DNA Computing

Dr Giuditta Franco

Department of Computer Science, University of Verona, Italy
giuditta.franco@univr.it

Overview

Information is stored in bio-polymers, enzymes (molecular biology and genetic engineering techniques) manipulate them in a *massively parallel way*, according to strategies producing **universal computation** [G.F., M. Margenstern, TCS 404, 2008].

$$\text{Input data} \overset{encoding}{\longleftrightarrow} \text{DNA} \quad \overset{bio\text{-}steps}{\rightsquigarrow} \quad \text{DNA} \overset{decoding}{\longleftrightarrow} \text{Output data}$$

Operations designed over a (*dry*) multiset of strings in Σ^*, performed over a (*wet*) pool of DNA sequences: content of a test tube.

Main goals of DNA Computing

- Efficiently solving NP-Complete problems (*initial* goal: Lipton, Jonoska, Sakamoto..)
- Innovative procedures, namely for biotechnological applications (*recent* trend: Komiya, Manca, Reif, Yamamura).
- Theoretical models of DNA computation (*traditional* trend: Head, Rozenberg, Benenson, Keinan, Seeman..)
- DNA Self-assembly process (*vivid* trend: Jonoska, Seeman, Winfree..)
- Encoding issues, generated by experimental mismatch problems (*basic* trend: Jonoska, Kari, Rothemund..).

Solving NP complete problems

DNA = {nano + computing} material

Computations by linear number of bio-steps

- Solution of a toy size Hamiltonian Path Problem (*Adleman 1994*)
- Solution of 3×3 Knight problem (*Faulhammer et al. 1998*)
- Solution of 20 variable 3-SAT problem (*Braich et al. 2002*)
- Solution of a max clique problem (*Ouyang et al. 1997*), max independent set (*Head et al. 2000*)
- Several other instances of toy size solutions and techniques demonstrations.

Operations on DNA pools

1. Mix, split, heat, cool: pairing/unpairing (hybr./den.)

   $P := Mix(P_1, P_2); (P_1, P_2) := Split(P); P := H(P); P := C(P)$

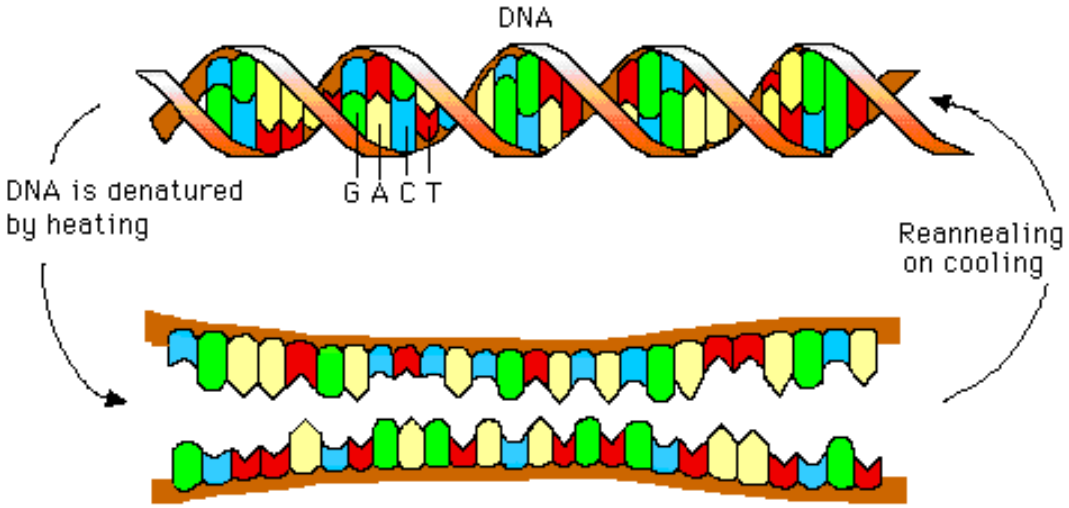
2. Enzymatic operations (*efficiency 100%*): cuts, nick repair, chemical modifications (methylation, phosphorilation, oxydrilation), writing/synthesis as guided elongation
3. Lengths measure, by gel electrophoresis + + recover of sequences from the gel (eluition, *scarse efficiency*)
4. amplification of (sub)strings
5. Reading sequences (by sequencing algorithms)

