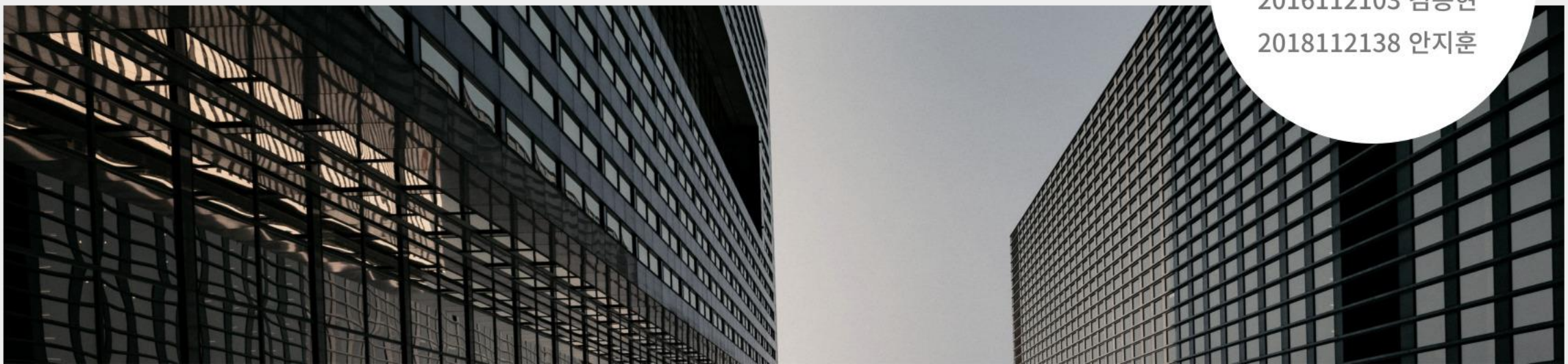
dongguk UNIVERSITY

Team USB.

Multiple Sequence Alignment with Genetic Algorithm

With. Doungguk Univ.

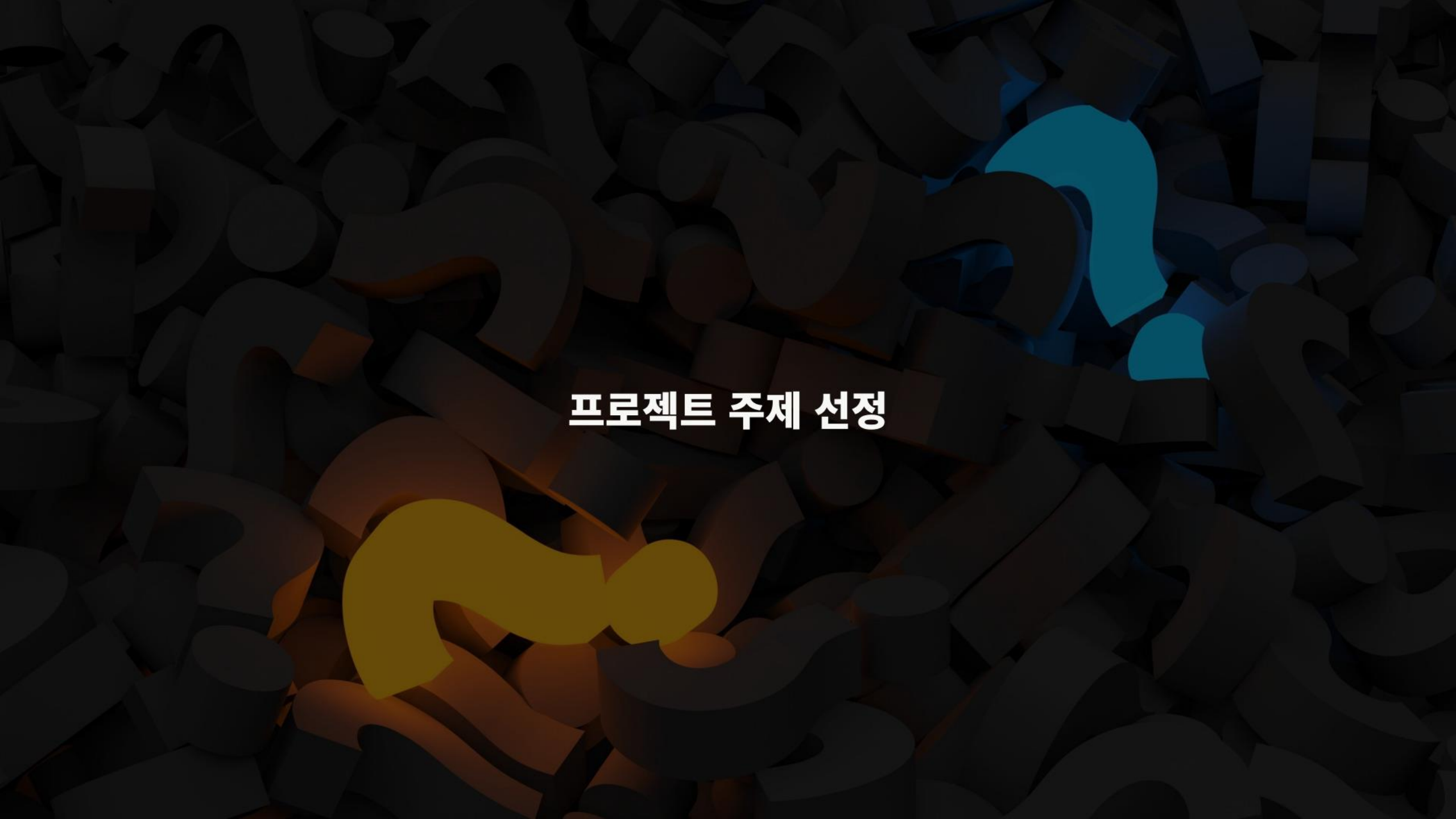
2017112139 김규열
2017112068 김관중
2016112103 김동현
2018112138 안지훈



dongguk UNIVERSITY

Contents

1. 주제 선정
2. Genetic Algorithm
3. SAGA
   (Sequecnce Alignment by Genetic Algorithm)
4. Experiment & Result



프로젝트 주제 선정

프로젝트 주제 선정

1. Global Alignment

```
--T--CC-C-AGT--TATGT-CAGGGGACACG--A-GCATGCAGA-GAC
  |   || |   ||   | | | |||     || |   | |   | ||||    |
AATTGCCGCC-GTCGT-T-TTCAG----CA-GTTATG--T-CAGAT--C
```

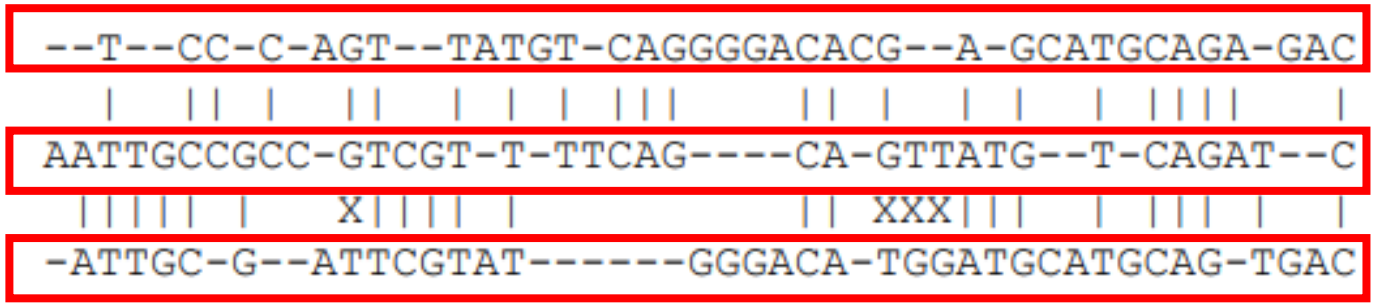
2. Local Alignment

```
                      tccCAGTTATGTCAGgggacacgagcatgcagagac
                         ||||||||||||
aattgccgccgtcgttttcagCAGTTATGTCAGatc
```

2개의 sequence에 대하여 alignment 수행 (Pairwise alignment)
→ 서열의 유사도 분석에 이용

프로젝트 주제 선정

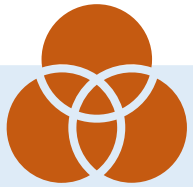
Multiple Alignment



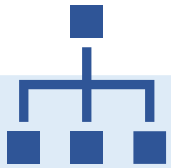
3개 이상의 sequence에 대하여 alignment 수행

프로젝트 주제 선정

Pairwise Alignment보다 더 많은 생물학적 정보를 나타낼 수 있음



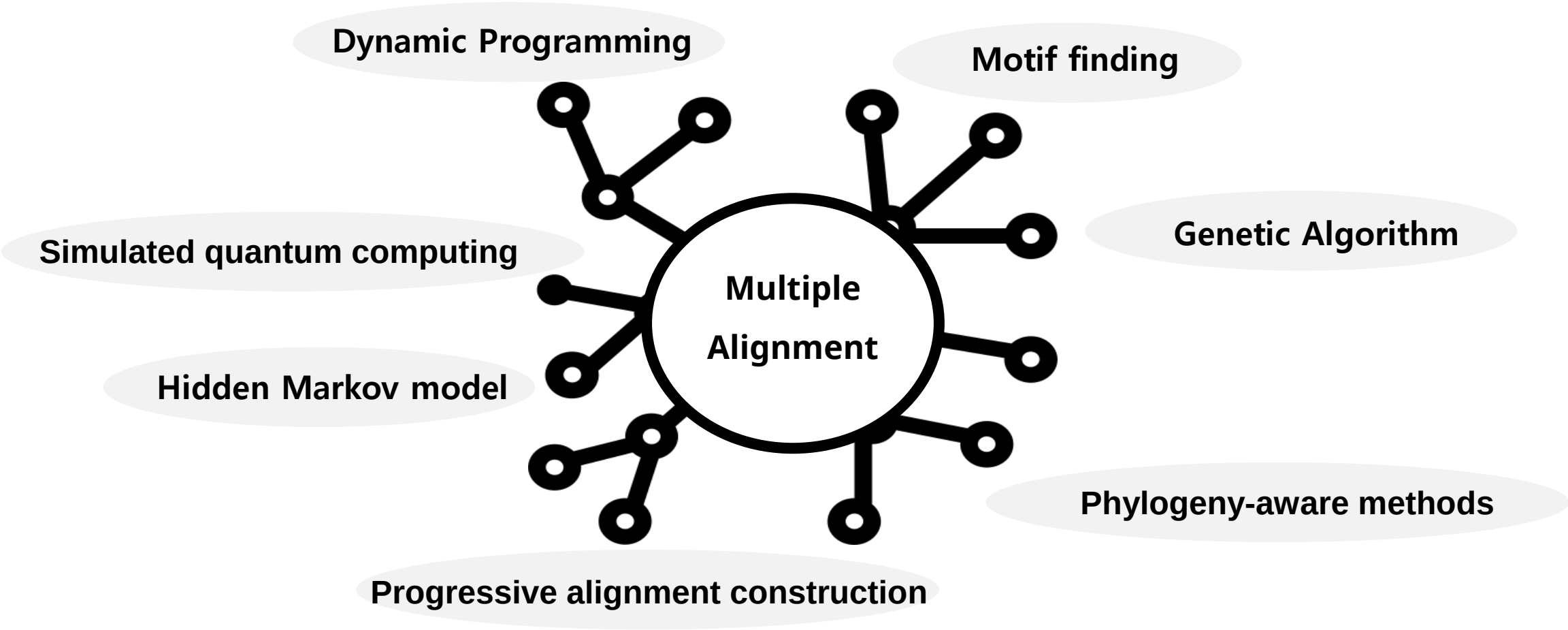
단백질, 서열 군에서
가변성 또는 보존 영역을 탐지



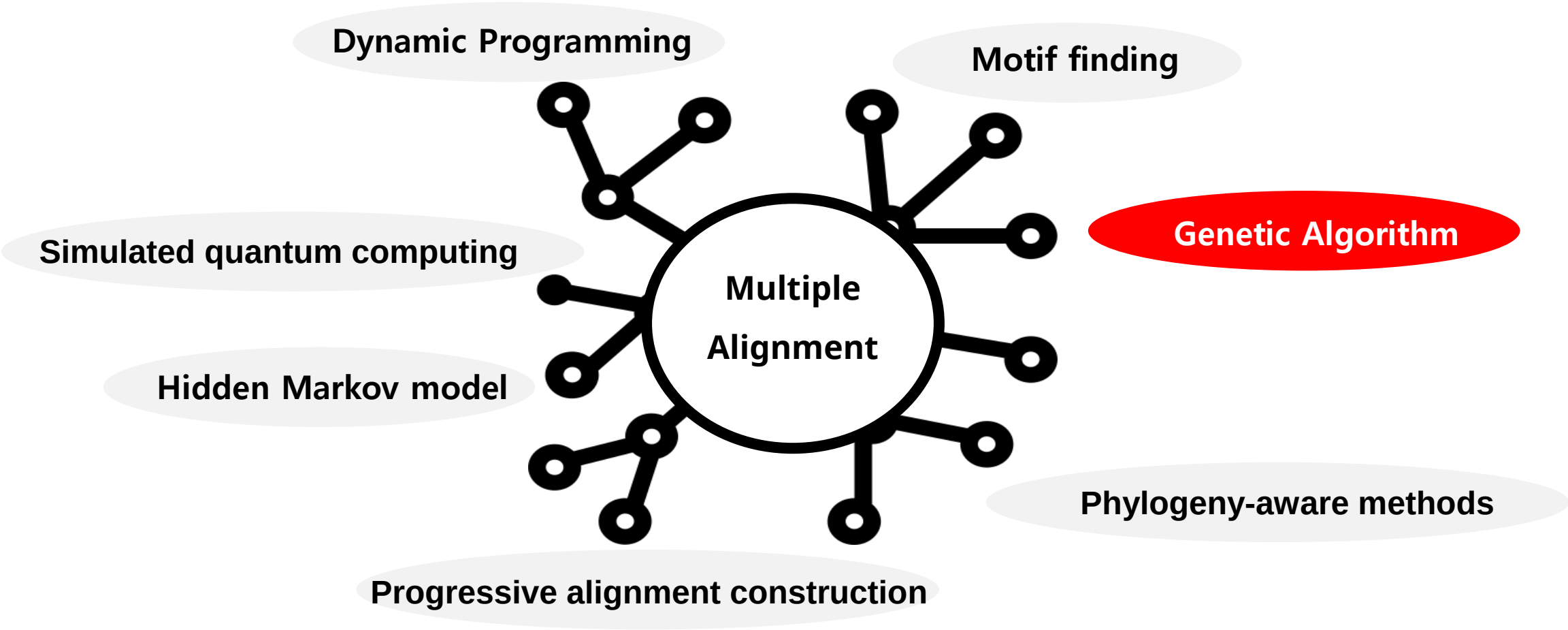
계통발생학적 분석을 통한
진화율 추정

기존 서열군 단백질 구조 예측
새롭게 서열화된 유전자 사이의
상동성 검출

프로젝트 주제 선정


Dynamic Programming
Motif finding
Simulated quantum computing
Multiple
Alignment
Genetic Algorithm
Hidden Markov model
Phylogeny-aware methods
Progressive alignment construction

프로젝트 주제 선정


Dynamic Programming
Motif finding
Simulated quantum computing
Multiple
Alignment
Genetic Algorithm
Hidden Markov model
Phylogeny-aware methods
Progressive alignment construction

프로젝트 주제 선정

Why? Genetic Algorithm..

1.

다윈의 진화론을 모방한 최적화 알고리즘

→ 바이오를 주제로 하는 수업과 그 연관이 있다고 판단

2.

관련 논문, 저널 기사 존재

→ 프로젝트 구현, 결과 분석시 활용 가능

3.

다양한 분야, 특히 인공지능 분야에서 활발히 사용

→ 추후 다른 설계 프로젝트에 활용 여지가 높음

프로젝트 주제 선정

Genetic Algorithm의 다양한 응용분야

응용 분야	응용 사례
최적화	수치적 함수의 최적화, 유전자 정보 해석 등
설계	비행기 날개의 공기 역학적 설계, 컴퓨터 통신망의 최적 설계 등
인공지능	문제해결 규칙의 자동 습득, 신경망 합성 및 학습, 패턴 인식, 자연언어 처리 등
시스템 분석 및 예측	시스템 동정, 케이오틱 시계열의 예측, 환율 변화 예측, 단백질 구조 분석 등
제어 및 로보틱스	이동 로봇의 경로 계획, 신경망 및 퍼지로직과 유전알고리즘의 결합에 의한 제어

Genetic Algorithm은 풀고자 하는 문제에 대한 가능해들을 일정한 형태의 자료구조(**유전자**)로 표현한 후, 점차 변형하여 더 나은 해를 만들어 내는 과정을 거침

프로젝트 주제 선정

주제.

