DNA Recombination and Extraction Methods by XPCR

Giuditta Franco

Department of Computer Science
University of Verona

MNC, Final Lecture - 24 May 2006

Premise

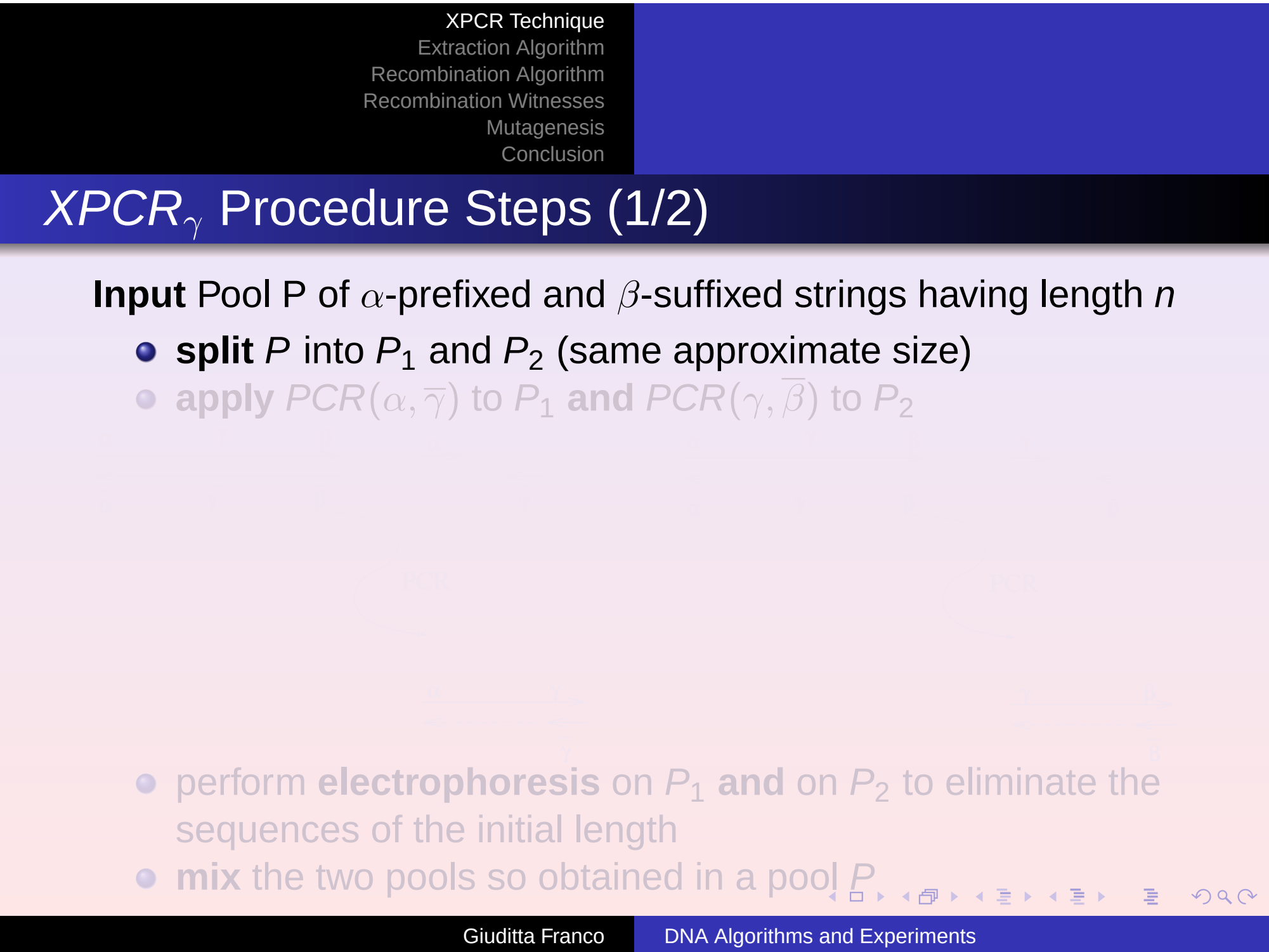
“No amount of experimentation can ever prove me right; a single experiment can prove me wrong.”

Albert Einstein

$XPCR_\gamma$ Procedure Steps (1/2)

Input Pool P of α-prefixed and β-suffixed strings having length n

- **split** P into P_1 and P_2 (same approximate size)
- **apply** $PCR(\alpha, \overline{\gamma})$ to P_1 **and** $PCR(\gamma, \overline{\beta})$ to P_2



- perform **electrophoresis** on P_1 **and** on P_2 to eliminate the sequences of the initial length
- **mix** the two pools so obtained in a pool P

$XPCR_\gamma$ Procedure Steps (1/2)

Input Pool P of α-prefixed and β-suffixed strings having length *n*

- **split** *P* into P_1 and P_2 (same approximate size)
- **apply** $PCR(\alpha, \overline{\gamma})$ to P_1 **and** $PCR(\gamma, \overline{\beta})$ to P_2

- perform **electrophoresis** on P_1 **and** on P_2 to eliminate the sequences of the initial length
- **mix** the two pools so obtained in a pool *P*

$XPCR_\gamma$ Procedure Steps (1/2)

Input Pool P of α-prefixed and β-suffixed strings having length *n*

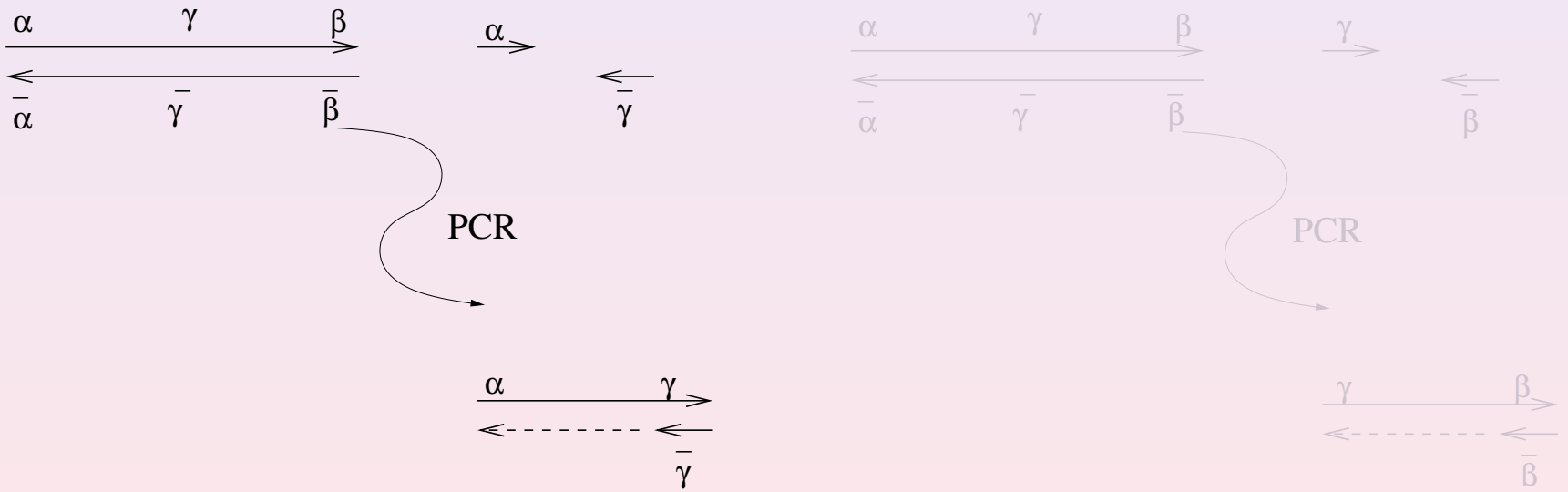
- **split** *P* into P_1 and P_2 (same approximate size)
- **apply** $PCR(\alpha, \overline{\gamma})$ to P_1 **and** $PCR(\gamma, \overline{\beta})$ to P_2

- perform **electrophoresis** on P_1 **and** on P_2 to eliminate the sequences of the initial length
- **mix** the two pools so obtained in a pool *P*

$XPCR_\gamma$ Procedure Steps (1/2)

Input Pool P of α-prefixed and β-suffixed strings having length *n*

- **split** *P* into P_1 and P_2 (same approximate size)
- **apply** $PCR(\alpha, \overline{\gamma})$ to P_1 **and** $PCR(\gamma, \overline{\beta})$ to P_2