6. Extraction/Separation (*efficiency 85%*)

Heating/Cooling

Manchester Metropolitan University

The molecule of life



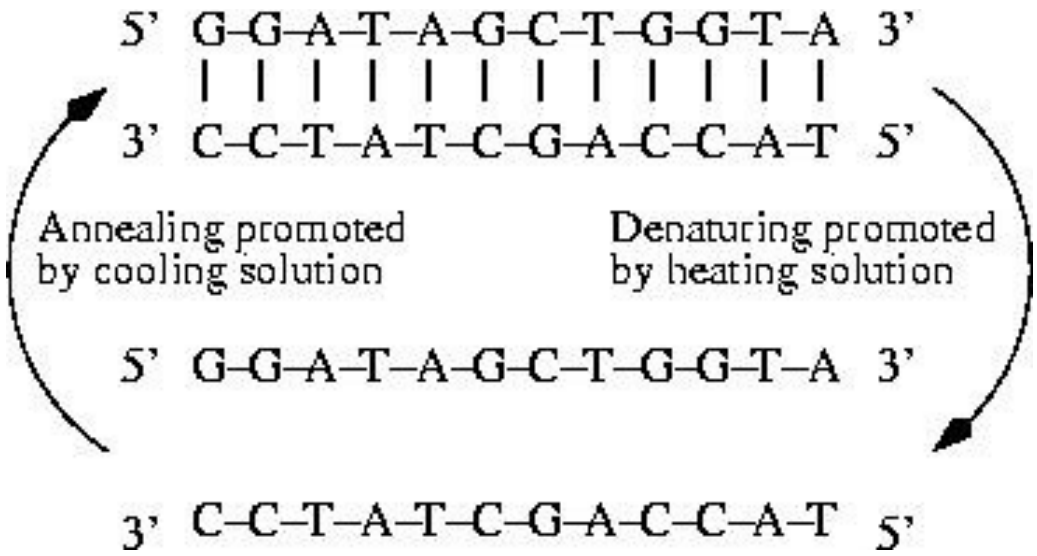
Access Excellence

novel computation group

Slide courtesy of prof. Martin Amos, Manchester Metropolitan University, UK

Heating/Cooling

Another view



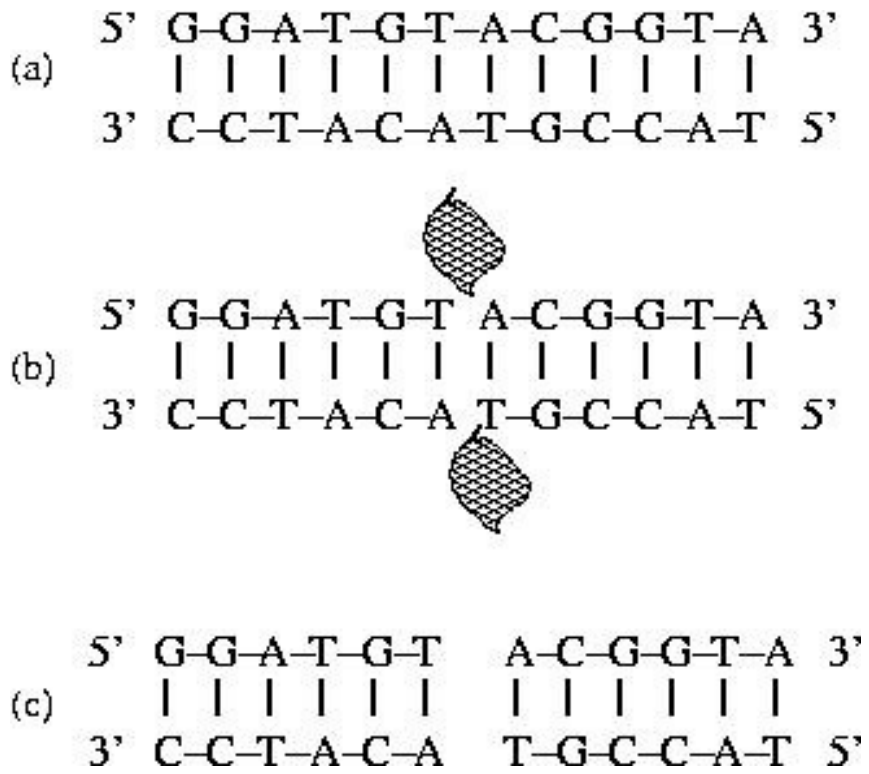
novel computation group

Slide courtesy of prof. Martin Amos, Manchester Metropolitan University, UK

Cut by endonucleases

Manchester Metropolitan University

Restriction enzymes



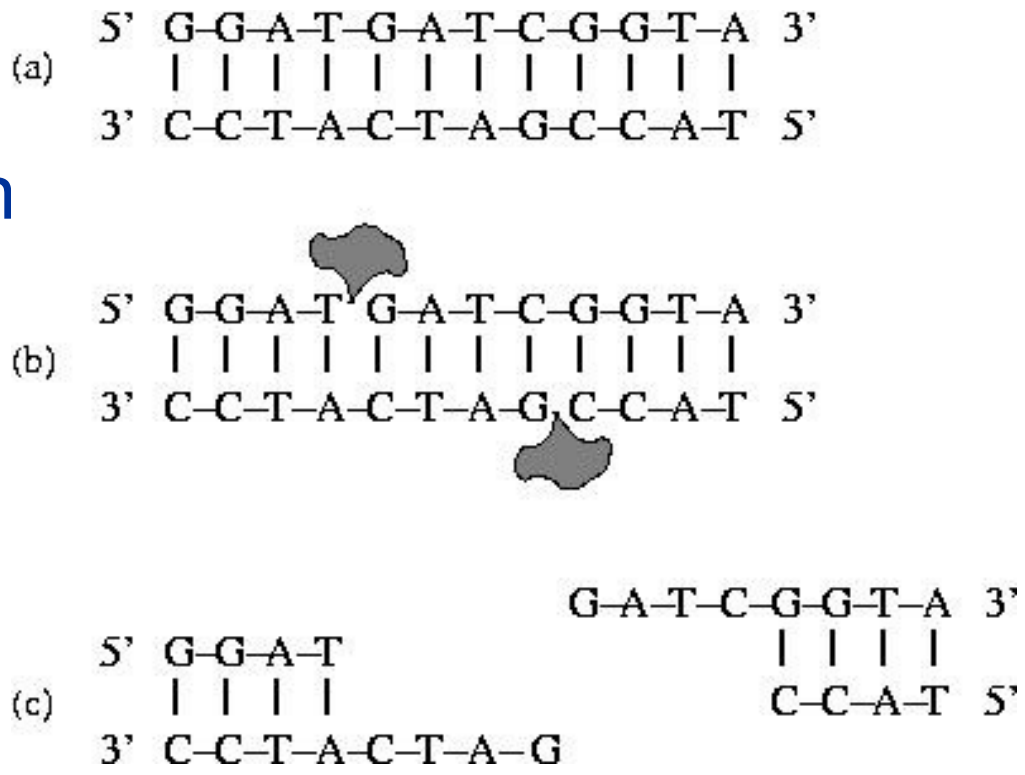
novel computation group

Slide courtesy of prof. Martin Amos, Manchester Metropolitan University, UK

Cut by endonucleases

Manchester Metropolitan University

Restriction enzymes



novel computation group

Slide courtesy of prof. Martin Amos, Manchester Metropolitan University, UK

Ligation

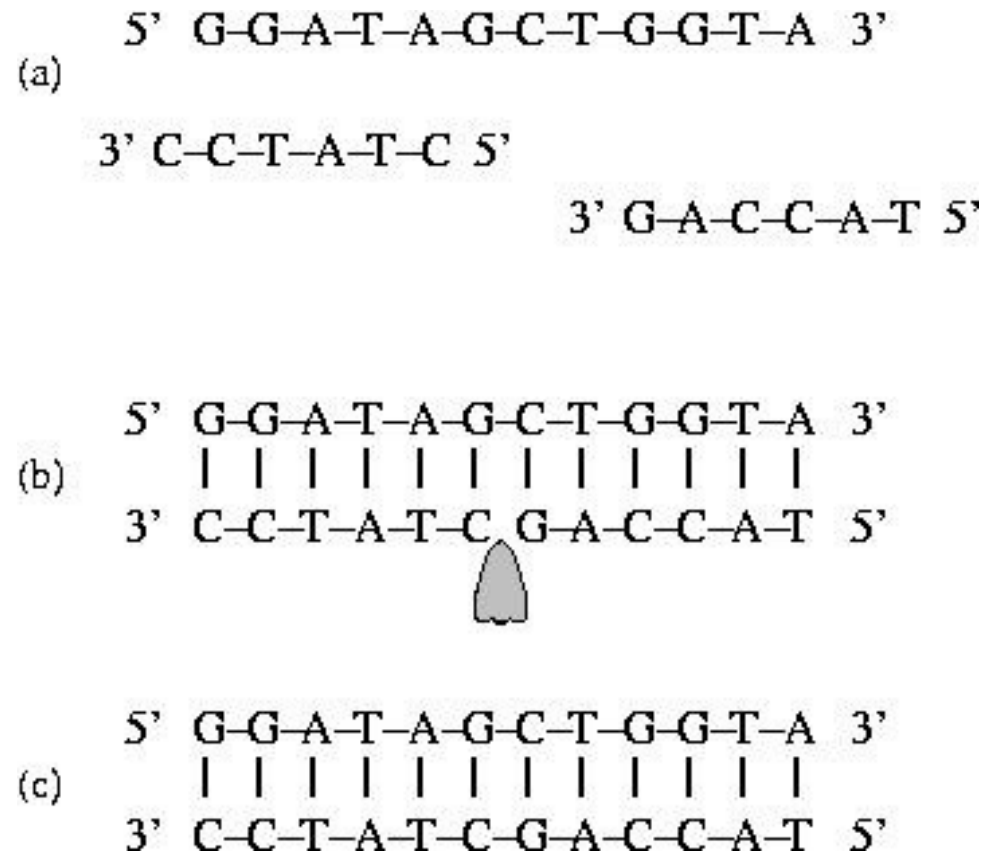
- If a single strand contains a “nick” in it, this is known as a *discontinuity*
- Can be repaired by a class of enzymes known as *ligases*
- Allows us to create double-stranded complexes out of several different single strands – important for later

Slide courtesy of prof. Martin Amos, Manchester Metropolitan University, UK

Concatenation by ligase

Manchester Metropolitan University

Ligase



novel computation group

Slide courtesy of prof. Martin Amos, Manchester Metropolitan University, UK

Operations on DNA pools

1. Mix, split, heat, cool: pairing/unpairing (hybr./den.)

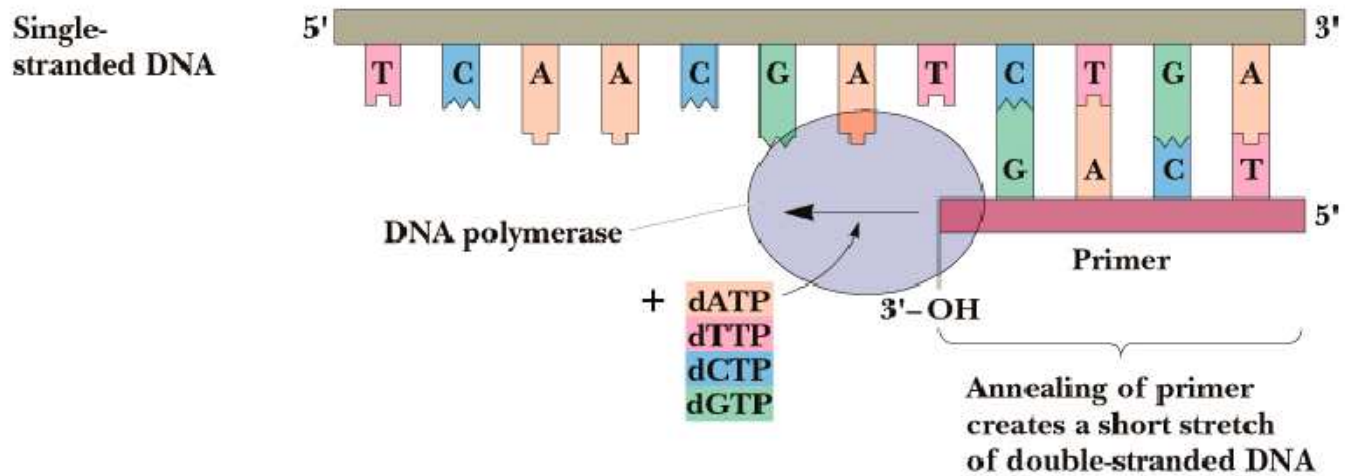
   $P := Mix(P_1, P_2); (P_1, P_2) := Split(P); P := H(P); P := C(P)$

2. Enzymatic operations (efficiency 100%): cuts, nick repair, chemical modifications (methylation, phosphorilation, oxydrilation), writing/synthesis as guided elongation
3. Lengths measure, by gel electrophoresis + recover of sequences from the gel (eluition)
4. amplification of (sub)strings
5. Reading sequences (by sequencing algorithms)

6. Extraction/Separation

Elongation by polymerase

Garrett & Grisham: Biochemistry, 2/e
Figure 12.2



Saunders College Publishing

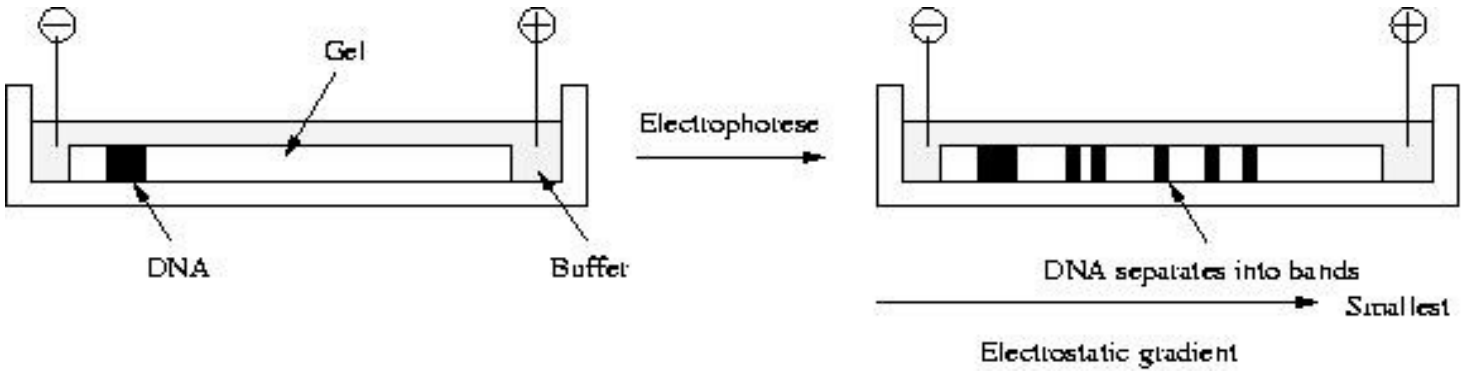
Sorting strands by length

It is used a technique known as gel electrophoresis (movement of molecules in a charged field)

DNA carries a negative charge: it tends to be attracted to the anode (positive charge)

Due to the friction with a gel (porous nature), strands move at a rate that is proportional to their length (longer strands move more slowly than short strands)

Gel electrophoresis



novel computation group

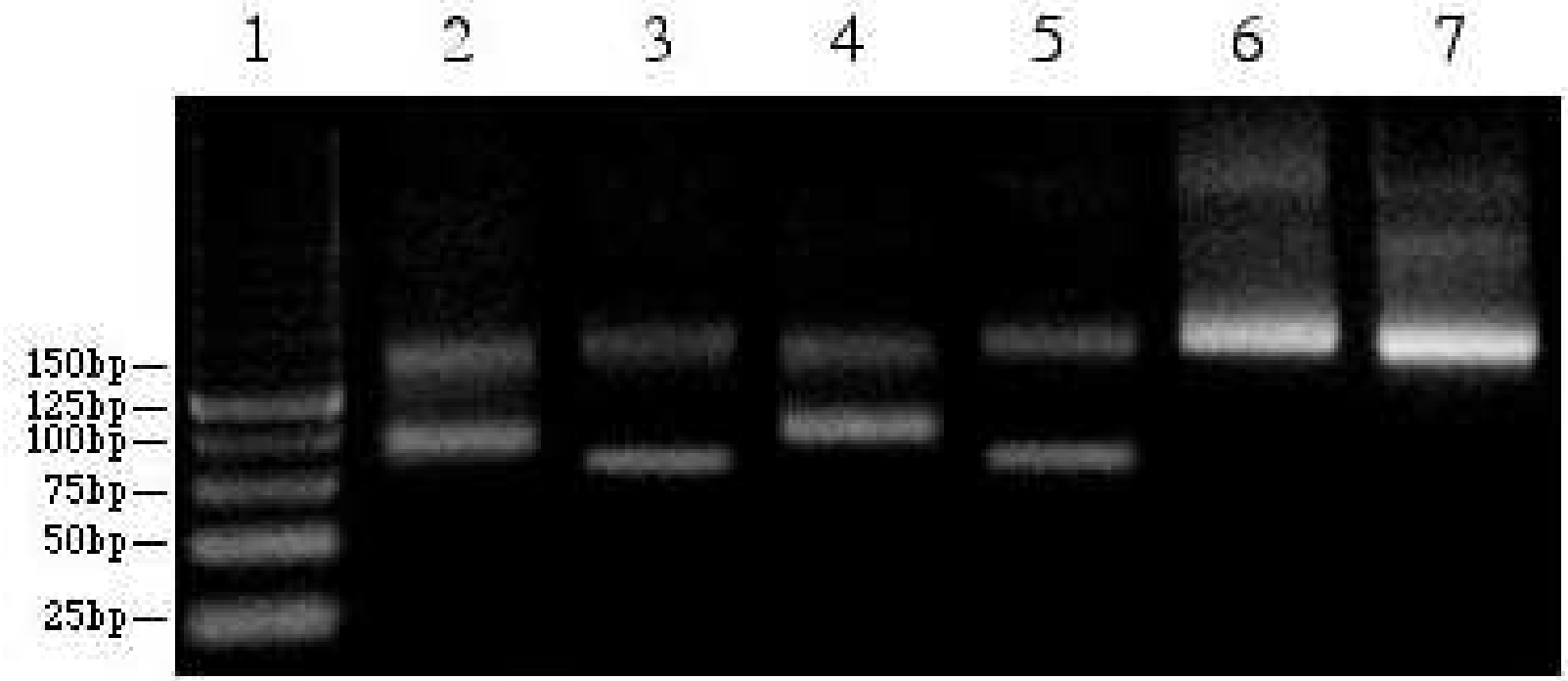
Slide courtesy of prof. Martin Amos, Manchester Metropolitan University, UK