“Multiple Sequence Alignment with Genetic Algorithm”

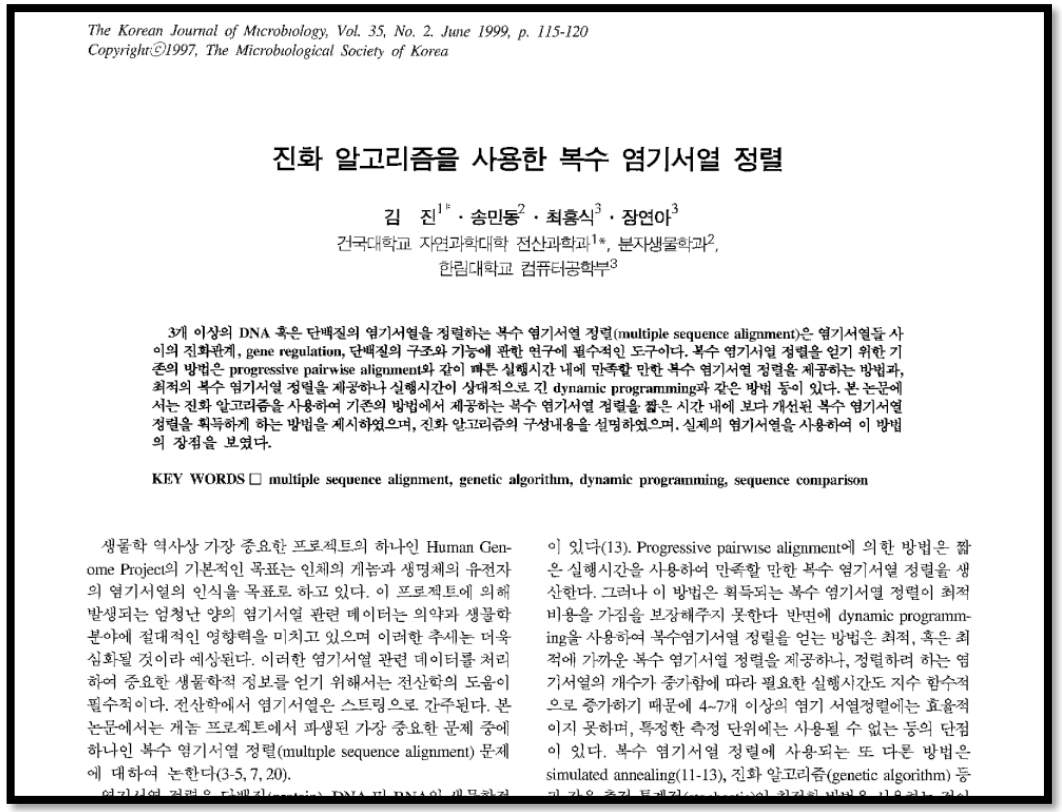
→ <Genetic Algorithm, MSA 연구 수행>

→ <Python을 활용한 개발 진행>

→ <Clustal W 기법과 결과 비교 진행>

프로젝트 주제 선정

1. 진화 알고리즘을 사용한 복수 염기서열 정렬



The Korean Journal of Microbiology, Vol. 35, No. 2. June 1999, p. 115-120

진화 알고리즘을 사용한 복수 염기서열 정렬

김 진[1*] · 송민동[2] · 최흥식[3] · 장연아[3]
건국대학교 자연과학대학 전산과학과[1*], 분자생물학과[2],
한림대학교 컴퓨터공학부[3]

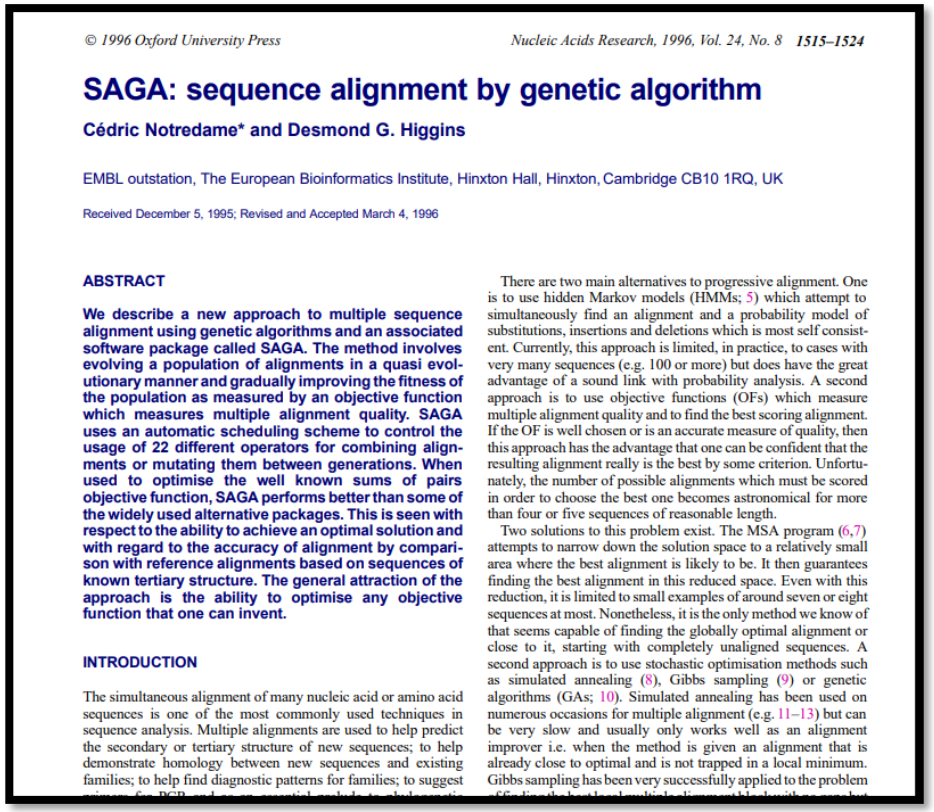
3개 이상의 DNA 혹은 단백질의 염기서열을 정렬하는 복수 염기서열 정렬(multiple sequence alignment)은 염기서열들 사이의 진화관계, gene regulation, 단백질의 구조와 기능에 관한 연구에 필수적인 도구이다. 복수 염기서열 정렬을 얻기 위한 기존의 방법은 progressive pairwise alignment와 같이 빠른 실행시간 내에 만족할 만한 복수 염기서열 정렬을 제공하는 방법과, 최적의 복수 염기서열 정렬을 제공하나 실행시간이 상대적으로 긴 dynamic programming과 같은 방법 등이 있다. 본 논문에서는 진화 알고리즘을 사용하여 기존의 방법에서 제공하는 복수 염기서열 정렬을 짧은 시간 내에 보다 개선된 복수 염기서열 정렬을 획득하게 하는 방법을 제시하였으며, 진화 알고리즘의 구성내용을 설명하였으며, 실제의 염기서열을 사용하여 이 방법의 장점을 보였다.

KEY WORDS □ multiple sequence alignment, genetic algorithm, dynamic programming, sequence comparison

생물학 역사상 가장 중요한 프로젝트의 하나인 Human Genome Project의 기본적인 목표는 인체의 게놈과 생명체의 유전자의 염기서열의 인식을 목표로 하고 있다. 이 프로젝트에 의해 발생되는 엄청난 양의 염기서열 관련 데이터는 의약과 생물학 분야에 절대적인 영향력을 미치고 있으며 이러한 추세는 더욱 심화될 것이라 예상된다. 이러한 염기서열 관련 데이터를 처리하여 중요한 생물학적 정보를 얻기 위해서는 전산학의 도움이 필수적이다. 전산학에서 염기서열은 스트링으로 간주된다. 본 논문에서는 게놈 프로젝트에서 파생된 가장 중요한 문제 중에 하나인 복수 염기서열 정렬(multiple sequence alignment) 문제에 대하여 논한다(3-5, 7, 20).

이 있다(13). Progressive pairwise alignment에 의한 방법은 짧은 실행시간을 사용하여 만족할 만한 복수 염기서열 정렬을 생산한다. 그러나 이 방법은 획득되는 복수 염기서열 정렬이 최적 비용을 가짐을 보장해주지 못한다. 반면에 dynamic programming을 사용하여 복수염기서열 정렬을 얻는 방법은 최적, 혹은 최적에 가까운 복수 염기서열 정렬을 제공하나, 정렬하려 하는 염기서열의 개수가 증가함에 따라 필요한 실행시간도 지수 함수적으로 증가하기 때문에 4~7개 이상의 염기 서열정렬에는 효율적이지 못하며, 특정한 측정 단위에는 사용될 수 없는 등의 단점이 있다. 복수 염기서열 정렬에 사용되는 또 다른 방법은 simulated annealing(11-13), 진화 알고리즘(genetic algorithm) 등

2. SAGA: Sequence Alignment by Genetic Algorithm



Nucleic Acids Research, 1996, Vol. 24, No. 8 **1515–1524**

SAGA: sequence alignment by genetic algorithm

Cédric Notredame* and Desmond G. Higgins

EMBL outstation, The European Bioinformatics Institute, Hinxton Hall, Hinxton, Cambridge CB10 1RQ, UK

Received December 5, 1995; Revised and Accepted March 4, 1996

ABSTRACT

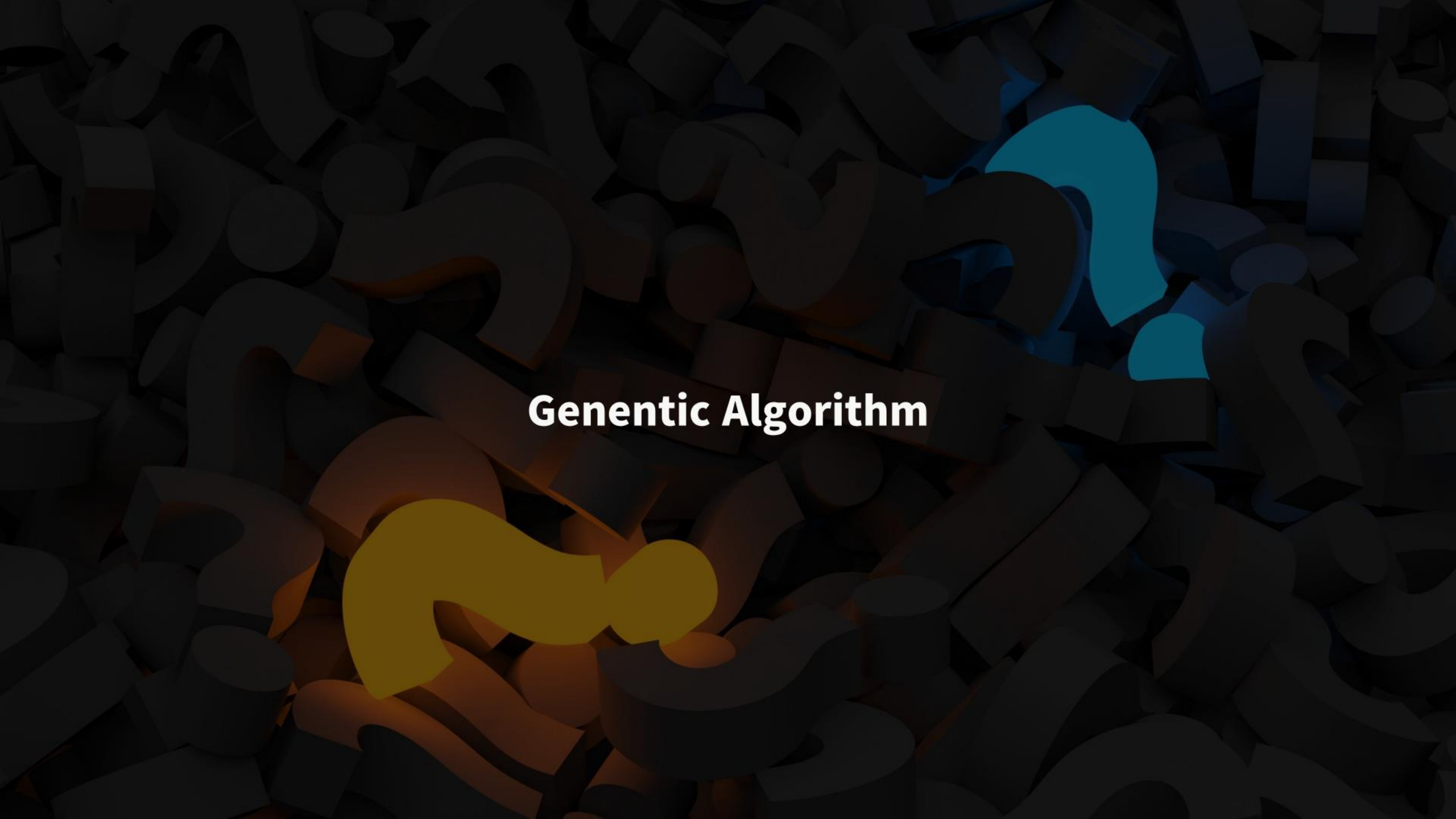
We describe a new approach to multiple sequence alignment using genetic algorithms and an associated software package called SAGA. The method involves evolving a population of alignments in a quasi evolutionary manner and gradually improving the fitness of the population as measured by an objective function which measures multiple alignment quality. SAGA uses an automatic scheduling scheme to control the usage of 22 different operators for combining alignments or mutating them between generations. When used to optimise the well known sums of pairs objective function, SAGA performs better than some of the widely used alternative packages. This is seen with respect to the ability to achieve an optimal solution and with regard to the accuracy of alignment by comparison with reference alignments based on sequences of known tertiary structure. The general attraction of the approach is the ability to optimise any objective function that one can invent.

INTRODUCTION

The simultaneous alignment of many nucleic acid or amino acid sequences is one of the most commonly used techniques in sequence analysis. Multiple alignments are used to help predict the secondary or tertiary structure of new sequences; to help demonstrate homology between new sequences and existing families; to help find diagnostic patterns for families; to suggest

There are two main alternatives to progressive alignment. One is to use hidden Markov models (HMMs; 5) which attempt to simultaneously find an alignment and a probability model of substitutions, insertions and deletions which is most self consistent. Currently, this approach is limited, in practice, to cases with very many sequences (e.g. 100 or more) but does have the great advantage of a sound link with probability analysis. A second approach is to use objective functions (OFs) which measure multiple alignment quality and to find the best scoring alignment. If the OF is well chosen or is an accurate measure of quality, then this approach has the advantage that one can be confident that the resulting alignment really is the best by some criterion. Unfortunately, the number of possible alignments which must be scored in order to choose the best one becomes astronomical for more than four or five sequences of reasonable length.