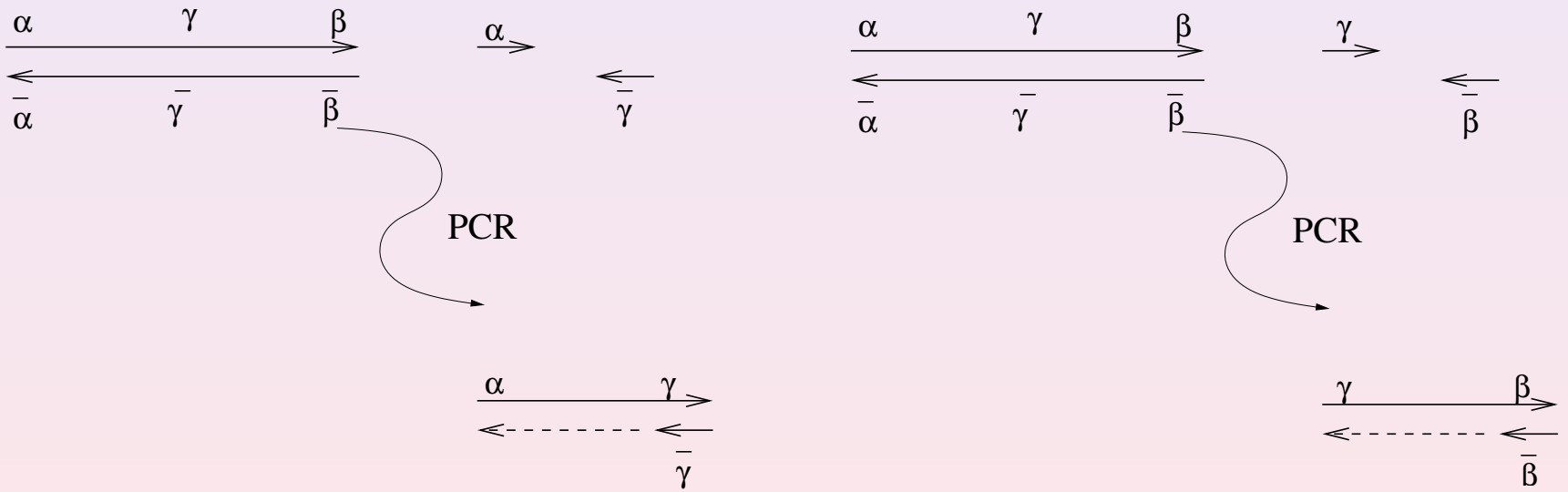


- perform **electrophoresis** on P_1 **and** on P_2 to eliminate the sequences of the initial length
- **mix** the two pools so obtained in a pool *P*

$XPCR_\gamma$ Procedure Steps (1/2)

Input Pool P of α-prefixed and β-suffixed strings having length *n*

- **split** *P* into P_1 and P_2 (same approximate size)
- **apply** $PCR(\alpha, \overline{\gamma})$ to P_1 **and** $PCR(\gamma, \overline{\beta})$ to P_2

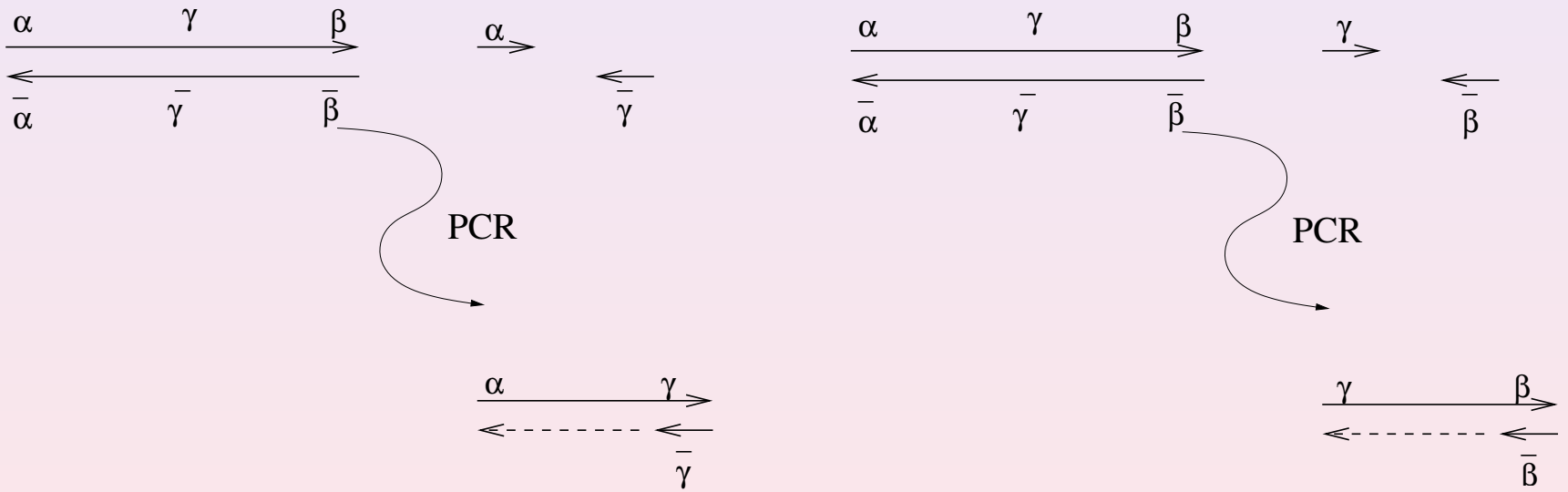


- perform **electrophoresis** on P_1 **and** on P_2 to eliminate the sequences of the initial length
- **mix** the two pools so obtained in a pool *P*

$XPCR_\gamma$ Procedure Steps (1/2)

Input Pool P of α-prefixed and β-suffixed strings having length *n*

- **split** *P* into P_1 and P_2 (same approximate size)
- **apply** $PCR(\alpha, \overline{\gamma})$ to P_1 **and** $PCR(\gamma, \overline{\beta})$ to P_2

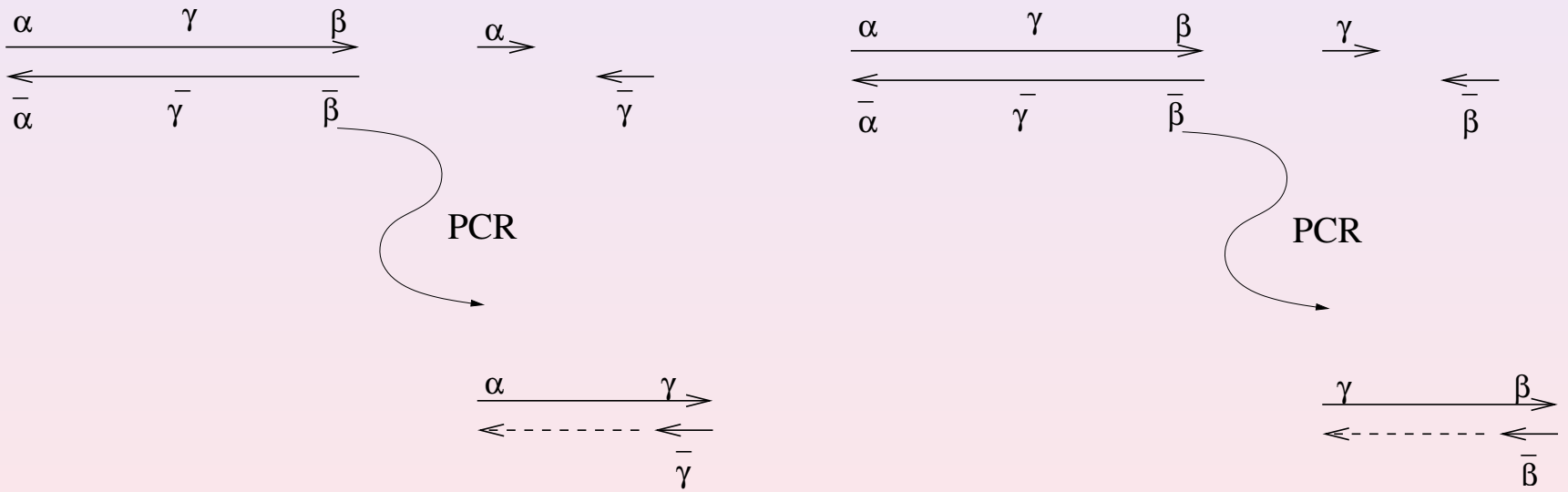


- perform **electrophoresis** on P_1 **and** on P_2 to eliminate the sequences of the initial length
- **mix** the two pools so obtained in a pool *P*

$XPCR_\gamma$ Procedure Steps (1/2)

Input Pool P of α-prefixed and β-suffixed strings having length *n*

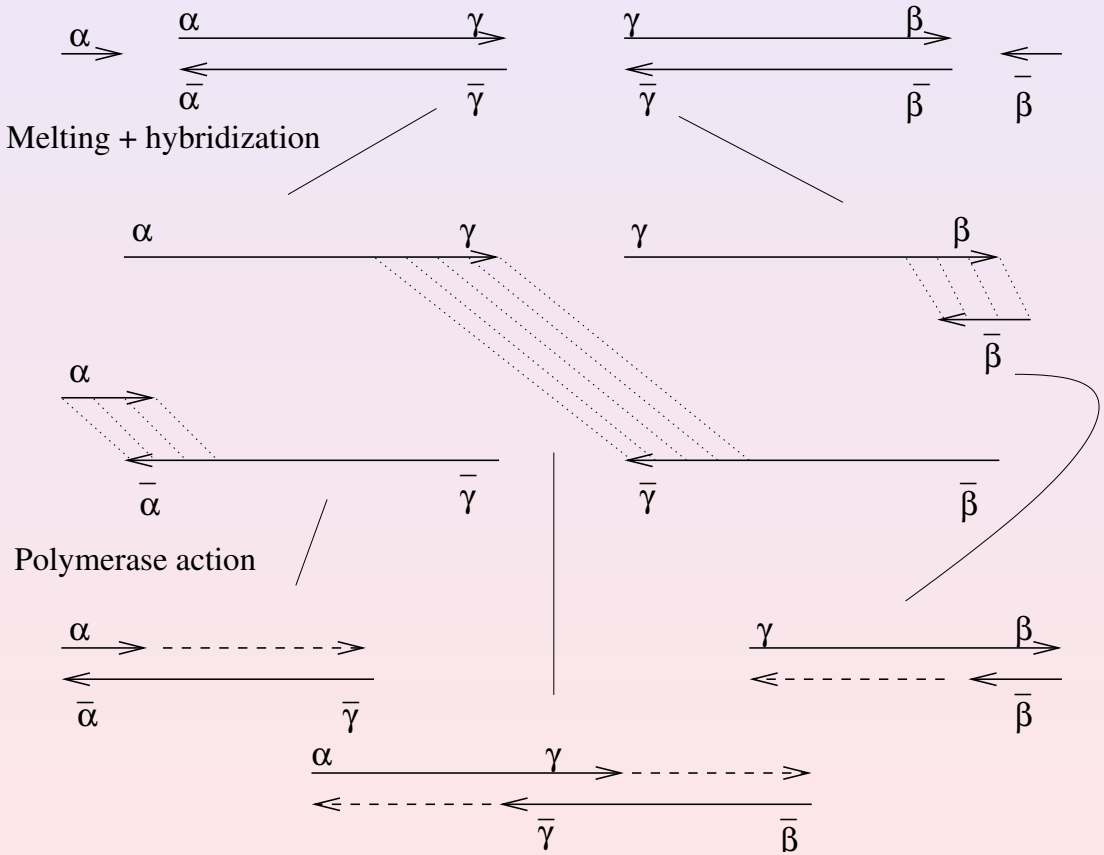
- **split** *P* into P_1 and P_2 (same approximate size)
- **apply** $PCR(\alpha, \overline{\gamma})$ to P_1 **and** $PCR(\gamma, \overline{\beta})$ to P_2



