



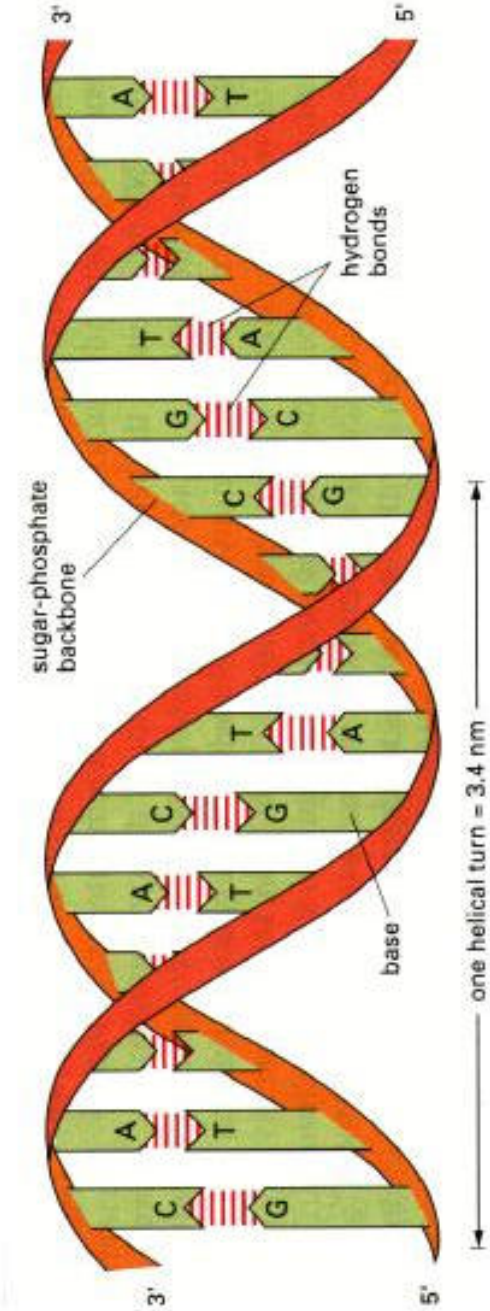
# Prototyping Nucleic Acid Computations

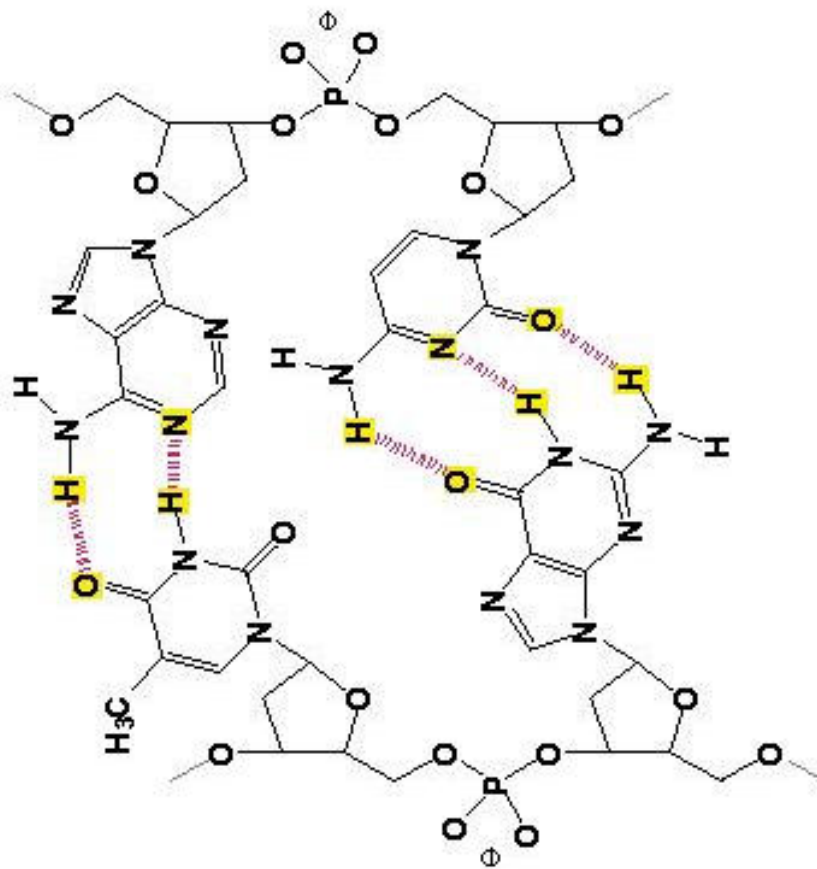
D. Faulhammer, R. J. Lipton, L. F. Landweber