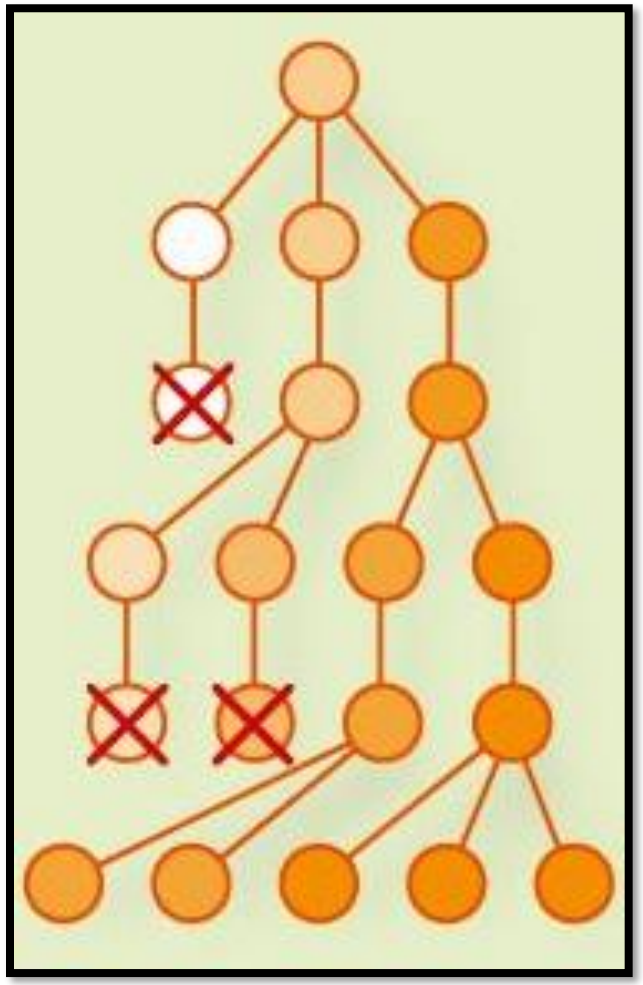
Two solutions to this problem exist. The MSA program (6,7) attempts to narrow down the solution space to a relatively small area where the best alignment is likely to be. It then guarantees finding the best alignment in this reduced space. Even with this reduction, it is limited to small examples of around seven or eight sequences at most. Nonetheless, it is the only method we know of that seems capable of finding the globally optimal alignment or close to it, starting with completely unaligned sequences. A second approach is to use stochastic optimisation methods such as simulated annealing (8), Gibbs sampling (9) or genetic algorithms (GAs; 10). Simulated annealing has been used on numerous occasions for multiple alignment (e.g. 11–13) but can be very slow and usually only works well as an alignment improver i.e. when the method is given an alignment that is already close to optimal and is not trapped in a local minimum. Gibbs sampling has been very successfully applied to the problem


Genentic Algorithm

Genetic Algorithm

Genetic Algorithm

- 다윈의 자연선택 이론을 적용한 탐색, 최적화 알고리즘
- 문제에 대한 가능한 해들을 나열한 뒤, 교배, 변이를 통해 정확도가 높고 좋은 해를 생산하게 된다
- 최적의 해를 찾는 방법론

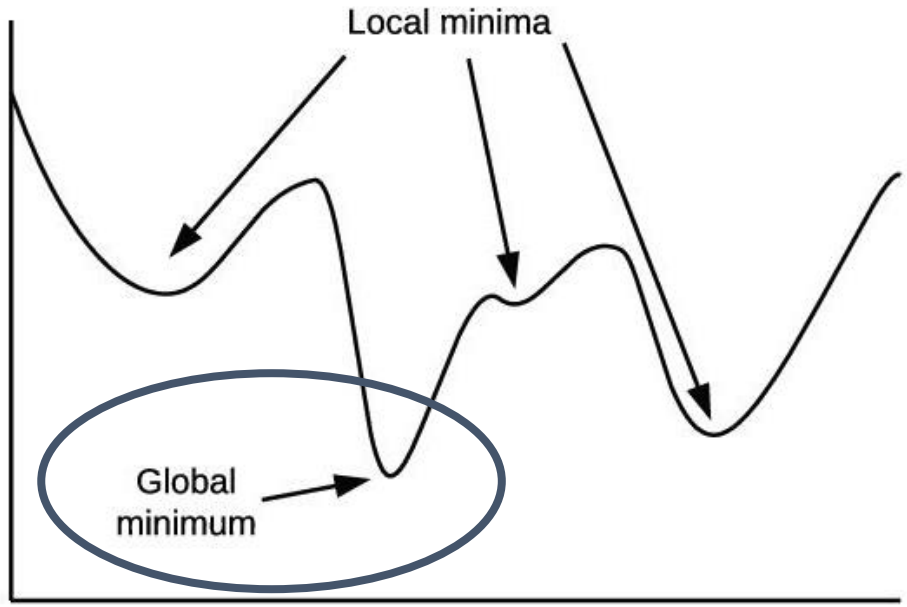


Genetic Algorithm

최적화 기법 중 전체 탐색영역에서 가장 좋은 해를 찾는 것을 목적으로 하는 **전역 최적화 기법**

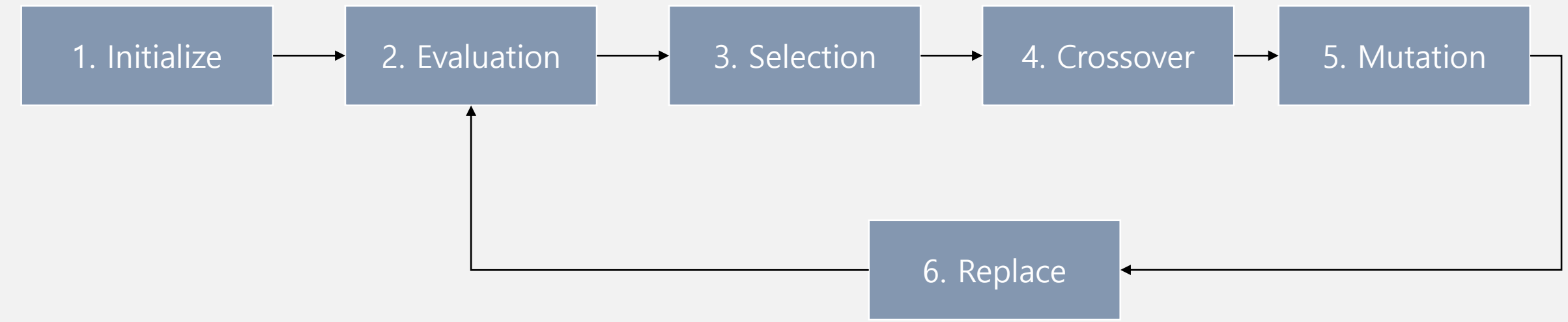
진화를 모방한 일종의 탐색 알고리즘

유전 알고리즘은 문제를 풀기위한 알고리즘이기 보단 문제 풀이를 위한 **접근 방법**에 가까움



Genetic Algorithm

Genetic Algorithm Flow Chart



Genetic Algorithm

1. Initialize

- 개체들로부터 모집단 생성 과정

1	0	0	1
1	1	0	1
0	0	1	1
1	1	0	0

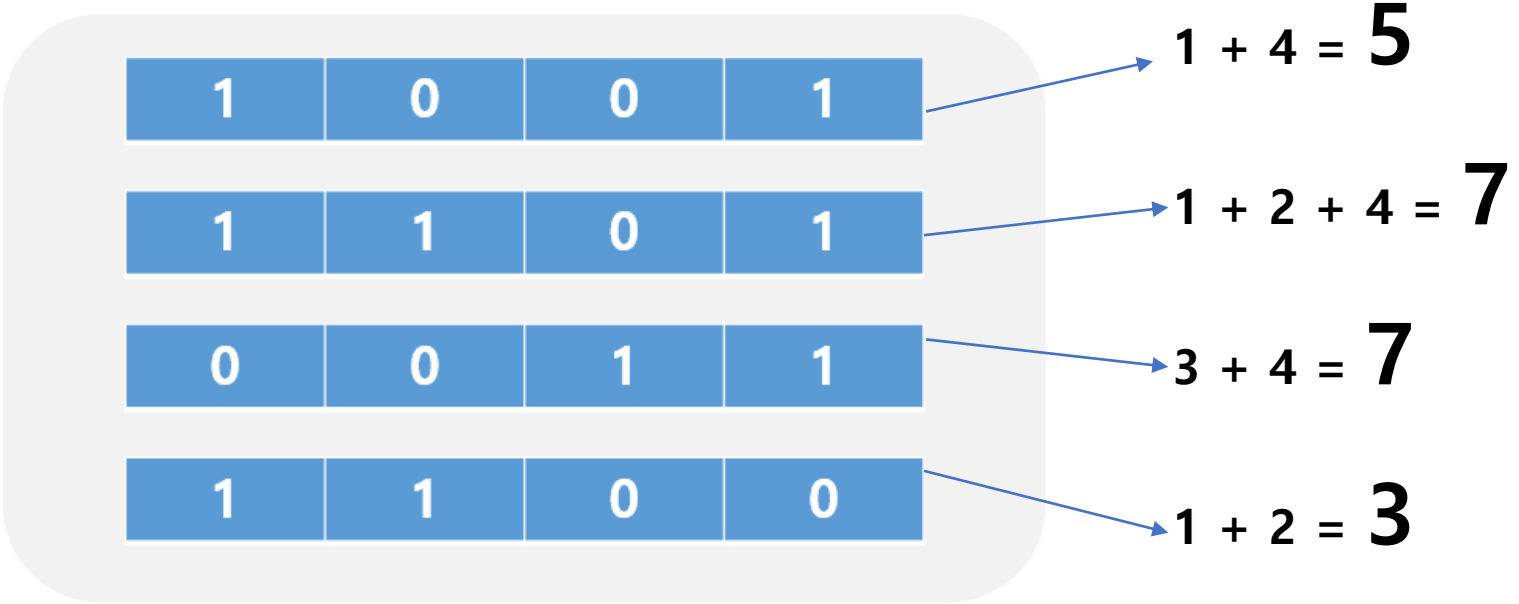
Genetic Algorithm

2. Evaluation

- 모집단 내 각 개체에 대해 적합도 함수를 통해 적합도 측정

→ 일정 적합도 이상의 개체가 탄생하면, Termination

$$f(x) = \sum_1^n i \times arr[i]$$



Genetic Algorithm

적합도

- 평가하고자 하는 요소와 얼마나 부합하는지를 나타내는 척도

적합도 함수

- 현재 세대의 개체들을 평가하는 함수
- 다음 세대에 포함될 개체들을 선택하는 기준
- 유전 알고리즘을 적용하고자 하는 문제에 따라 다르게 설계

$$f(x) = \sum_{1}^{n} i \times arr[i]$$

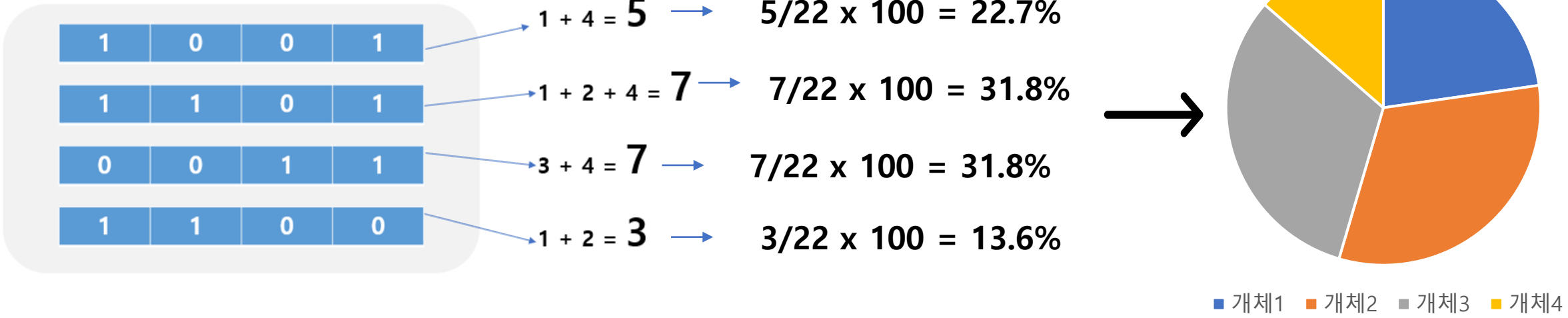
$$\text{교체비용}(A) = \sum_{i=1}^{n-1} \sum_{j=i}^{n} \text{교체비용}(S_i', S_j')$$

$$c(\alpha) = \sum_{0<i<j\leq n} c_2(\alpha_{ij}).$$

Genetic Algorithm

3. Selection

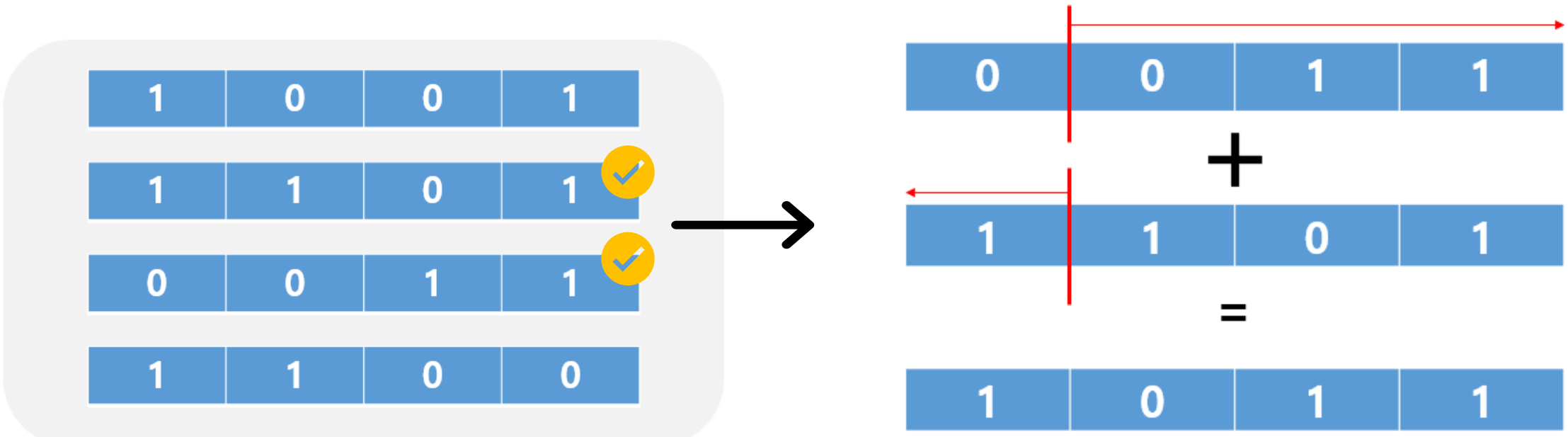
- 적합도에 따라 부여된 확률에 기반하여, 교배를 진행할 개체 선택하는 Roulette-wheel selection을 진행



Genetic Algorithm

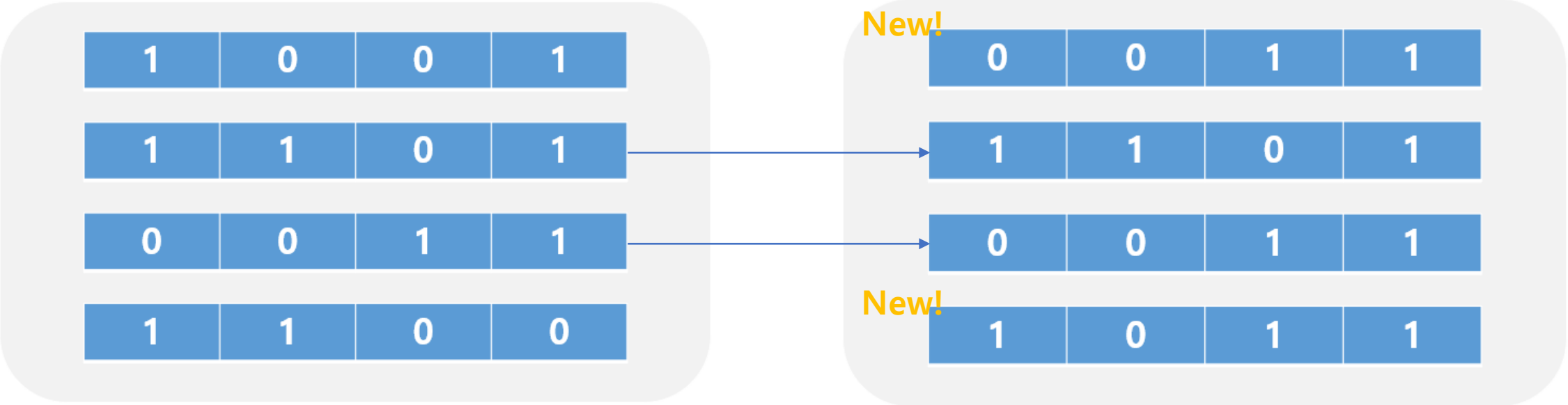
4. Crossover

- 선택된 두 개체의 일부를 교배(조합)하여, 다음 세대의 모집단에 추가



Genetic Algorithm

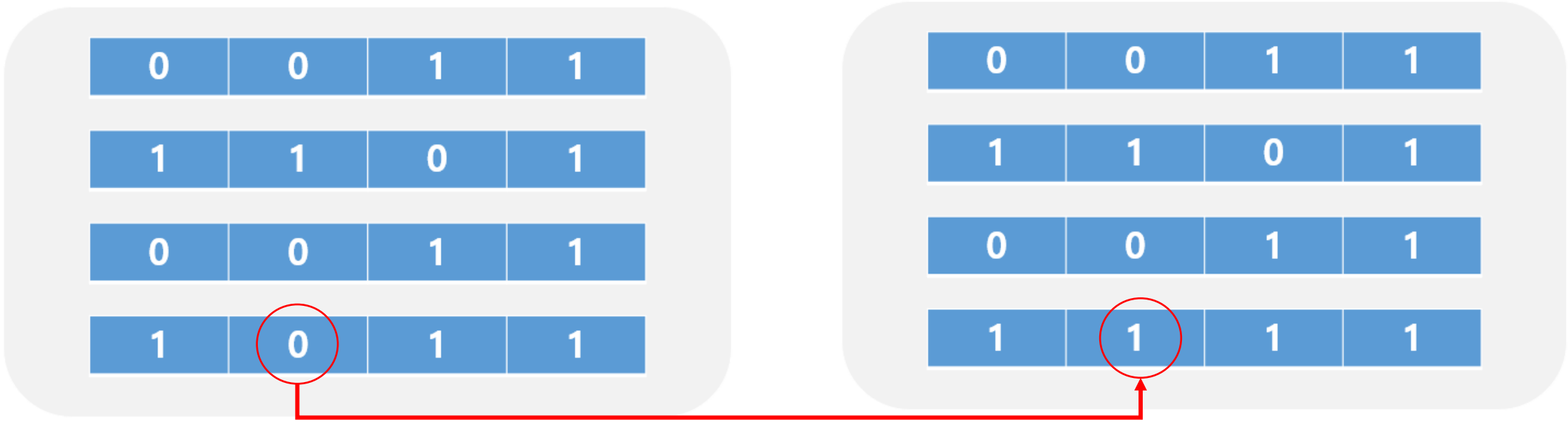
- 적합도 상위 50% 개체는 그대로 다음 세대로 전이, 하위 50% 개체는 교배를 통해 대체됨