- perform **electrophoresis** on P_1 **and** on P_2 to eliminate the sequences of the initial length
- **mix** the two pools so obtained in a pool *P*

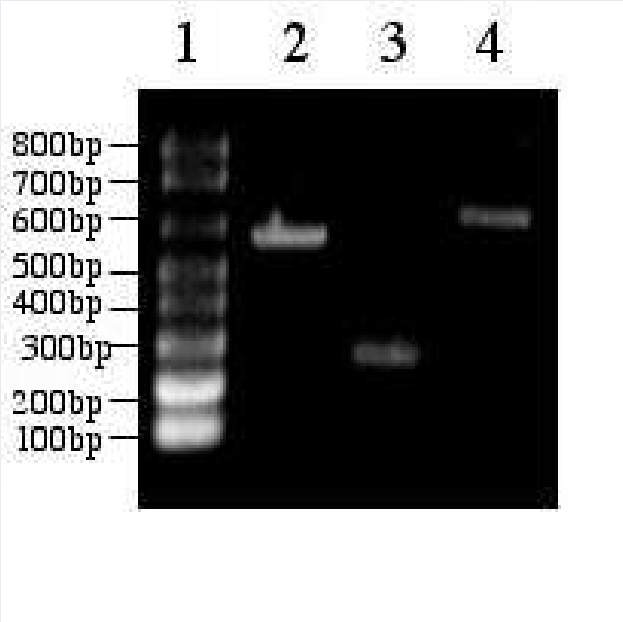
$XPCR_\gamma$ Procedure Steps (2/2)

- **apply** $PCR(\alpha, \overline{\beta})$ to P



Output The pool P resulting from the previous step.

Experiment for Testing the XPCR



Electrophoresis results. 1: molecular size marker ladder (100b). 2: $\alpha\phi\gamma\psi$-strands of human RhoA (582bp), 3: $\gamma\psi\beta$-strands (253bp), 4: cross pairing amplification of $\alpha\phi\gamma\psi\beta$-strands (606bp): 606 = 582 +253 - 229.

$XPCR - Extract(P, \gamma)$

1. $S := \emptyset$; $L := length(P)$;
2. **for each** $n \in L$ **do**
   1. $R_1 := \emptyset, R_2 := \emptyset, Q := \emptyset, P_1 := \emptyset, P_2 := \emptyset$;
   2. $P := separate(P, n)$;
   3. $P := infix(P, \alpha, \beta)$;
   4. $(P_1, P_2) := split(P)$;
   5. $P_1 := PCR(P_1, \alpha, \bar{\gamma})$;
   6. **for each** $m < n$ **do** $R_1 := mix(R_1, separate(P_1, m))$;
   7. $P_2 := PCR(P_2, \gamma, \bar{\beta})$
   8. **for each** $m < n$ **do** $R_2 := mix(R_2, separate(P_2, m))$;
   9. $Q := mix(R_1, R_2)$;
   10. $Q := PCR(Q, \alpha, \bar{\beta})$;
   11. $Q := separate(Q, n + |\alpha| + |\beta|)$;
   12. $S := mix(S, Q)$;
3. **output** S.

$XPCR - Extract(P, \gamma)$

1. $S := \emptyset$; $L := length(P)$;
2. **for each** $n \in L$ **do**
   1. $R_1 := \emptyset, R_2 := \emptyset, Q := \emptyset, P_1 := \emptyset, P_2 := \emptyset$;
   2. $P := separate(P, n)$;
   3. $P := infix(P, \alpha, \beta)$;
   4. $(P_1, P_2) := split(P)$;
   5. $P_1 := PCR(P_1, \alpha, \bar{\gamma})$;
   6. **for each** $m < n$ **do** $R_1 := mix(R_1, separate(P_1, m))$;
   7. $P_2 := PCR(P_2, \gamma, \bar{\beta})$
   8. **for each** $m < n$ **do** $R_2 := mix(R_2, separate(P_2, m))$;
   9. $Q := mix(R_1, R_2)$;
   10. $Q := PCR(Q, \alpha, \bar{\beta})$;
   11. $Q := separate(Q, n + |\alpha| + |\beta|)$;
   12. $S := mix(S, Q)$;
3. **output** S.

$XPCR - Extract(P, \gamma)$

1. $S := \emptyset$; $L := length(P)$;
2. **for each** $n \in L$ **do**
   1. $R_1 := \emptyset, R_2 := \emptyset, Q := \emptyset, P_1 := \emptyset, P_2 := \emptyset$;
   2. $P := separate(P, n)$;
   3. $P := infix(P, \alpha, \beta)$;
   4. $(P_1, P_2) := split(P)$;
   5. $P_1 := PCR(P_1, \alpha, \bar{\gamma})$;
   6. **for each** $m < n$ **do** $R_1 := mix(R_1, separate(P_1, m))$;
   7. $P_2 := PCR(P_2, \gamma, \bar{\beta})$
   8. **for each** $m < n$ **do** $R_2 := mix(R_2, separate(P_2, m))$;
   9. $Q := mix(R_1, R_2)$;
   10. $Q := PCR(Q, \alpha, \bar{\beta})$;
   11. $Q := separate(Q, n + |\alpha| + |\beta|)$;
   12. $S := mix(S, Q)$;
3. **output** S.

$XPCR - Extract(P, \gamma)$

1. $S := \emptyset$; $L := length(P)$;
2. **for each** $n \in L$ **do**
    1. $R_1 := \emptyset, R_2 := \emptyset, Q := \emptyset, P_1 := \emptyset, P_2 := \emptyset$;
    2. $P := separate(P, n)$;
    3. $P := infix(P, \alpha, \beta)$;
    4. $(P_1, P_2) := split(P)$;
    5. $P_1 := PCR(P_1, \alpha, \bar{\gamma})$;
    6. **for each** $m < n$ **do** $R_1 := mix(R_1, separate(P_1, m))$;
    7. $P_2 := PCR(P_2, \gamma, \bar{\beta})$
    8. **for each** $m < n$ **do** $R_2 := mix(R_2, separate(P_2, m))$;
    9. $Q := mix(R_1, R_2)$;
    10. $Q := PCR(Q, \alpha, \bar{\beta})$;
    11. $Q := separate(Q, n + |\alpha| + |\beta|)$;
    12. $S := mix(S, Q)$;
3. **output** S.

$XPCR - Extract(P, \gamma)$

1. $S := \emptyset$; $L := length(P)$;
2. **for each** $n \in L$ **do**
    1. $R_1 := \emptyset, R_2 := \emptyset, Q := \emptyset, P_1 := \emptyset, P_2 := \emptyset$;
    2. $P := separate(P, n)$;
    3. $P := infix(P, \alpha, \beta)$;
    4. $(P_1, P_2) := split(P)$;
    5. $P_1 := PCR(P_1, \alpha, \bar{\gamma})$;
    6. **for each** $m < n$ **do** $R_1 := mix(R_1, separate(P_1, m))$;
    7. $P_2 := PCR(P_2, \gamma, \bar{\beta})$
    8. **for each** $m < n$ **do** $R_2 := mix(R_2, separate(P_2, m))$;
    9. $Q := mix(R_1, R_2)$;
    10. $Q := PCR(Q, \alpha, \bar{\beta})$;
    11. $Q := separate(Q, n + |\alpha| + |\beta|)$;
    12. $S := mix(S, Q)$;
3. **output** S.

$XPCR - Extract(P, \gamma)$

1. $S := \emptyset$; $L := length(P)$;
2. **for each** $n \in L$ **do**
   1. $R_1 := \emptyset, R_2 := \emptyset, Q := \emptyset, P_1 := \emptyset, P_2 := \emptyset$;
   2. $P := separate(P, n)$;
   3. $P := infix(P, \alpha, \beta)$;
   4. $(P_1, P_2) := split(P)$;
   5. $P_1 := PCR(P_1, \alpha, \bar{\gamma})$;
   6. **for each** $m < n$ **do** $R_1 := mix(R_1, separate(P_1, m))$;
   7. $P_2 := PCR(P_2, \gamma, \bar{\beta})$
   8. **for each** $m < n$ **do** $R_2 := mix(R_2, separate(P_2, m))$;
   9. $Q := mix(R_1, R_2)$;
   10. $Q := PCR(Q, \alpha, \bar{\beta})$;
   11. $Q := separate(Q, n + |\alpha| + |\beta|)$;
   12. $S := mix(S, Q)$;
3. **output** S.