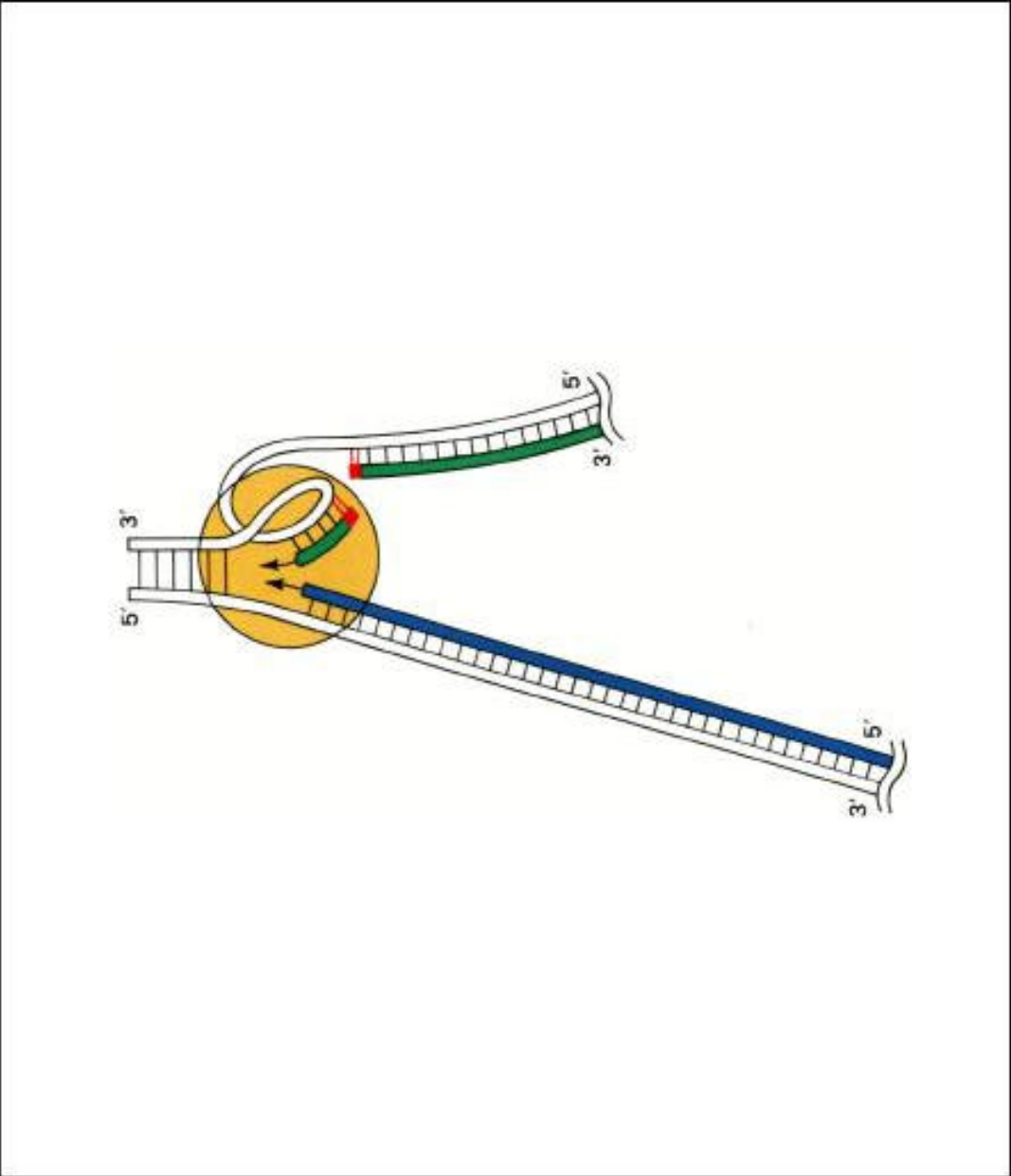
# Prototyping Nucleic Acid Computations

---

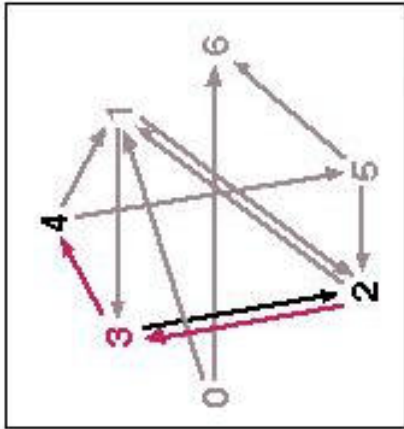
- Testing  $DNA^2$  DNA Computation
- RNA Solutions to SAT
  - New Method for Creating Libraries
  - Construction of a 10-bit Library
  - The ‘Knight Problem’







# The 'Traveling Salesman Problem'



$O_2$  TATCGGATCGGTATATCCGA  
 $O_3$  GCTATTCGAGCTTAAAGCTA  
 $O_4$  GGCTAGGTACCAGCATGCTT  
 $O_{2 \rightarrow 3}$  GTATATCCGAGCTATTCGAG  
 $O_{3 \rightarrow 4}$  CTTAAAGCTAGGCTAGGTAC  
 $O_3$  CGATAAGCTCGAATTTCGAT

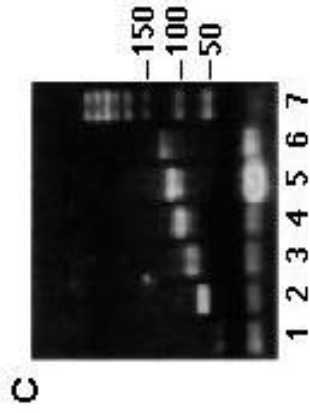
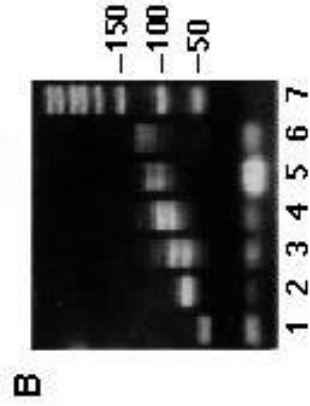
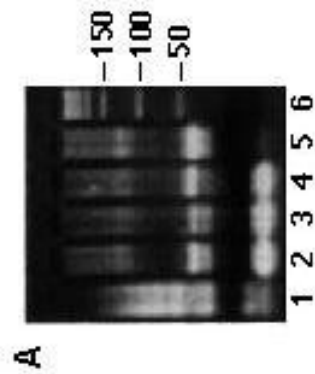
$O_{2 \rightarrow 3}$

$O_{3 \rightarrow 4}$



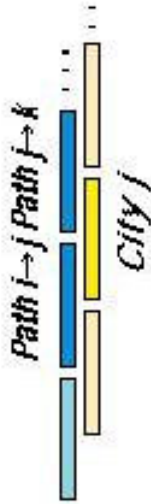
GTATATCCGAGCTATTCGAGCTTAAAGCTAEGCTAGGTAC  
 CGATAAGCTCGAATTTCGAT

$O_3$

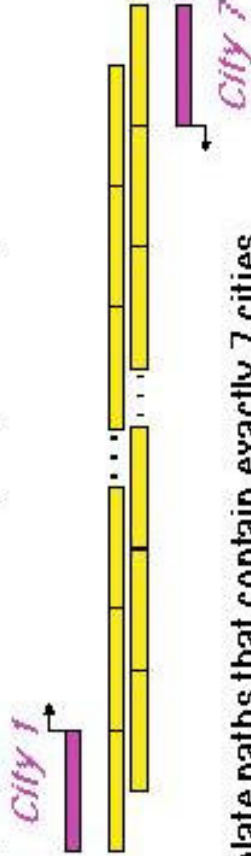


# The 'Traveling Salesman Problem'

1) Generate random paths through the graph.



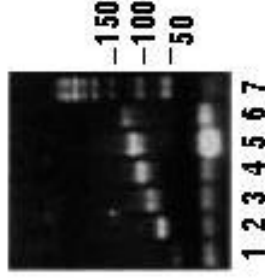
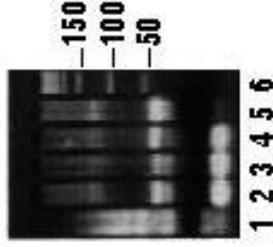
2) Amplify paths containing City 1 and City 7



3) Isolate paths that contain exactly 7 cities.

4) Keep only those paths enter all of the cities at least once.

5) If any paths/products remain, say "YES", otherwise say "NO"







## Other Examples

---

- DNA-based addition (Guarnieriet *al.* 1996)**
- Maximal Clique Problem (Ouyang *et al.* 1997)**
- In vitro* selection of oligonucleotide libraries**
- RNA editing and other biological systems.**

## **Features**

---

- Solve *computationally hard problems* with simple protocols
- Demonstrate the power of *massive parallelism*
- Solve NP-complete problems in *linear time*
- High *energy efficiency* per calculation step

## **Features**

---

- Solve *computationally hard problems* with simple protocols
- Demonstrate the power of *massive parallelism*
- Solve NP-complete problems in *linear time*
- High *energy efficiency* per calculation step

# Testing $DNA^2DNA$ Computation

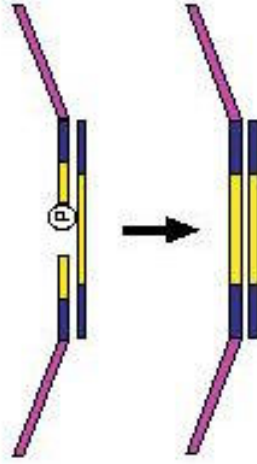
---

How well do DNA ligases perform?

