Modeling Update: Software product

Rohit Malyala 26-Jan-2018

McMaster Biomod

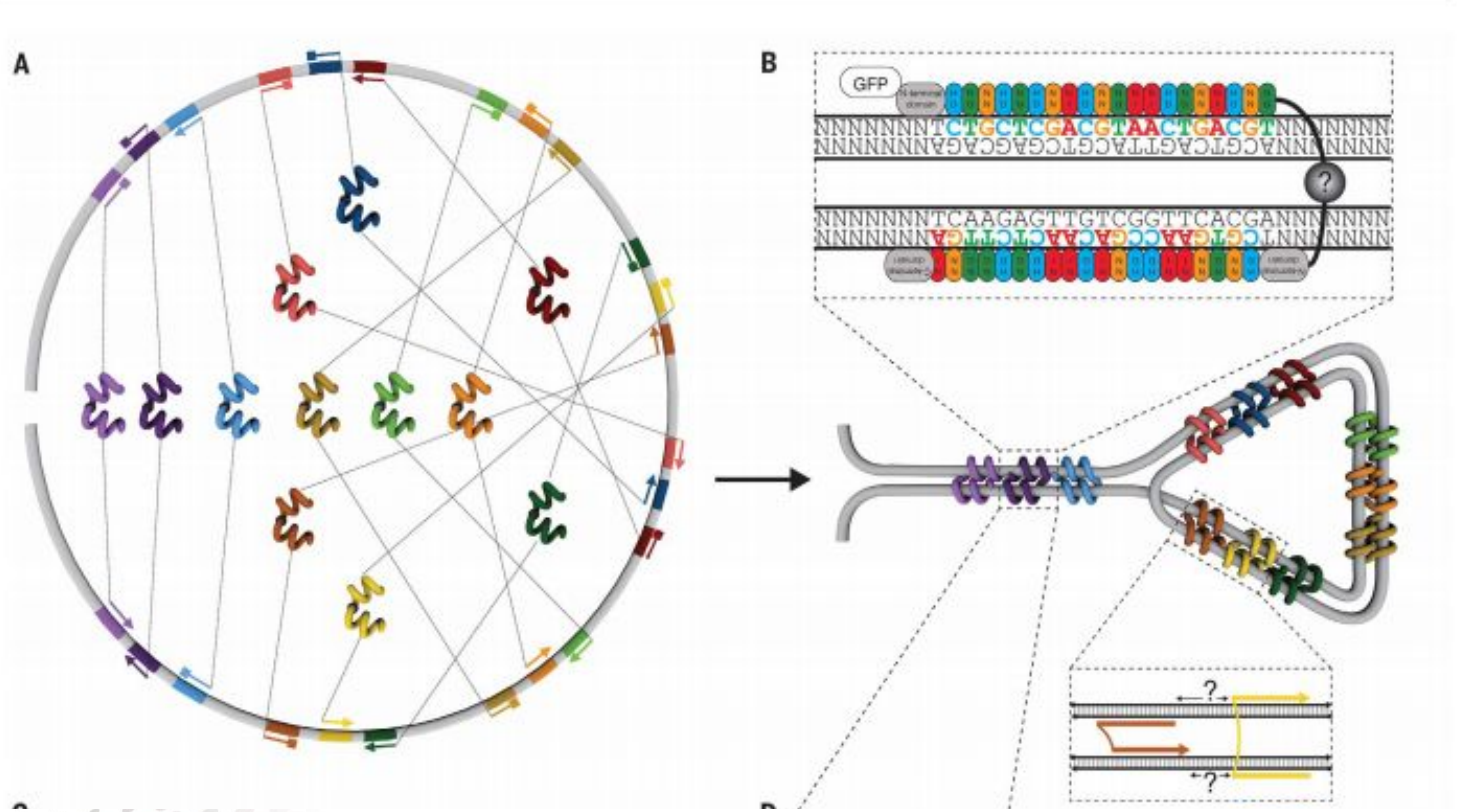
Agenda for discussion

1. Background from the Praetorius paper
2. Why is this software program needed?
3. What has been accomplished so far?
4. How it was done
5. Future directions