$XPCR - Extract(P, \gamma)$

1. $S := \emptyset$; $L := length(P)$;
2. **for each** $n \in L$ **do**
    1. $R_1 := \emptyset, R_2 := \emptyset, Q := \emptyset, P_1 := \emptyset, P_2 := \emptyset$;
    2. $P := separate(P, n)$;
    3. $P := infix(P, \alpha, \beta)$;
    4. $(P_1, P_2) := split(P)$;
    5. $P_1 := PCR(P_1, \alpha, \bar{\gamma})$;
    6. **for each** $m < n$ **do** $R_1 := mix(R_1, separate(P_1, m))$;
    7. $P_2 := PCR(P_2, \gamma, \bar{\beta})$
    8. **for each** $m < n$ **do** $R_2 := mix(R_2, separate(P_2, m))$;
    9. $Q := mix(R_1, R_2)$;
    10. $Q := PCR(Q, \alpha, \bar{\beta})$;
    11. $Q := separate(Q, n + |\alpha| + |\beta|)$;
    12. $S := mix(S, Q)$;
3. **output** S.

$XPCR - Extract(P, \gamma)$

1. $S := \emptyset$; $L := length(P)$;
2. **for each** $n \in L$ **do**
   1. $R_1 := \emptyset, R_2 := \emptyset, Q := \emptyset, P_1 := \emptyset, P_2 := \emptyset$;
   2. $P := separate(P, n)$;
   3. $P := infix(P, \alpha, \beta)$;
   4. $(P_1, P_2) := split(P)$;
   5. $P_1 := PCR(P_1, \alpha, \bar{\gamma})$;
   6. **for each** $m < n$ **do** $R_1 := mix(R_1, separate(P_1, m))$;
   7. $P_2 := PCR(P_2, \gamma, \bar{\beta})$
   8. **for each** $m < n$ **do** $R_2 := mix(R_2, separate(P_2, m))$;
   9. $Q := mix(R_1, R_2)$;
   10. $Q := PCR(Q, \alpha, \bar{\beta})$;
   11. $Q := separate(Q, n + |\alpha| + |\beta|)$;
   12. $S := mix(S, Q)$;
3. **output** S.

$XPCR - Extract(P, \gamma)$

1. $S := \emptyset$; $L := length(P)$;
2. **for each** $n \in L$ **do**
   1. $R_1 := \emptyset, R_2 := \emptyset, Q := \emptyset, P_1 := \emptyset, P_2 := \emptyset$;
   2. $P := separate(P, n)$;
   3. $P := infix(P, \alpha, \beta)$;
   4. $(P_1, P_2) := split(P)$;
   5. $P_1 := PCR(P_1, \alpha, \bar{\gamma})$;
   6. **for each** $m < n$ **do** $R_1 := mix(R_1, separate(P_1, m))$;
   7. $P_2 := PCR(P_2, \gamma, \bar{\beta})$
   8. **for each** $m < n$ **do** $R_2 := mix(R_2, separate(P_2, m))$;
   9. $Q := mix(R_1, R_2)$;
   10. $Q := PCR(Q, \alpha, \bar{\beta})$;
   11. $Q := separate(Q, n + |\alpha| + |\beta|)$;
   12. $S := mix(S, Q)$;
3. **output** S.

$XPCR - Extract(P, \gamma)$

1. $S := \emptyset$; $L := length(P)$;
2. **for each** $n \in L$ **do**
   1. $R_1 := \emptyset, R_2 := \emptyset, Q := \emptyset, P_1 := \emptyset, P_2 := \emptyset$;
   2. $P := separate(P, n)$;
   3. $P := infix(P, \alpha, \beta)$;
   4. $(P_1, P_2) := split(P)$;
   5. $P_1 := PCR(P_1, \alpha, \bar{\gamma})$;
   6. **for each** $m < n$ **do** $R_1 := mix(R_1, separate(P_1, m))$;
   7. $P_2 := PCR(P_2, \gamma, \bar{\beta})$
   8. **for each** $m < n$ **do** $R_2 := mix(R_2, separate(P_2, m))$;
   9. $Q := mix(R_1, R_2)$;
   10. $Q := PCR(Q, \alpha, \bar{\beta})$;
   11. $Q := separate(Q, n + |\alpha| + |\beta|)$;
   12. $S := mix(S, Q)$;
3. **output** S.

$XPCR - Extract(P, \gamma)$

1. $S := \emptyset$; $L := length(P)$;
2. **for each** $n \in L$ **do**
   1. $R_1 := \emptyset, R_2 := \emptyset, Q := \emptyset, P_1 := \emptyset, P_2 := \emptyset$;
   2. $P := separate(P, n)$;
   3. $P := infix(P, \alpha, \beta)$;
   4. $(P_1, P_2) := split(P)$;
   5. $P_1 := PCR(P_1, \alpha, \bar{\gamma})$;
   6. **for each** $m < n$ **do** $R_1 := mix(R_1, separate(P_1, m))$;
   7. $P_2 := PCR(P_2, \gamma, \bar{\beta})$
   8. **for each** $m < n$ **do** $R_2 := mix(R_2, separate(P_2, m))$;
   9. $Q := mix(R_1, R_2)$;
   10. $Q := PCR(Q, \alpha, \bar{\beta})$;
   11. $Q := separate(Q, n + |\alpha| + |\beta|)$;
   12. $S := mix(S, Q)$;
3. **output** S.

$XPCR - Extract(P, \gamma)$

1. $S := \emptyset$; $L := length(P)$;
2. **for each** $n \in L$ **do**
    1. $R_1 := \emptyset, R_2 := \emptyset, Q := \emptyset, P_1 := \emptyset, P_2 := \emptyset$;
    2. $P := separate(P, n)$;
    3. $P := infix(P, \alpha, \beta)$;
    4. $(P_1, P_2) := split(P)$;
    5. $P_1 := PCR(P_1, \alpha, \bar{\gamma})$;
    6. **for each** $m < n$ **do** $R_1 := mix(R_1, separate(P_1, m))$;
    7. $P_2 := PCR(P_2, \gamma, \bar{\beta})$
    8. **for each** $m < n$ **do** $R_2 := mix(R_2, separate(P_2, m))$;
    9. $Q := mix(R_1, R_2)$;
    10. $Q := PCR(Q, \alpha, \bar{\beta})$;
    11. $Q := separate(Q, n + |\alpha| + |\beta|)$;
    12. $S := mix(S, Q)$;
3. **output** S.

$XPCR - Extract(P, \gamma)$

1. $S := \emptyset$; $L := length(P)$;
2. **for each** $n \in L$ **do**
    1. $R_1 := \emptyset, R_2 := \emptyset, Q := \emptyset, P_1 := \emptyset, P_2 := \emptyset$;
    2. $P := separate(P, n)$;
    3. $P := infix(P, \alpha, \beta)$;
    4. $(P_1, P_2) := split(P)$;
    5. $P_1 := PCR(P_1, \alpha, \bar{\gamma})$;
    6. **for each** $m < n$ **do** $R_1 := mix(R_1, separate(P_1, m))$;
    7. $P_2 := PCR(P_2, \gamma, \bar{\beta})$
    8. **for each** $m < n$ **do** $R_2 := mix(R_2, separate(P_2, m))$;
    9. $Q := mix(R_1, R_2)$;
    10. $Q := PCR(Q, \alpha, \bar{\beta})$;
    11. $Q := separate(Q, n + |\alpha| + |\beta|)$;
    12. $S := mix(S, Q)$;
3. **output** S.

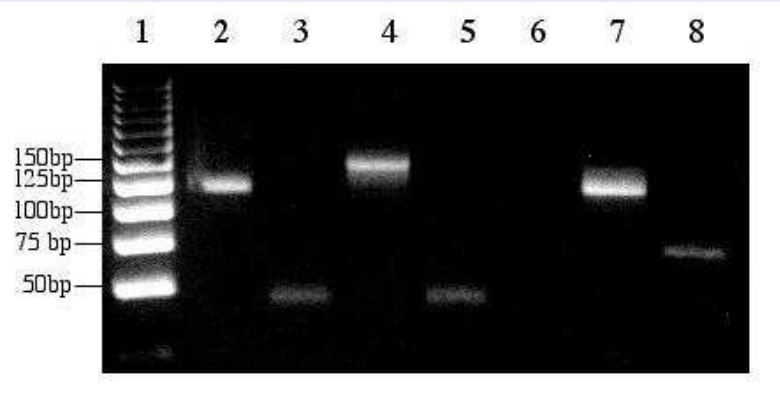
$XPCR - Extract(P, \gamma)$

1. $S := \emptyset$; $L := length(P)$;
2. **for each** $n \in L$ **do**
    1. $R_1 := \emptyset, R_2 := \emptyset, Q := \emptyset, P_1 := \emptyset, P_2 := \emptyset$;
    2. $P := separate(P, n)$;
    3. $P := infix(P, \alpha, \beta)$;
    4. $(P_1, P_2) := split(P)$;
    5. $P_1 := PCR(P_1, \alpha, \bar{\gamma})$;
    6. **for each** $m < n$ **do** $R_1 := mix(R_1, separate(P_1, m))$;
    7. $P_2 := PCR(P_2, \gamma, \bar{\beta})$
    8. **for each** $m < n$ **do** $R_2 := mix(R_2, separate(P_2, m))$;
    9. $Q := mix(R_1, R_2)$;
    10. $Q := PCR(Q, \alpha, \bar{\beta})$;
    11. $Q := separate(Q, n + |\alpha| + |\beta|)$;
    12. $S := mix(S, Q)$;
3. **output** S.