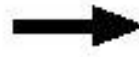
# Testing Ligation Efficiency and Fidelity

---

1. Ligation



2. Gel electrophoresis of ligation products



Efficiency

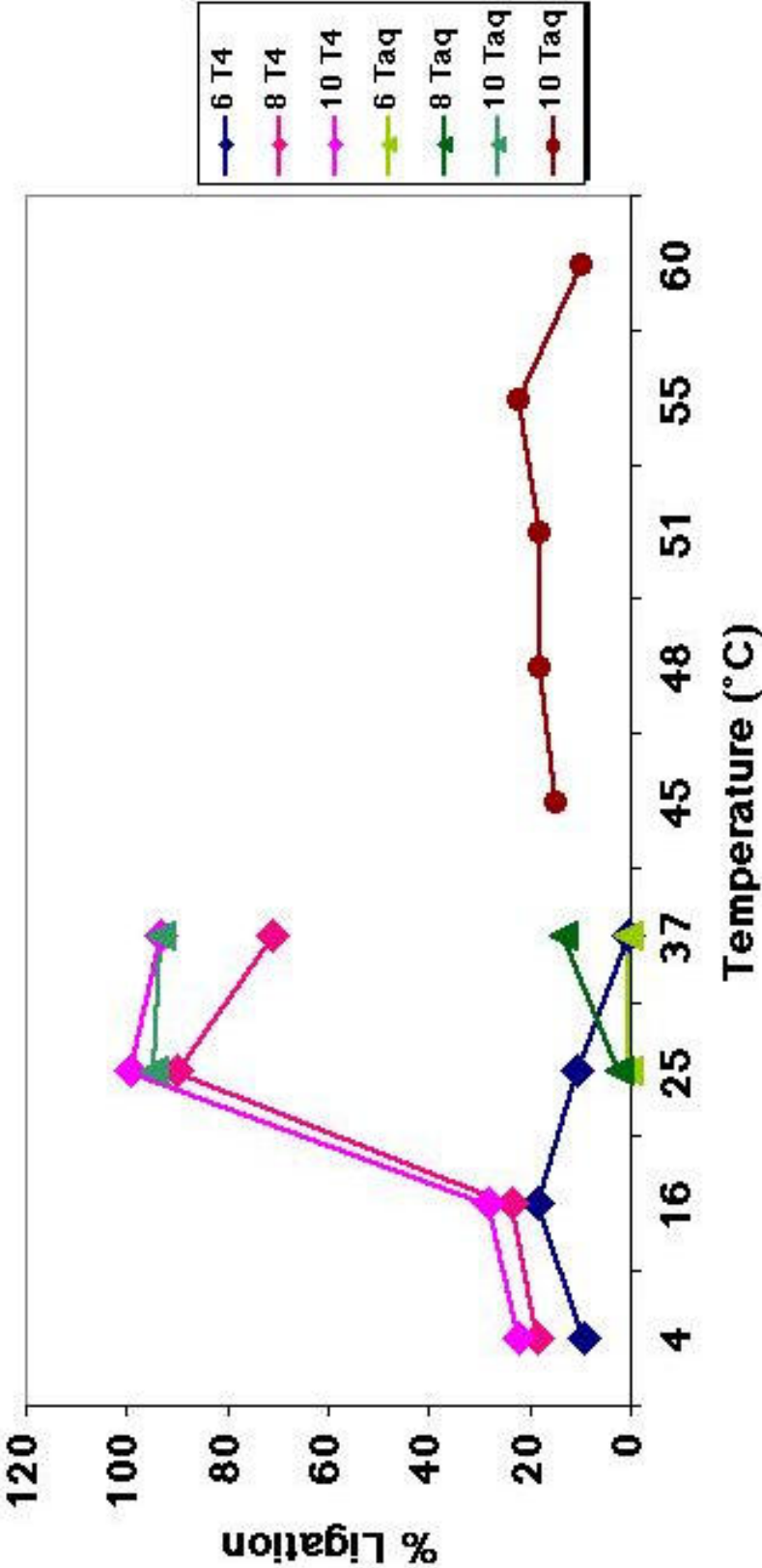
3. Amplification of ligation products by PCR

4. Treatment of PCR products with restriction enzymes and gel electrophoresis

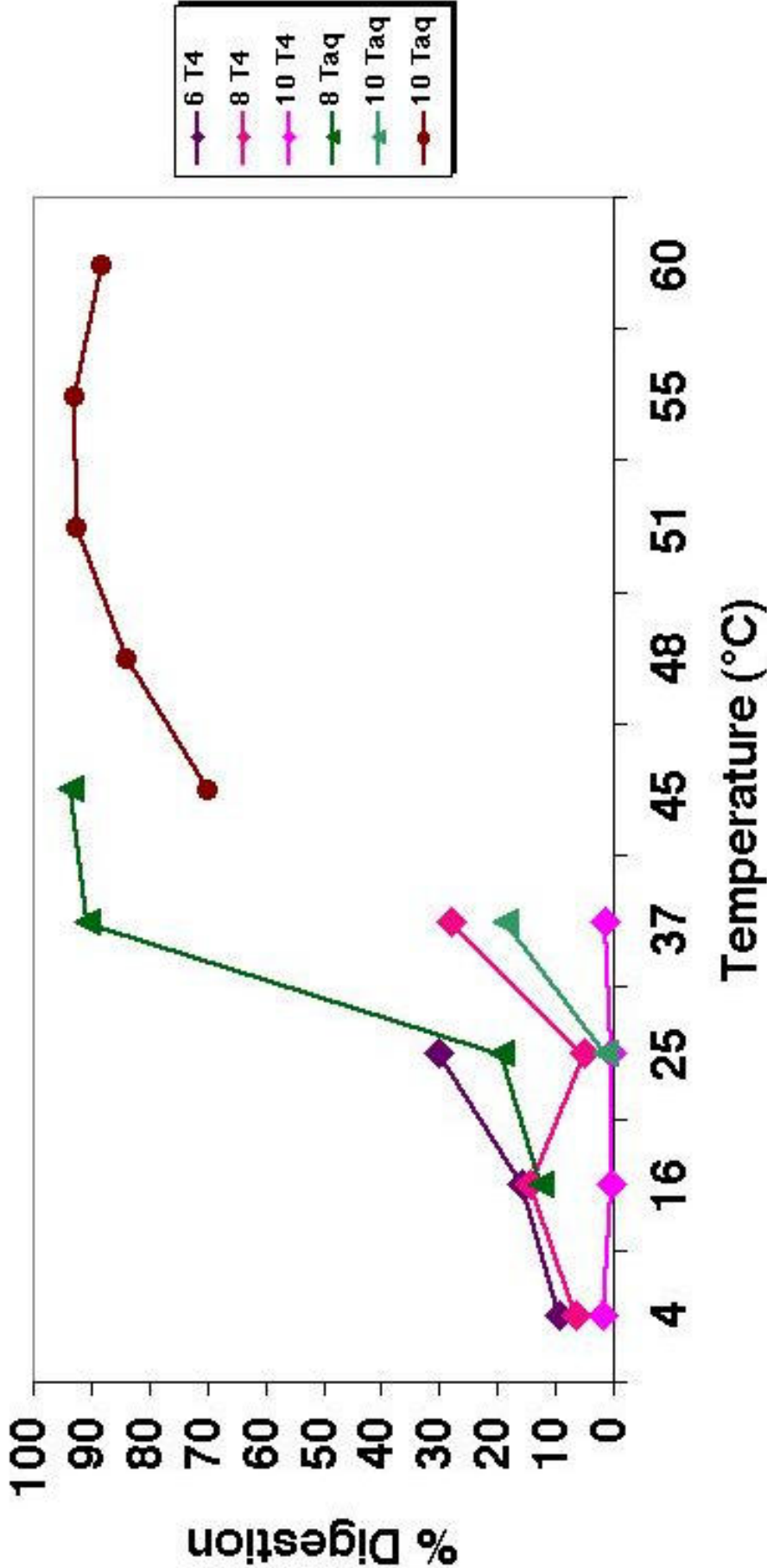


Fidelity

# Ligation Efficiency



# Ligation Fidelity



## Summary

**Ligases tolerate mismatches.**

**Crucial parameters are:**

- **position relative to site of ligation;**
- **length of the overlap -- higher fidelity for short oligonucleotides;**
- **annealing temperature -- higher fidelity for higher temperature;**
- **time -- higher fidelity for shorter times.**