McMaster Biomod

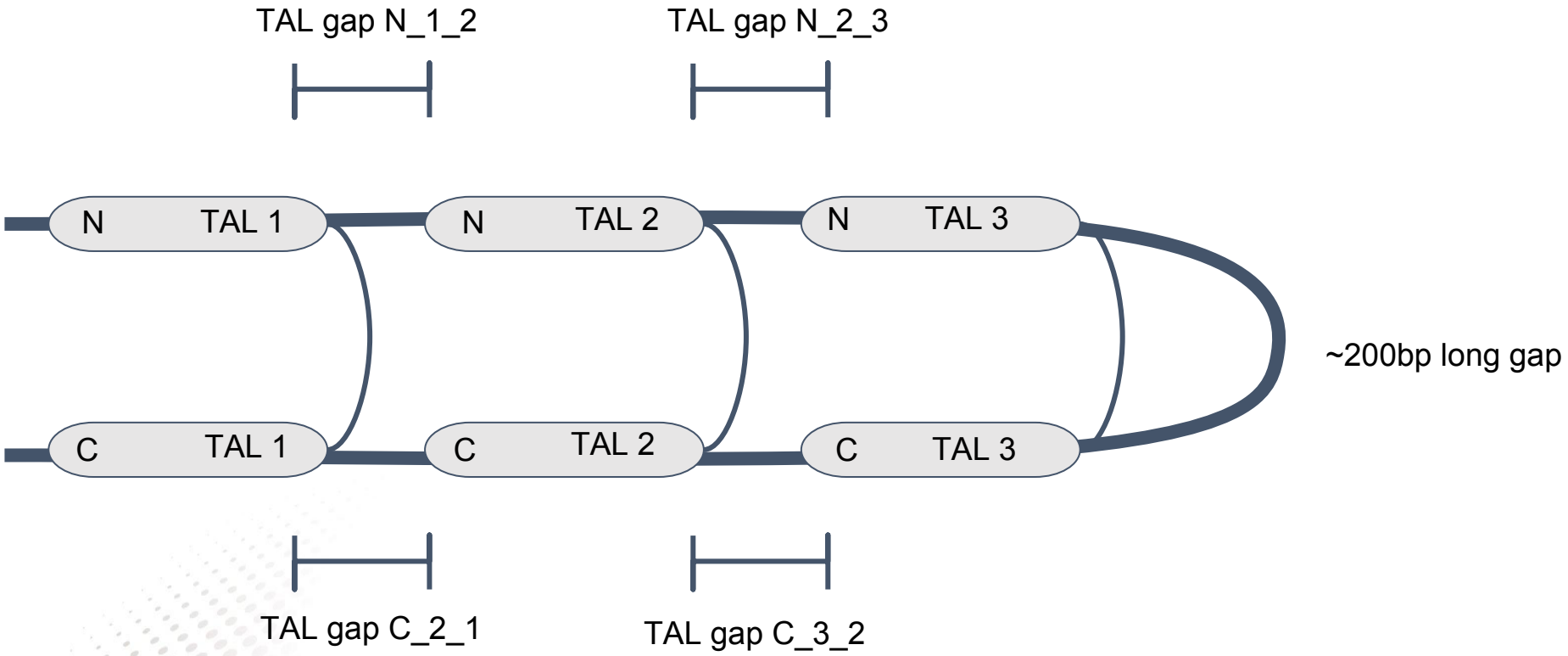
1. The Praetorius Paper

This paper outlined a method for creating DNA origami by stapling linear or circular dsDNA with TAL-protein staples.