$XPCR - Extract(P, \gamma)$

1. $S := \emptyset$; $L := length(P)$;
2. **for each** $n \in L$ **do**
   1. $R_1 := \emptyset, R_2 := \emptyset, Q := \emptyset, P_1 := \emptyset, P_2 := \emptyset$;
   2. $P := separate(P, n)$;
   3. $P := infix(P, \alpha, \beta)$;
   4. $(P_1, P_2) := split(P)$;
   5. $P_1 := PCR(P_1, \alpha, \bar{\gamma})$;
   6. **for each** $m < n$ **do** $R_1 := mix(R_1, separate(P_1, m))$;
   7. $P_2 := PCR(P_2, \gamma, \bar{\beta})$
   8. **for each** $m < n$ **do** $R_2 := mix(R_2, separate(P_2, m))$;
   9. $Q := mix(R_1, R_2)$;
   10. $Q := PCR(Q, \alpha, \bar{\beta})$;
   11. $Q := separate(Q, n + |\alpha| + |\beta|)$;
   12. $S := mix(S, Q)$;
3. **output** S.

Experiment for Testing the Extraction



Electrophoresis results. 1: molecular size marker ladder (50 bp). 2: amplification of $\alpha \ldots \gamma$ strands (120 bp); 3: amplification of $\gamma \ldots \beta$ strands (45 bp); 4: cross pairing amplification of $\alpha \ldots \gamma$ and $\gamma \ldots \beta$ (150 bp). 5: positive control by PCR($\gamma, \bar{\beta}$), with γ at position -45; 6: negative control by PCR($\gamma', \bar{\beta}$); 7, 8: positive controls by PCR($\gamma_1, \bar{\beta}$) and PCR($\gamma_2, \bar{\beta}$) respectively, with γ_1 at position -125 and γ_2 at position -75.

XPCR-Based Recombination Algorithm

Goal: to generate the combinatorial library of n binary numbers

$$\{\alpha_1 \cdots \alpha_n \mid \alpha_i \in \{X_i, Y_i\},\ i = 1, \ldots, n\}.$$

The method starts from I_1, I_2, I_3 I_4 (extended by prefix α and suffix β), and generates the whole library, in linear time and by using essentially polymerase extension.

1. *Positive*: $I_1 = X_1X_2X_3X_4X_5X_6$,
2. *Negative*: $I_2 = Y_1Y_2Y_3Y_4Y_5Y_6$,
3. *Positive-Negative*: $I_3 = X_1Y_2X_3Y_4X_5Y_6$,
4. *Negative-Positive*: $I_4 = Y_1X_2Y_3X_4Y_5X_6$.

XPCR-Based Recombination Algorithm

Goal: to generate the combinatorial library of n binary numbers

$$\{\alpha_1 \cdots \alpha_n \mid \alpha_i \in \{X_i, Y_i\},\ i = 1, \ldots, n\}.$$

The method starts from I_1, I_2, I_3 I_4 (extended by prefix α and suffix β), and generates the whole library, in linear time and by using essentially polymerase extension.

1. *Positive*: $I_1 = X_1X_2X_3X_4X_5X_6$,
2. *Negative*: $I_2 = Y_1Y_2Y_3Y_4Y_5Y_6$,
3. *Positive-Negative*: $I_3 = X_1Y_2X_3Y_4X_5Y_6$,
4. *Negative-Positive*: $I_4 = Y_1X_2Y_3X_4Y_5X_6$.

XPCR-Based Recombination Algorithm

Goal: to generate the combinatorial library of n binary numbers

$$\{\alpha_1 \cdots \alpha_n \mid \alpha_i \in \{X_i, Y_i\},\ i = 1, \ldots, n\}.$$

The method starts from I_1, I_2, I_3 I_4 (extended by prefix α and suffix β), and generates the whole library, in linear time and by using essentially polymerase extension.

1. *Positive*: $I_1 = X_1X_2X_3X_4X_5X_6$,
2. *Negative*: $I_2 = Y_1Y_2Y_3Y_4Y_5Y_6$,
3. *Positive-Negative*: $I_3 = X_1Y_2X_3Y_4X_5Y_6$,
4. *Negative-Positive*: $I_4 = Y_1X_2Y_3X_4Y_5X_6$.

XPCR-Based Recombination Algorithm

Goal: to generate the combinatorial library of n binary numbers

$$\{\alpha_1 \cdots \alpha_n \mid \alpha_i \in \{X_i, Y_i\},\ i = 1, \ldots, n\}.$$

The method starts from I_1, I_2, I_3 I_4 (extended by prefix α and suffix β), and generates the whole library, in linear time and by using essentially polymerase extension.

1. *Positive*: $I_1 = X_1X_2X_3X_4X_5X_6$,
2. *Negative*: $I_2 = Y_1Y_2Y_3Y_4Y_5Y_6$,
3. *Positive-Negative*: $I_3 = X_1Y_2X_3Y_4X_5Y_6$,
4. *Negative-Positive*: $I_4 = Y_1X_2Y_3X_4Y_5X_6$.

XPCR-Based Recombination Algorithm

Goal: to generate the combinatorial library of n binary numbers

$$\{\alpha_1 \cdots \alpha_n \mid \alpha_i \in \{X_i, Y_i\},\ i = 1, \ldots, n\}.$$

The method starts from I_1, I_2, I_3 I_4 (extended by prefix α and suffix β), and generates the whole library, in linear time and by using essentially polymerase extension.

1. *Positive*: $I_1 = X_1X_2X_3X_4X_5X_6$,
2. *Negative*: $I_2 = Y_1Y_2Y_3Y_4Y_5Y_6$,
3. *Positive-Negative*: $I_3 = X_1Y_2X_3Y_4X_5Y_6$,
4. *Negative-Positive*: $I_4 = Y_1X_2Y_3X_4Y_5X_6$.

XPCR-Based Recombination Algorithm

Goal: to generate the combinatorial library of n binary numbers

$$\{\alpha_1 \cdots \alpha_n \mid \alpha_i \in \{X_i, Y_i\},\ i = 1, \ldots, n\}.$$

The method starts from I_1, I_2, I_3 I_4 (extended by prefix α and suffix β), and generates the whole library, in linear time and by using essentially polymerase extension.

1. *Positive*: $I_1 = X_1X_2X_3X_4X_5X_6$,
2. *Negative*: $I_2 = Y_1Y_2Y_3Y_4Y_5Y_6$,
3. *Positive-Negative*: $I_3 = X_1Y_2X_3Y_4X_5Y_6$,
4. *Negative-Positive*: $I_4 = Y_1X_2Y_3X_4Y_5X_6$.

Recombination Algorithm

- **Let** P_1 and P_2 be two copies of the pool

$$\{\alpha I_1 \beta,\ \alpha I_2 \beta,\ \alpha I_3 \beta,\ \alpha I_4 \beta\}$$

- **for** $i = 2, 3, 4, 5$ **do**
  - **perform** $XPCR_{X_i}$ on P_1 and $XPCR_{Y_i}$ on P_2
  - **mix** the two pools obtained in the previous step in a pool $P := P_1 \cup P_2$, then **split** P randomly in two new pools P_1 and P_2 (with the same approximate size)

Example: $I_1,\ I_4 \xrightarrow{r_{X_2}} X_1X_2Y_3X_4Y_5X_6,\ Y_1X_2X_3X_4X_5X_6,$
$I_2,\ X_1X_2Y_3X_4Y_5X_6 \xrightarrow{r_{Y_5}} Y_1Y_2Y_3Y_4Y_5X_6,\ \mathbf{X_1X_2Y_3X_4Y_5Y_6}.$

Recombination Algorithm

- **Let** P_1 and P_2 be two copies of the pool

$$\{\alpha I_1 \beta,\ \alpha I_2 \beta,\ \alpha I_3 \beta,\ \alpha I_4 \beta\}$$

- **for** $i = 2, 3, 4, 5$ **do**
  - **perform** $XPCR_{X_i}$ on P_1 and $XPCR_{Y_i}$ on P_2
  - **mix** the two pools obtained in the previous step in a pool $P := P_1 \cup P_2$, then **split** P randomly in two new pools P_1 and P_2 (with the same approximate size)

Example: $I_1,\ I_4 \xrightarrow{r_{X_2}} X_1X_2Y_3X_4Y_5X_6,\ Y_1X_2X_3X_4X_5X_6,$
$I_2,\ X_1X_2Y_3X_4Y_5X_6 \xrightarrow{r_{Y_5}} Y_1Y_2Y_3Y_4Y_5X_6,\ \mathbf{X_1X_2Y_3X_4Y_5Y_6}.$

Recombination Algorithm

- **Let** P_1 and P_2 be two copies of the pool

$$\{\alpha I_1 \beta,\ \alpha I_2 \beta,\ \alpha I_3 \beta,\ \alpha I_4 \beta\}$$

- **for** $i = 2, 3, 4, 5$ **do**
  - **perform** $XPCR_{X_i}$ on P_1 and $XPCR_{Y_i}$ on P_2
  - **mix** the two pools obtained in the previous step in a pool $P := P_1 \cup P_2$, then **split** P randomly in two new pools P_1 and P_2 (with the same approximate size)

Example: $I_1,\ I_4 \xrightarrow{r_{X_2}} X_1X_2Y_3X_4Y_5X_6,\ Y_1X_2X_3X_4X_5X_6,$
$I_2,\ X_1X_2Y_3X_4Y_5X_6 \xrightarrow{r_{Y_5}} Y_1Y_2Y_3Y_4Y_5X_6,\ \mathbf{X_1X_2Y_3X_4Y_5Y_6}.$