## **Conclusions**

**High fidelity might be achieved at the expense of efficiency.**

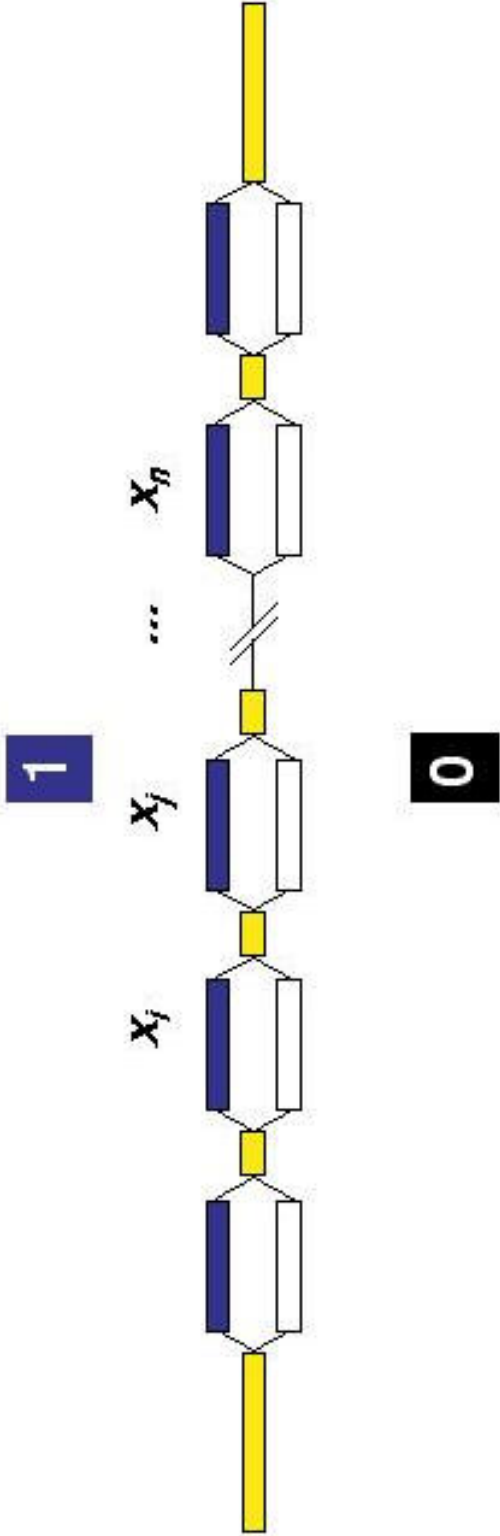
**Both efficiency and fidelity need to be optimized for DNA computation.**

**Algorithms and encodings must be robust.**

# RNA Solutions to SAT

---

# Model of a Binary Nucleic Acid Library

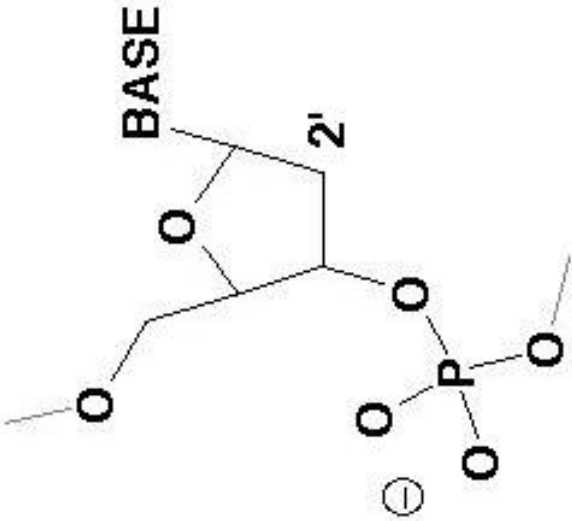


## **Destruction vs. Extraction**

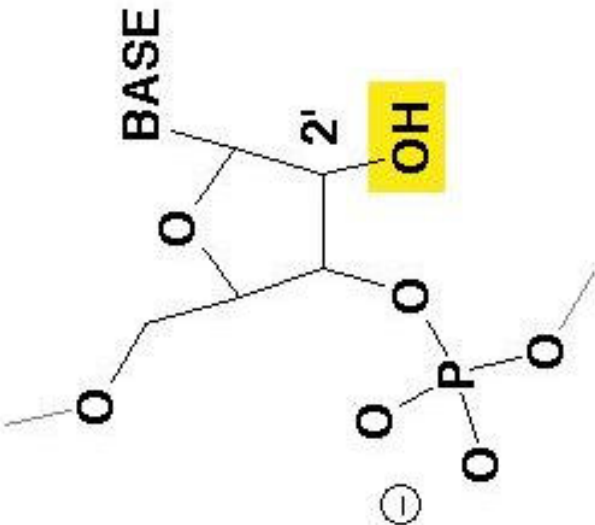
- 1 pmol of a 1,000 strand library contains  $6 \times 10^8$  copies of each strand.
- The main error is the presence of contaminating DNA.

### **Two possibilities:**

- Isolate solutions by extraction hybridization (biotin/avidin).
- Remove contaminating DNA by destruction (restriction enzymes).



DNA

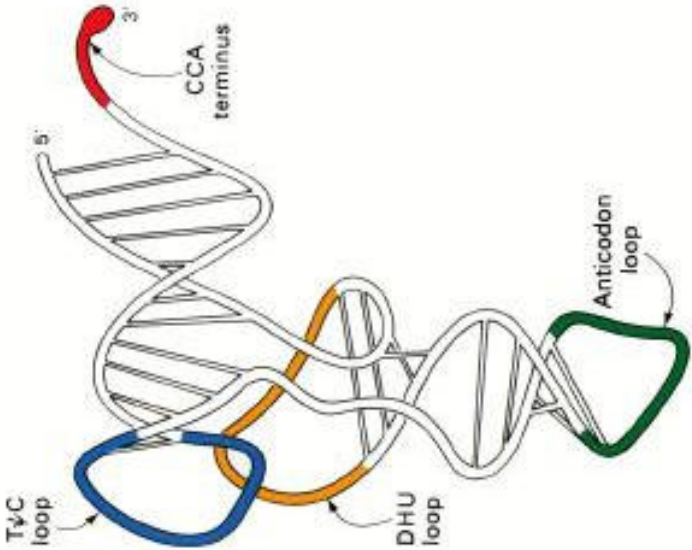
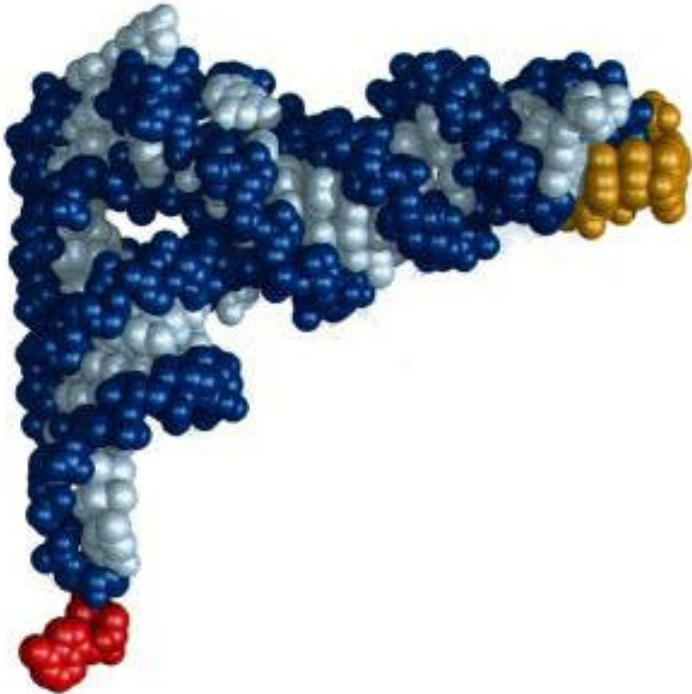


RNA

## RNA Is Suited for a Destructive Algorithm...

---

- **...because:**
- RNA's 2' hydroxyl group makes it more prone to hydrolysis.
- RNA strands can be marked for destruction by introducing complementary DNA oligonucleotides.
- Rnase H serves as a "universal RNA restriction enzyme".



