2D tRNA Graph Generator

By Rongzhong Li

Sequence Information

```
In[1]:= sequence =
          "GCCCGGAUAGCUCAGUCGGUAGAGCAGGGGAUUGAAAAUCCCCGUGUCCU
             UGGUUCGAUUCCGAGUCCGGGCACCA";
        base = Table[StringTake[sequence, {i}],
           {i, StringLength[sequence]}];
```

Basepair Information Format

- **Total Number, Acceptor1, Length_acc, D1, Length_d, D2, Anticodon1, Length_ant, Anticodon2, Y1, Length_t, Y2, Acceptor2**

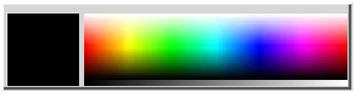
```
In[3]:= info = {StringLength[sequence],
           1, 7, 10, 4, 25, 27, 5, 43, 49, 5, 65, 72};
```

Generator

```
In[26]:= Text[Style["RGBColor Picker", 18]]
         {ColorSlider[Dynamic[x]], Dynamic[x]}
         Text[Style["Adjust the parameters.", 18]]
         Seq2D[N_, Acc1_, Lacc_, D1_, Ld_, D2_, Ant1_, Lant_, Ant2_, Y1_, Ly_, Y2_, Acc2_] :=
          (anti = (Ant1 + Ant2) / 2;
           seq = Table[n → n + 1, {n, N - 1}];
           arm1 = Table[D1 + k - 1 → D2 - k + 1, {k, Ld}];
           arm2 = Table[Ant1 + k - 1 → Ant2 - k + 1, {k, Lant}];
           arm3 = Table[Y1 + k - 1 → Y2 - k + 1, {k, Ly}];
           arm4 = Table[Acc1 + k - 1 → Acc2 - k + 1, {k, Lacc}];
           loop = {Acc1 + Lacc - 1 → D1, D2 → Ant1, Ant2 → Y1};
           graph = Join[seq, arm1, arm2, arm3, arm4, loop];
          )
         Seq2D @@ info;
         plot[StepLength_, Iterations_, Rotation_, Size_] :=
          GraphPlot[graph, VertexRenderingFunction →
             ({EdgeForm[Black], RGBColor[1.`, 1.`, 0.5019607843137255`], Disk[#1, 0.4], Black,
                Text[If[Divisible[#2, 5] && #2 ≠ anti, Style[#2, Bold, Red], 1] If[Abs[#2 - anti] < 2,
                   Style[base[[#2]], Medium, Bold, Red], base[[#2]]], #1]} &), EdgeRenderingFunction →
             (If [Or @@ Table[First[#2] == loop[[i]][[1]] && Last[#2] == loop[[i]][[2]], {i, 3}],
                {}, If[First[#2] == Last[#2] - 1, {Black, Thickness[0.005], Line[#1]},
                 {Blue, Thickness[0.005], Line[#1]}]] &),
           "Method" → {"SpringEmbedding", "RecursionMethod" → "Multilevel", "StepLength" → StepLength,
             "MaxIterations" → Iterations, "Rotation" → Rotation}, ImageSize → Size]
         Manipulate[plot[StepLength, Iterations, Rotation, Size], {StepLength, 5, 20, 1},
          {Iterations, 10, 200, 5}, {Rotation, 0, 2 Pi, 2 Pi / 18}, {Size, 600, 900, 50}]
```

