Advanced Computational Techniques for Triplex-DNA Engineering

Zhuan Chen
and
Anthony Kusalik

Department of Computer Science
University of Saskatchewan
Saskatoon, Saskatchewan

Credits

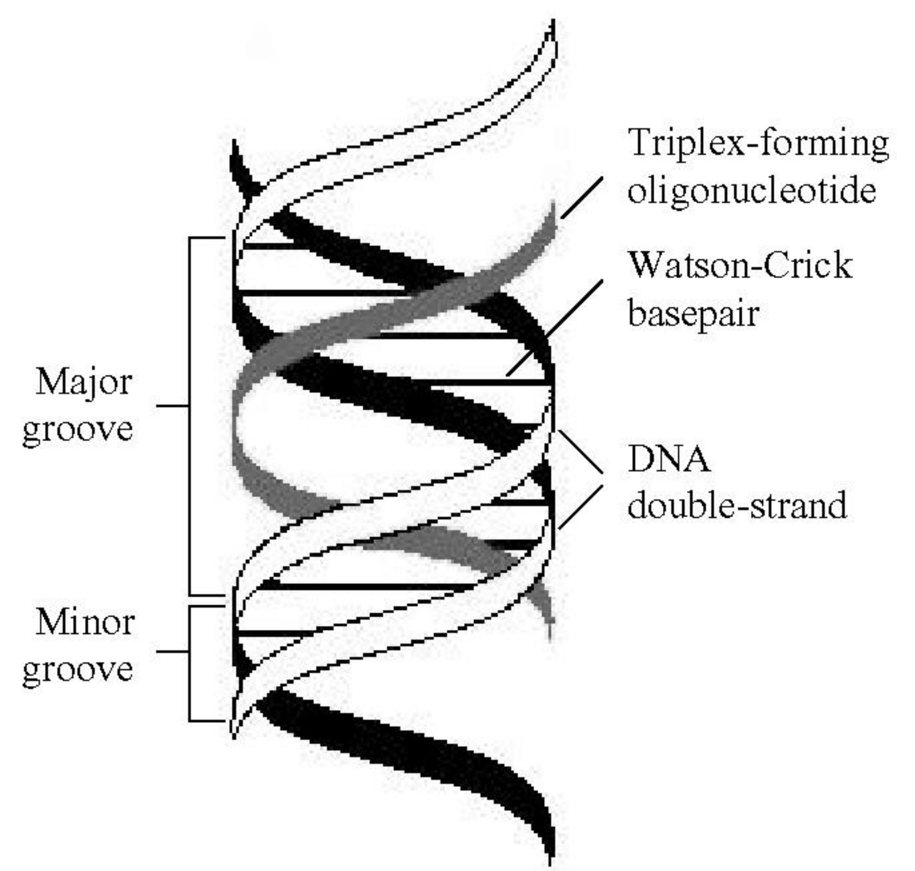
Dr. Robert Hickie
Pharmacology

University of Saskatchewan
Saskatoon, Saskatchewan

Outline

- Introduction
- Objectives
- System Design
- Experiment
- Results
- Discussion

Triplex DNA

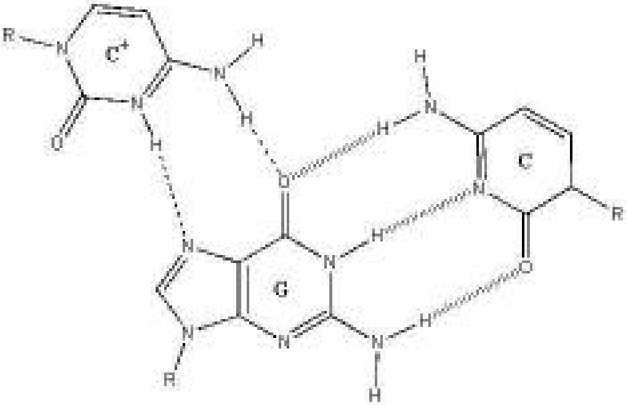


TFO

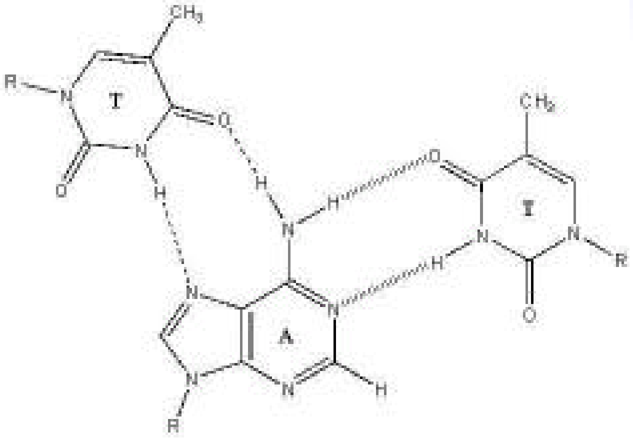
TFR

Triplex DNA — continued

```
      5′ TCTTCTTTCC 3′                 Pyrimidine motif TFO
         **********                    Hoogsteen basepair
5′ TGTCAGAAGAAAGGTAGA 3′               Polypurine strand of DNA (TFR)
   ||||||||||||||||||                  Watson- Crick basepair
3′ ACAGTCTTCTTTCCATCT 5′               Polypyrimidine strand of DNA
```



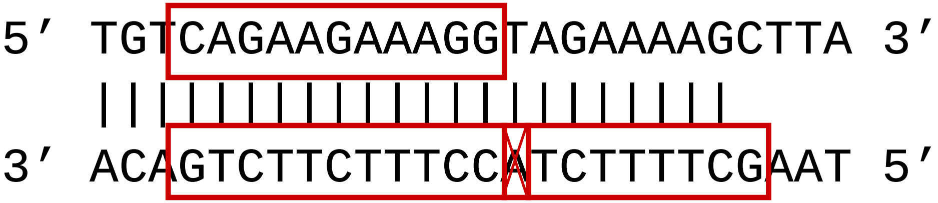
CG*C+



TA*T

Triplex DNA — continued

- intermolecular and intramolecular triplexes
- can inhibit DNA transcription and replication
- various types:
  - continuous (CP*n*)