Recombination Algorithm

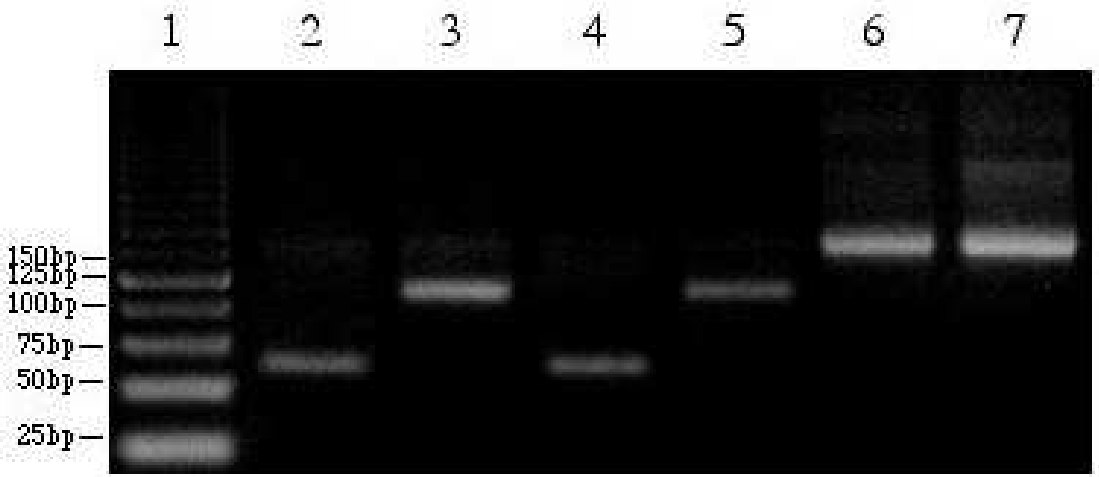
- **Let** P_1 and P_2 be two copies of the pool

$$\{\alpha I_1 \beta,\ \alpha I_2 \beta,\ \alpha I_3 \beta,\ \alpha I_4 \beta\}$$

- **for** $i = 2, 3, 4, 5$ **do**
  - **perform** $XPCR_{X_i}$ on P_1 and $XPCR_{Y_i}$ on P_2
  - **mix** the two pools obtained in the previous step in a pool $P := P_1 \cup P_2$, then **split** P randomly in two new pools P_1 and P_2 (with the same approximate size)

Example: $I_1,\ I_4 \xrightarrow{r_{X_2}} X_1X_2Y_3X_4Y_5X_6,\ Y_1X_2X_3X_4X_5X_6,$
$I_2,\ X_1X_2Y_3X_4Y_5X_6 \xrightarrow{r_{Y_5}} Y_1Y_2Y_3Y_4Y_5X_6,\ \mathbf{X_1X_2Y_3X_4Y_5Y_6}.$

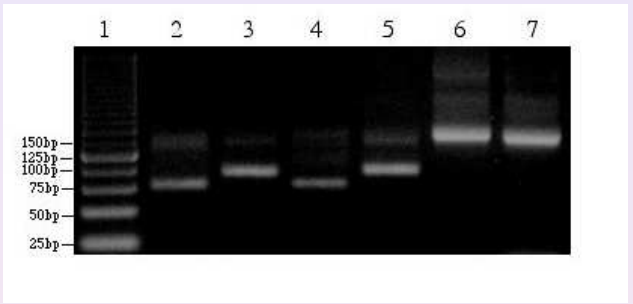
Experiment for Testing Recombination (1/2)



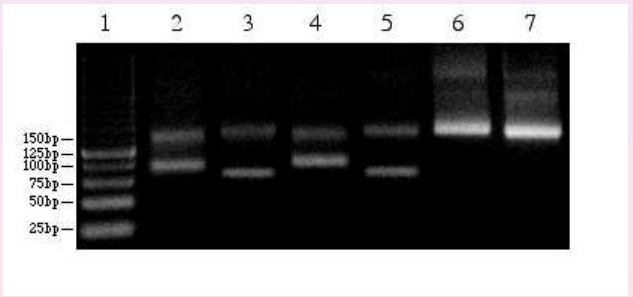
1: Marker (25bp). 2: $\alpha \cdots X_2$ (60bp), 3: $X_2 \cdots \beta$ (105bp), 4:$\alpha \cdots Y_2$ (60bp), 5: $Y_2 \cdots \beta$ (105bp). 6: $XPCR_{X_2}$ and 7: $XPCR_{Y_2}$ with X_2 and Y_2 15 long.

Experiment for Testing Recombination (2/2)

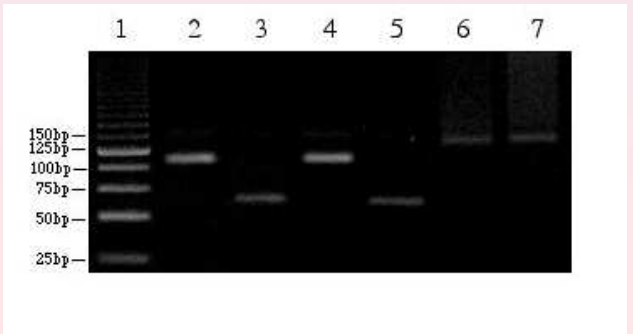
Results $XPCR_{X_i}$ and $XPCR_{Y_i}$ for i=3:



Results $XPCR_{X_i}$ and $XPCR_{Y_i}$ for i=4:

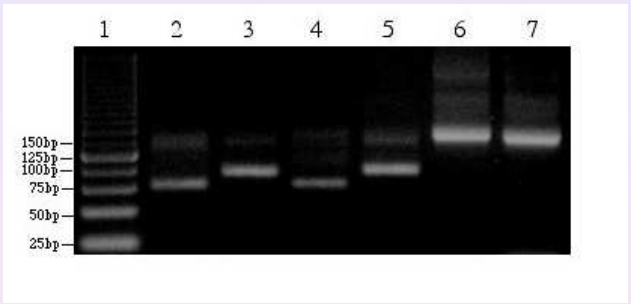


Results $XPCR_{X_i}$ and $XPCR_{Y_i}$ for i=5:

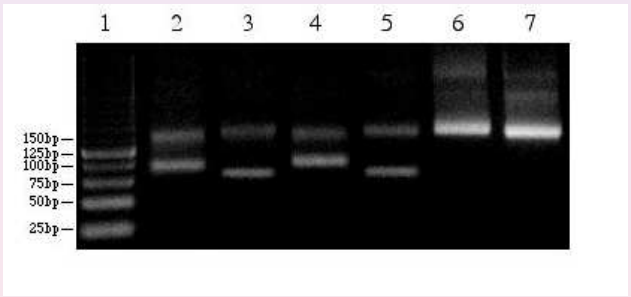


Experiment for Testing Recombination (2/2)

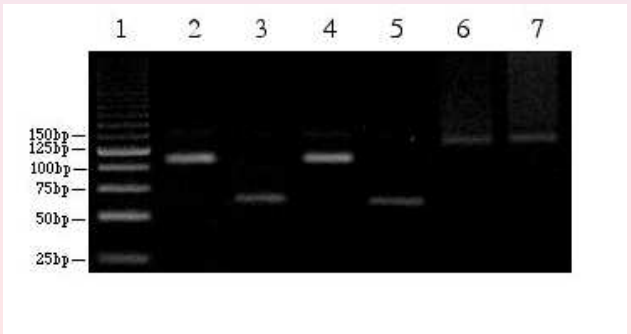
Results $XPCR_{X_i}$ and $XPCR_{Y_i}$ for i=3:



Results $XPCR_{X_i}$ and $XPCR_{Y_i}$ for i=4:



Results $XPCR_{X_i}$ and $XPCR_{Y_i}$ for i=5:



Experiment for Testing The Witnesses Presence

For each $Z_1Z_2Z_3Z_4Z_5Z_6$ of the recombination witnesses W_1 and W_2, its presence was checked in the final pool by means of the following steps:

1. **perform** $PCR(Z_1, \overline{Z_6})$, then **electrophoresis** selecting shortest strands

$$\alpha \underline{Z_1 \ldots Z_6} \beta$$

2. **perform** $PCR(Z_2, \overline{Z_5})$, then **electrophoresis** selecting shortest strands

$$Z_1 \underline{Z_2 \ldots Z_5} Z_6$$

3. **perform** $PCR(Z_3, \overline{Z_4})$

$$Z_2 \underline{Z_3 Z_4} Z_5$$

Experiment for Testing The Witnesses Presence

For each $Z_1Z_2Z_3Z_4Z_5Z_6$ of the recombination witnesses W_1 and W_2, its presence was checked in the final pool by means of the following steps:

1. **perform** $PCR(Z_1, \overline{Z_6})$, then **electrophoresis** selecting shortest strands

$$\alpha \underline{Z_1 \ldots Z_6} \beta$$

2. **perform** $PCR(Z_2, \overline{Z_5})$, then **electrophoresis** selecting shortest strands

$$Z_1 \underline{Z_2 \ldots Z_5} Z_6$$

3. **perform** $PCR(Z_3, \overline{Z_4})$

$$Z_2 \underline{Z_3 Z_4} Z_5$$

Experiment for Testing The Witnesses Presence

For each $Z_1Z_2Z_3Z_4Z_5Z_6$ of the recombination witnesses W_1 and W_2, its presence was checked in the final pool by means of the following steps:

1. **perform** $PCR(Z_1, \overline{Z_6})$, then **electrophoresis** selecting shortest strands

$$\alpha \underline{Z_1 \ldots Z_6} \beta$$

2. **perform** $PCR(Z_2, \overline{Z_5})$, then **electrophoresis** selecting shortest strands

$$Z_1 \underline{Z_2 \ldots Z_5} Z_6$$

3. **perform** $PCR(Z_3, \overline{Z_4})$

$$Z_2 \underline{Z_3 Z_4} Z_5$$

Experiment for Testing The Witnesses Presence

For each $Z_1Z_2Z_3Z_4Z_5Z_6$ of the recombination witnesses W_1 and W_2, its presence was checked in the final pool by means of the following steps:

1. **perform** $PCR(Z_1, \overline{Z_6})$, then **electrophoresis** selecting shortest strands

$$\alpha \underline{Z_1 \ldots Z_6} \beta$$

2. **perform** $PCR(Z_2, \overline{Z_5})$, then **electrophoresis** selecting shortest strands

$$Z_1 \underline{Z_2 \ldots Z_5} Z_6$$

3. **perform** $PCR(Z_3, \overline{Z_4})$

$$Z_2 \underline{Z_3 Z_4} Z_5$$

Success

Marker (25bp). 2: PCR $(X_1, \overline{X_6})$ (90bp), 3: $PCR(X_2, \overline{X_5})$ (60bp), 4: $PCR(Y_3, \overline{Y_4})$ (30bp), 5: $PCR(Y_1, \overline{Y_6})$ (90bp), 6: $PCR(Y_2, \overline{Y_5})$ (60pb), 7: $PCR(X_3, \overline{X_4})$ (30pb).