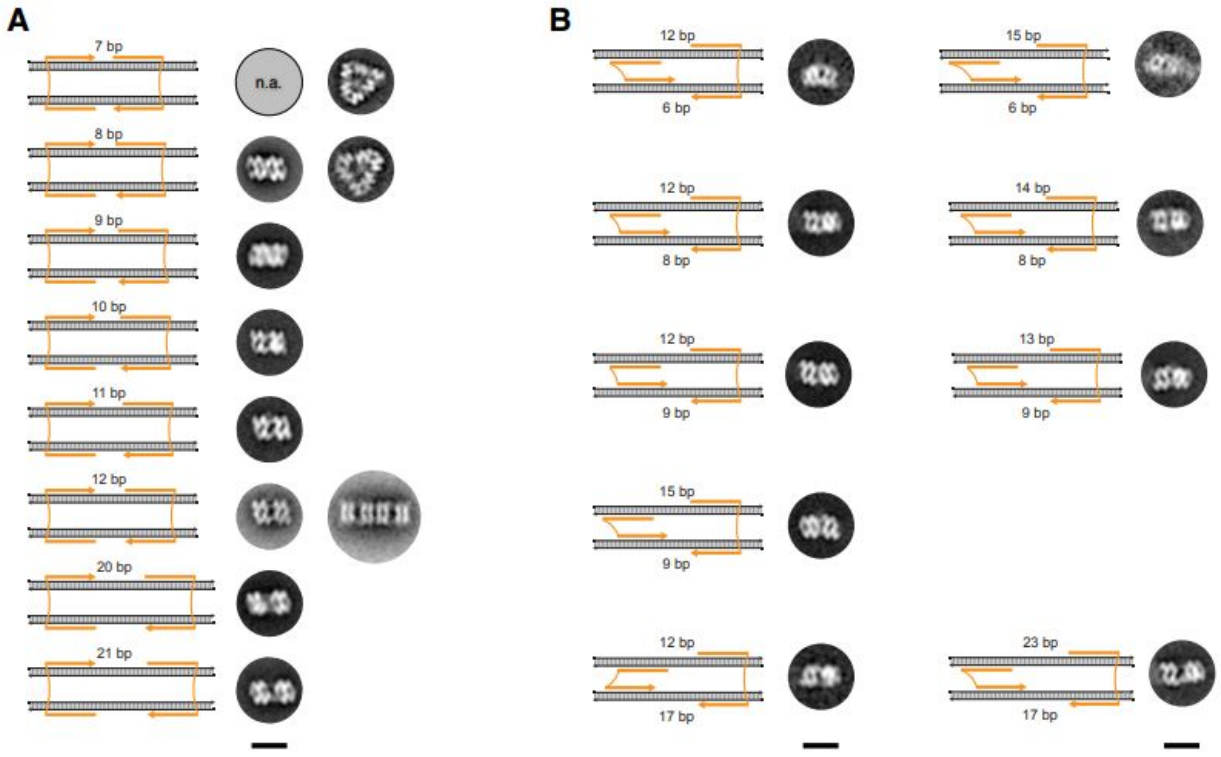
1. The Praetorius Paper

A general visual representation of the method they used.



2. Why is a software product needed?

Outline the huge number of potential ways in which staples pinch together dsDNA to create various shapes. A large amount of visual information needs to be put forward in order for anybody to understand what is going on.



McMaster Biomod

2. Why is a software product needed?

Each of these shapes has a unique primary sequence. A great deal of experimental work was put into figuring out optimal sequences of primitive shapes in the supplementary of the original paper. But there are two problems:

1) How does someone easily reconcile all sorts of competing physical considerations when designing a DNA nanostructure by hand?
2) How do you deviate appropriately from the provided sequences/shapes?

3. What has been accomplished so far?

A command line script that takes base pair gap inputs as integers, and series of printouts are given.

1) Linear length b/n mounted proteins
2) *Real* distance between proteins
3) Degree of curve
4) Dihedral angle
5) Primary sequence

(really bad rn tbh pls don't be upset)

```
DNA ORIGAMI TWO-TAL SEQUENCE STRUCTURE

Input base pair distance between N-terminal TAL 1 and N-terminal TAL 2:8

Input base pair distance between N-terminal TAL 2 and N-terminal TAL 3:13

Input base pair distance between C-terminal TAL 1 and C-terminal TAL 2:8

Input base pair distance between C-terminal TAL 2 and C-terminal TAL 3:13


The distance between your mounted proteins, assuming both are chimerized to the n-terminal of each
double-TAL:
linear protein distance, bp
29
linear protein distance, nm
9.860000000000001
angled protein distance, nm
2.4650000000000003


The estimated dihedral angle between your mounted enzymes along the axis of the DNA sequence where the
TALs are mounted is provided below. This estimate becomes less meaningful the greater inter-enzyme
distance is.


Dihedral between TAL 1 and TAL 2 in degrees
334.8000000000002
Dihedral between TAL 2 and TAL 3 in degrees
280.80000000000007


This is the degree of curvature.
-243.74999999999997


TAL(#8) tCTGCTCGACGTAACTGACGT | CGCTCCAC | tCCGGTGAATCCACATTTACC | AATAGATCACCCG | tCCGGTCCGTGTTAGCGATGC
|
GGAAAAAGTCGCCAATGAGCCACTTGTAaCACAATTGGGCATtTCAGATGATCGCTGTCGTTCCTGGTTGCTTACCAGTACATGTATAACTaCTCAGTCAGTAAA
tATAAATGTCGGTTGCCGTCTTTGTCCATTTCGAGATCAATCGATAGTAaGACCTATCACAAGtTCGACGAAACGTTACGCCTCGGACTAACTTCGAGATCTATG
```

IPython console | History log

Permissions: RW | End-of-lines: CRLF | Encoding: ASCII | Line: 126 | Column: 9 | Memory: 45 %