  - single discontinuity (SD*n*)
  - alternate-strand region



```
5′ TGTCAGAAGAAAGGTAGAAAAGCTTA 3′
   ||||||||||||||||||||||
3′ ACAGTCTTCTTTCCATCTTTTCGAAT 5′
```

Calmodulin

- Calmodulin (*CaM*): a main regulator of Ca^{2+}-dependent signaling in eukaryotic cells.
- Higher eukaryotes possess three *CaM* genes encoding identical proteins.

```
CaMI     atg gct gat cag ctg acc gaa gaa cag att
CaMII    *** *** **c **a *** **t **a **g *** ***
CaMIII   *** *** **c **g *** **t **g **g *** ***
          M   A   D   Q   L   T   E   E   Q   I
                    ..................

CaMI     gaa ttc gta cag atg atg act gca aaa tga
CaMII    *** **t *** **a *** *** **a *** **g ***
CaMIII   *** **t *** **g *** *** **t *** **g ***
          E   F   V   Q   M   M   T   A   K  ///
```

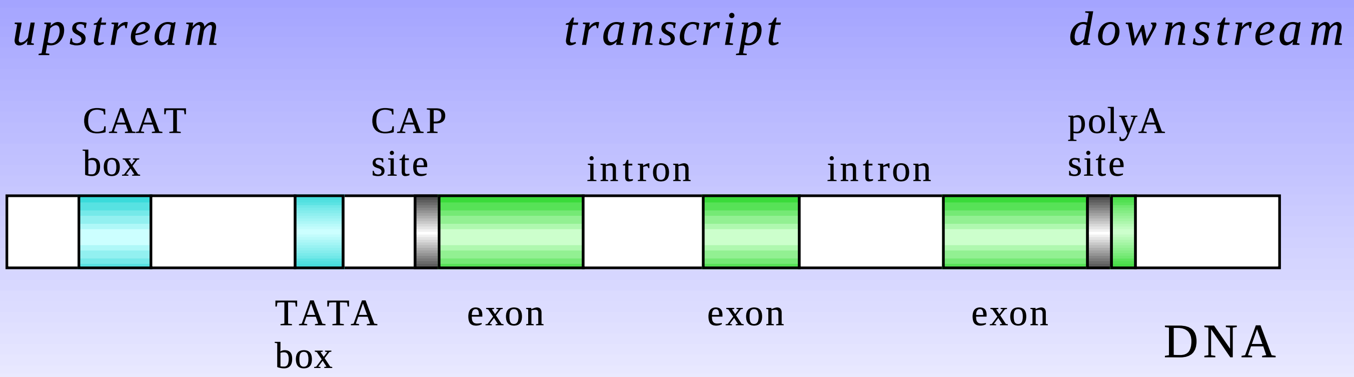
Calmodulin — continued

- active research: regulation of expression and function of each gene
- selectively “shut off” any pair
- transcriptional inhibition by triplex DNA
- hard to engineer TFO
- harder in CaM case because the gene sequences are highly conserved