Sorting strands by length

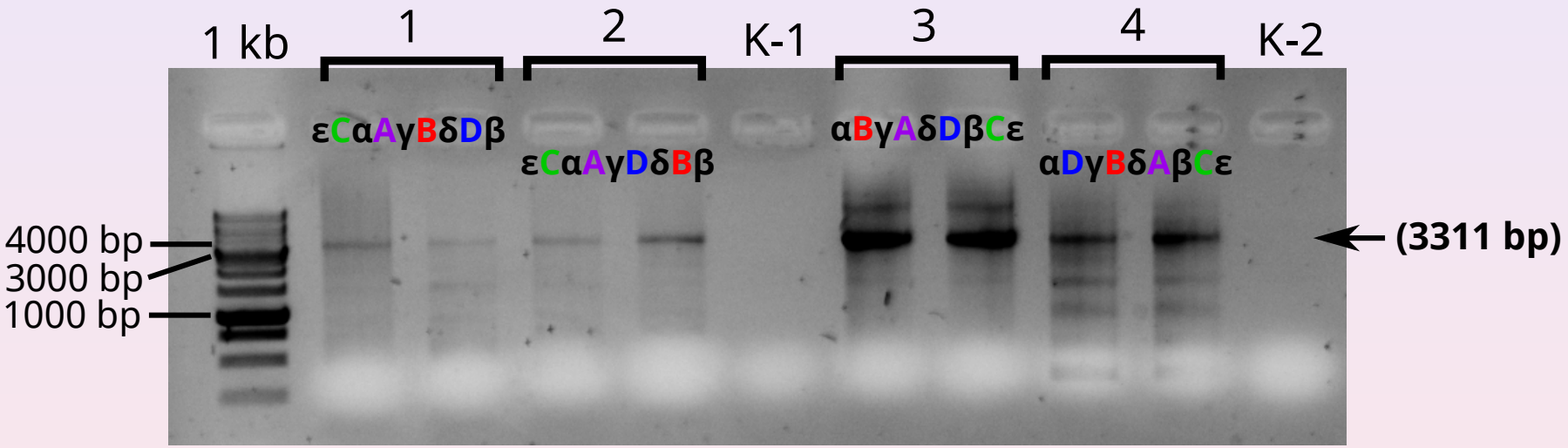
Once the gel has run, we can see (stained with a fluorescent dye) different bands of DNA under UV light

Eluition: We can cut out bands, thus retaining only DNA of a certain length; this can then be removed from the gel by soaking

Gel photograph



Gel photograph



Operations on DNA pools

1. Mix, split, heat, cool: pairing/unpairing (hybr./den.)

   $P := Mix(P_1, P_2); (P_1, P_2) := Split(P); P := H(P); P := C(P)$

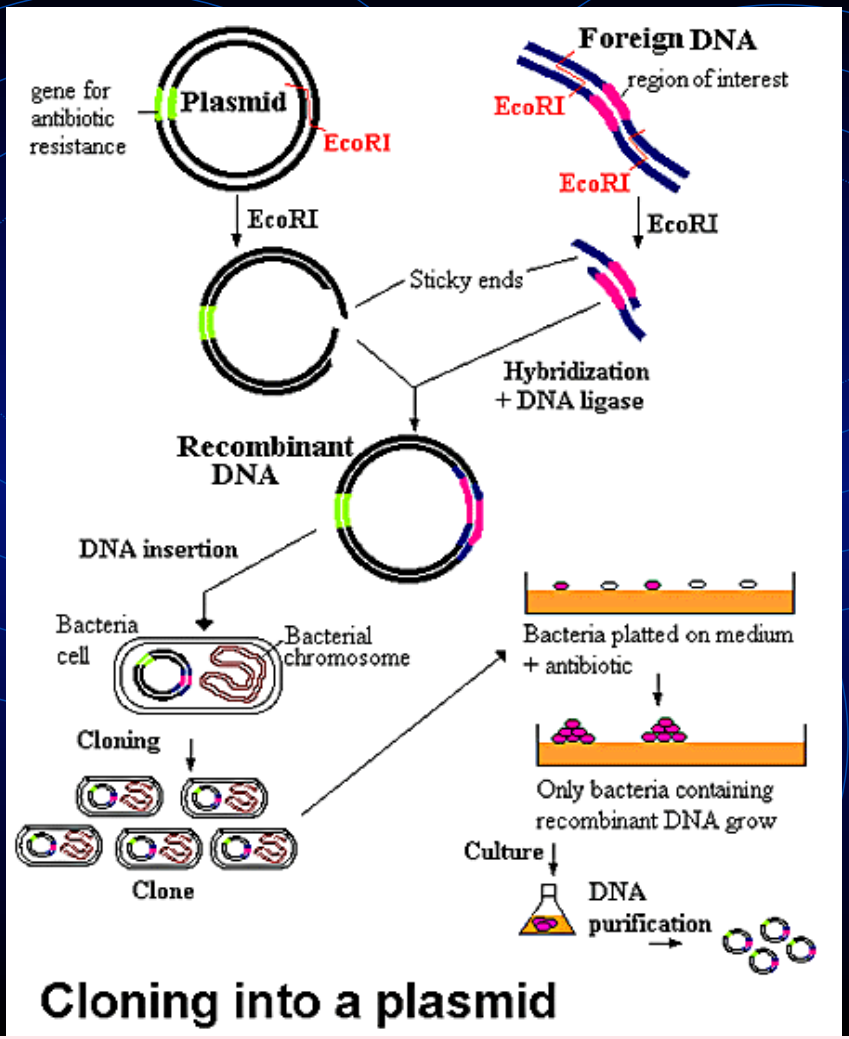
2. Enzymatic operations (efficiency 100%): cuts, nick repair, chemical modifications (methylation, phosphorilation, oxydrilation), writing/synthesis as guided elongation
3. Lengths measure, by gel electrophoresis + recover of sequences from the gel (eluition)
4. amplification of (sub)strings
5. Reading sequences (by NGS sequencing algorithms)

6. Extraction/Separation

Amplifying DNA sequences - two main methods

Cloning: for α-sequence long 2-200 kb.
Clone(α)

PCR (Polymerase Chair Reaction) : for α-sequence shorter than several thousand bps