McMaster Biomod

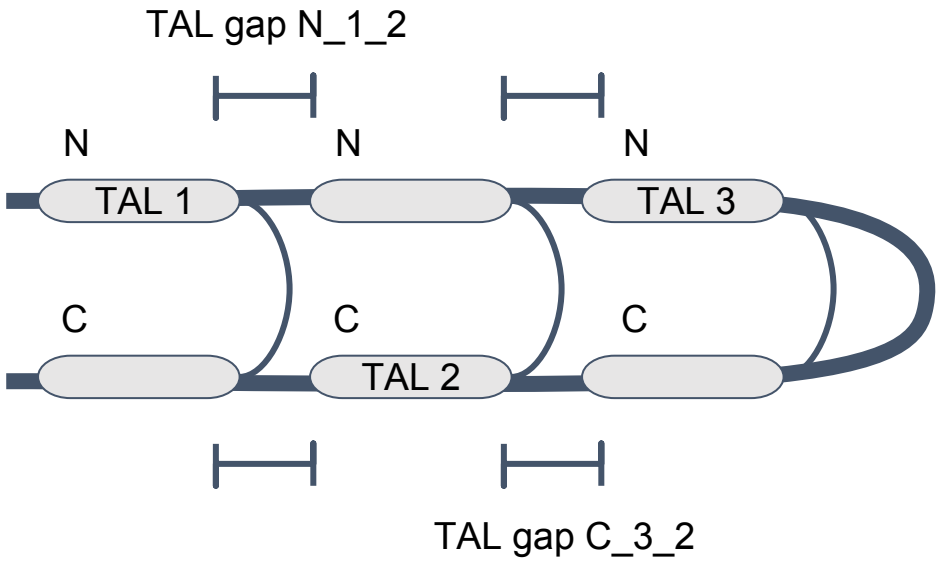
3. What has been accomplished so far?

Currently, a script has been created that, for a custom three-TAL monomer system only, a series of physical and informational outputs are generated.

You are able to input the number of base pairs making the gaps in between N-TALs and C-TALs, as integers.

The series of outputs includes:

- Distance between N-terminuses (which hold the enzymes)
- The dihedral angle between enzymes (the helix normal vector)
- The monomer's degree of curvature
- <u>The primary DNA sequence</u>



McMaster Biomod

4. How was this achieved?

There are three sets of principles that are used to give rise to the formulas used in the script.

1) Praetorius' design rules (outlined in the paper and supplementary)

E.g. When generating strings that represent the DNA sequences, respect must be given to the rules illustrated in figure 2. But not too much respect. Ultimately the base pair gaps between TALs must be up to the user, I guess.

I definitely need someone who understands the Praetorius paper to help fix this

Empirically derived optimal distances between binding sites in all six possible relative orientations of two adjacent double-TAL staple proteins. All binding sites are 21 bp long. We found that twohelix bundles built using repetitions of the doublecrossover motifs in Fig. 2, A and D, tended to form multistranded higher-order structures, whereas the bundles based on the motifs in Fig. 2, B and C, remained mostly monomeric...

McMaster Biomod

4. How was this achieved?

2) The Dietz lab's understanding of the physical properties of DNA

E.g. 10bp == 3.4nm == 3400pm == 34 Å == one 360° turn of the helix.

∴ proteinLinearDistance % 10 * 180) % 360 = dihedral angle between the n-terminuses of two TAL proteins, assuming the DNA helix is linearly aligned.

∴ proteinLinearDistance*0.34 = converts bp integer to nm float distance

McMaster Biomod

4. How was this achieved?

3) Trigonometry

E.g. calculating the degree of curvature given a 21bp difference between the top and bottom TAL-staple gaps is a geometric system of equations

