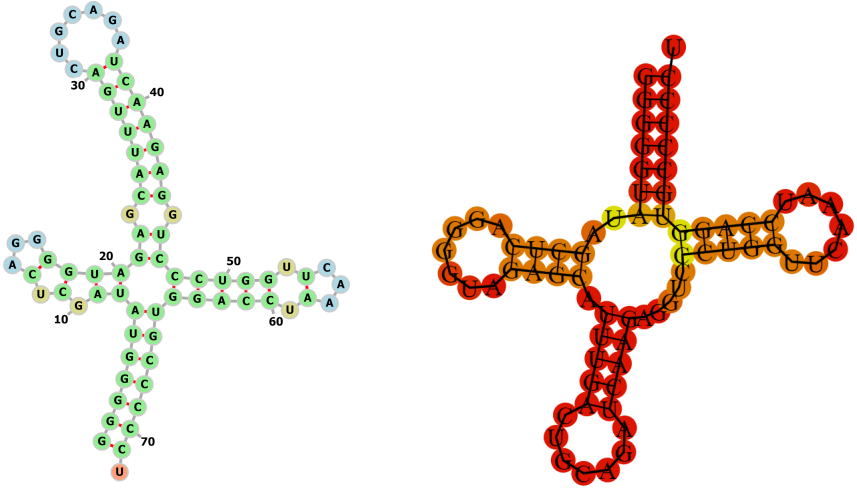

Figure 2: Sketch of the constrained prediction (left) vs the RNAfold server prediction (right) with a minimum free energy of -24.50 kcal/mol

The unconstrained structure consists of two stem loops attached to an internal loop on one side, two internal loops in the center and a stem loop on the other side. The introduction of the constraint changes the structure quite dramatically. This new structure has 3 stem loops around an internal loop and the overall structure is similar to that of the tRNA cloverleaf structure. Annotation of the sequence reveals that it is indeed a tRNA with Rfam Accession RF00005.

4 Conclusion

It can be seen that the constrained RNAfold structure is the closest to the true structure of a tRNA. The unconstrained RNAfold structure is similar but has a large unpaired region that is not present in the tRNA structure. Both structures, however, are significantly better compared to the structures predicted by the Nussinov algorithm, specially when the constraint is not applied.

One of the causes of the difference in structures is the energy model used. The RNAfold webserver uses the Minimum Free Energy model which takes into account base pair stacking, loop energies and the optimal multi-loop energy. These features are all missing in the Nussinov implementation. Another disadvantage of the Nussinov algorithm is the ambiguous decomposition, i.e. multiple structures can give the same score, so multiple paths can be taken during the traceback. However, the disadvantage of this more advanced energy model is that its space complexity is $O(n^4)$ whereas the simpler Nussinov model is $O(n^2)$. [1]

References

[1] J. Gorodkin. *RNA Sequence, Structure, and Function: Computational and Bioinformatic Methods*. Humana Press Inc, 2013.

[2] Kerpedjiev P. Forna (force-directed rna): Simple and effective online rna secondary structure diagrams. *Bioinformatics*, 2015.