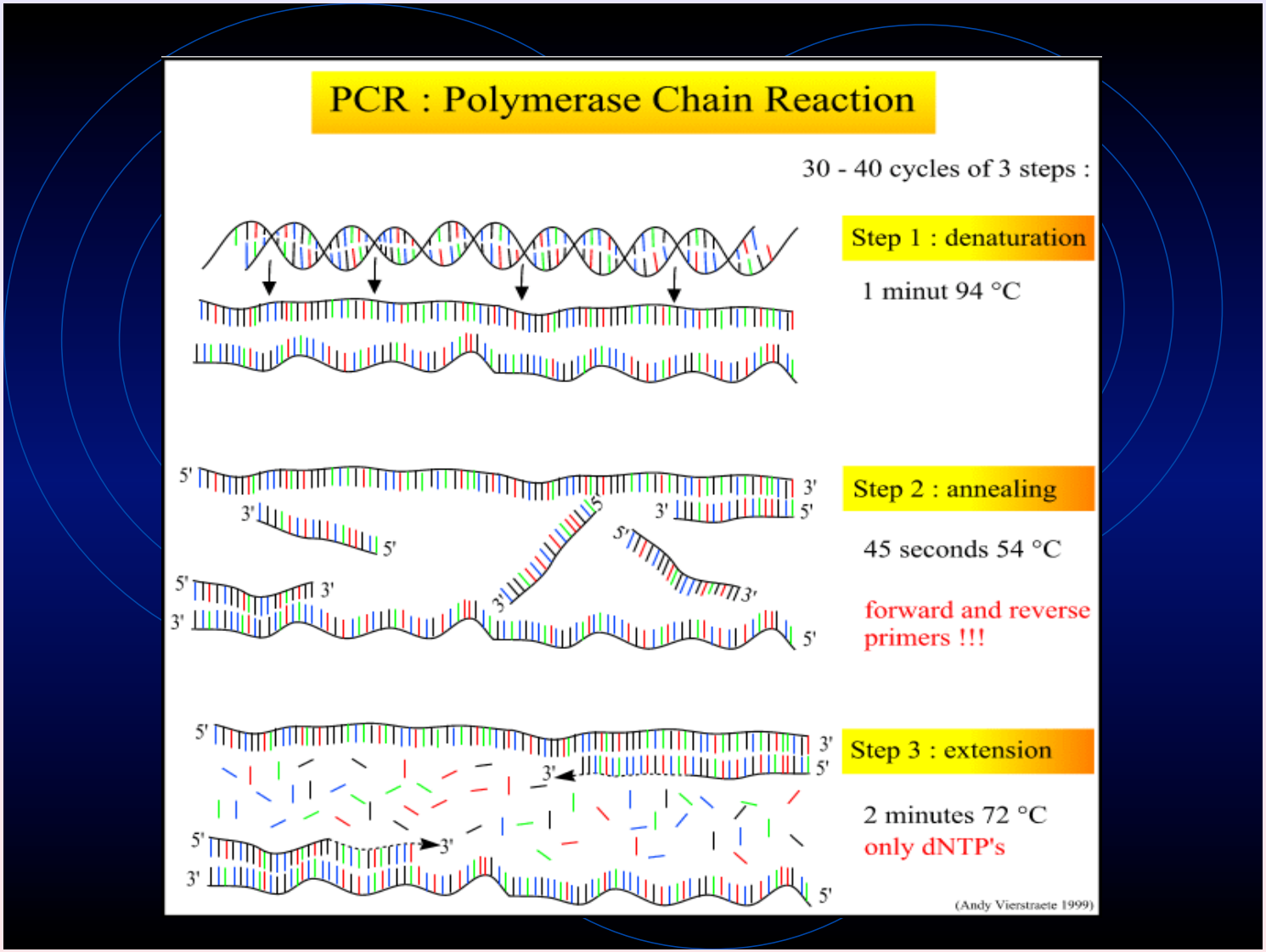


Slide courtesy of prof. Martin Amos, Manchester Metropolitan University, UK

DNA replication

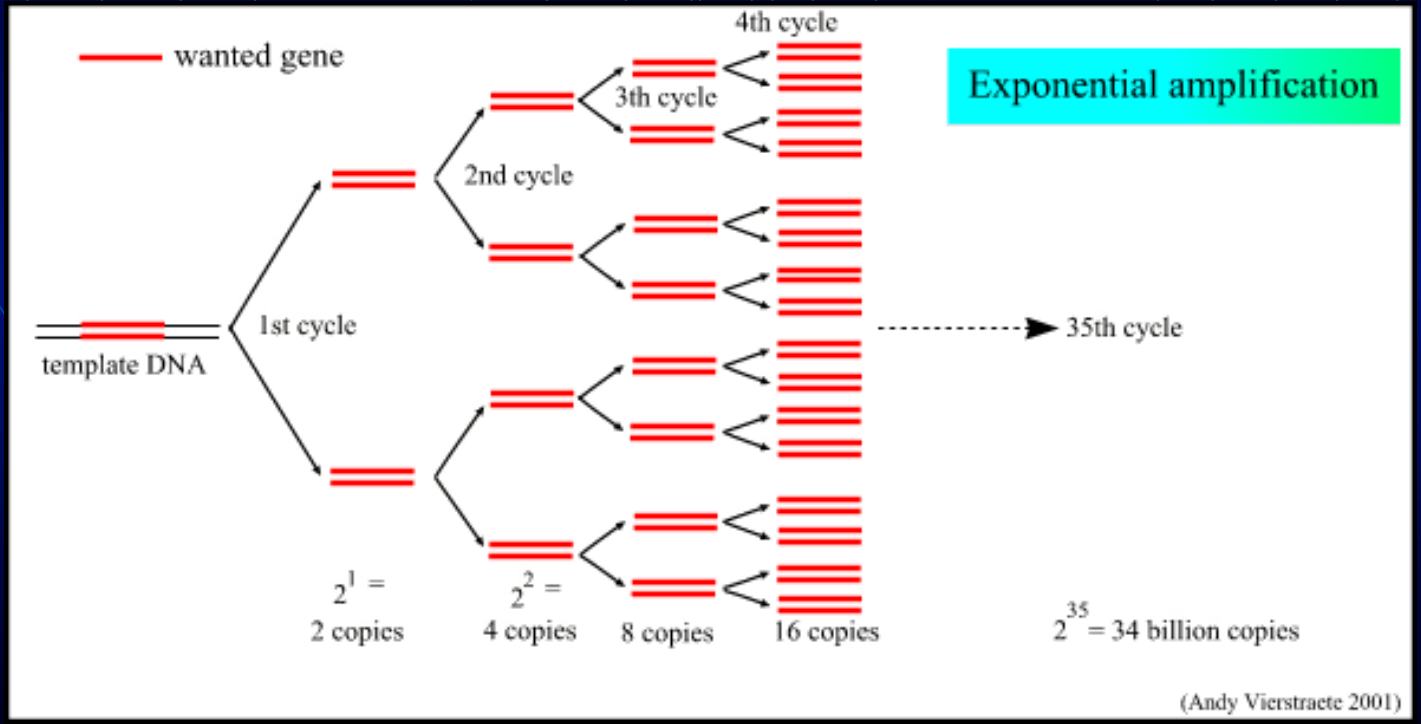
- DNA can also be replicated, taking a single molecule and multiplying it a thousand-fold (litres, if necessary)
- Useful in forensics, as well as in general molecular biology
- We use a technique known as the *polymerase chain reaction* (PCR)
- Kary Mullis, its inventor, won the Nobel Prize for its discovery
- Uses enzymes known as *polymerases*, which, given an “anchor” point and free bases (“spare nucleotides”), extend the anchor point, creating DS DNA as it goes

Slide courtesy of prof. Martin Amos, Manchester Metropolitan University, UK



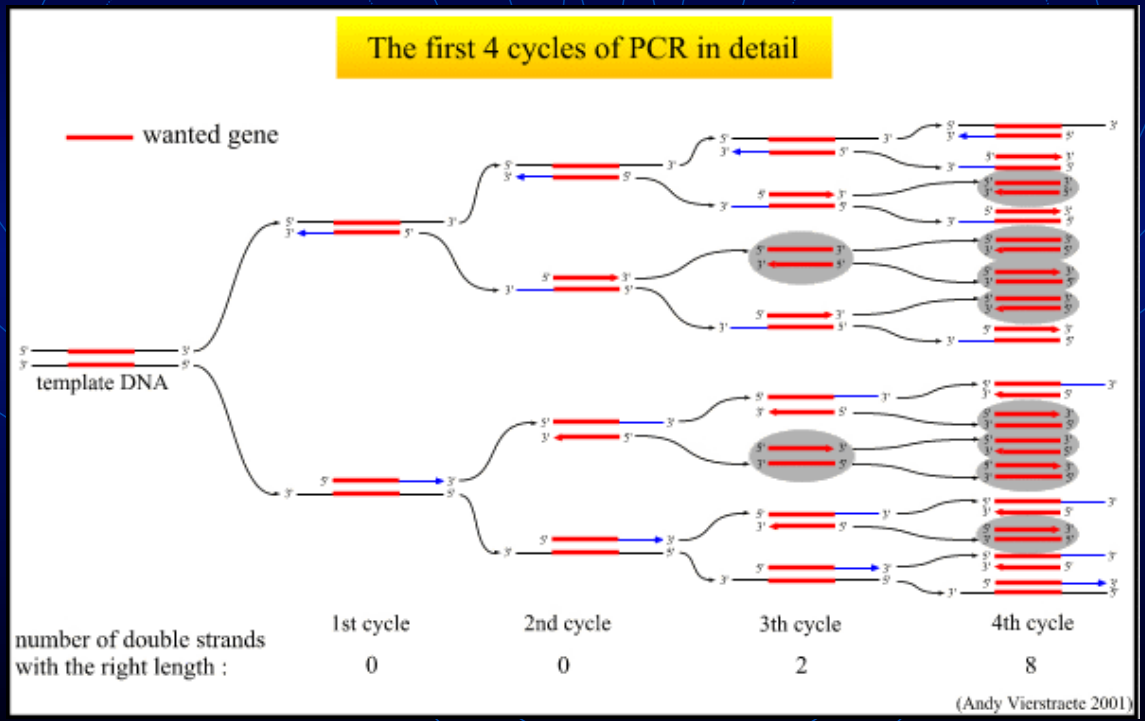
Slide courtesy of prof. Bin Ma, University of Waterloo, CA

PCR



Slide courtesy of prof. Bin Ma, University of Waterloo, CA

PCR



Slide courtesy of prof. Bin Ma, University of Waterloo, CA

Formal framework

Theorem[1]: If an exponential amplification occurs, then, at most at the third setp of the process, a blunt string appears which is a seed for the amplification.

$PCR_{(\alpha,\overline{\beta})}(P)$ or $PCR(\alpha,\overline{\beta})(P)$, over a pool P, by primers $(\alpha,\overline{\beta})$

$El(P) = \{d_1, d_2, \ldots, d_k\}$, and $El_n(P) = \{\alpha \mid \alpha \in P, |\alpha| = n\}$

$Enz_x(P), Lig(P), Taq(P)$

[1] pag 61, Infobiotics. DNA comp: pp 43-68

$PCR(\alpha, \overline{\beta})(P)$

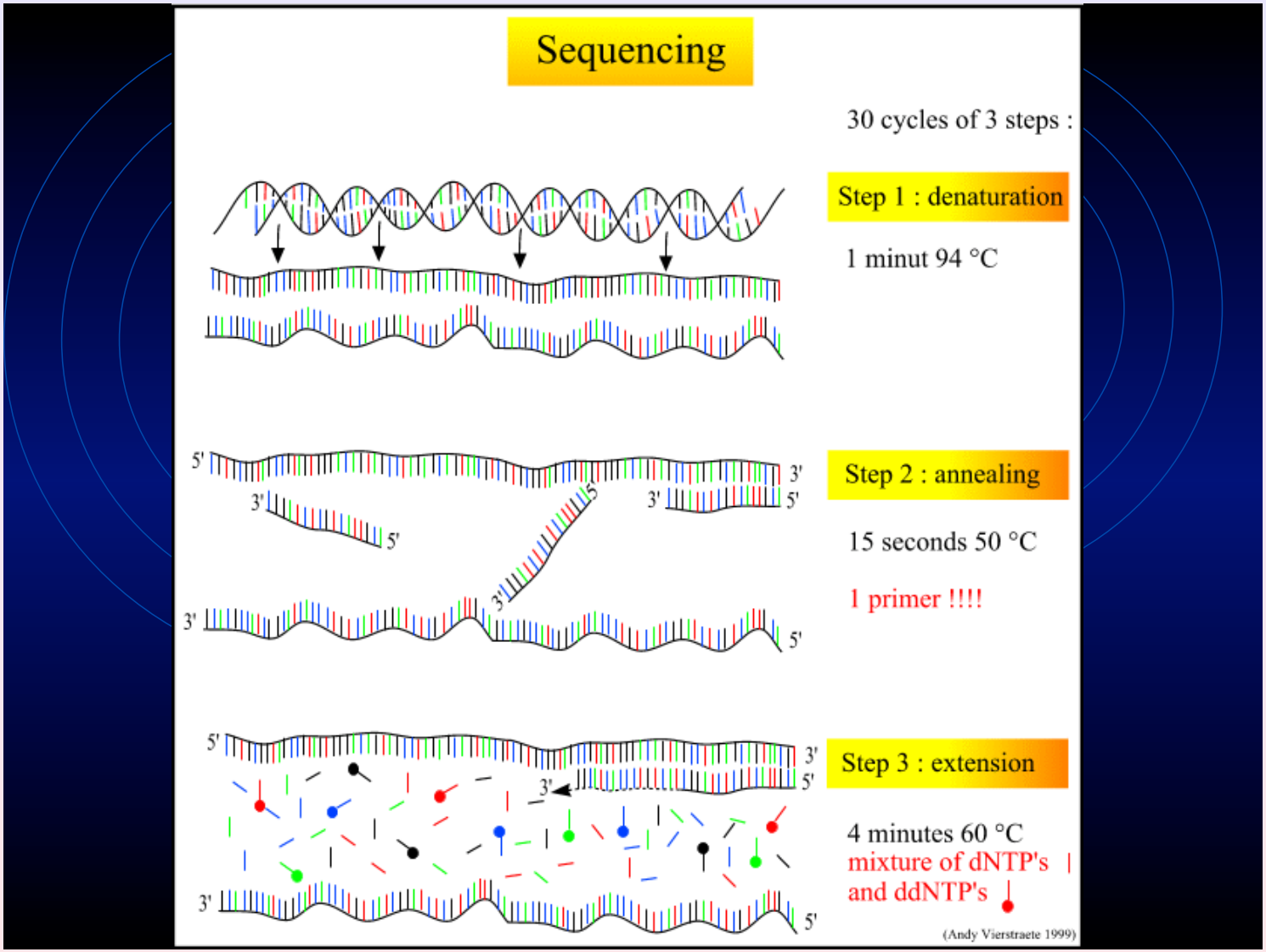
Input pool: P. % strands multiset (template) + buffer + dNTPs