$XPCR - Mutagenesis(P, \gamma, \delta)$

let $Type(P) = \{< \alpha\gamma\beta >\}$;

1. **input** $Q : \{< \alpha[-18, -1]\ \delta\ \beta[1, 20] >\}$;
2. $(P_1, P_2) := split(P)$;
3. $P_1 := PCR(P_1, \alpha[1, 18], mir(\alpha[-18, -1]))$;
4. $P_2 := PCR(P_2, \beta[1, 20], mir(\beta[-20, -1]))$;
5. $P_1 := separate(P_1, |\alpha|)$; $P_2 := separate(P_2, |\beta|)$;
6. $P_1 := mix(P_1, Q)$;
7. $P_1 := PCR(P_1, \alpha[1, 18], mir(\beta[1, 20]))$;
8. $P_1 := separate(P_1, |\alpha| + |\delta| + 20)$;
9. $P := mix(P_1, P_2)$;
10. $P := PCR(P, \alpha[1, 18], mir(\beta[-20, -1]))$;
11. $P := separate(P, |\alpha| + |\beta| + |\delta|)$;
12. **output** P.

$XPCR - Mutagenesis(P, \gamma, \delta)$

let $Type(P) = \{< \alpha\gamma\beta >\}$;

1. **input** $Q : \{< \alpha[-18, -1]\ \delta\ \beta[1, 20] >\}$;
2. $(P_1, P_2) := split(P)$;
3. $P_1 := PCR(P_1, \alpha[1, 18], mir(\alpha[-18, -1]))$;
4. $P_2 := PCR(P_2, \beta[1, 20], mir(\beta[-20, -1]))$;
5. $P_1 := separate(P_1, |\alpha|)$; $P_2 := separate(P_2, |\beta|)$;
6. $P_1 := mix(P_1, Q)$;
7. $P_1 := PCR(P_1, \alpha[1, 18], mir(\beta[1, 20]))$;
8. $P_1 := separate(P_1, |\alpha| + |\delta| + 20)$;
9. $P := mix(P_1, P_2)$;
10. $P := PCR(P, \alpha[1, 18], mir(\beta[-20, -1]))$;
11. $P := separate(P, |\alpha| + |\beta| + |\delta|)$;
12. **output** P.

$XPCR - Mutagenesis(P, \gamma, \delta)$

let $Type(P) = \{< \alpha\gamma\beta >\}$;

1. **input** $Q : \{< \alpha[-18, -1]\ \delta\ \beta[1, 20] >\}$;
2. $(P_1, P_2) := split(P)$;
3. $P_1 := PCR(P_1, \alpha[1, 18], mir(\alpha[-18, -1]))$;
4. $P_2 := PCR(P_2, \beta[1, 20], mir(\beta[-20, -1]))$;
5. $P_1 := separate(P_1, |\alpha|)$; $P_2 := separate(P_2, |\beta|)$;
6. $P_1 := mix(P_1, Q)$;
7. $P_1 := PCR(P_1, \alpha[1, 18], mir(\beta[1, 20]))$;
8. $P_1 := separate(P_1, |\alpha| + |\delta| + 20)$;
9. $P := mix(P_1, P_2)$;
10. $P := PCR(P, \alpha[1, 18], mir(\beta[-20, -1]))$;
11. $P := separate(P, |\alpha| + |\beta| + |\delta|)$;
12. **output** P.

$XPCR - Mutagenesis(P, \gamma, \delta)$

let $Type(P) = \{< \alpha\gamma\beta >\}$;

1. **input** $Q : \{< \alpha[-18, -1]\ \delta\ \beta[1, 20] >\}$;
2. $(P_1, P_2) := split(P)$;
3. $P_1 := PCR(P_1, \alpha[1, 18], mir(\alpha[-18, -1]))$;
4. $P_2 := PCR(P_2, \beta[1, 20], mir(\beta[-20, -1]))$;
5. $P_1 := separate(P_1, |\alpha|)$; $P_2 := separate(P_2, |\beta|)$;
6. $P_1 := mix(P_1, Q)$;
7. $P_1 := PCR(P_1, \alpha[1, 18], mir(\beta[1, 20]))$;
8. $P_1 := separate(P_1, |\alpha| + |\delta| + 20)$;
9. $P := mix(P_1, P_2)$;
10. $P := PCR(P, \alpha[1, 18], mir(\beta[-20, -1]))$;
11. $P := separate(P, |\alpha| + |\beta| + |\delta|)$;
12. **output** P.

$XPCR - Mutagenesis(P, \gamma, \delta)$

let $Type(P) = \{< \alpha\gamma\beta >\}$;

1. **input** $Q : \{< \alpha[-18, -1]\ \delta\ \beta[1, 20] >\}$;
2. $(P_1, P_2) := split(P)$;
3. $P_1 := PCR(P_1, \alpha[1, 18], mir(\alpha[-18, -1]))$;
4. $P_2 := PCR(P_2, \beta[1, 20], mir(\beta[-20, -1]))$;
5. $P_1 := separate(P_1, |\alpha|)$; $P_2 := separate(P_2, |\beta|)$;
6. $P_1 := mix(P_1, Q)$;
7. $P_1 := PCR(P_1, \alpha[1, 18], mir(\beta[1, 20]))$;
8. $P_1 := separate(P_1, |\alpha| + |\delta| + 20)$;
9. $P := mix(P_1, P_2)$;
10. $P := PCR(P, \alpha[1, 18], mir(\beta[-20, -1]))$;
11. $P := separate(P, |\alpha| + |\beta| + |\delta|)$;
12. **output** P.

$XPCR - Mutagenesis(P, \gamma, \delta)$

let $Type(P) = \{< \alpha\gamma\beta >\}$;

1. **input** $Q : \{< \alpha[-18,-1]\ \delta\ \beta[1,20] >\}$;
2. $(P_1, P_2) := split(P)$;
3. $P_1 := PCR(P_1, \alpha[1,18], mir(\alpha[-18,-1]))$;
4. $P_2 := PCR(P_2, \beta[1,20], mir(\beta[-20,-1]))$;
5. $P_1 := separate(P_1, |\alpha|)$; $P_2 := separate(P_2, |\beta|)$;
6. $P_1 := mix(P_1, Q)$;
7. $P_1 := PCR(P_1, \alpha[1,18], mir(\beta[1,20]))$;
8. $P_1 := separate(P_1, |\alpha| + |\delta| + 20)$;
9. $P := mix(P_1, P_2)$;
10. $P := PCR(P, \alpha[1,18], mir(\beta[-20,-1]))$;
11. $P := separate(P, |\alpha| + |\beta| + |\delta|)$;
12. **output** P.

$XPCR - Mutagenesis(P, \gamma, \delta)$

let $Type(P) = \{< \alpha\gamma\beta >\}$;

1. **input** $Q : \{< \alpha[-18, -1]\ \delta\ \beta[1, 20] >\}$;
2. $(P_1, P_2) := split(P)$;
3. $P_1 := PCR(P_1, \alpha[1, 18], mir(\alpha[-18, -1]))$;
4. $P_2 := PCR(P_2, \beta[1, 20], mir(\beta[-20, -1]))$;
5. $P_1 := separate(P_1, |\alpha|)$; $P_2 := separate(P_2, |\beta|)$;
6. $P_1 := mix(P_1, Q)$;
7. $P_1 := PCR(P_1, \alpha[1, 18], mir(\beta[1, 20]))$;
8. $P_1 := separate(P_1, |\alpha| + |\delta| + 20)$;
9. $P := mix(P_1, P_2)$;
10. $P := PCR(P, \alpha[1, 18], mir(\beta[-20, -1]))$;
11. $P := separate(P, |\alpha| + |\beta| + |\delta|)$;
12. **output** P.