# Design of an 10-bit RNA Library

---

## Significant complexity (1,024 molecules)

5' PREFIX Bit 1 (0/1) | Spacer 1 | Bit 2 (0/1) | Spacer 2 |...| Spacer 9 | Bit 10 (0/1) | SUFFIX 3'

## The PERMUTE program

### Criteria:

- Bits must be fundamentally different (maximal Hamming Distance).
- Average melting temperature 45 °C.
- Minimal secondary structure: three letter alphabet (A, C, U).
- Avoid intra- and/or intermolecular hybridization.



# Sequences for Bits and Spacers

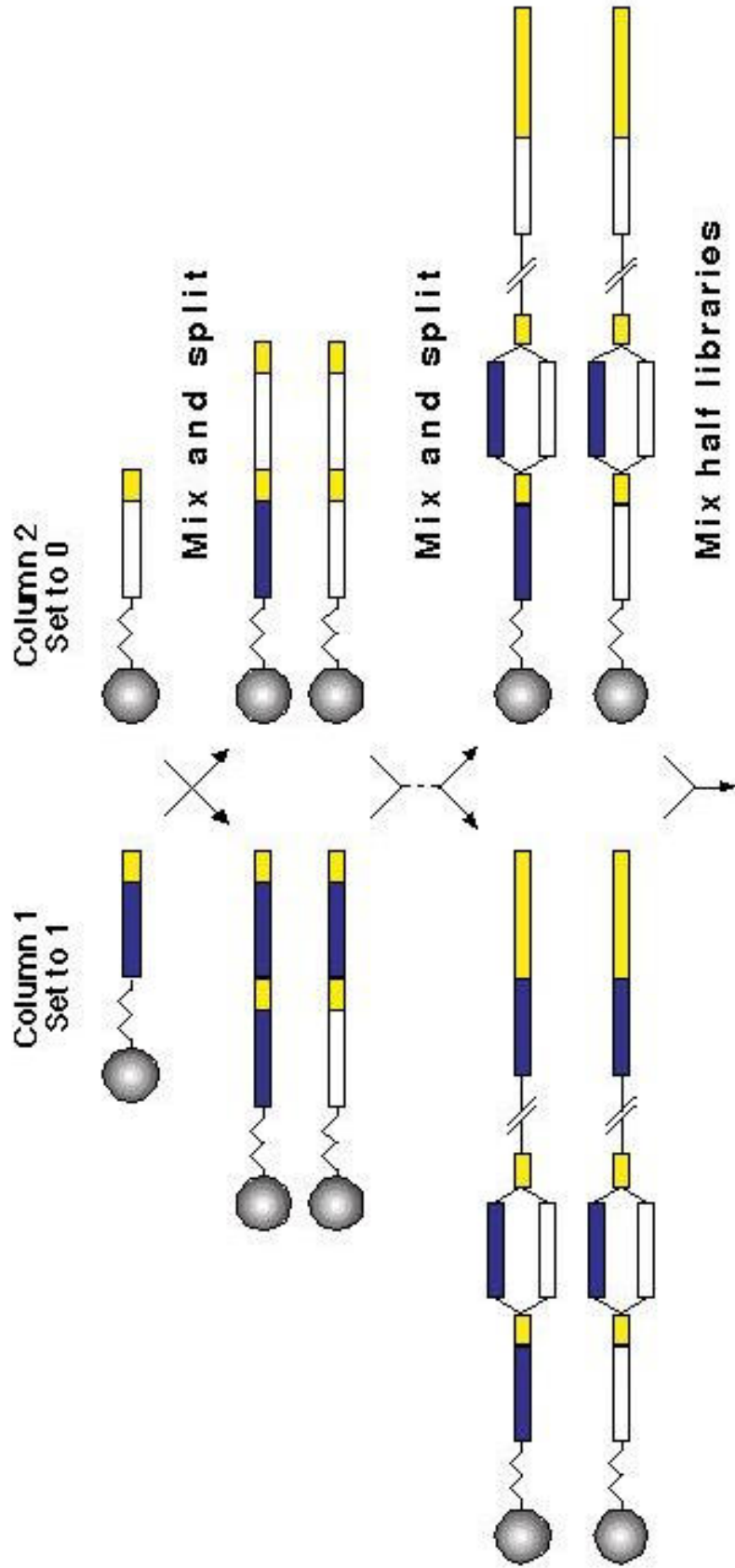
Bit	Knight Absent <i>Bit set to 0</i>	Knight Present <i>Bit set to 1</i>	Spacer
<i>a</i>	CTCTTACTCAATTCT	TCCTCACATTACTTA	TCTAC
<i>b</i>	CATAATCAACATCTTA	ACTTCCCTTATATCC	ATAAC
<i>c</i>	ATCCTCCACTTCACA	TTATAACAACATCC	CTTAA
<i>d</i>	TTAAAAATCTTCCCTC	ACATAACCCTCTTCA	TTTAC
<i>e</i>	CATTTTATCCACATT	ACCTTACTTCCATA	TACAA
<i>f</i>	GCTTCAAACAATTCC	GTACATTCTCCCTAC	TCCTT
<i>g</i>	AACTCTCAAATTCAA	CATAATCTTATATTC	TCAAT
<i>h</i>	CTAACCTTTACTTCA	ATAATCACATACTTC	TCCAA
<i>i</i>	CATTCCCTTATCCCAC	TCCACCAACTACCTA	ACACA
<i>j</i>	CACCCTTCTCCTCT	TTTTAAATTTACAAA	SUFFIX