For $i = 1, 30$

1. H (Mix $(P, \{\alpha, \overline{\beta}\})$);
2. C(P);
3. Taq(P);

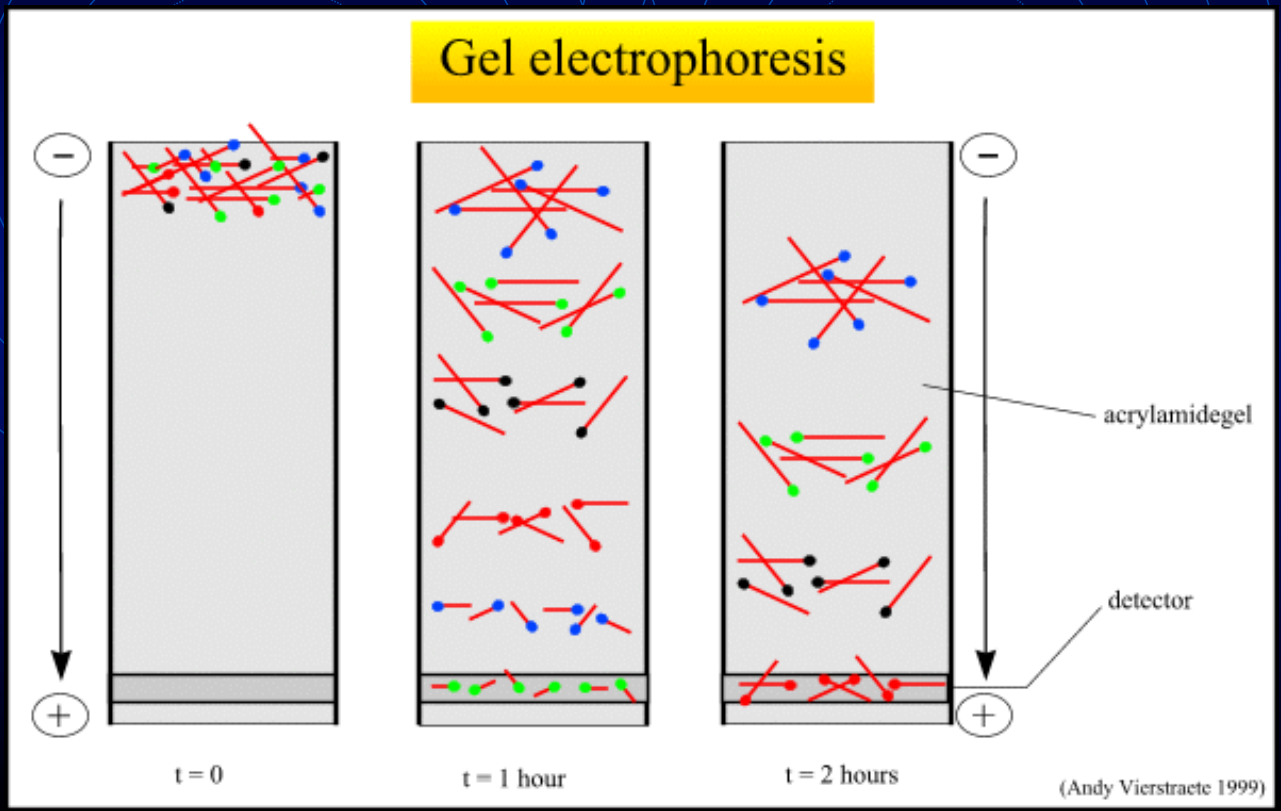
end for

Output: P with exponential amount (2^{30}) of all α-prefixed and β-suffixed strands already present within the sequences of P.



Slide courtesy of prof. Bin Ma, University of Waterloo, CA

Sequencing 2



Slide courtesy of prof. Bin Ma, University of Waterloo, CA

Limits of Sanger method

Only a sequence from an homogeneous pool may be sequenced.

Initial primer has to be artificial, otherwise very long and difficoult to design (since the sequence is not known).

Quantities of modified nucleotides have to be balanced, hard for long sequences.

Piro-sequencing, and recently NGS (heterogeneous pool).

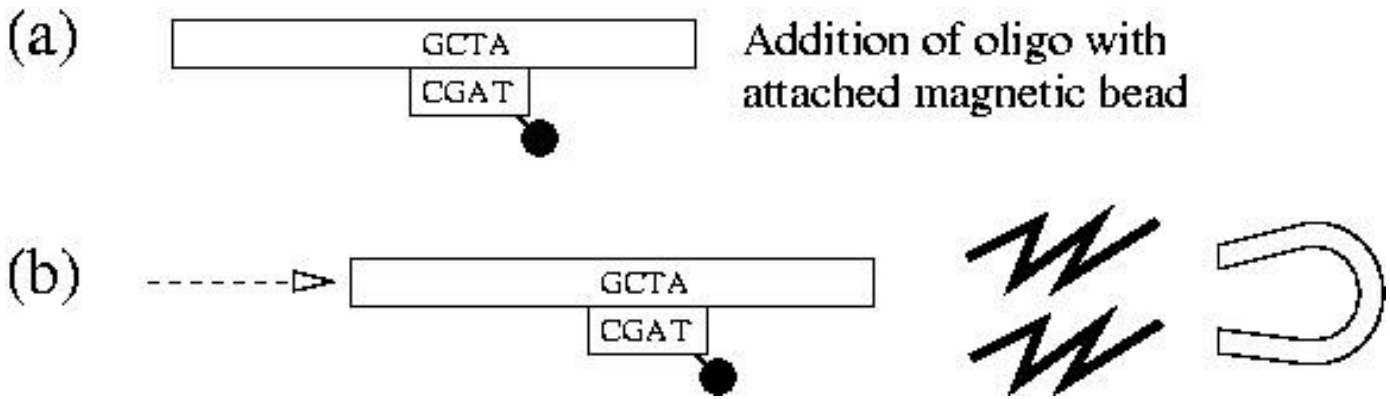
Notation: $Read(P)$ = P support

Separation

- It is sometimes useful to extract from a “pot” of DNA strands only those containing a certain sequence
- Rather like doing a Unix “grep”on a file, it only returns the lines of text containing the sequence you’re looking for
- Can be achieved using a technique known as magnetic bead separation, or affinity purification

Slides courtesy of prof. Martin Amos, Manchester Metropolitan University, UK

Affinity purification



novel computation group

Slide courtesy of prof. Martin Amos, Manchester Metropolitan University, UK

Specification of a DNA Extraction Problem

Given an input pool P of heterogeneous DNA strands, with the same length and with the same prefix and suffix, and given a string γ, provide an output pool P_γ containing all and only the γ-superstrands [2] of P.

This operation is denoted by $Ext(P, \gamma)$ or $Separate(P, \gamma)$

[2] strings with at least one occurrence of γ as a substring

Test Tube Operations in DNAC

- Denaturation (Melting)
- Renaturation (Hybridization, Annealing, Ligation)
- Amplification (Polymerase PCR)
- Sequencing
- Synthesis (Oligos, Affix Extension)
- Clonation (Plasmide Transinfection)
- Gel Electrophoresis
- Merging
- Splitting (Random, Subtractive)
- Restriction (R. Enzymes)
- Selection by Affinity
- Detection

0

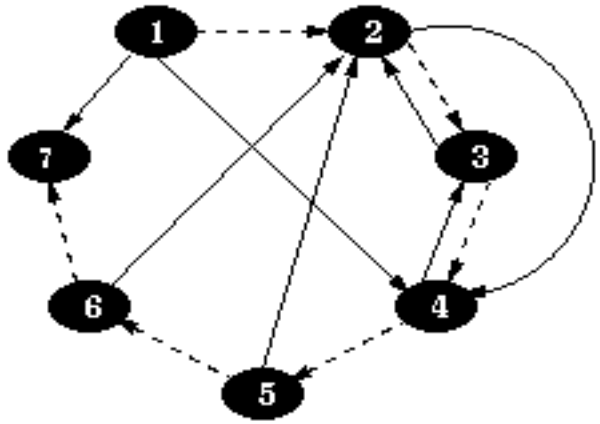
Slide courtesy of prof. Vincenzo Manca, Verona University, IT

A few DNA Algoritms

- Adleman (TSP-DHPP, Science, '94)
- Lipton (SAT, Science, '95)
- Jonoska (SAT, '99)
- Sakamoto et al., Takenaka et al. (SAT, '00 - '03)..
- Braich et al. (SAT, Science, '02)

The computation

- Adleman solved a small instance of a variant of the Travelling Salesman Problem, the *Hamiltonian Path Problem*
- Given a set of cities connected by roads, is there a tour starting at one city and ending at another that visits each city once and only once?



novel computation group

Slide courtesy of prof. Martin Amos, Manchester Metropolitan University, UK

Complexity

- The HPP is an archetypal *NP-complete* problem
- Such problems are characterised by their having an exponential-sized search space (possible solutions)
- There may be trillions of possible solutions, the vast majority of which are incorrect, but a few of which might be valid

Slide courtesy of prof. Martin Amos, Manchester Metropolitan University, UK

Adleman's solution

- The ultimate search for a "needle in a haystack" – generate all possible solutions to the problem, then throw away the ones that fail to meet certain criteria
- This is formally known as a *massively-parallel random search*
- Each possible solution is, in this case, represented as a strand of DNA

Slide courtesy of prof. Martin Amos, Manchester Metropolitan University, UK

Adleman's algorithm

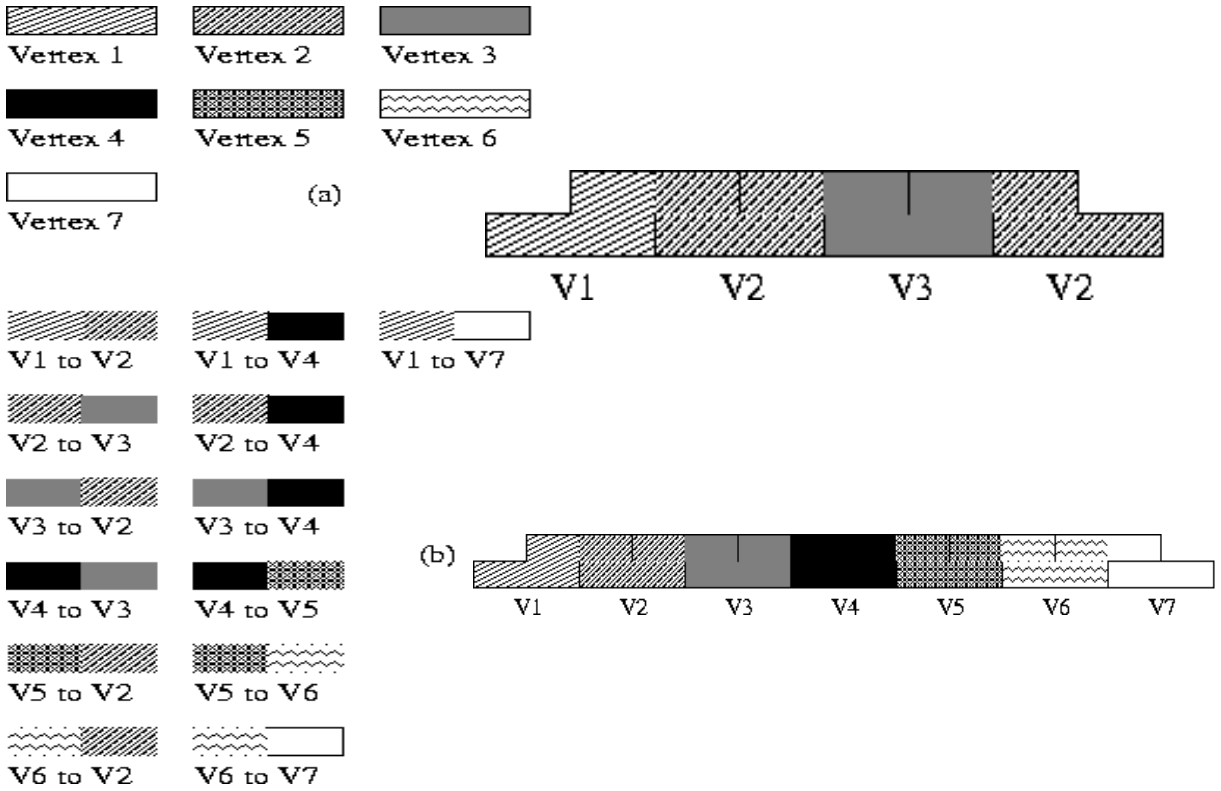
1) Generate strands encoding random paths, such that the HP is represented with high probability (use sufficient DNA to ensure this)
2) Remove all strands that *do not* encode a HP
3) Sequence what is left to discover the result