Out[26]= RGBColor Picker

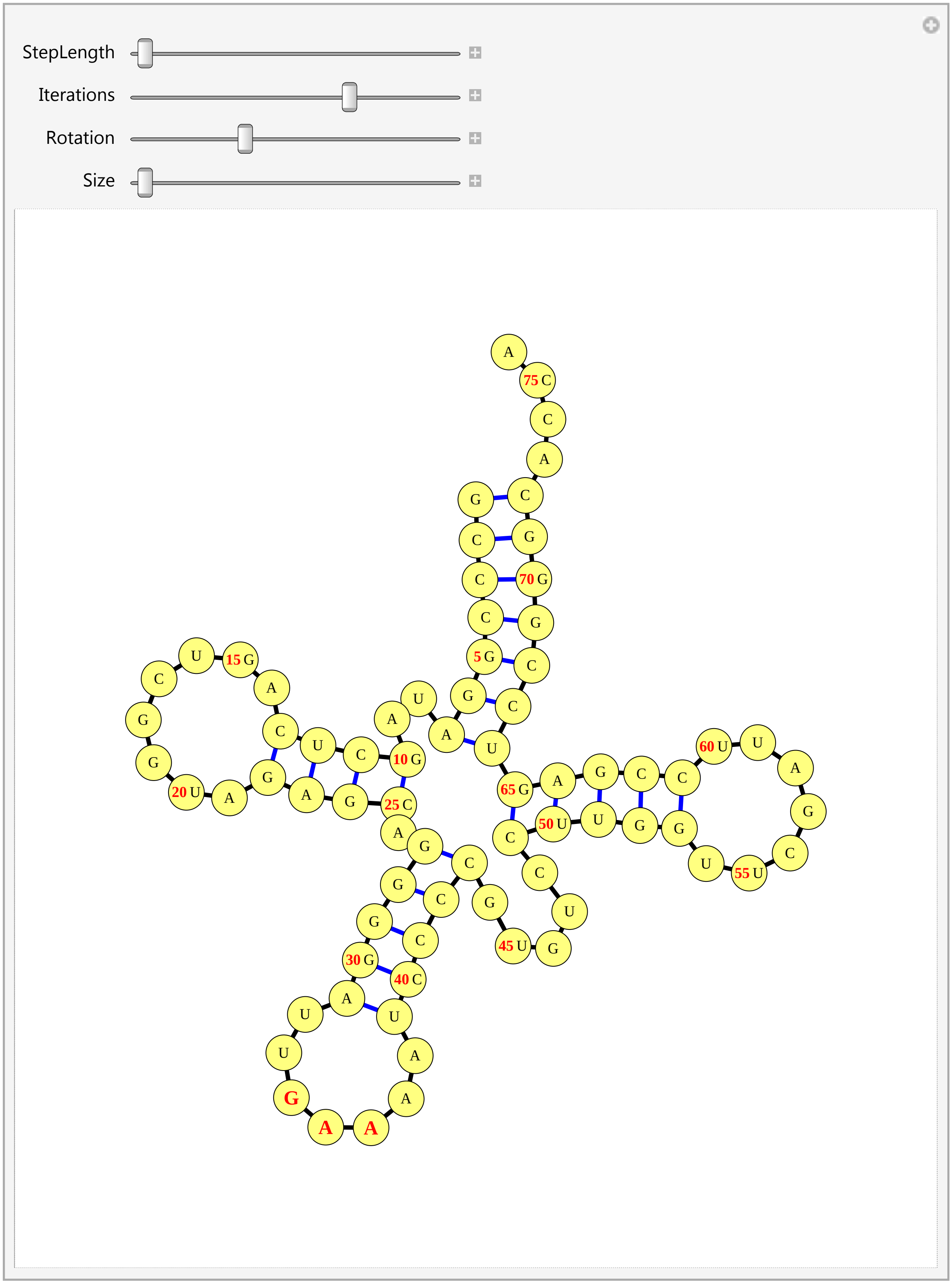
Out[27]= {



, RGBColor[0., 0., 0.]}

Out[28]= Adjust the parameters.

Out[32]=


StepLength
Iterations
Rotation
Size
A
75 C
C
A
C
G
70 G
G
C
C
U
65 G
A
G
C
C
60 U
U
A
G
C
55 U
U
G
G
U
50 U
C
C
U
G
45 U
G
C
C
C
40 C
U
A
A
A
G
U
U
A
30 G
G
G
G
A
25 C
G
A
G
A
20 U
G
G
C
U
15 G
A
C
U
C
10 G
A
U
A
G
5 G
C
C
C
G

Export 2D Image File

```
In[11]:= Export["D:\userdata\desktop\phe.jpg", plot[5, 140, 2 π/3, 600], "jpg"];
```