# Library Construction by Assembly PCR

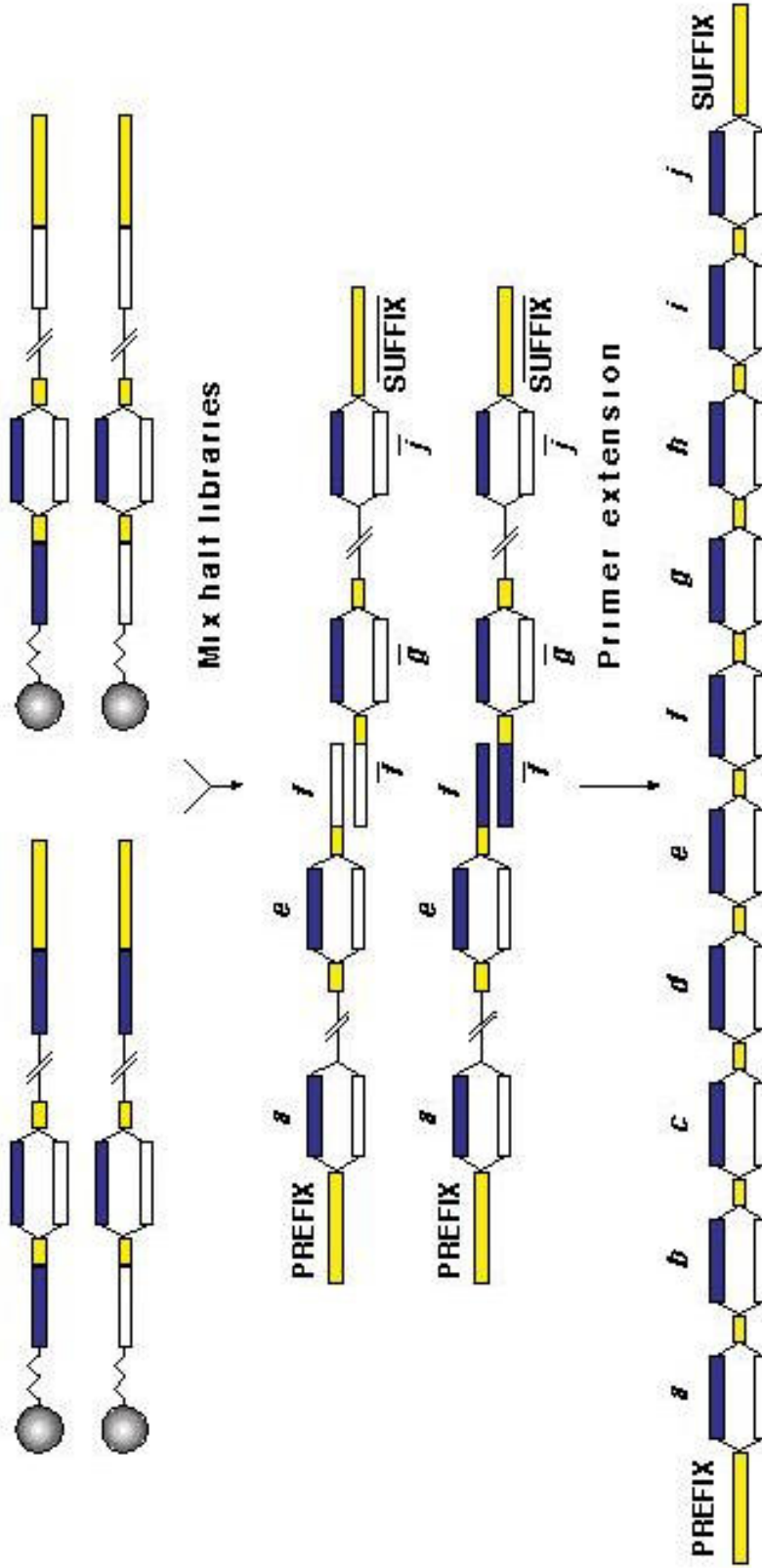
---



# Mix and Split Synthesis



# Mix and Split Synthesis



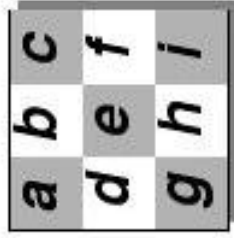
## The 'Knight Problem'

---

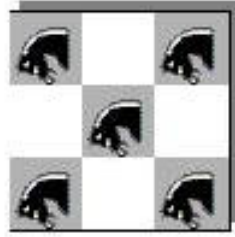
- Given an  $n \times n$  chess board, what position can a knight occupy such that no knight can attack another knight.
- An example for SAT.
- NP-complete for infinite boards.

# The 'Knight Problem': 3 x 3 Board

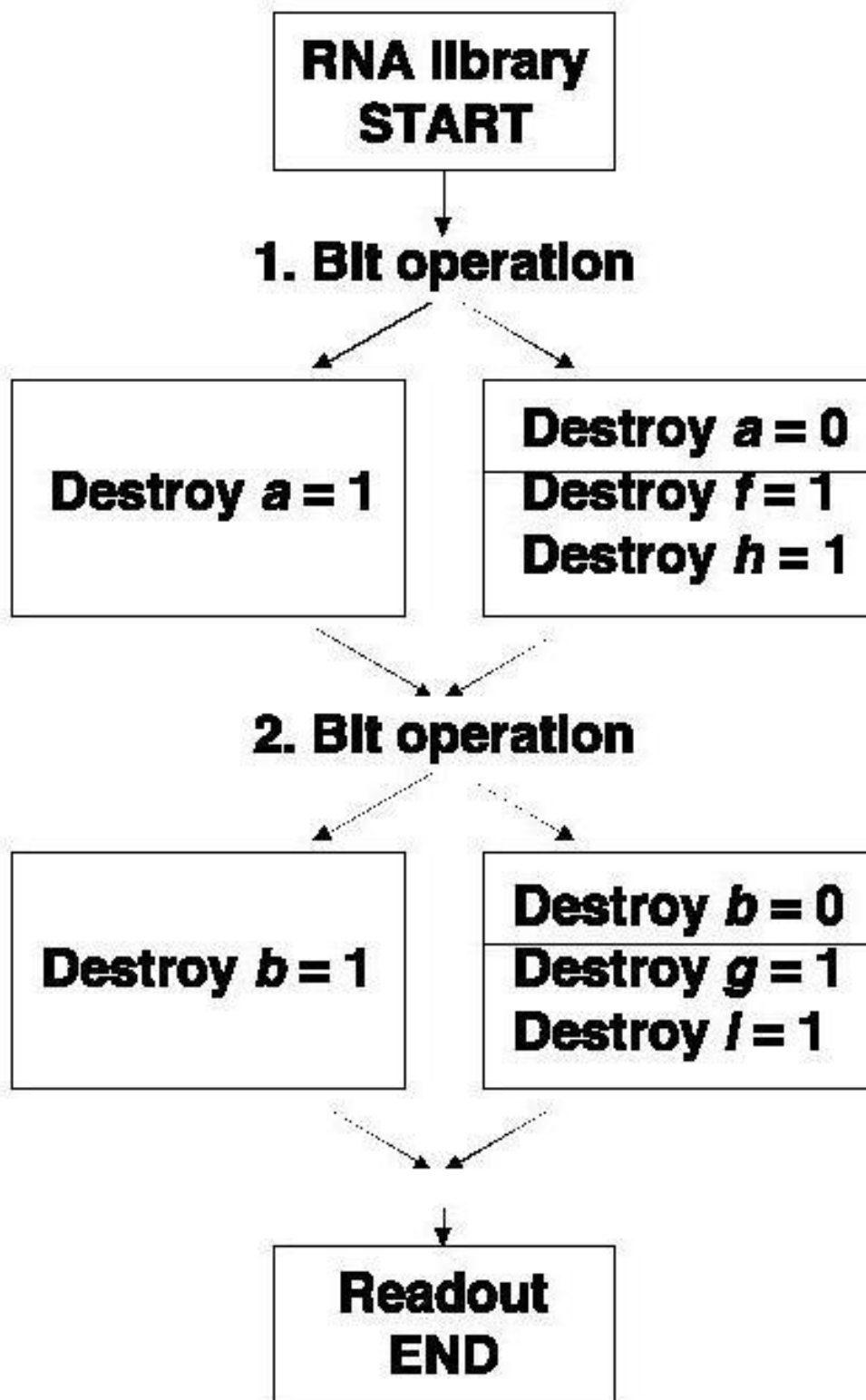
---



$$((\neg h \wedge \neg f) \vee \neg a) \wedge (\neg g \wedge \neg f) \vee \neg b) \wedge ((\neg d \wedge \neg h) \vee \neg c) \wedge ((\neg c \wedge \neg f) \vee \neg d) \wedge (\neg a \wedge \neg g) \vee \neg f) \wedge$$
$$(\neg b \wedge \neg f) \vee \neg g) \wedge ((\neg a \wedge \neg c) \vee \neg h) \wedge (\neg d \wedge \neg b) \vee \neg i)$$



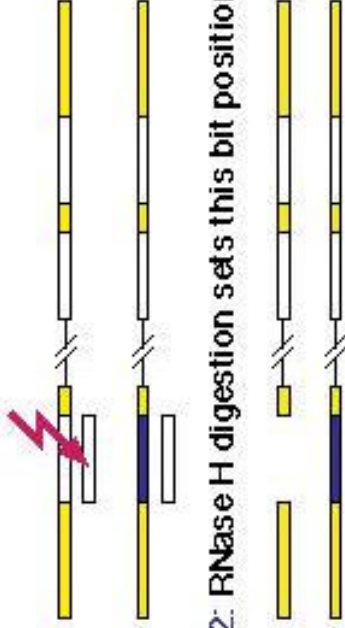
# The RNA Algorithm



# Implementing the RNA Algorithm

---

**Step 1:** Set bits to 1 or 0. Split the RNA pool into 2 tubes and anneal DNA oligonucleotides complementary to 'bits' to be digested. (For example, if bit position  $a$  is set to 1, then bit positions  $a$  and  $a+1$  must be set to 0.)