Slide courtesy of prof. Martin Amos, Manchester Metropolitan University, UK

Encoded data + Lig (C (H(P)))

Generating paths

Manchester Metropolitan University



novel computation group

Slide courtesy of prof. Martin Amos, Manchester Metropolitan University, UK

Extraction by PCRs, EI_{140}, and iterate separation

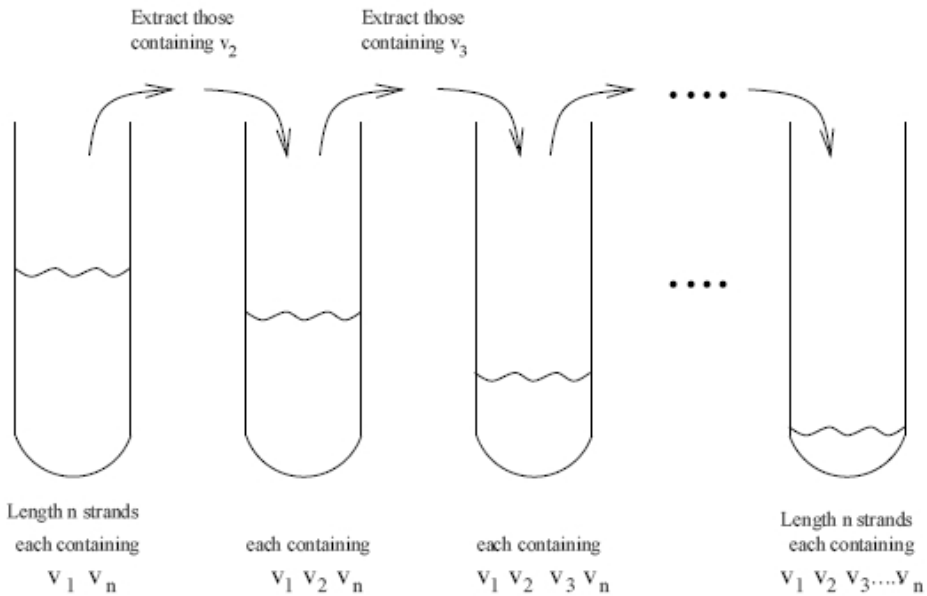
2. Remove illegal solutions

- Remove all strands that do not encode the HP
  - Wrong start/end point
  - Wrong length
  - Cities visited
- We know that the path must start at city 1 and end at city 7
- We therefore massively amplify only those strands that encode solutions that begin with the sequence encoding city 1 and end with the sequence encoding city 7
- How do we achieve this?

Slide courtesy of prof. Martin Amos, Manchester Metropolitan University, UK

Selecting paths with all cities

Manchester Metropolitan University



novel computation group

Slide courtesy of prof. Martin Amos, Manchester Metropolitan University, UK

Separation to test all cities

Use magnetic bead separation

- We perform a series of extractions, one each for cities 2, 3, 4, 5 and 6
- We already know that cities 1 and 7 are represented
- Use the complementary sequence of city 2 as the "bait" on the magnetic fishing line, remove strands containing that sequence, and then use only those strands for the next separation
- Continue with an ever-shrinking tube until we are left only with strands that contain the sequences for cities 2, 3, 4, 5 and 6 (in addition to 1 and 7)

novel computation group

Slide courtesy of prof. Martin Amos, Manchester Metropolitan University, UK

Inefficient separation

False positive. Adleman's experiment worked, but he failed to carry it out on a graph that did not contain a HP. Not a reliable algorithm.

At that time, no way to read the result in case of multiple Hamiltonian paths.

Linear time for execution, exponential volumes of DNA. Scaled to 200 vertices, Adleman's algorithm would require a prohibitive amount of DNA to solve the problem.

Adleman - Lipton's Extract Model

- The Generation of all possible solutions
- The Extraction of true solutions

Extraction is performed in a number of sub-steps and each of them selects all the strands that include a sub-strand of a given type

0

Slides courtesy of prof. Vincenzo Manca, Verona University, IT

Post-Adleman

- Richard Lipton followed up on Adleman's work by showing how the *Satisfiability* problem may, in principle, be solved using a similar approach
- SAT is the "gold standard" of NP-complete problems
- Decide if there exists a combination of assignments to the terms in a propositional formula such that the overall formula is true

Richard J. Lipton. DNA solution of hard computational problems. *Science*, **268**:542-545, 1995.

Slide courtesy of prof. Martin Amos, Manchester Metropolitan University, UK

3-SAT(n,m): a decision problem

Given a first order propositional formula ϕ, we may assume it to be the conjunction of *m* clauses, each of which is the disjunction of at most three *literals*, where each literal is a variable or its negation.

Example:

$$\phi = (x_1 \vee \neg x_2 \vee x_4) \wedge (\neg x_2 \vee \neg x_3 \vee x_5) \wedge (x_1 \vee \neg x_4 \vee \neg x_5) \wedge (x_3 \vee \neg x_4 \vee \neg x_5)$$

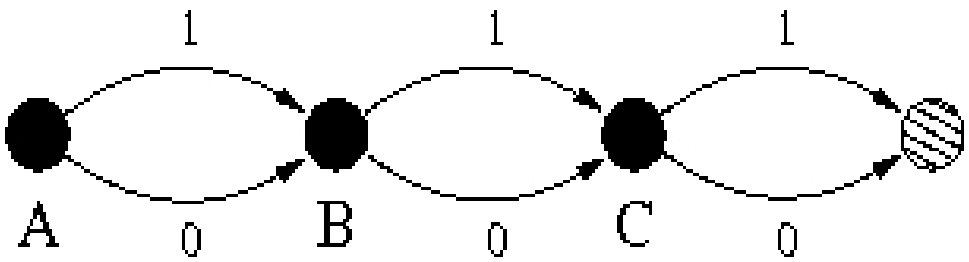
ϕ is satisfiable iff there exists an assignment of truth values to the *n* variables making the formula true.

Generating exponential solution space for 3 variables: P, Q, R

Manchester Metropolitan University

Lipton's solution

- Use a graph, where each vertex represents a bit
- Choice of edge dictates value of bit



novel computation group

Slide courtesy of prof. Martin Amos, Manchester Metropolitan University, UK

Lipton's Algorithm 3-Sat(n, m)

Generate *n*-space solutions in *T*;

For j = 1,m % for each clause [3]

T1 := Ext (T, L(1,j)); Z := T - T1

T2 := Ext (Z, L(2,j)); W:= Z - T2

T3 := Ext(W, L(3,j)); T := Merge(T1, T2, T3)

Detect T:

If T $\neq \emptyset$, then formula is satisfiable

[3] L(i,j) denotes i-th literal of j-th clause

We will see an example, of the algorithm at work, on formula

$$(P \vee \neg Q) \wedge (Q \vee R) \wedge (\neg R \vee \neg P)$$

Still exponential space complexity, with linear time complexity. How to read the solution?

Time (bio)complexity is counted by number of "laborious" bio-steps.

An example

- Starting pool of solution strands:

P	Q	R
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

Slide courtesy of prof. Martin Amos, Manchester Metropolitan University, UK

First clause

- P OR (NOT Q)
- Keep *only* strands that encode P==1 or Q==0
- Lose PQR=010 and PQR=011

P	Q	R
0	0	0
0	0	1
1	0	0
1	0	1
1	1	0
1	1	1

Slide courtesy of prof. Martin Amos, Manchester Metropolitan University, UK

Second clause

- Q OR R
- Retain only strands that encode 1 for Q or R
- Lose PQR=000 and PQR=100

P	Q	R
0	0	1
1	0	1
1	1	0
1	1	1

Slide courtesy of prof. Martin Amos, Manchester Metropolitan University, UK

Final clause

- (NOT R) OR (NOT P)
- Retain only strands that encode 0 for R or P
- Lose PQR=101 and PQR=111

P	Q	R
0	0	1
1	1	0

Slide courtesy of prof. Martin Amos, Manchester Metropolitan University, UK

Confirmation

P	Q	R	P OR (NOT Q)	Q OR R	(NOT R) OR (NOT P)	F
0	0	0	1	0	1	0
0	0	1	1	1	1	1
0	1	0	0	1	1	0
0	1	1	0	1	1	0
1	0	0	1	0	1	0
1	0	1	1	1	0	0
1	1	0	1	1	1	1
1	1	1	1	1	0	0