$S_1 = r * \theta$

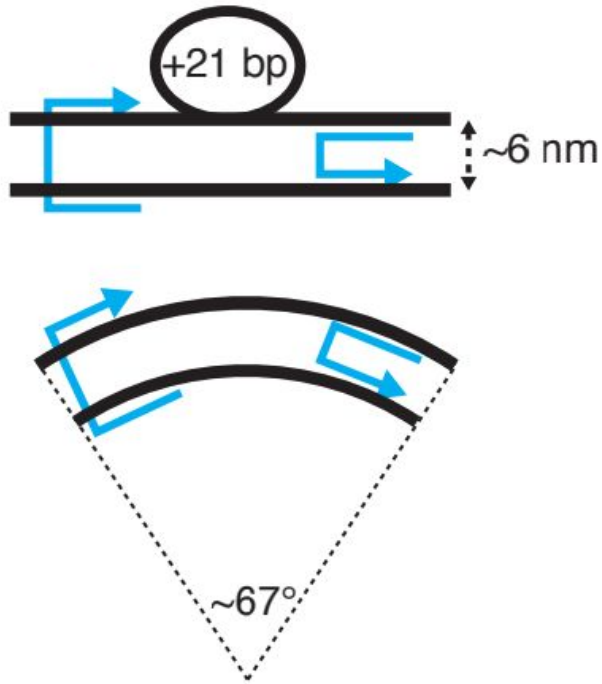
$S_2 = (r + 6nm) * \theta$

$\longrightarrow$ solve for r

$(6 * S_2)/(S_1 - S_2) = r$

Degree of curvature = $(180 * S_1)/(\pi * r)$

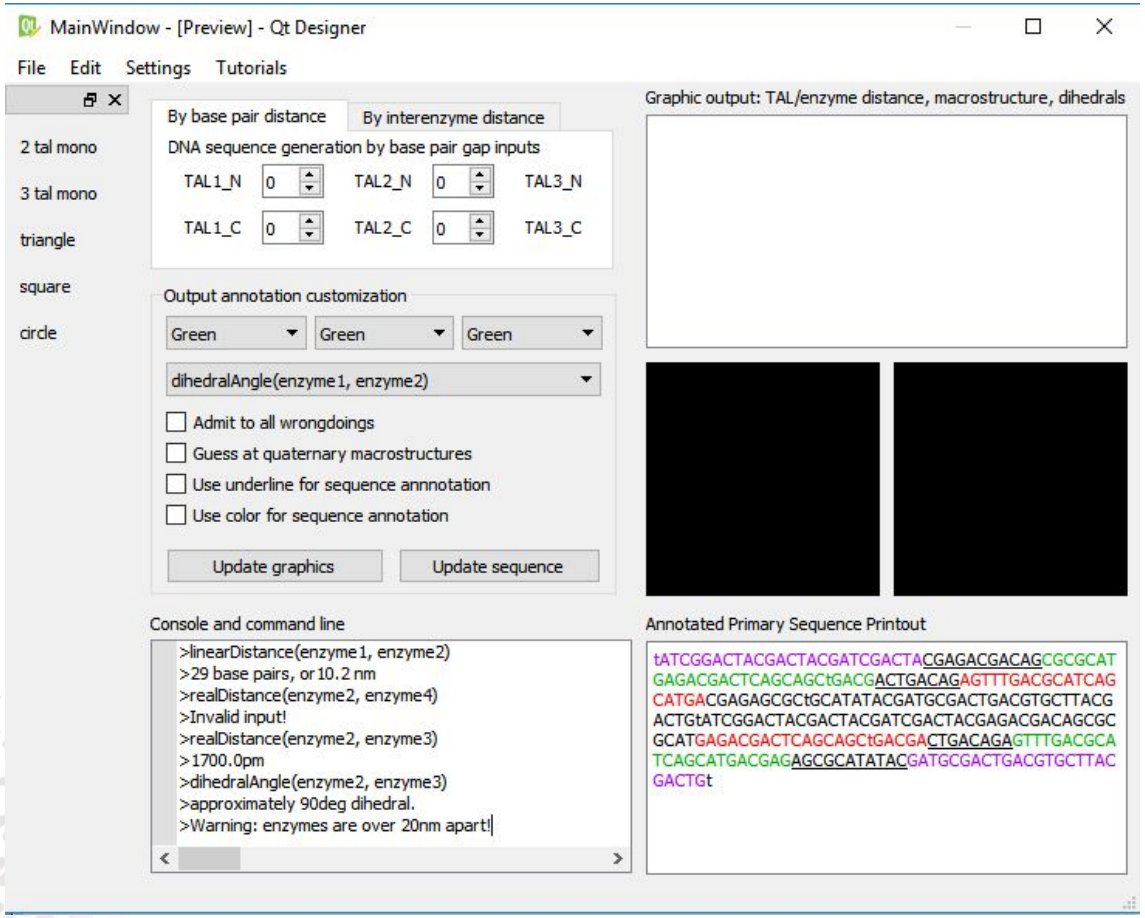
Real distance between enzymes = chord between N-terminuses of both TALs



McMaster Biomod

5. Future Steps

This GUI pretty much summarizes everything we need to achieve… including the fact that we probably want a GUI since the entire project is so visual.



McMaster Biomod

5. Other Future Steps

Make less spaghetti-like code for my three-TAL system, so that it can be more easily adapted to new “primitive” objects.

An *a posteriori* approach that allows someone to input desired distances or dihedrals, and get the base pair gap lengths automatically generated.

A good deal of visualization and vector representation of the system should be developed

Most importantly, empirical demonstration that the code output isn’t B.S.