Genetic Algorithm

5. Mutation

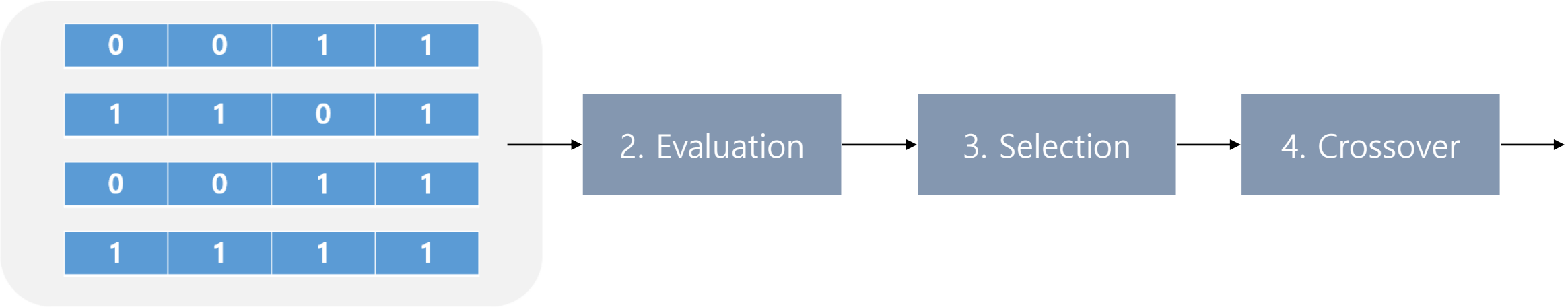
- 매우 낮은 확률로, 일정 개체에는 무작위 변이 진행

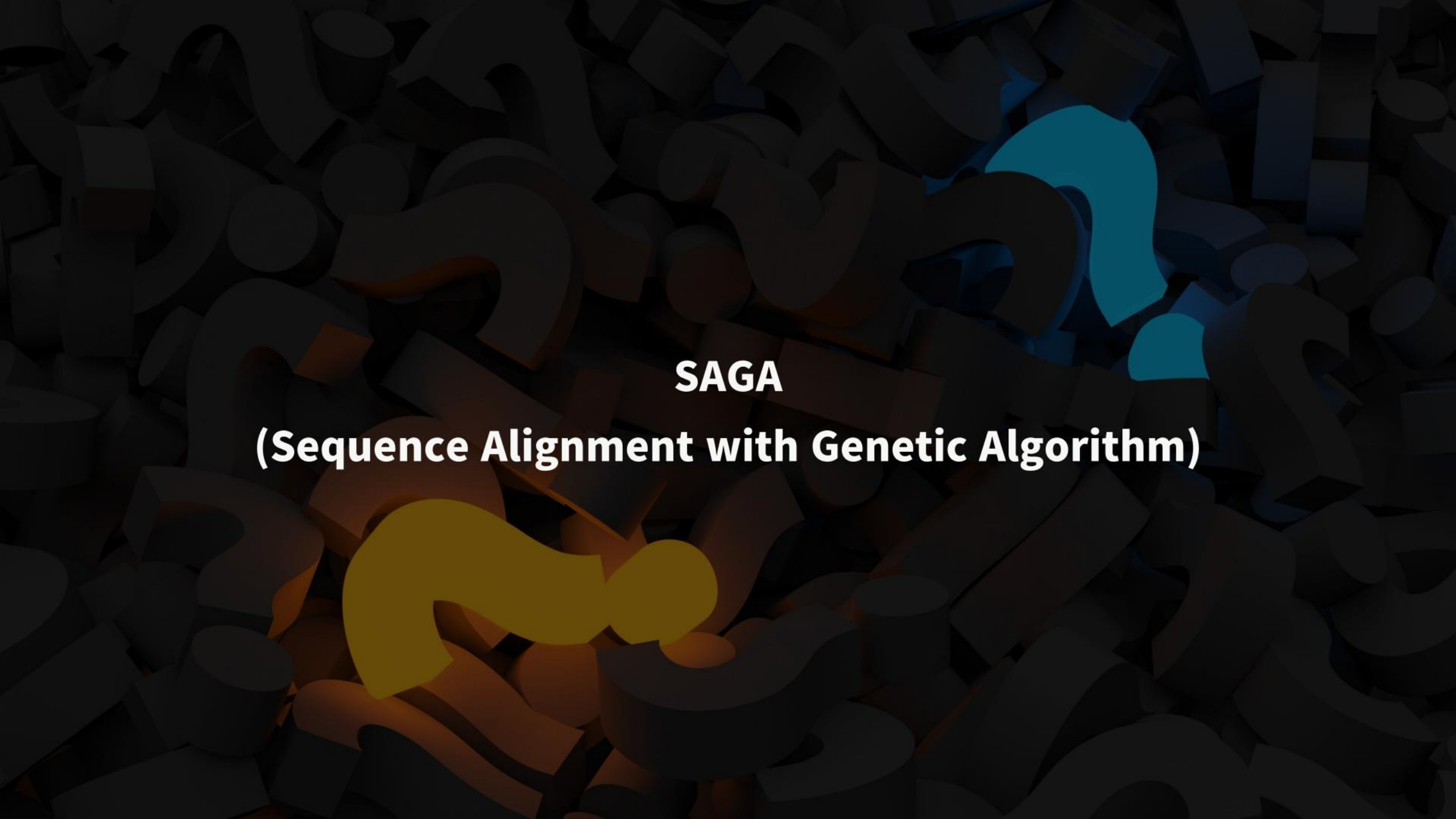


Genetic Algorithm

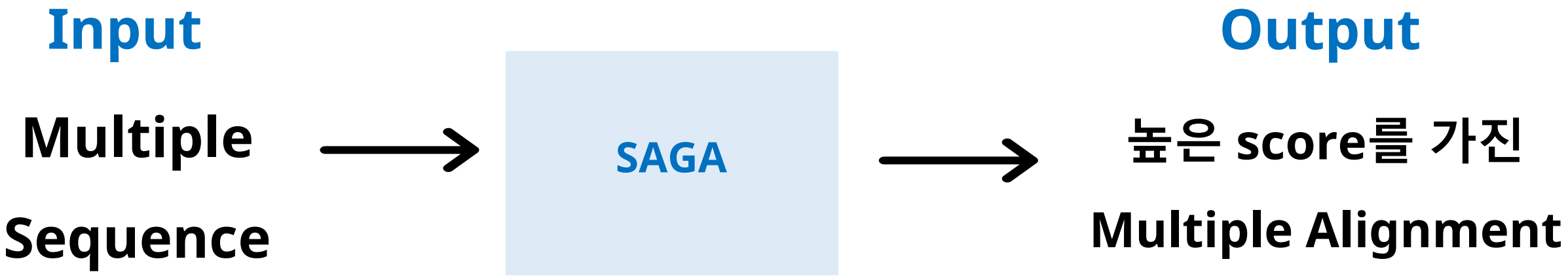
6. Replace
- 다음 세대로 전이

7. 2~6 단계 반복




SAGA
(Sequence Alignment with Genetic Algorithm)