P	Q	R
0	0	1
1	1	0

novel computation group

Slide courtesy of prof. Martin Amos, Manchester Metropolitan University, UK

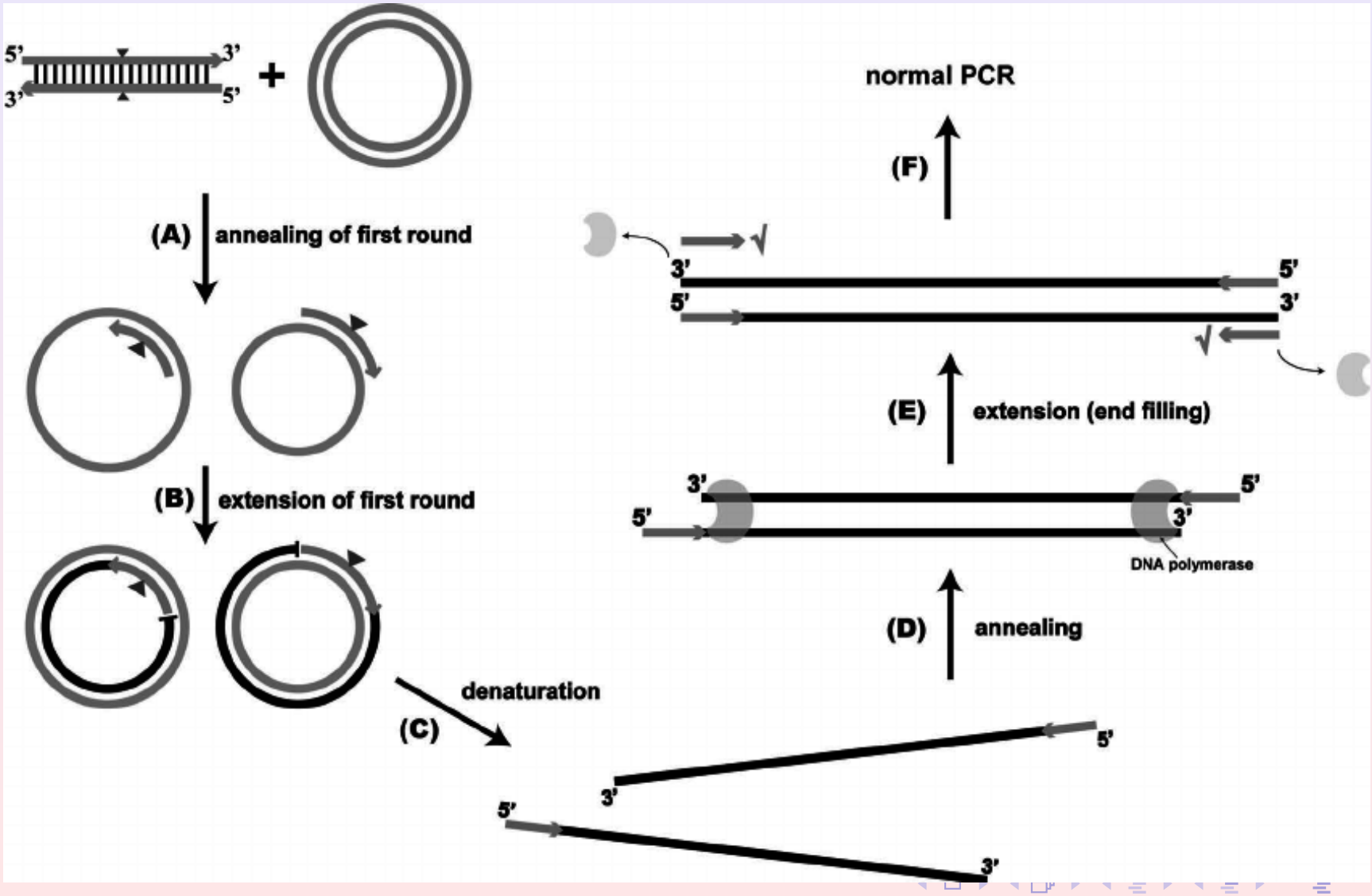
N. Jonoska's algorithm, '99

A smarter strategy then the brute force, and a very efficient extraction phase, based on (linear) enzymatic cuts

Interesting way to read the output, by circular PCR [4]

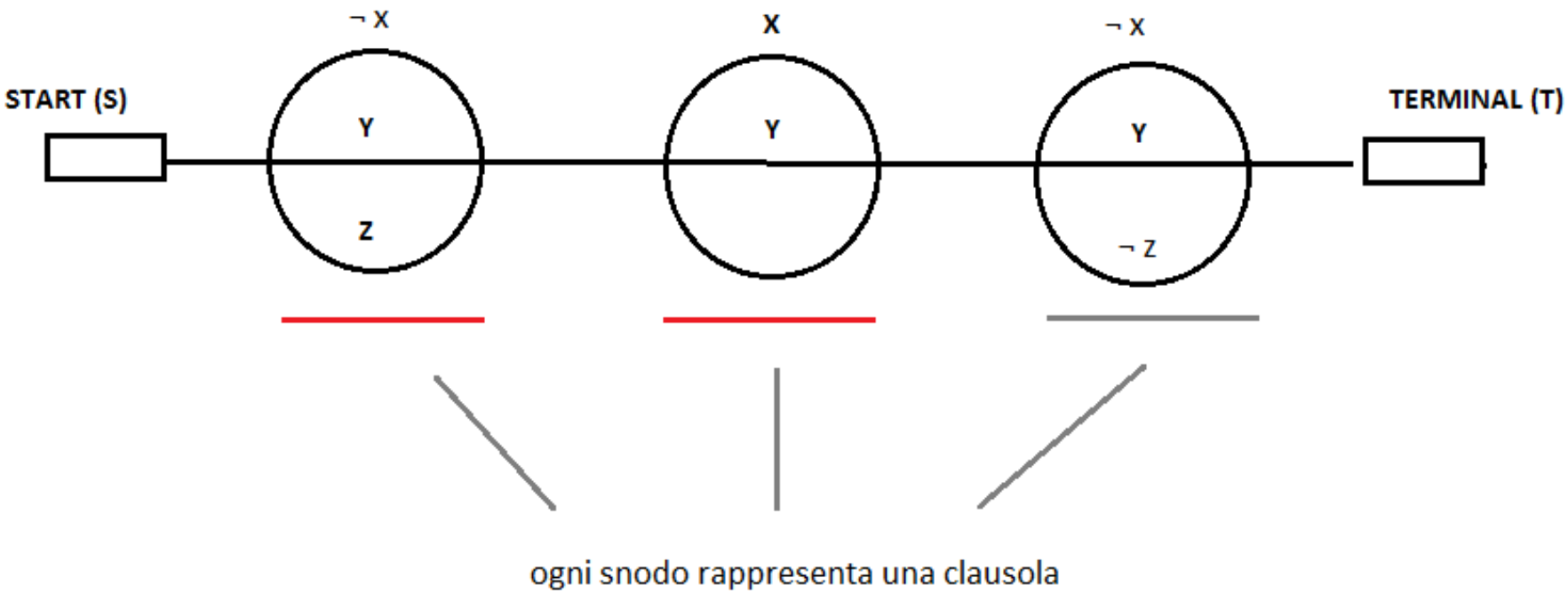
[4] Z. Chen et al. Amplification of closed circular DNA in vitro, 1998

Circular amplification



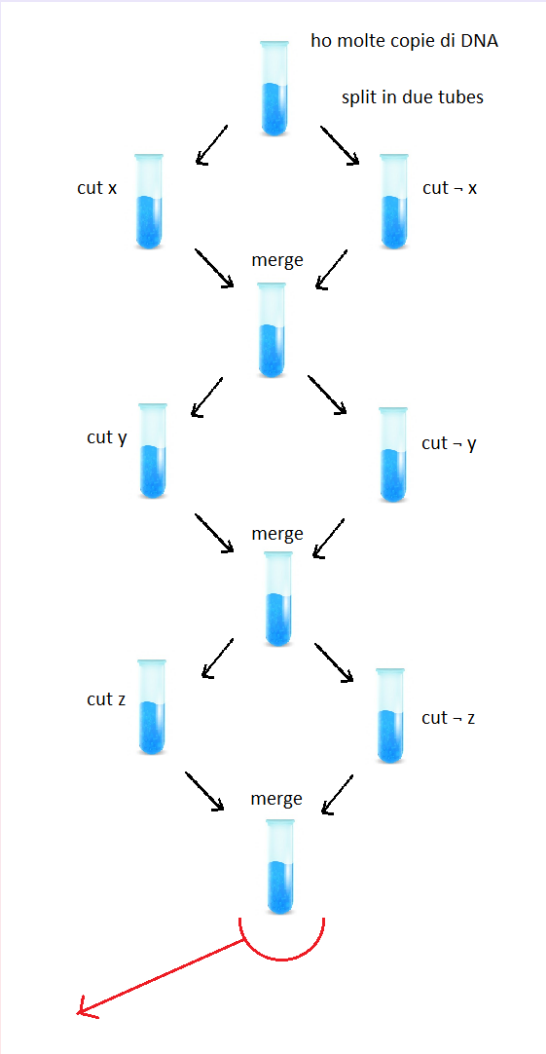
Data encoding

$$\phi = (\neg x \vee y \vee z) \wedge (x \vee y) \wedge (\neg x \vee y \vee \neg z)$$

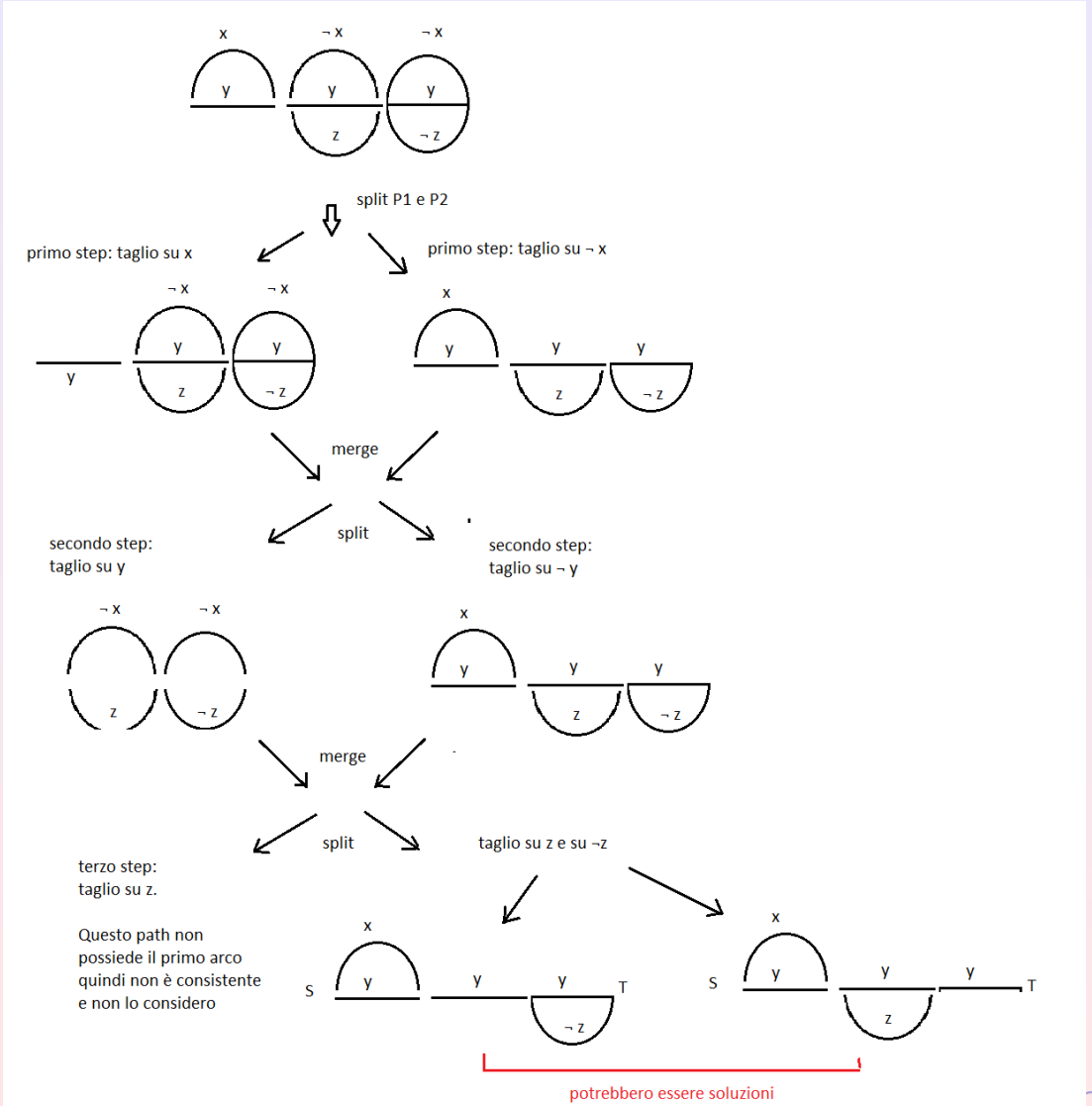


ϕ is satisfiable iff there exists a consistent path from s to t.