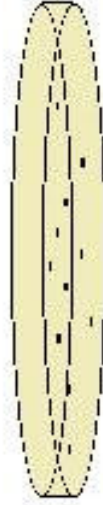


**Step 2:** RNase H digestion sets this bit position to 1.

**Step 3:** Remove RNA oligonucleotides and short RNA products by PAGE; reamplify RNA by RT-PCR and in vitro transcription.

**Step 4:** Combine the 2 tubes and repeat steps 1 - 3 for the next bit position.

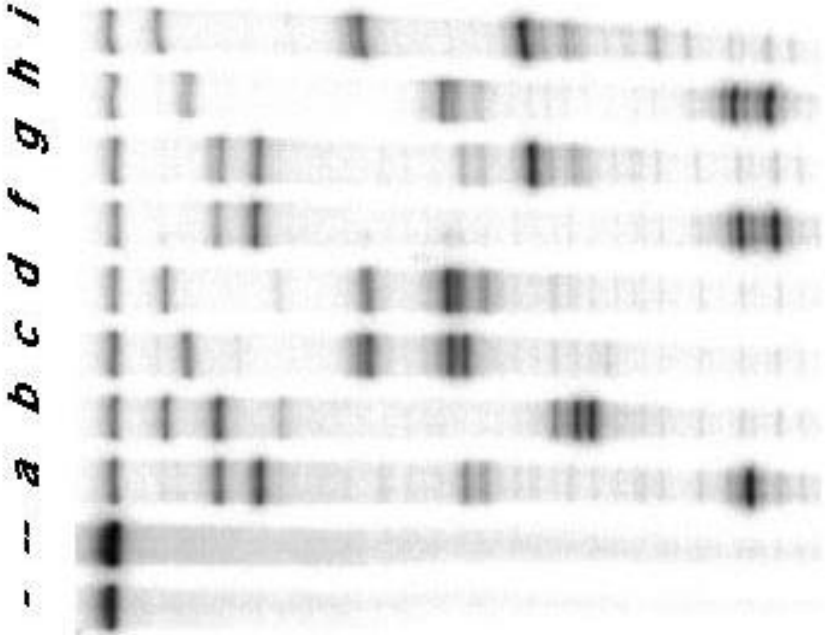
**Step 5:** After the last digestion, RT-PCR and clone representative molecules from the enriched pool.