SAGA


Input
Multiple
Sequence
SAGA
Output
높은 score를 가진
Multiple Alignment

SAGA

Input

Multiple

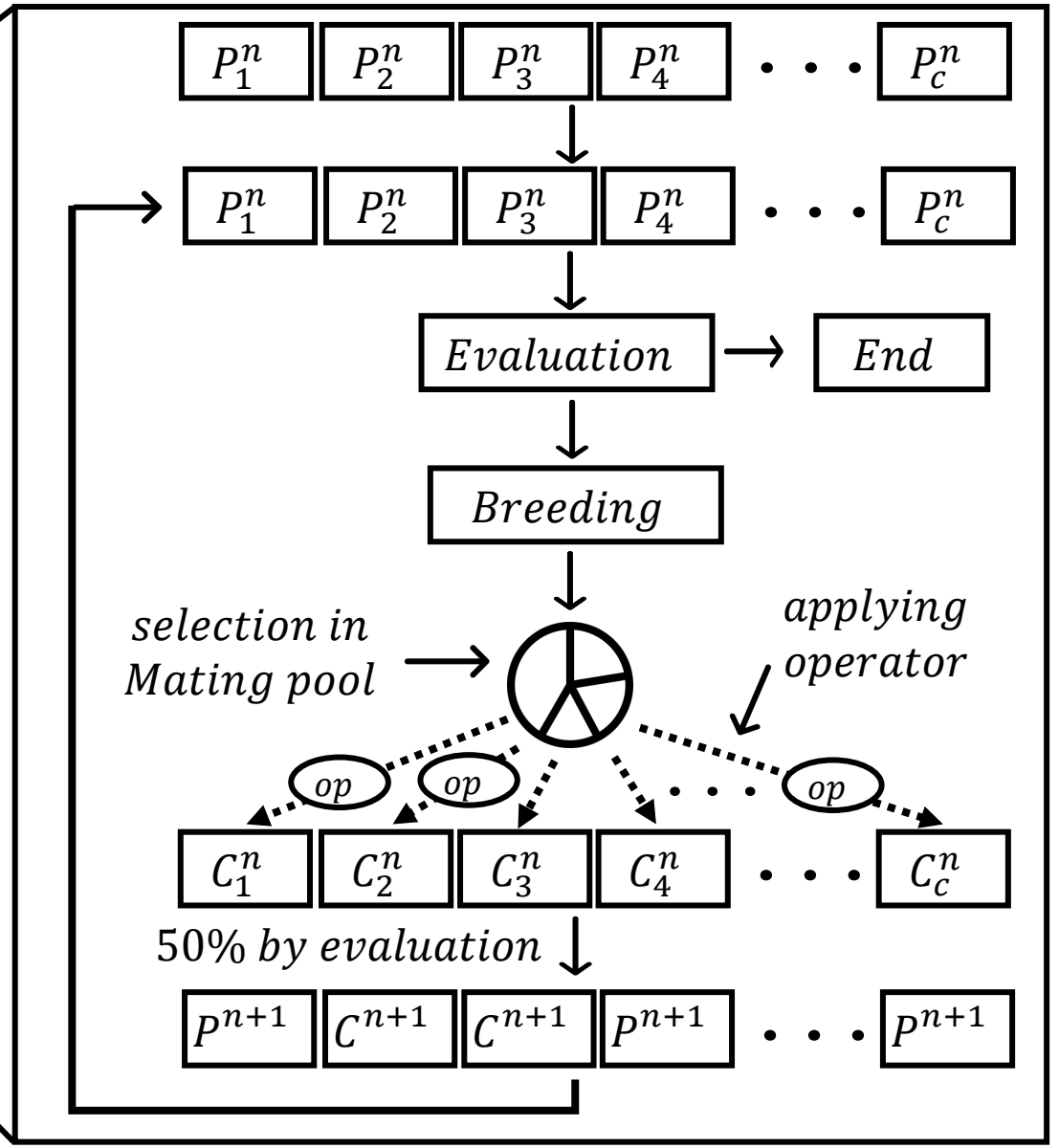
Sequence

SAGA

Output

높은 score를 가진

Multiple Alignment



SAGA

Input

Multiple

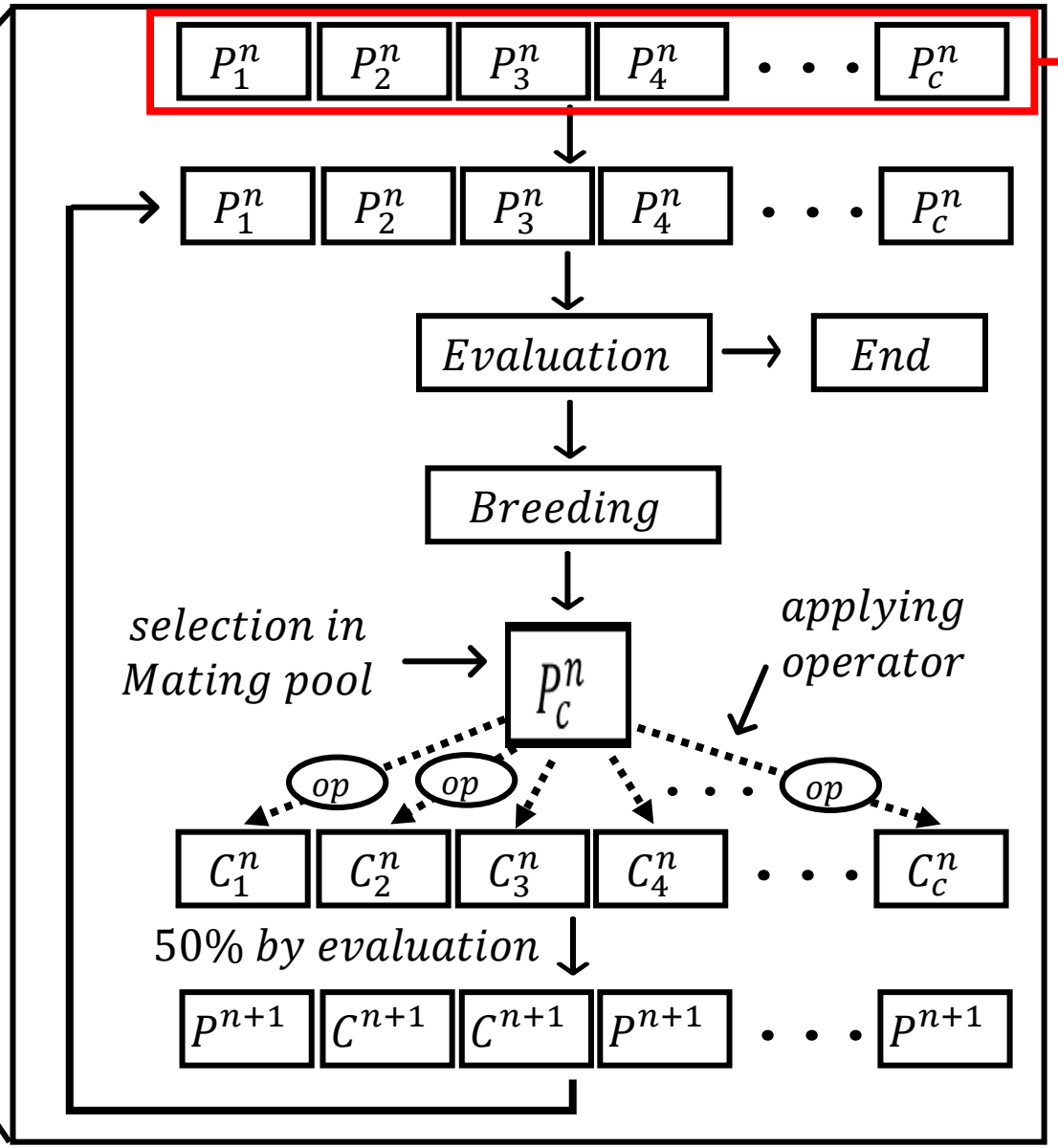
Sequence

SAGA

Output

높은 score를 가진

Multiple Alignment



SAGA

Input

Multiple

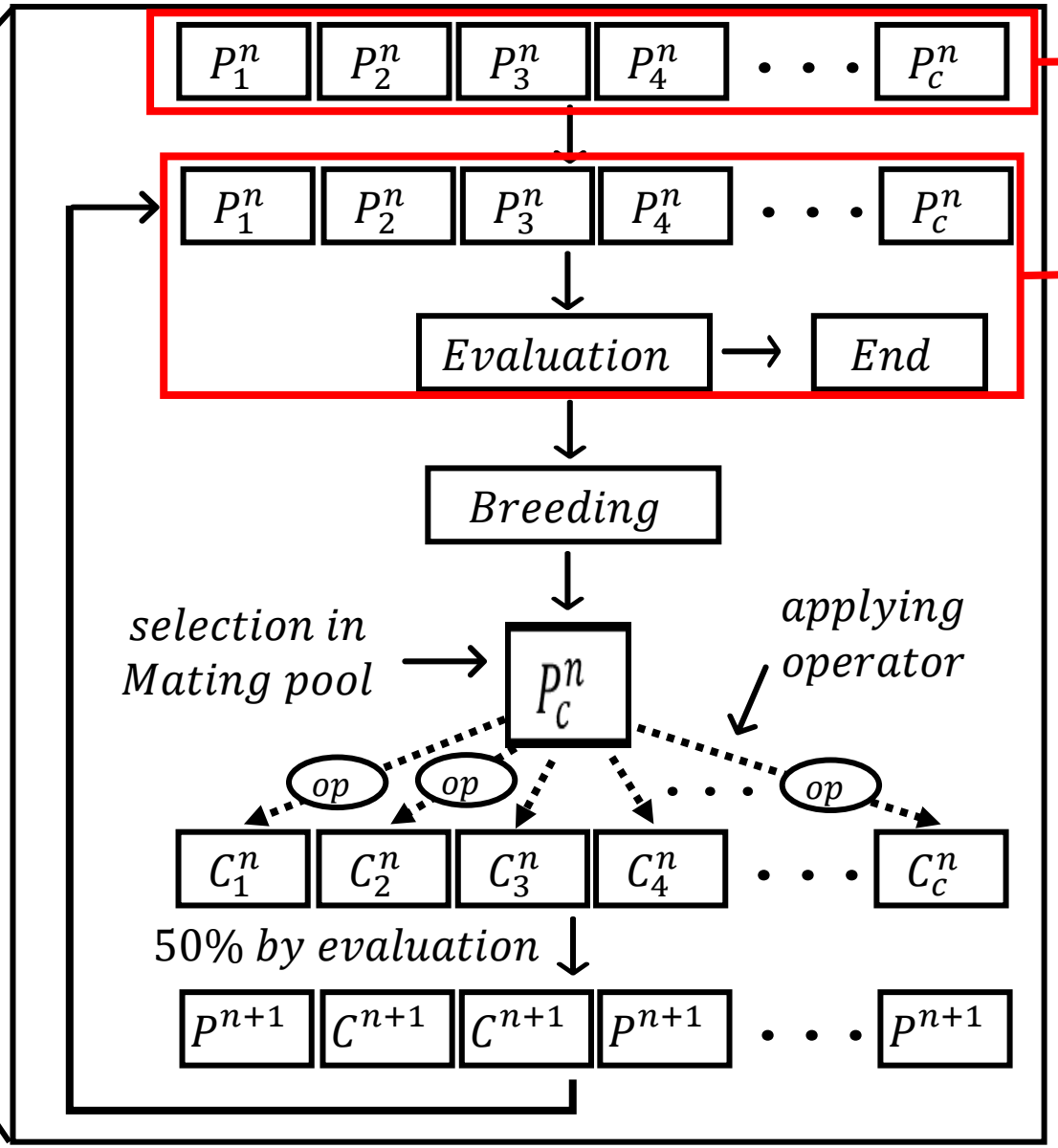
Sequence

SAGA

Output

높은 score를 가진

Multiple Alignment



SAGA

Input

Multiple

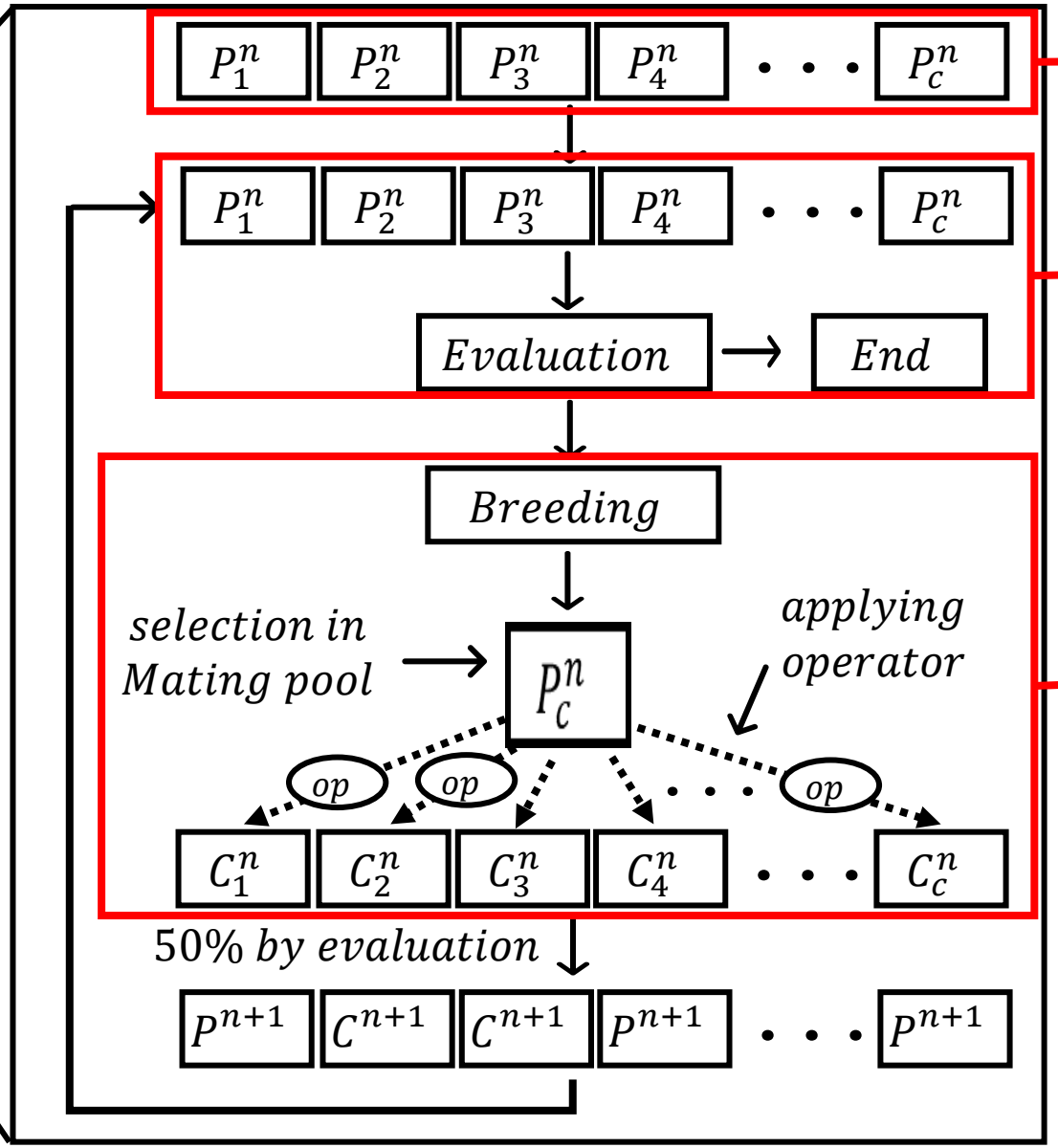
Sequence

SAGA

Output

높은 score를 가진

Multiple Alignment



1. Initialize

2. Evaluation

3. Breeding

SAGA

Input

Multiple

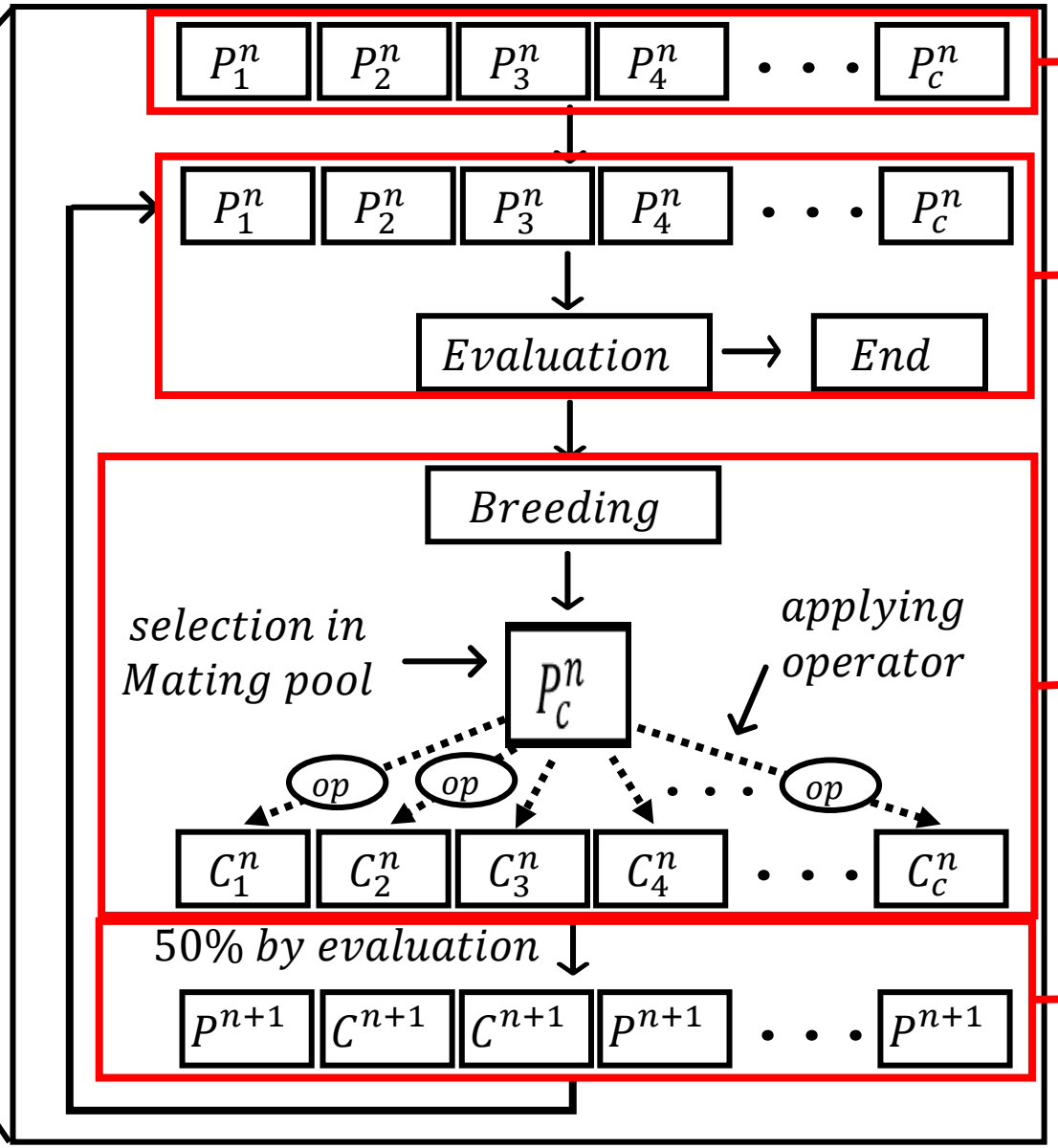