3D Extention and Rough Structure Predictor

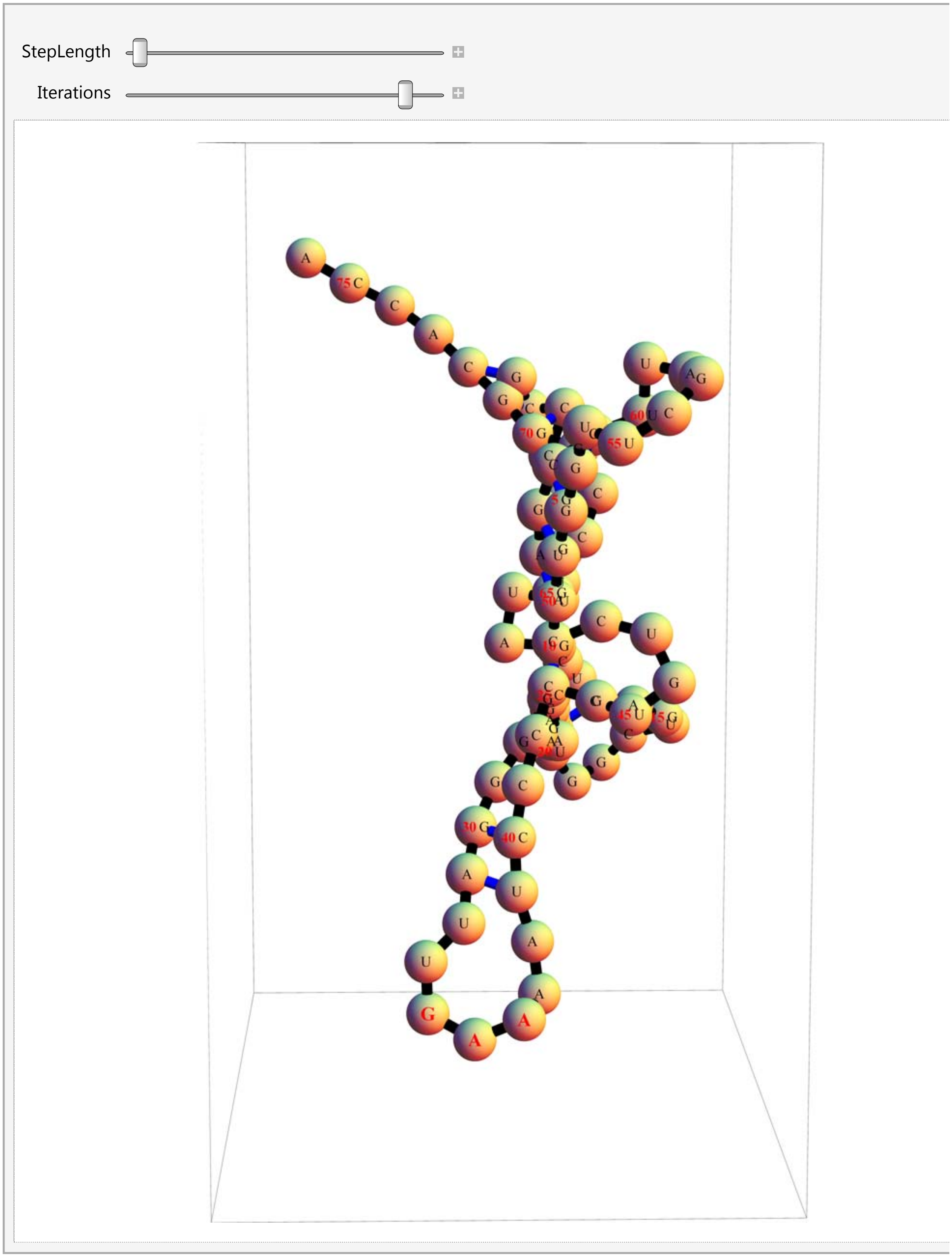
- **Add additional contacts to the graph**

```
In[12]:= contacts = {};
         link = Table[contacts[[i]][[1]] → contacts[[i]][[2]], {i, Length[contacts]}];
         newgraph = Union[graph, link];
```

- **Prediction (Just for fun. By no chance should it be trusted)**

```
In[33]:= plot3D[StepLength_, Iterations_] := GraphPlot3D[newgraph,
           VertexRenderingFunction → ({RGBColor[1.`, 1.`, 0.5019607843137255`], Sphere[#1, 0.4], Black,
                Text[If[Divisible[#2, 5] && #2 ≠ anti, Style[#2, Bold, Red], 1] If[Abs[#2 - anti] < 2,
                    Style[base[[#2]], Medium, Bold, Red], base[[#2]]], #1]} &), EdgeRenderingFunction →
             (If [Or @@ Table[First[#2] == loop[[i]][[1]] && Last[#2] == loop[[i]][[2]], {i, 3}] || ! Or @@
                    Table[First[#2] == graph[[i]][[1]] && Last[#2] == graph[[i]][[2]], {i, Length[graph]}],
                 {}, If[First[#2] == Last[#2] - 1, {RGBColor[0.`, 0.`, 0.`], Thickness[0.005],
                   Cylinder[#1, 0.1]}, {Blue, Thickness[0.005], Cylinder[#1, 0.1]}]] &),
           "Method" → {"SpringEmbedding", "RecursionMethod" → "Multilevel",
             "StepLength" → StepLength, "MaxIterations" → Iterations, "InferentialDistance" → 0.5}]
         Manipulate[plot3D[StepLength, Iterations], {StepLength, 5, 20, 1}, {Iterations, 10, 200, 5}]
```

Out[34]=


StepLength
Iterations
75
70
60
55
65
50
19
45
20
40
30
G
A
A