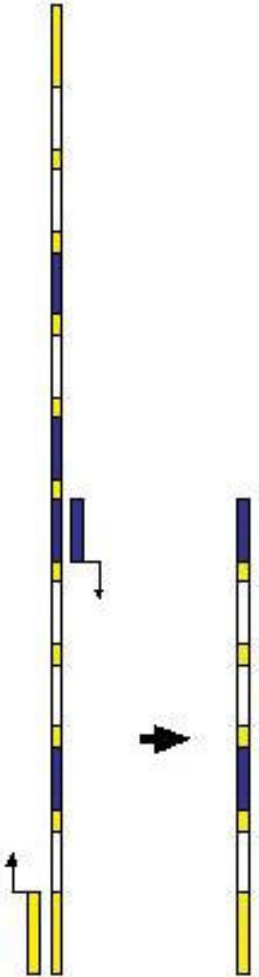


# Implementing the RNA Algorithm

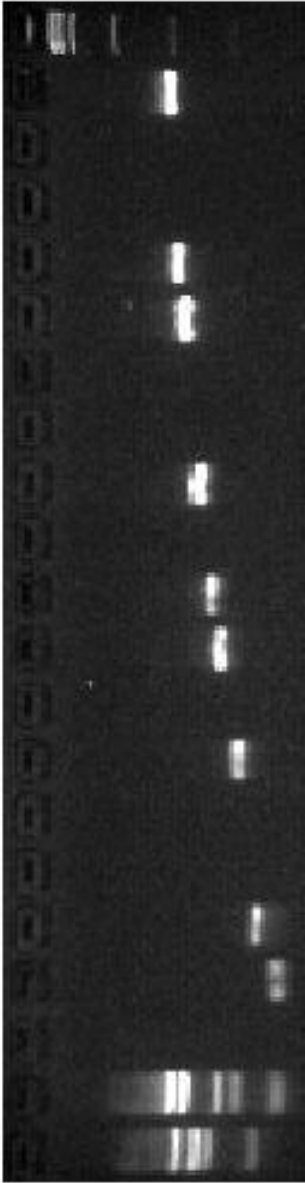
---



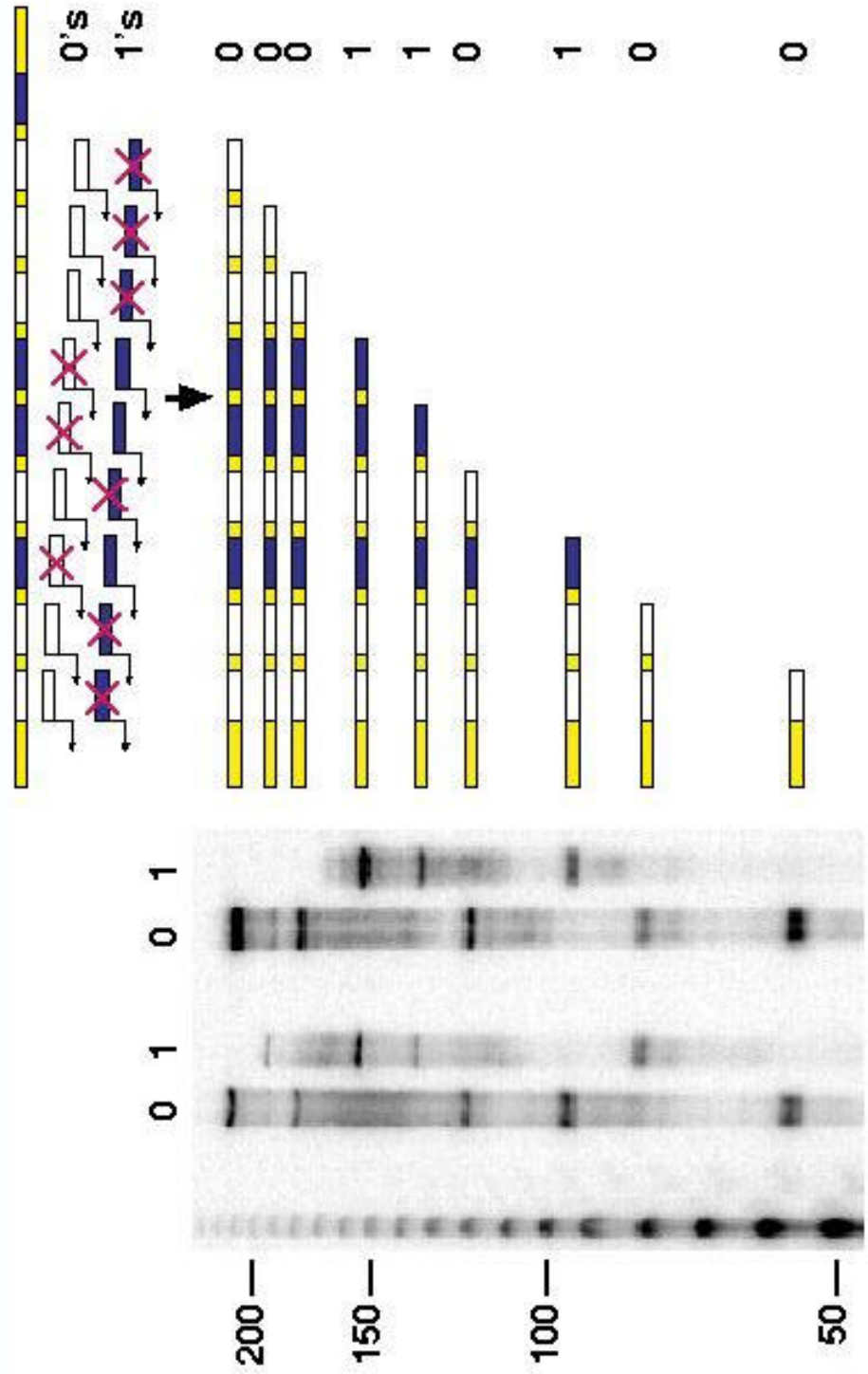
# Readout of a Single Clone by PCR



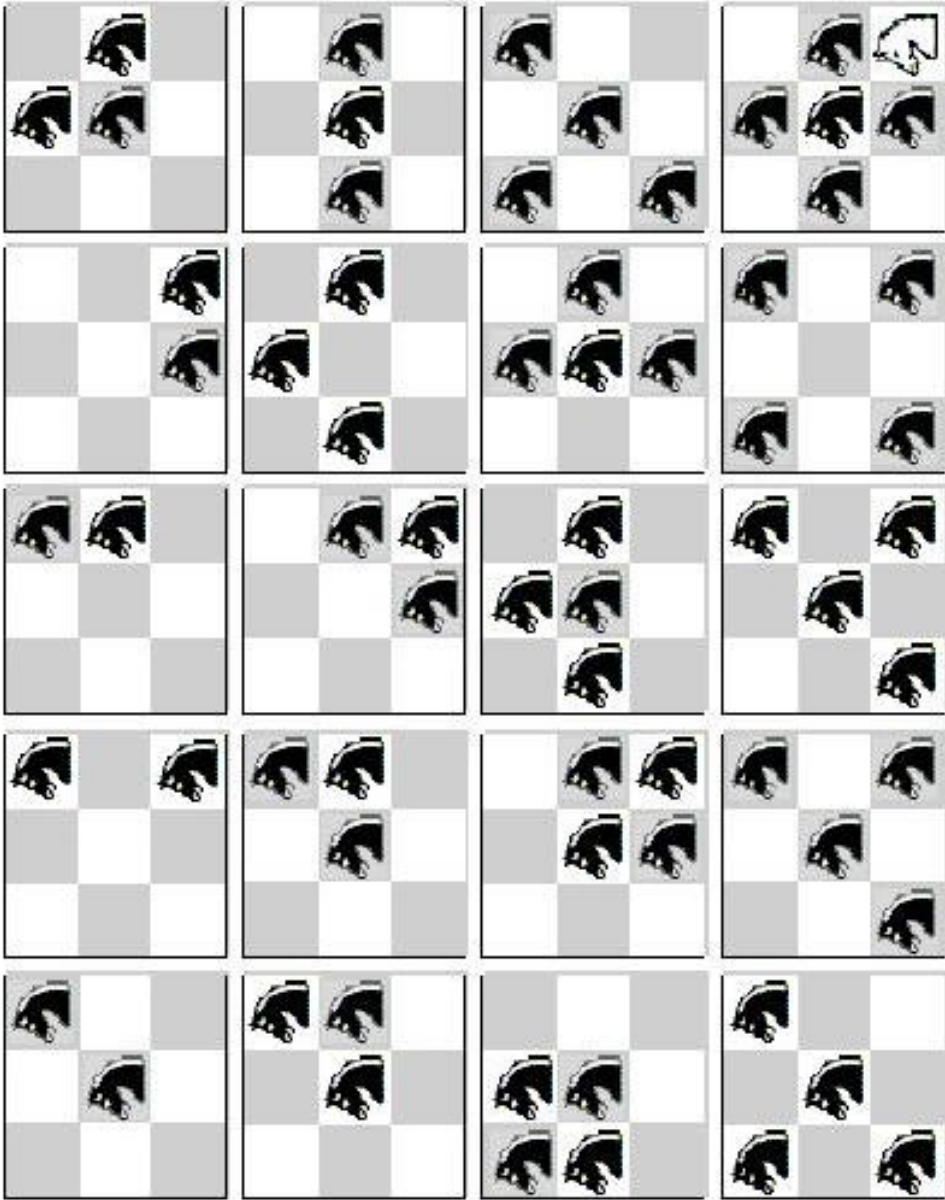
*a* *b* *c* *d* *e* *f* *g* *h* *i*  
1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0



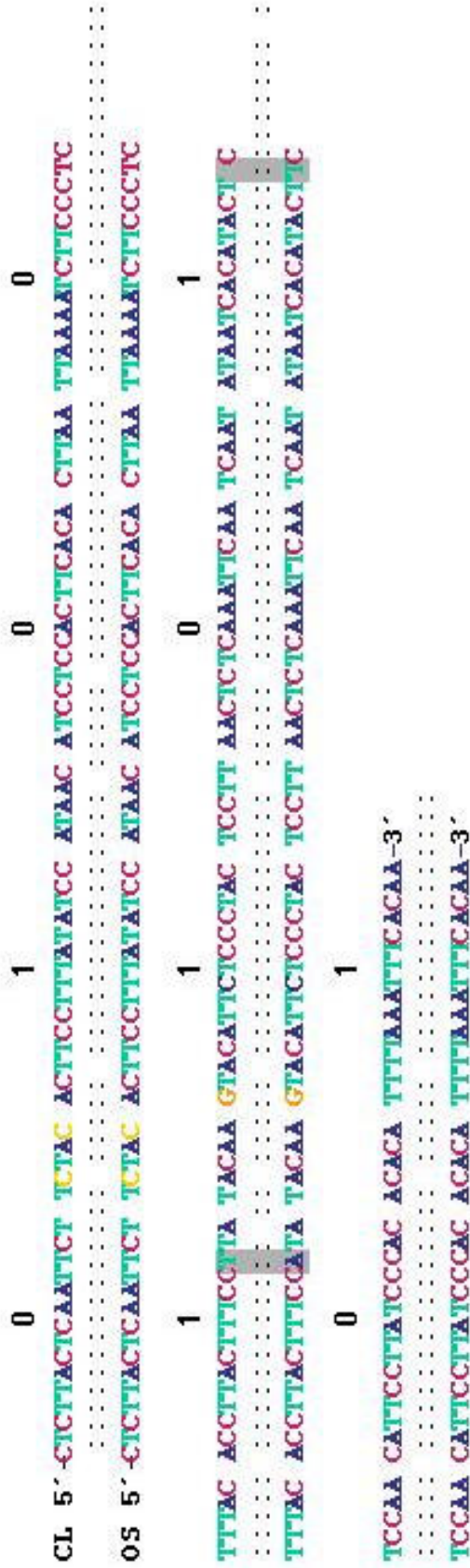
# Readout by Multiplex Primer Extension



# 19 Solutions to the 'Knight Problem'



## The RNA Algorithm and Encoding are Robust



## Summary

- 19 out of 20 clones (= 67 out of 68 knights) are correct solutions.
- In more than 95 % the answer is correct.
- Readout by multiplex primer extension is fast and efficient.
- Algorithm and encoding are robust.

# Thanks to:



**Laura Landweber**  
**The Landweber lab**  
**Richard J. Lipton**  
**Anthony R. Cukras**  
**Saw Kyin**  
**Steve Lawrie**  
**Vernadette Simon**  
**Erik Winfree**