$XPCR - Mutagenesis(P, \gamma, \delta)$

let $Type(P) = \{< \alpha\gamma\beta >\}$;

1. **input** $Q : \{< \alpha[-18,-1]\ \delta\ \beta[1,20] >\}$;
2. $(P_1, P_2) := split(P)$;
3. $P_1 := PCR(P_1, \alpha[1,18], mir(\alpha[-18,-1]))$;
4. $P_2 := PCR(P_2, \beta[1,20], mir(\beta[-20,-1]))$;
5. $P_1 := separate(P_1, |\alpha|)$; $P_2 := separate(P_2, |\beta|)$;
6. $P_1 := mix(P_1, Q)$;
7. $P_1 := PCR(P_1, \alpha[1,18], mir(\beta[1,20]))$;
8. $P_1 := separate(P_1, |\alpha| + |\delta| + 20)$;
9. $P := mix(P_1, P_2)$;
10. $P := PCR(P, \alpha[1,18], mir(\beta[-20,-1]))$;
11. $P := separate(P, |\alpha| + |\beta| + |\delta|)$;
12. **output** P.

$XPCR - Mutagenesis(P, \gamma, \delta)$

let $Type(P) = \{< \alpha\gamma\beta >\}$;

1. **input** $Q : \{< \alpha[-18, -1]\ \delta\ \beta[1, 20] >\}$;
2. $(P_1, P_2) := split(P)$;
3. $P_1 := PCR(P_1, \alpha[1, 18], mir(\alpha[-18, -1]))$;
4. $P_2 := PCR(P_2, \beta[1, 20], mir(\beta[-20, -1]))$;
5. $P_1 := separate(P_1, |\alpha|)$; $P_2 := separate(P_2, |\beta|)$;
6. $P_1 := mix(P_1, Q)$;
7. $P_1 := PCR(P_1, \alpha[1, 18], mir(\beta[1, 20]))$;
8. $P_1 := separate(P_1, |\alpha| + |\delta| + 20)$;
9. $P := mix(P_1, P_2)$;
10. $P := PCR(P, \alpha[1, 18], mir(\beta[-20, -1]))$;
11. $P := separate(P, |\alpha| + |\beta| + |\delta|)$;
12. **output** P.

$XPCR - Mutagenesis(P, \gamma, \delta)$

let $Type(P) = \{< \alpha\gamma\beta >\}$;

1. **input** $Q : \{< \alpha[-18, -1]\ \delta\ \beta[1, 20] >\}$;
2. $(P_1, P_2) := split(P)$;
3. $P_1 := PCR(P_1, \alpha[1, 18], mir(\alpha[-18, -1]))$;
4. $P_2 := PCR(P_2, \beta[1, 20], mir(\beta[-20, -1]))$;
5. $P_1 := separate(P_1, |\alpha|)$; $P_2 := separate(P_2, |\beta|)$;
6. $P_1 := mix(P_1, Q)$;
7. $P_1 := PCR(P_1, \alpha[1, 18], mir(\beta[1, 20]))$;
8. $P_1 := separate(P_1, |\alpha| + |\delta| + 20)$;
9. $P := mix(P_1, P_2)$;
10. $P := PCR(P, \alpha[1, 18], mir(\beta[-20, -1]))$;
11. $P := separate(P, |\alpha| + |\beta| + |\delta|)$;
12. **output** P.

$XPCR - Mutagenesis(P, \gamma, \delta)$

let $Type(P) = \{< \alpha\gamma\beta >\}$;

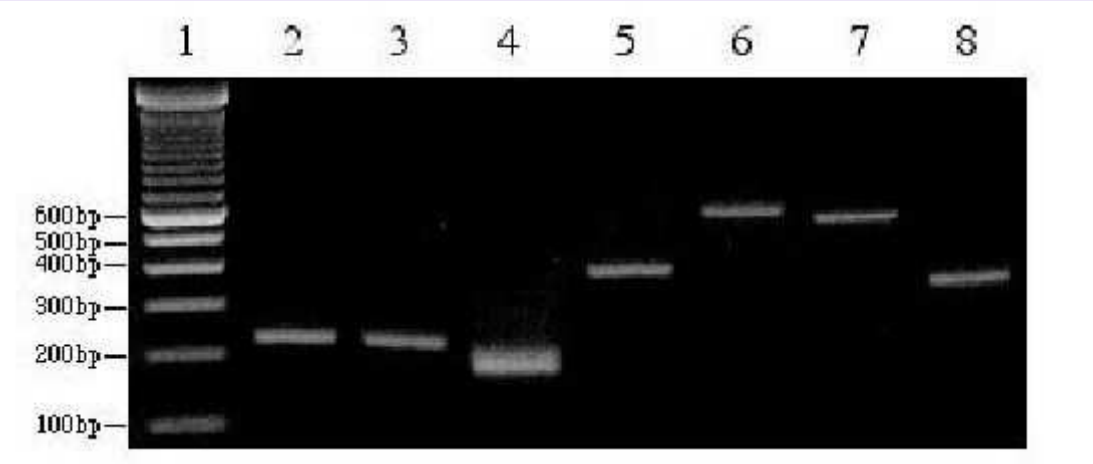
1. **input** $Q : \{< \alpha[-18, -1]\ \delta\ \beta[1, 20] >\}$;
2. $(P_1, P_2) := split(P)$;
3. $P_1 := PCR(P_1, \alpha[1, 18], mir(\alpha[-18, -1]))$;
4. $P_2 := PCR(P_2, \beta[1, 20], mir(\beta[-20, -1]))$;
5. $P_1 := separate(P_1, |\alpha|)$; $P_2 := separate(P_2, |\beta|)$;
6. $P_1 := mix(P_1, Q)$;
7. $P_1 := PCR(P_1, \alpha[1, 18], mir(\beta[1, 20]))$;
8. $P_1 := separate(P_1, |\alpha| + |\delta| + 20)$;
9. $P := mix(P_1, P_2)$;
10. $P := PCR(P, \alpha[1, 18], mir(\beta[-20, -1]))$;
11. $P := separate(P, |\alpha| + |\beta| + |\delta|)$;
12. **output** P.

$XPCR - Mutagenesis(P, \gamma, \delta)$

let $Type(P) = \{< \alpha\gamma\beta >\}$;

1. **input** $Q : \{< \alpha[-18, -1]\ \delta\ \beta[1, 20] >\}$;
2. $(P_1, P_2) := split(P)$;
3. $P_1 := PCR(P_1, \alpha[1, 18], mir(\alpha[-18, -1]))$;
4. $P_2 := PCR(P_2, \beta[1, 20], mir(\beta[-20, -1]))$;
5. $P_1 := separate(P_1, |\alpha|)$; $P_2 := separate(P_2, |\beta|)$;
6. $P_1 := mix(P_1, Q)$;
7. $P_1 := PCR(P_1, \alpha[1, 18], mir(\beta[1, 20]))$;
8. $P_1 := separate(P_1, |\alpha| + |\delta| + 20)$;
9. $P := mix(P_1, P_2)$;
10. $P := PCR(P, \alpha[1, 18], mir(\beta[-20, -1]))$;
11. $P := separate(P, |\alpha| + |\beta| + |\delta|)$;
12. **output** P.

Experiment for Testing Mutagenesis



DNA Mutagenesis. *1*: molecular size marker ladder (100bp). *2*: amplification of strand α (230bp). *3*: amplification of strand β (229bp). *4*: amplification of strand $\alpha[-18,-1]\delta\beta[1,20]$ (188bp). *5*: cross pairing amplification of α and $\alpha[-18,-1]\delta\beta[1,20]$ (400bp). *6*: cross pairing amplification of β and $\alpha\delta\beta[1,20]$ (609bp). *7*: RhoA (582bp). *8*: positive control by $PCR(\alpha[26,46],\beta[-20,-1])$ 354 bp. All PCRs at 58°C.

References


G. F., C. Giagulli, C. Laudanna, V. Manca (2005).
DNA Extraction by XPCR. DNA10, LNCS 3384, 104-112.

G. F., V. Manca, C. Giagulli, C. Laudanna (2006).
DNA Recombination by XPCR. DNA11, LNCS 3892, 55-66.

G. F. (2006).
Biomolecular Computing - Combinatorial Algorithms and Laboratory Experiments. PhD Thesis, Library of Computer Science Department, University of Verona.

References


G. F., C. Giagulli, C. Laudanna, V. Manca (2005).
DNA Extraction by XPCR. DNA10, LNCS 3384, 104-112.


G. F., V. Manca, C. Giagulli, C. Laudanna (2006).
DNA Recombination by XPCR. DNA11, LNCS 3892, 55-66.

G. F. (2006).
Biomolecular Computing - Combinatorial Algorithms and Laboratory Experiments. PhD Thesis, Library of Computer Science Department, University of Verona.

References


G. F., C. Giagulli, C. Laudanna, V. Manca (2005).
DNA Extraction by XPCR. DNA10, LNCS 3384, 104-112.


G. F., V. Manca, C. Giagulli, C. Laudanna (2006).
DNA Recombination by XPCR. DNA11, LNCS 3892, 55-66.


G. F. (2006).
Biomolecular Computing - Combinatorial Algorithms and Laboratory Experiments. PhD Thesis, Library of Computer Science Department, University of Verona.

Milner's quotation

In 1986, at the inauguration of the Laboratory for Foundation of Computer Science at the Department of Computer Science of the University of Edinburgh, Robin Milner titled his lecture with

Is Computing an Experimental Science?

A Positive Answer:

- Successful DNA Computing Experiments!

Milner's quotation

In 1986, at the inauguration of the Laboratory for Foundation of Computer Science at the Department of Computer Science of the University of Edinburgh, Robin Milner titled his lecture with

Is Computing an Experimental Science?

A Positive Answer:

- Successful DNA Computing Experiments!

Milner's quotation

In 1986, at the inauguration of the Laboratory for Foundation of Computer Science at the Department of Computer Science of the University of Edinburgh, Robin Milner titled his lecture with

Is Computing an Experimental Science?

A Positive Answer:

- Successful DNA Computing Experiments!