Sequence

SAGA

Output

높은 score를 가진

Multiple Alignment



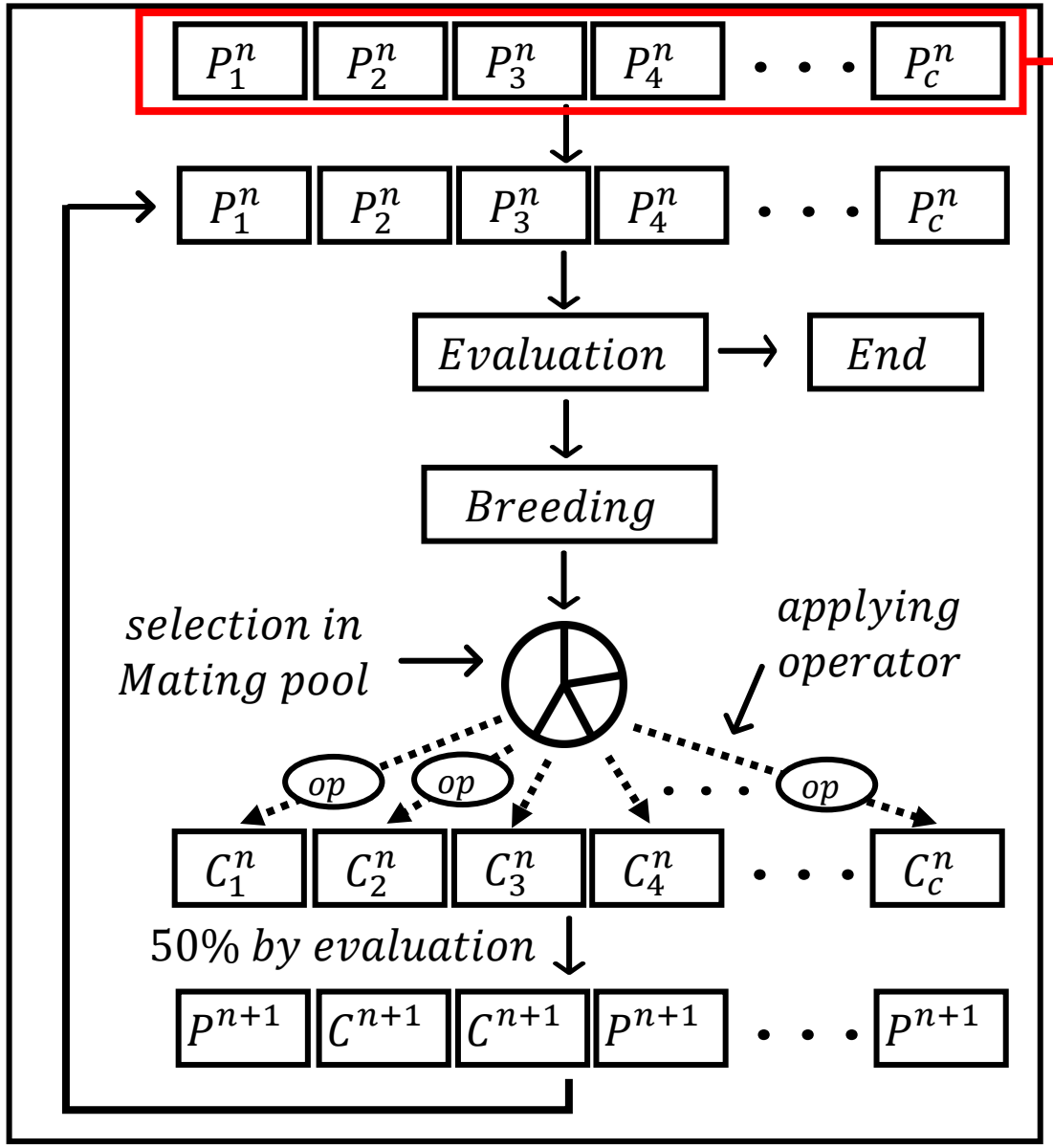
1. Initialize

2. Evaluation

3. Breeding

4. Generation

SAGA



1. Initialize

Input

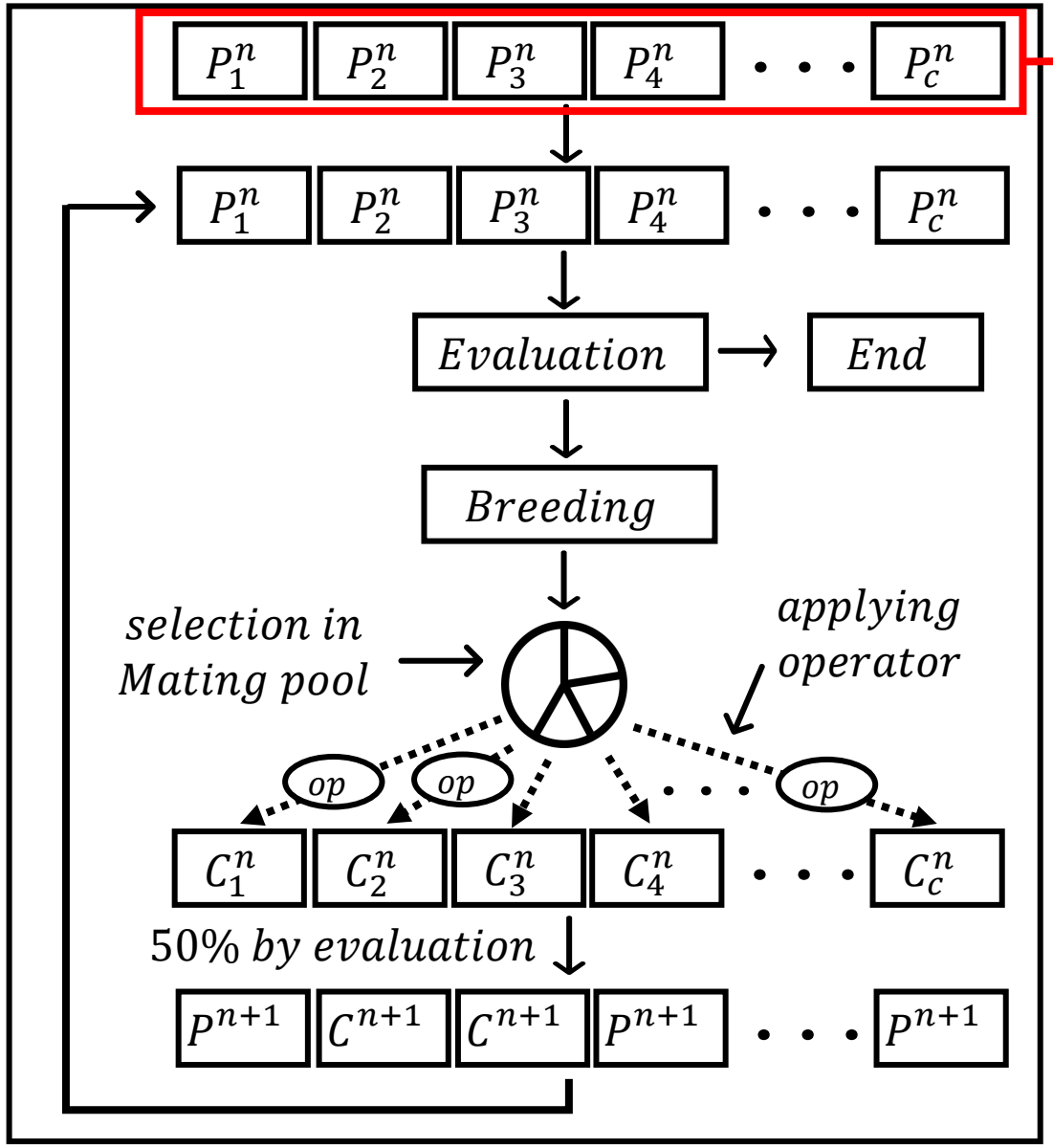
Multiple Sequence

Initialize

Output

Initial Population

SAGA



1. Initialize

Input

Multiple Sequence

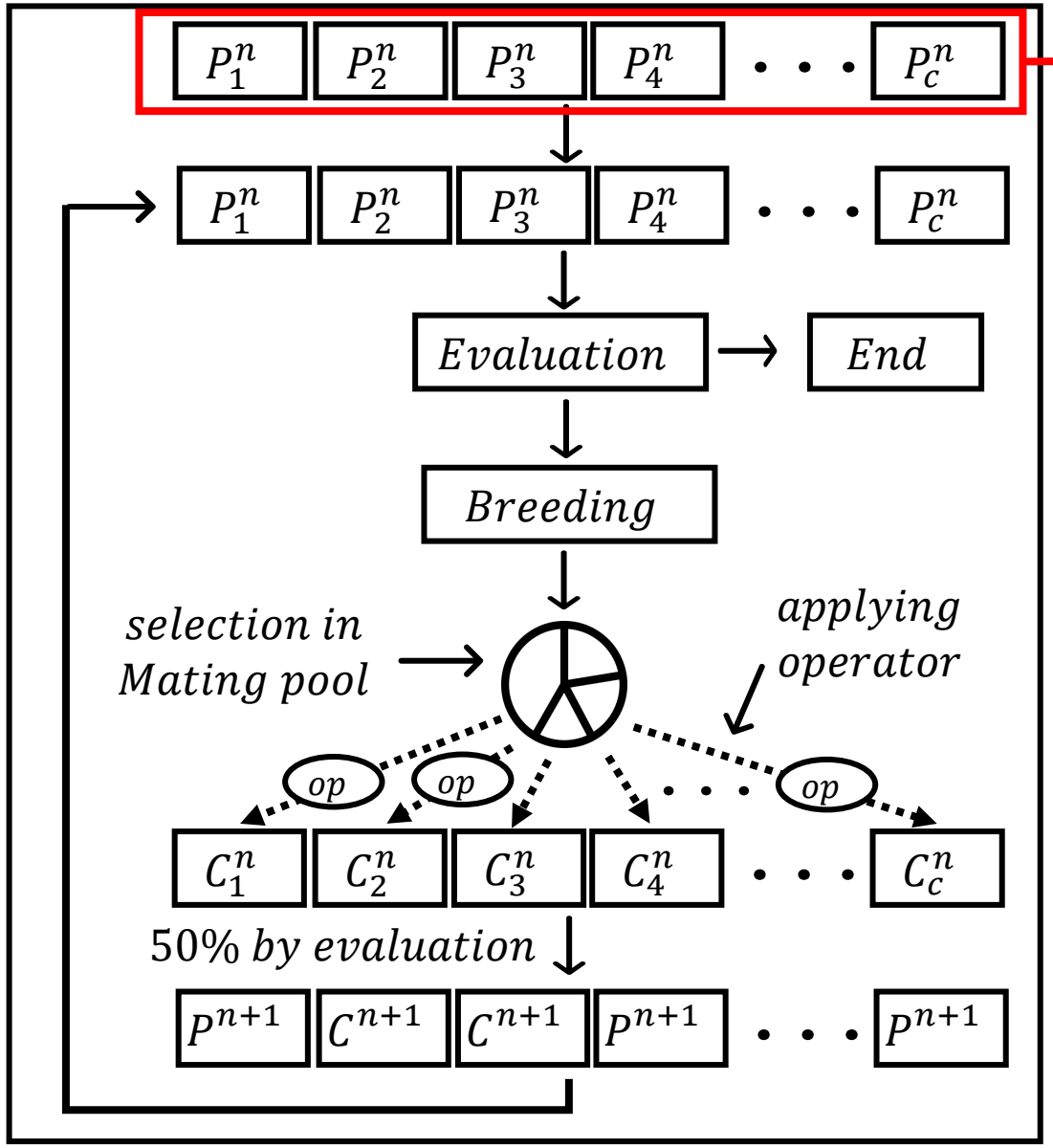
Initialize

Output

Initial Population

n ⌜ *sequence*
sequence
⋮
⌞ *sequence*

SAGA



1. Initialize

Input

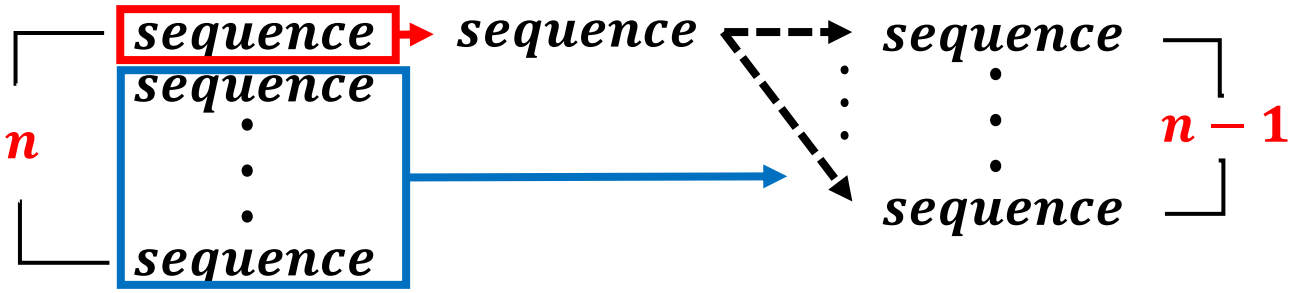
Multiple Sequence

Initialize

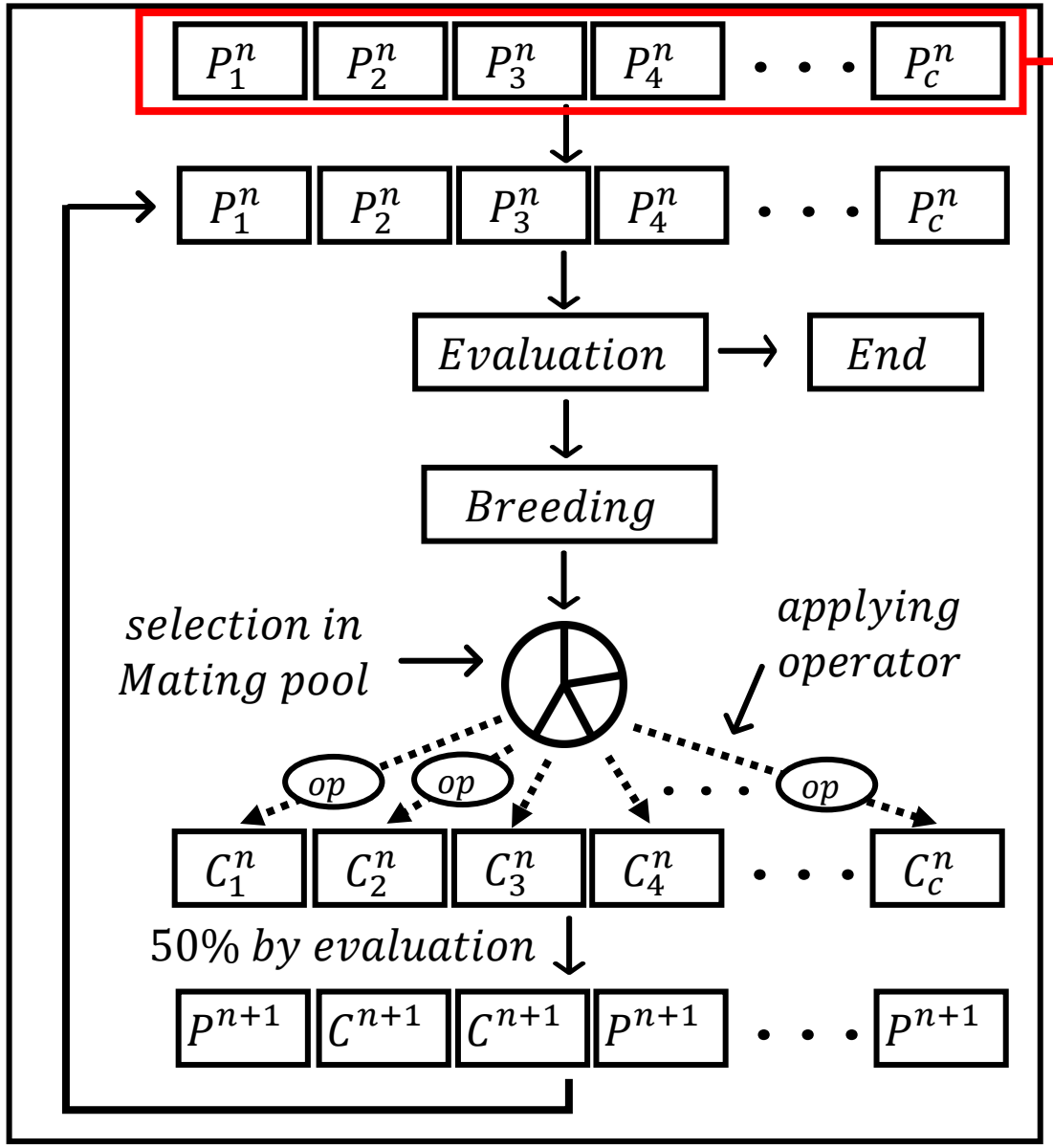
Output

Initial Population

Global alignment