Logic Programming & Grammars

- useful tool in language processing and pattern searching
- logic grammars, e.g. DCGs:
  - high-level, powerful
  - used for language (and biosequence) analysis
  - directly translated and executed

DCG example



```
gene -> upstream, transcript,
        downstream.
upstream -> caat_box, basepairs(N),
        tata_box, basepairs(M),
        {N >= 40, N <= 50,
         M >= 17, M <= 27}.
```

Constraint Logic Programming

- constraint satisfaction problems (CSP) from math, operations research, AI
- constraint logic programming (CLP) improves efficiency, expressivity, generality, and reusability of logic programs
- CLG: logic grammars incorporating CLP instead of LP

CLG example

```
upstream -> caat_box, basepairs(N),
            tata_box, basepairs(M),
            {N >= 40, N <= 50,
             M >= 17, M <= 27}.
```

```
upstream -> caat_box,
            { N :: [40 .. 50],
              M :: [17 .. 27] },
            basepairs(N), tata_box,
            basepairs(M).
```

Objectives

- identify and encode knowledge rules for
    - ranking “desirability” of TFRs
    - designing TFOs
- software to
    - identify TFRs with continuous Pu/Py (CP)
    - identify SD TFRs (with continuous Pu/Py region flanking)
    - rank desirability of TFRs
    - generate TFOs