Intuition

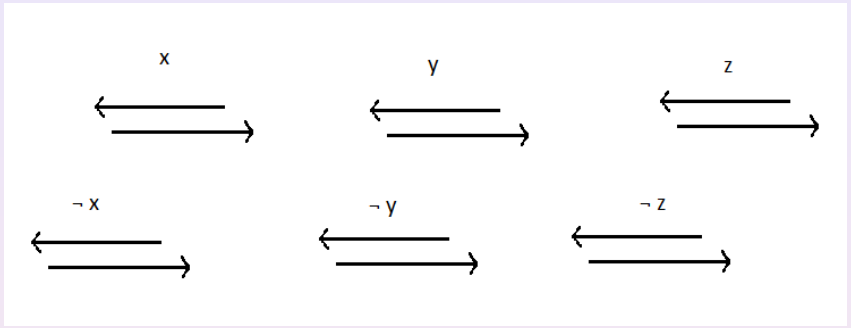


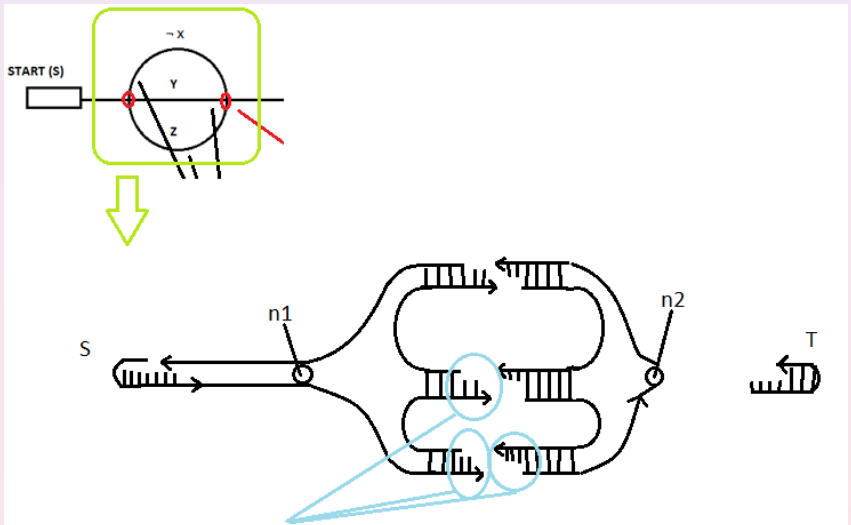
FINAL POOL



Implementation – input data

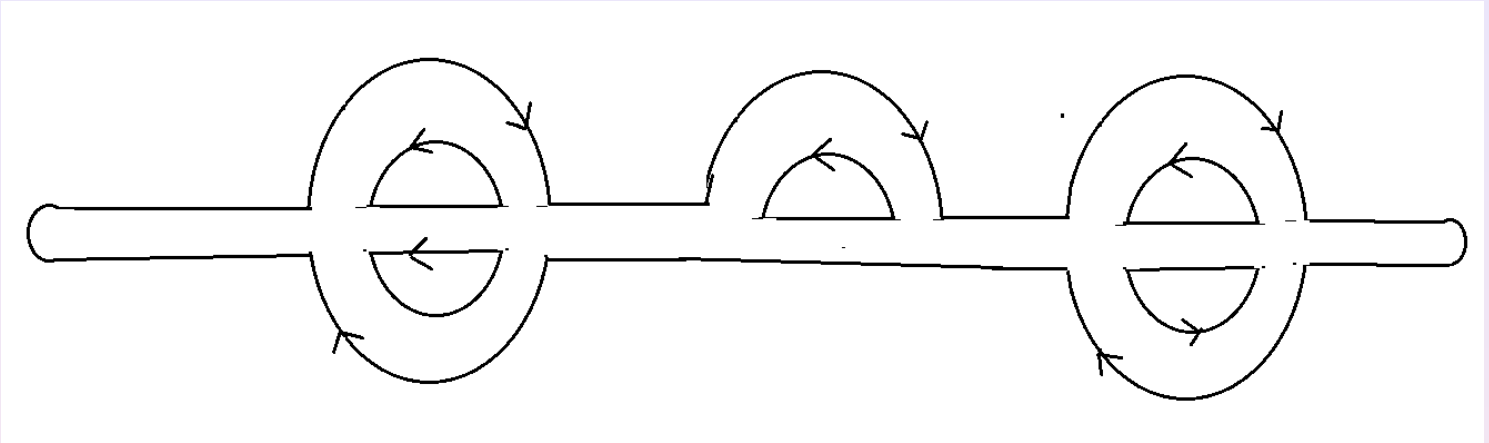
Different restriction sites for different literals

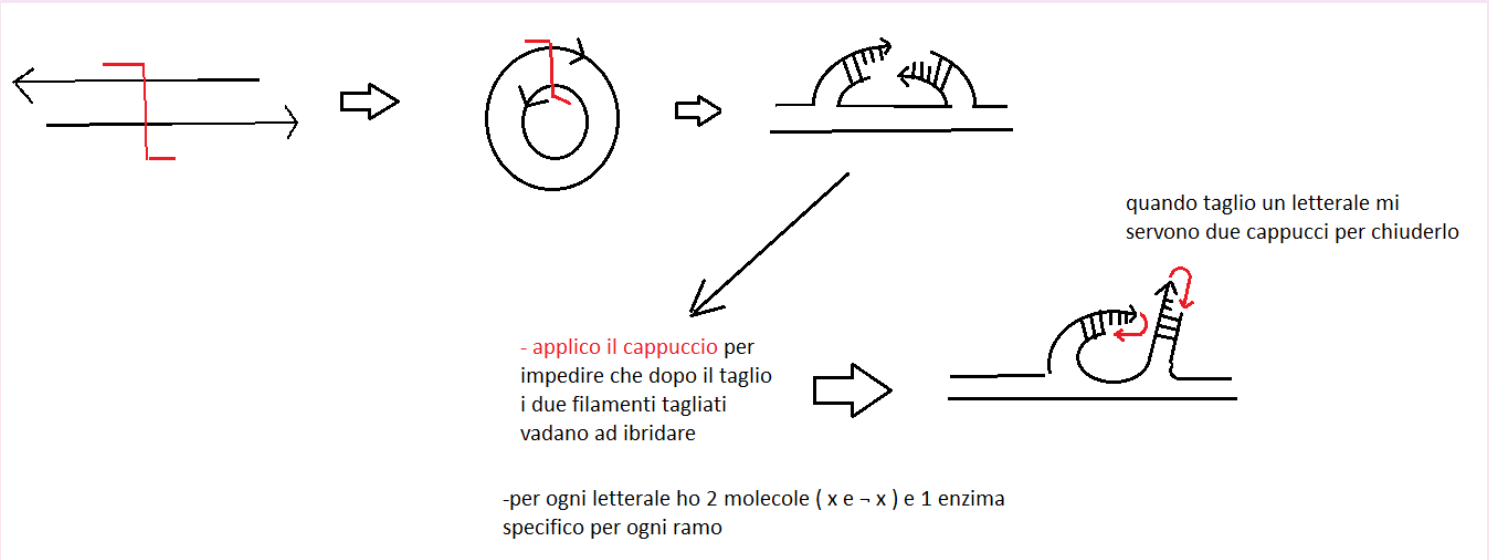




tutte queste sono sticky ends dove la parte attaccata al nodo a sx può ibridare con la parte legata al nodo a destra

Implementation





Slides courtesy of Alessandro Mainente, University of Verona, IT

N. Jonoska's Algorithm - SAT (n,m)

Input pool P_0 has m 3D structures for the nodes, $2n$ arcs for each literal with a different restriction site x, with heads T_x

P : = Exo(Lig (C(P_0))); % proper graph formations by hybridiz

For i = 1, n
(P1, P2):= split (P);
P1 : = Enz_{x_i} (P1); % enzymatic cut of edge x_i
P1 := Lig (C (mix (P1, T_{x_i}))) % heads T_{x_i} to close cut x_i
P2 := Enz_{x_2} (P2); % enzymatic cut of edge $\neg x_i$
P2 := Lig (C (mix (P2, $T_{\neg x_i}$))) % heads $T_{\neg x_i}$ to close cut x_i
P : = mix (P1, P2);
End For

P := PCR $(s, \bar{t})$(H(P));
$P \neq \emptyset$ iff the formula is satisfiable.

Problems of scalability

$2n$ enzymes acting in comparable times and conditions are necessary

more than 2^n initial copies of the graph, with big 3D structures long m blocks: again a problem of space complexity.

DNA algorithms in Japan

Hairpin formation to eliminate non-solutions in Sakamoto's algorithm: γ =ATCG (double) restriction site, inserted in all literal (negated variable are encoded by the mirror sequence):

$$PCR(s, \overline{c_m})(Enz_\gamma(C^*(H(Lig(C(P))))))) \qquad (Sakamoto)$$

Fluorescence Activated Cell Sorter to separate solutions in Takenaka's algorithm: beads of 5μm with 10^6 copies of each assignment, hybridization with complementary assignments fluoridated by Cy5, and with non-solutions by R110. FACS separates only Cy5 cells, sequencing by MPSS (3×10^6 beads x cm^2).

Identification of quasi-solutions for non satisfiable formulas. Better technology, still a lot of space: 3^m strands, and beads.

Solving NP-complete problems

After the seminal Adleman's experiment ('94), where the solution of an instance of Hamiltonian Path Problem was found within DNA sequences, Lipton ('95) showed that SAT can be solved by using essentially the same bio-techniques.

The exponential amount of DNA in such an *extract model* (brute force search) was shown to be prohibitive to scale-up the algorithms for real problems.

There have been several attempts to reduce the space and/or the time complexity of DNA algorithms solving NP-complete problems [e.g., X.Wang, *DNA Computing Solve the 3-SAT Problem with a Small Solution Space*].

Conclusion

It is yet clear that DNA computing is not competitive with *in silico* computers to solve NP-complete problems. Encoding problems due to mismatches are overcame as well.

Current trends focus on investigations on self-assembly phenomenon (namely construction of state machines, "DNA doctor"), as well as on improvements of bio-techniques (Whiplash PCR), and search for new procedures (**XPCR**).

Novel XPCR-based recombination (extraction, mutagenesis, concatenation) methods have been proposed as combinatorial algorithms, and validated by experiments.