SAGA



1. Initialize

Input

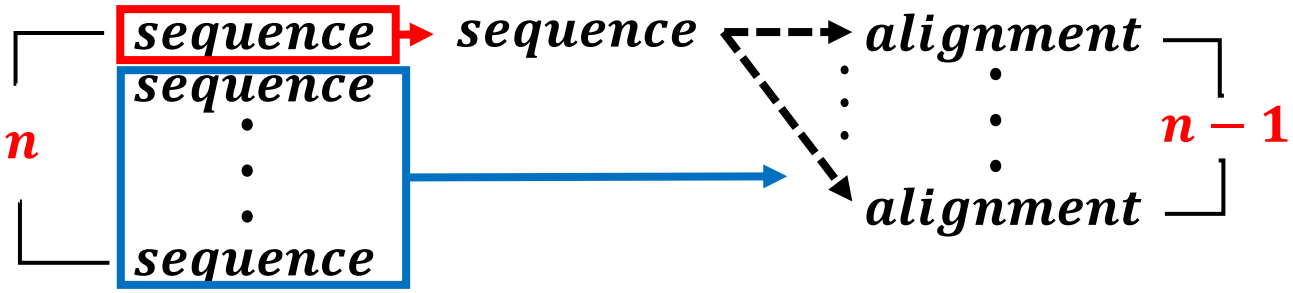
Multiple Sequence

Initialize

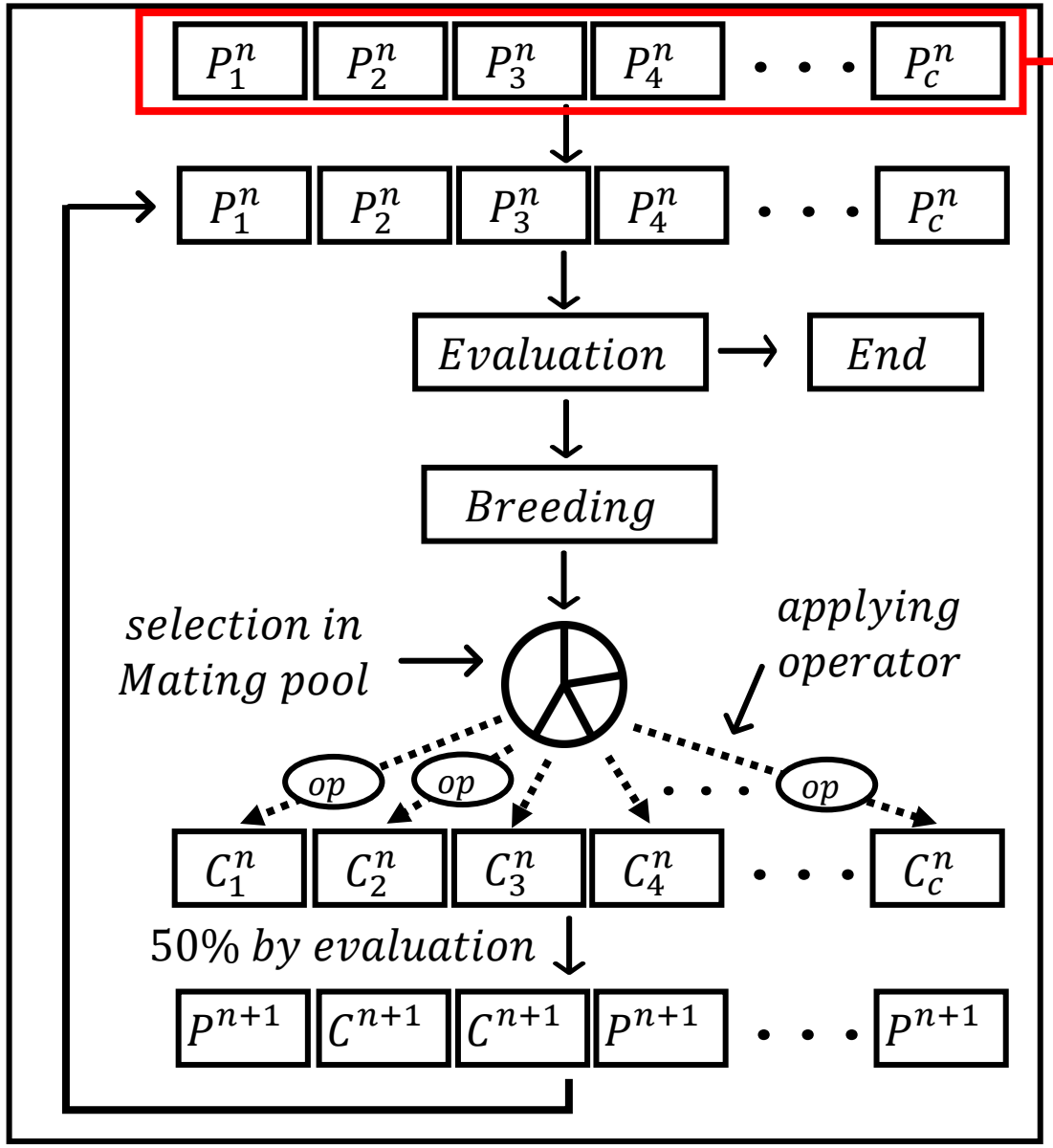
Output

Initial Population

Global alignment



SAGA



1. Initialize

Input

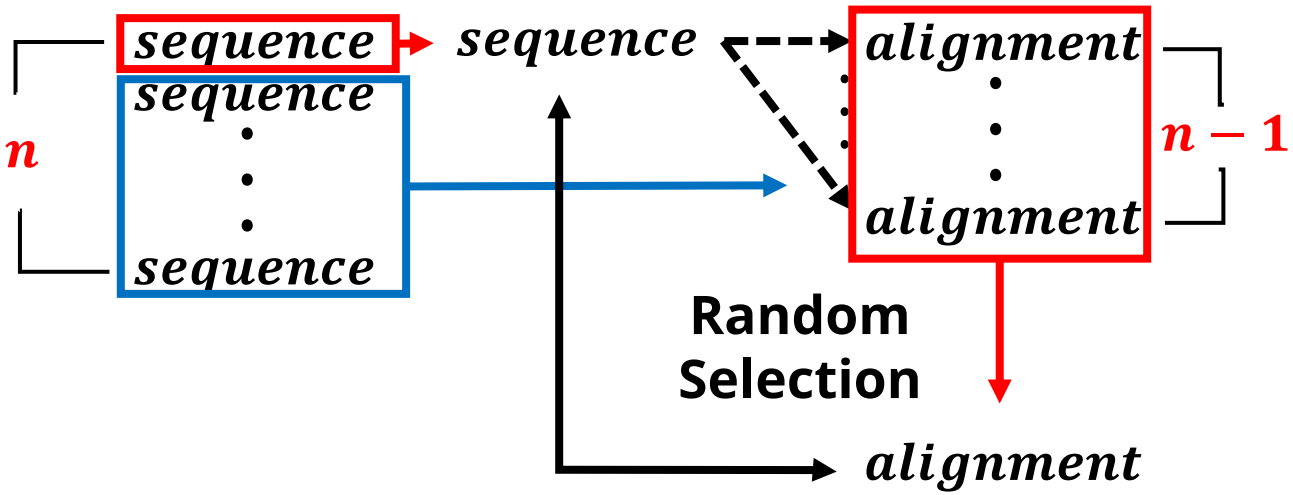
Multiple Sequence

Initialize

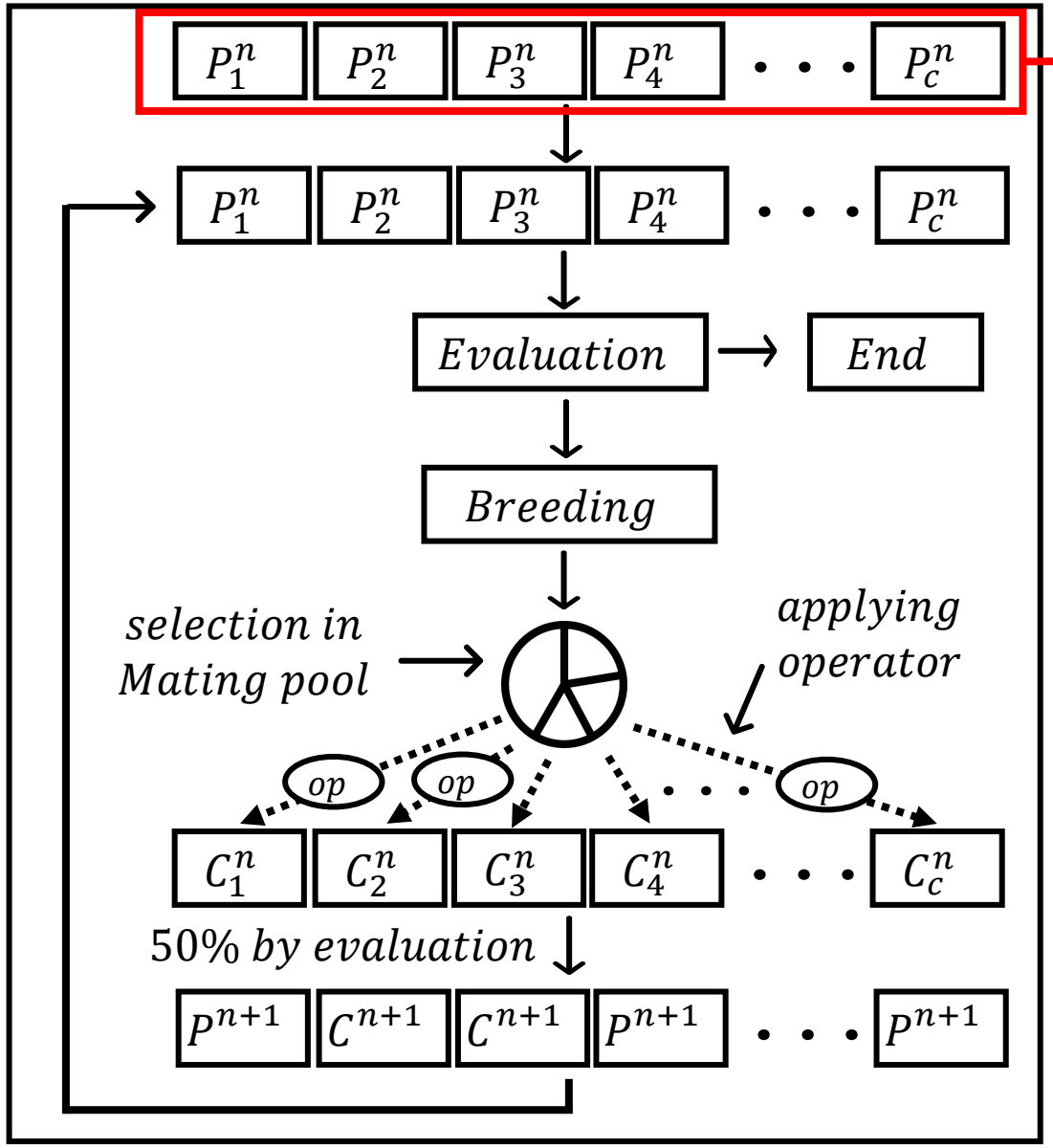
Output

Initial Population

Global alignment



SAGA



1. Initialize

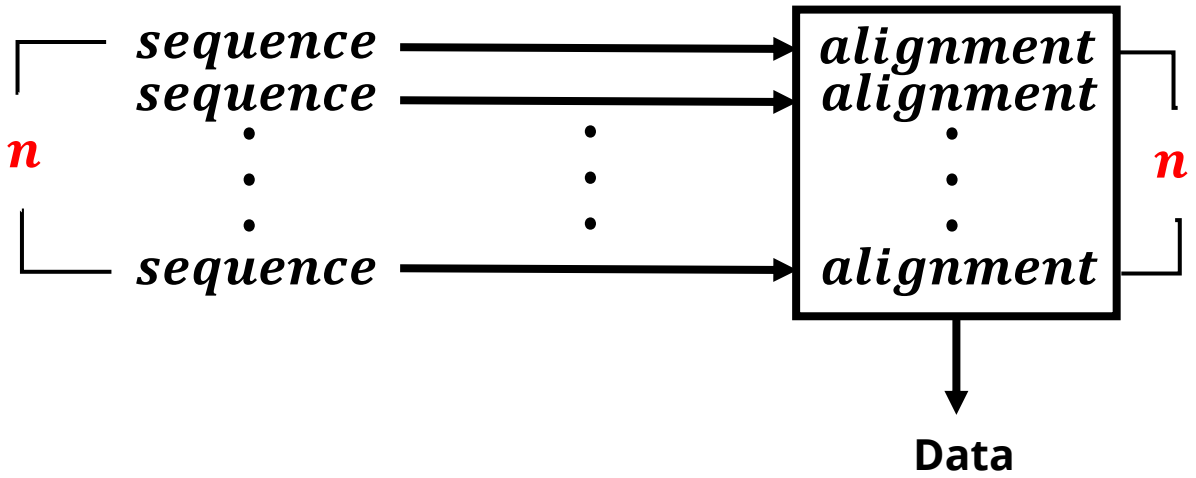
Input

Multiple Sequence

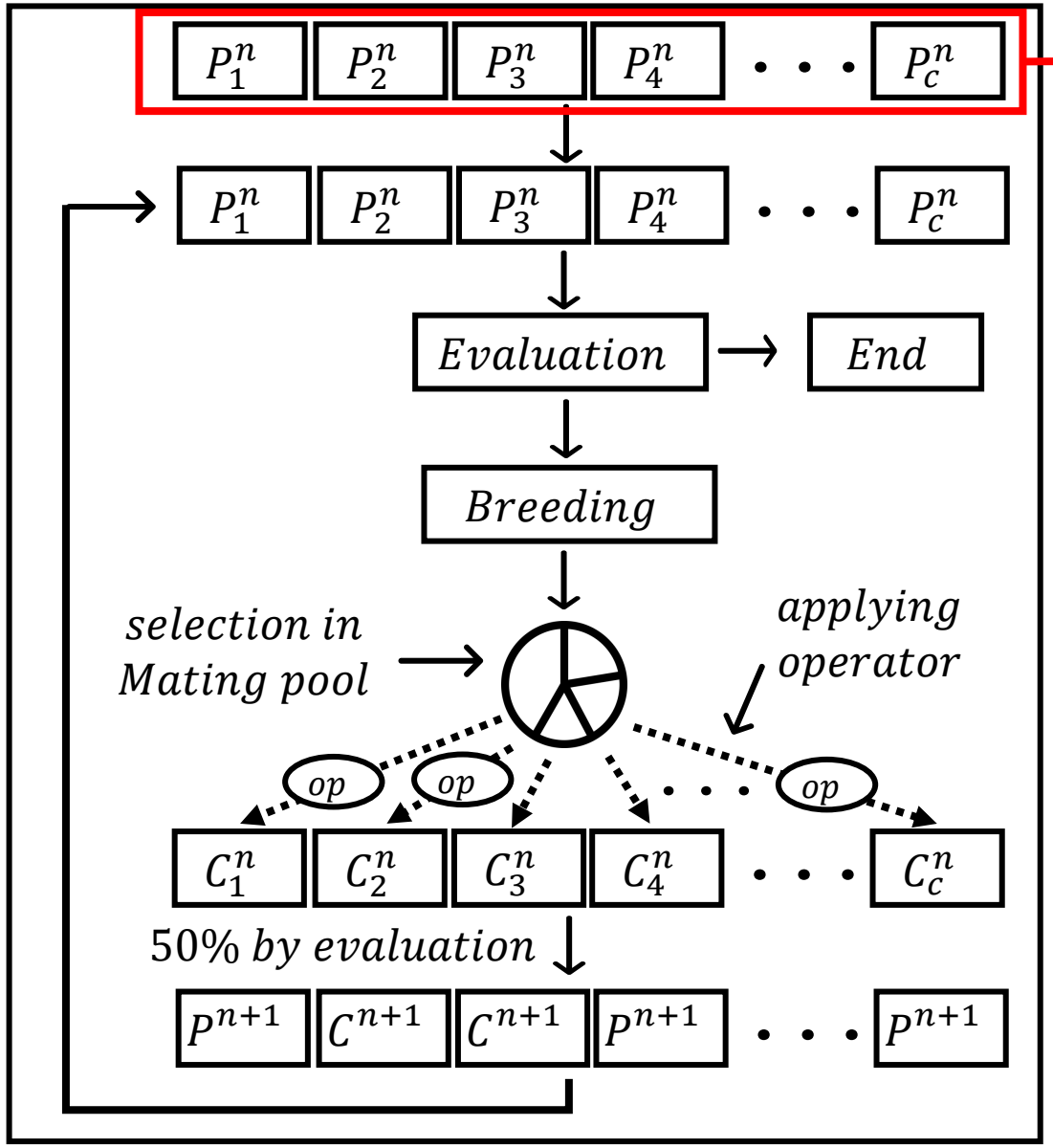
Initialize

Output

Initial Population



SAGA



1. Initialize

Input

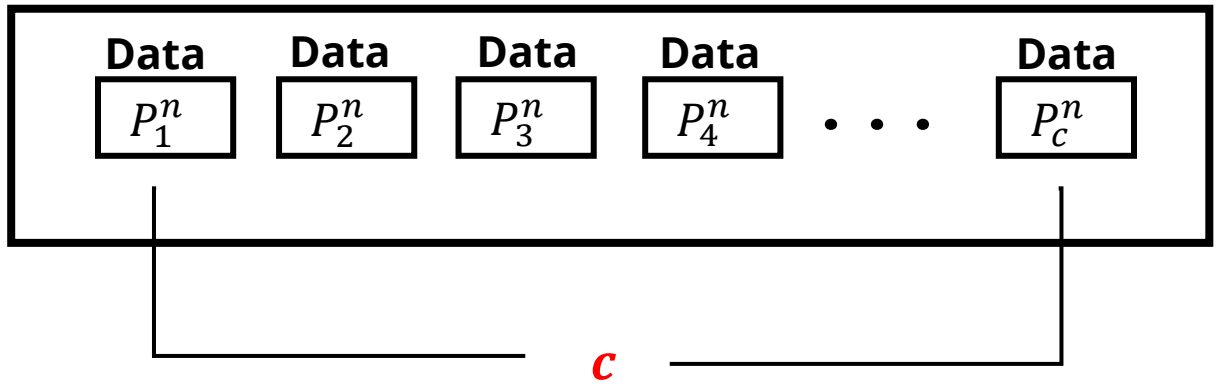
Multiple Sequence

Initialize