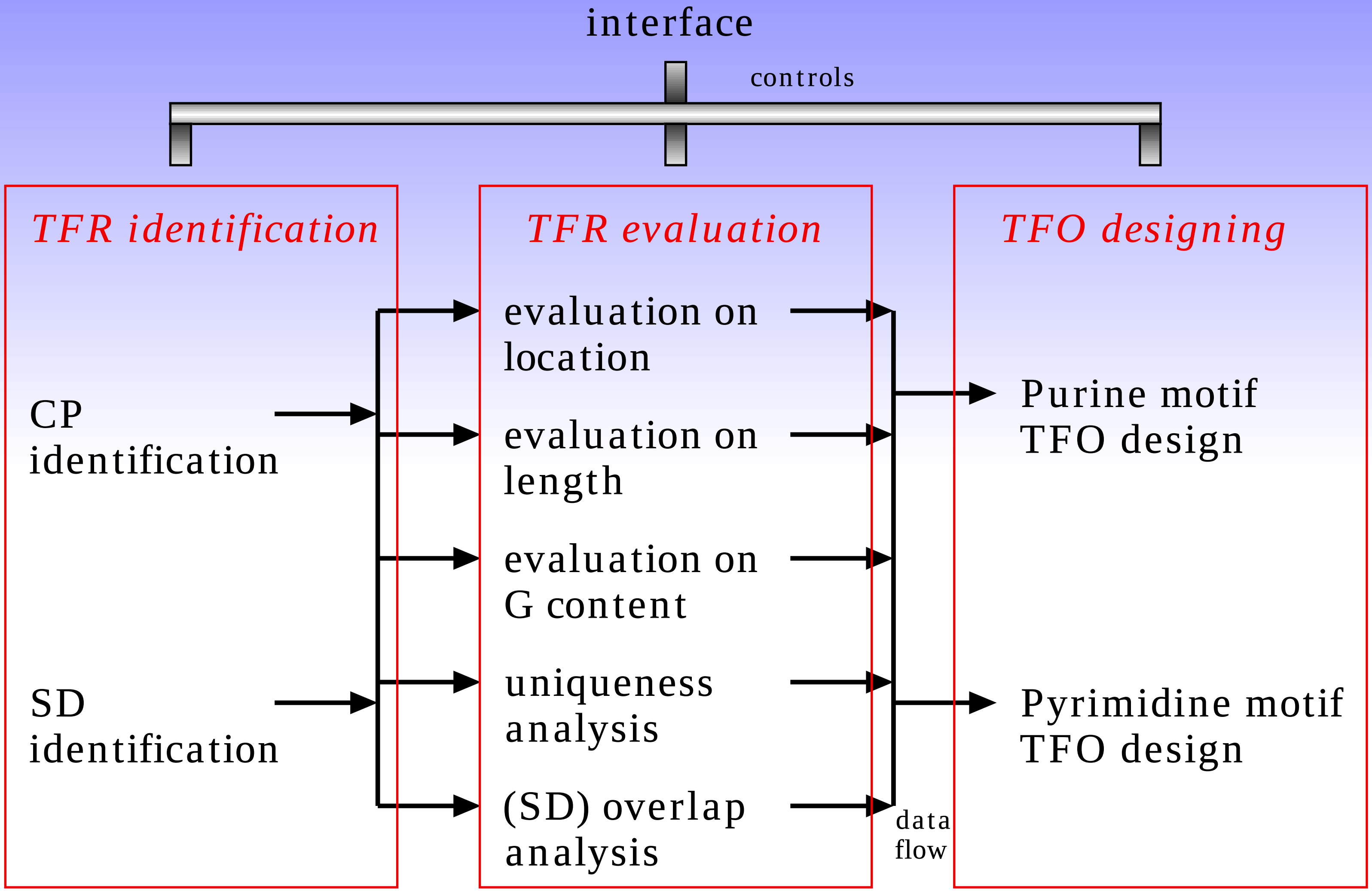
Knowledge Rules

- imprecise, and sometimes conflicting
- rules
    - TFR
        - length
        - ratio of G content

        binding proficiency

        - position
        - uniqueness
        - multiple overlaps
    - TFO
        - e.g. purine motif TFO: CG*C^+, TA*T

Software Modular Decomposition



interface

controls

TFR identification

CP identification

SD identification

TFR evaluation

evaluation on location

evaluation on length

evaluation on G content

uniqueness analysis

(SD) overlap analysis

data flow

TFO designing

Purine motif TFO design

Pyrimidine motif TFO design

Other Software Design Details

- parameters
    - length of continuous matching lengths
- text-based, menu-oriented interface
- uniqueness
    - within a gene: *gene sublist group*
    - across genes: *sublist group*
- conversion of py-motif to pu-motif
    - G content
    - uniqueness

    CTCCTTCCTT (5’->3’)   AAGGAAGGAG (5’->3’)

Experiment

- CP9 and SD4
- validation testing
    - with verification data
    - soundness and completeness
- data verification
- execution with *CaM* genes
- timing

Results

- validation test
- data verification
- CP TFRs

Gene	Count	Longest	No. Unique
CaMI	48	23 bp	20
CaMII	34	29 bp	13
CaMIII	49	25 bp	25
G content: 18%-81%			

E.g. of gene sublist group:

```
CaM2  231  240 10 no 60          gagagaggga
CaM2 1536 1555 20 no 60 gaaggaagggagagagggag
```

Results — continued

- SD4 TFRs:

Gene	Count	Longest	No. Unique	No. Overlaps
CaMI	70	27 bp	60	8
CaMII	39	37 bp	26	6
CaMIII	80	23 bp	68	9
G content: 0-90%				

E.g. of gene sublist group:

```
CaM2  493  503 11 no 27 gaaatgaagaa
CaM2 3203 3212 10 no 30 gaaatgaaga
```

E.g. of multiple overlapping SD TFRs

```
CaM3 5534 5545 12 py->pu 83 ggggtggggagg
CaM3 5542 5552 11 py->pu 90 ggggggtgggg
CaM3 5547 5559 13 py->pu 84 aggggggtgggggg
CaM3 5554 5569 16 py->pu 68 gaggggaagcaggggg
```

Results — continued

figure showing all the TFRs identified in the beginning portion of CaMIII

TFO Design

- Highly Ranked TFRs

CP9 by position

```
CaM1   48   58 11     no 18 aagaaaagaaa
CaM3  158  174 17 py->pu 29 agagaagaagagaaaaa
```

CP9 by length

```
CaM2 3253 3281 29 py->pu  0 aaaaaaaaaaaaaaaaaaaaaaa
                            aaaaaa
CaM3 5103 5127 25 py->pu 32 agagggaaagaaaaaaaaagag
                            aa
```

SD4 by both position and length

```
CaM3  153  174 22 py->pu 22 agagaagaagagaaaaataaaa
```

TFO Design — continued

CP9

```
 CaM3  158  174 17 py->pu 29 agagaagaagagaaaaa
Purine motif TFO             tgtgttgttgtgttttt
Pyrimidine motif TFO         tctcttcttctcttttt
```

SD4

```
 CaM3  153  174 22 py->pu 22 agagaagaagagaaaaataaaa
Purine motif TFO             –
Pyrimidine motif TFO         tctcttcttctctttttgtttt
```

Other Results

- timing
  - SUN Ultra5
  - no stage took more than 1 sec CPUtime

Discussion

- unique system for triplex DNA analysis
- straightforward programming effort
- system is extensible
  - e.g. 14-3-3 genes
- future work
  - inductive database
  - automatic ranking of TFR/TFO candidates
  - improve knowledge rules with lab feedback