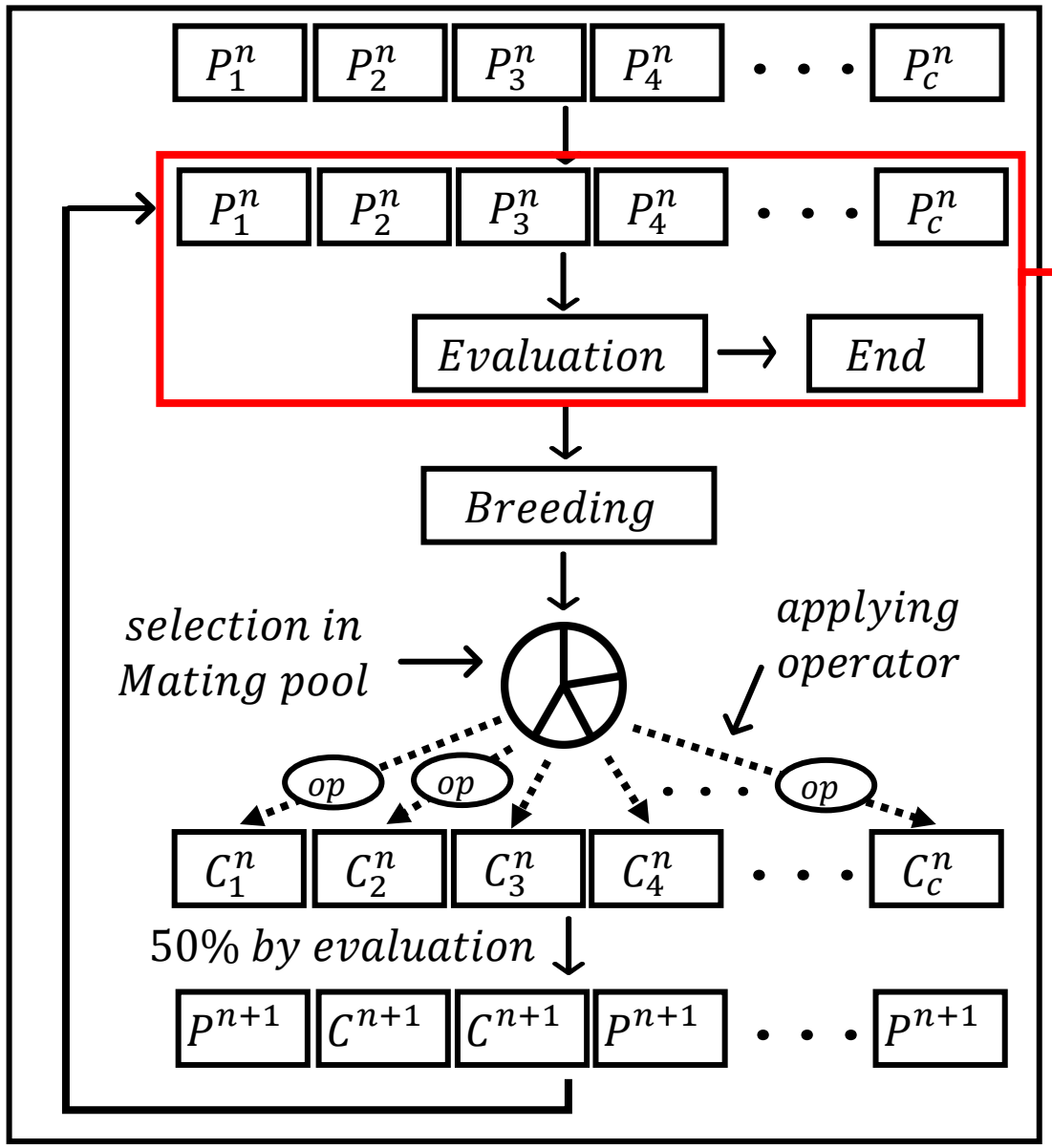
Output

Initial Population

Population



SAGA



2. Evaluation

Input

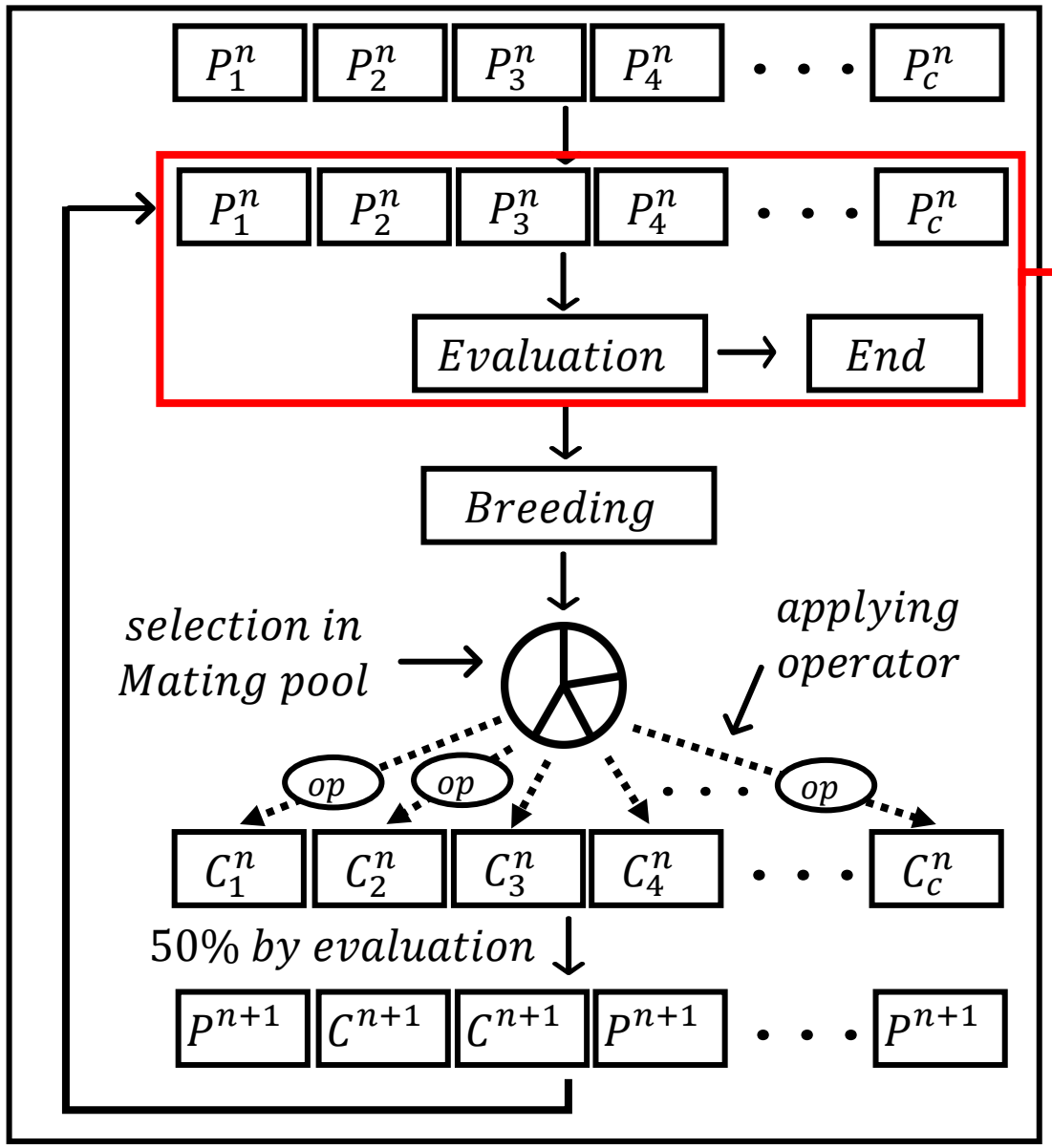
Population

Evaluation

Output

Population with Evaluation

SAGA



2. Evaluation

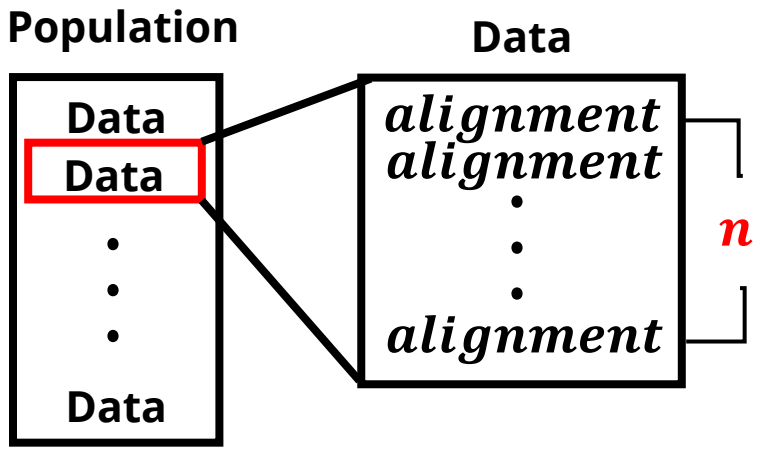
Input

Population

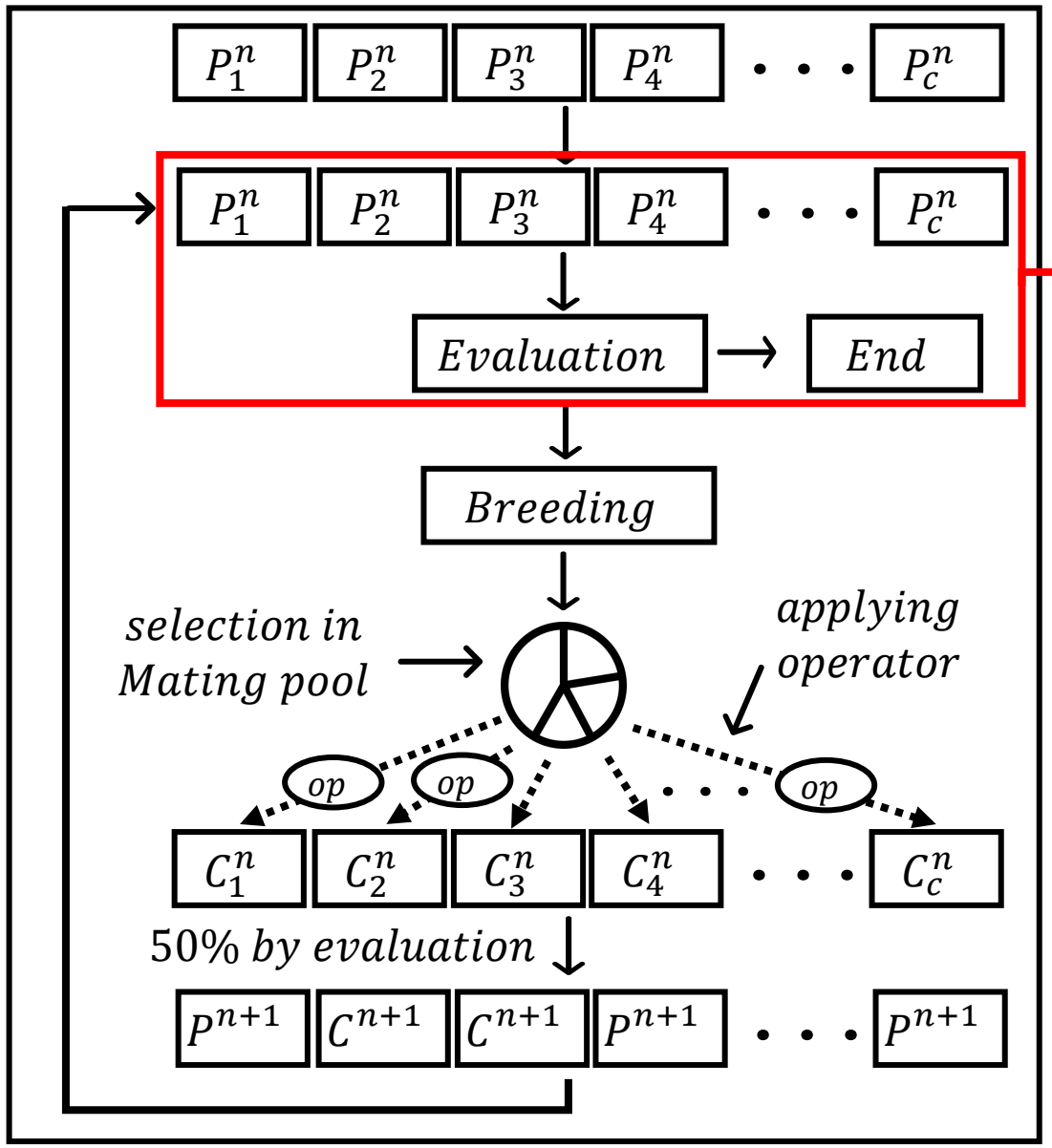
Evaluation

Output

Population with Evaluation



SAGA



2. Evaluation

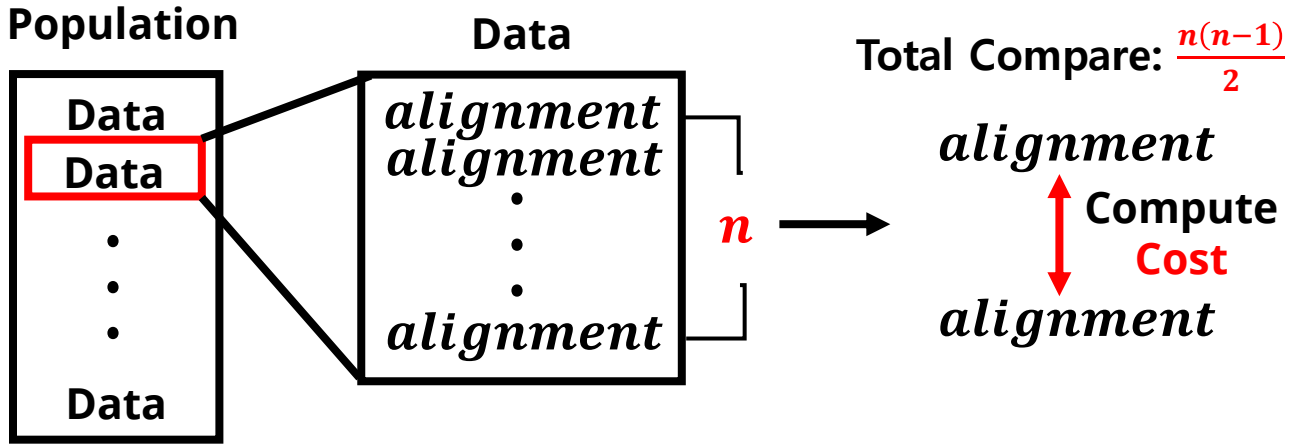
Input

Population

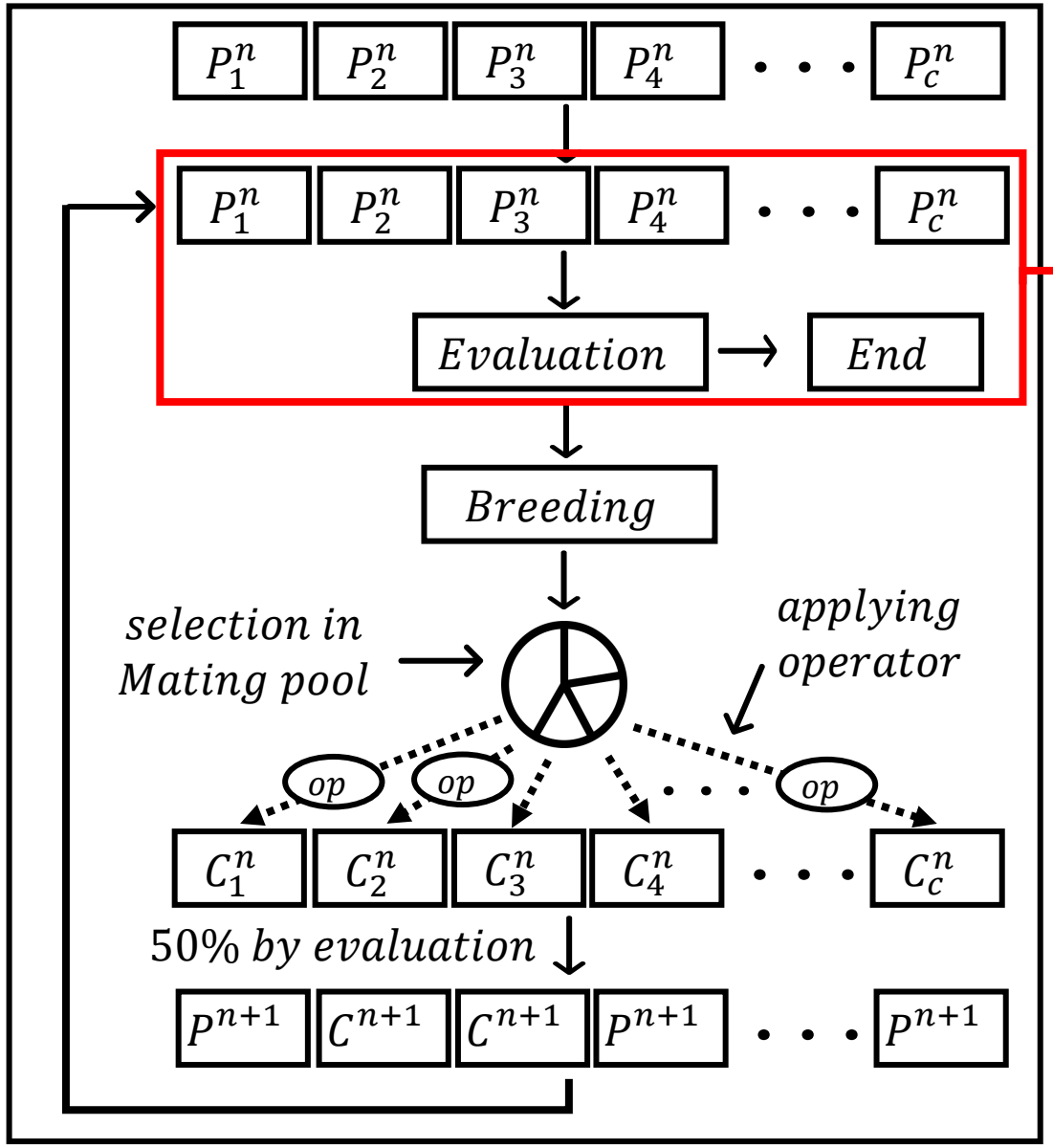
Evaluation

Output

Population with Evaluation



SAGA



2. Evaluation

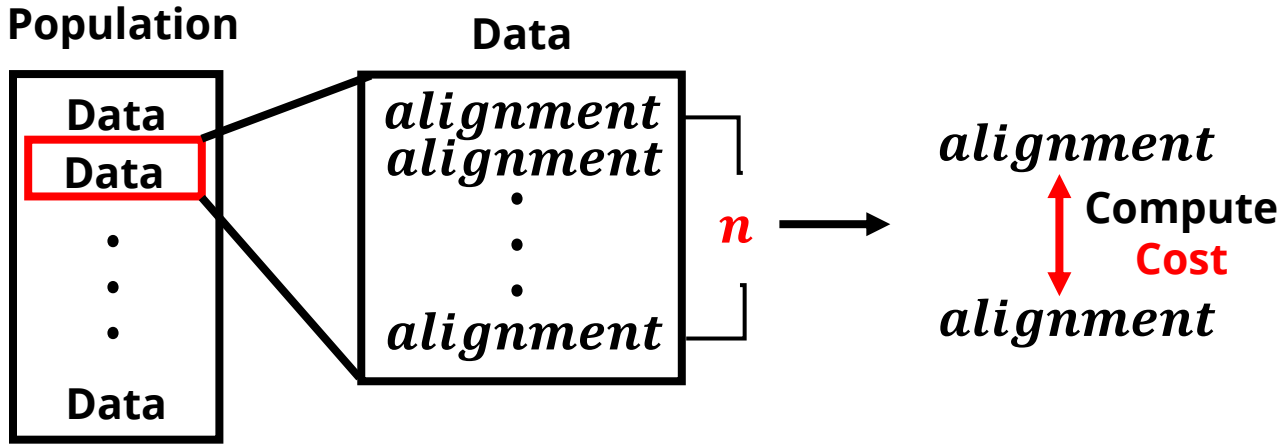
Input

Population

Evaluation

Output

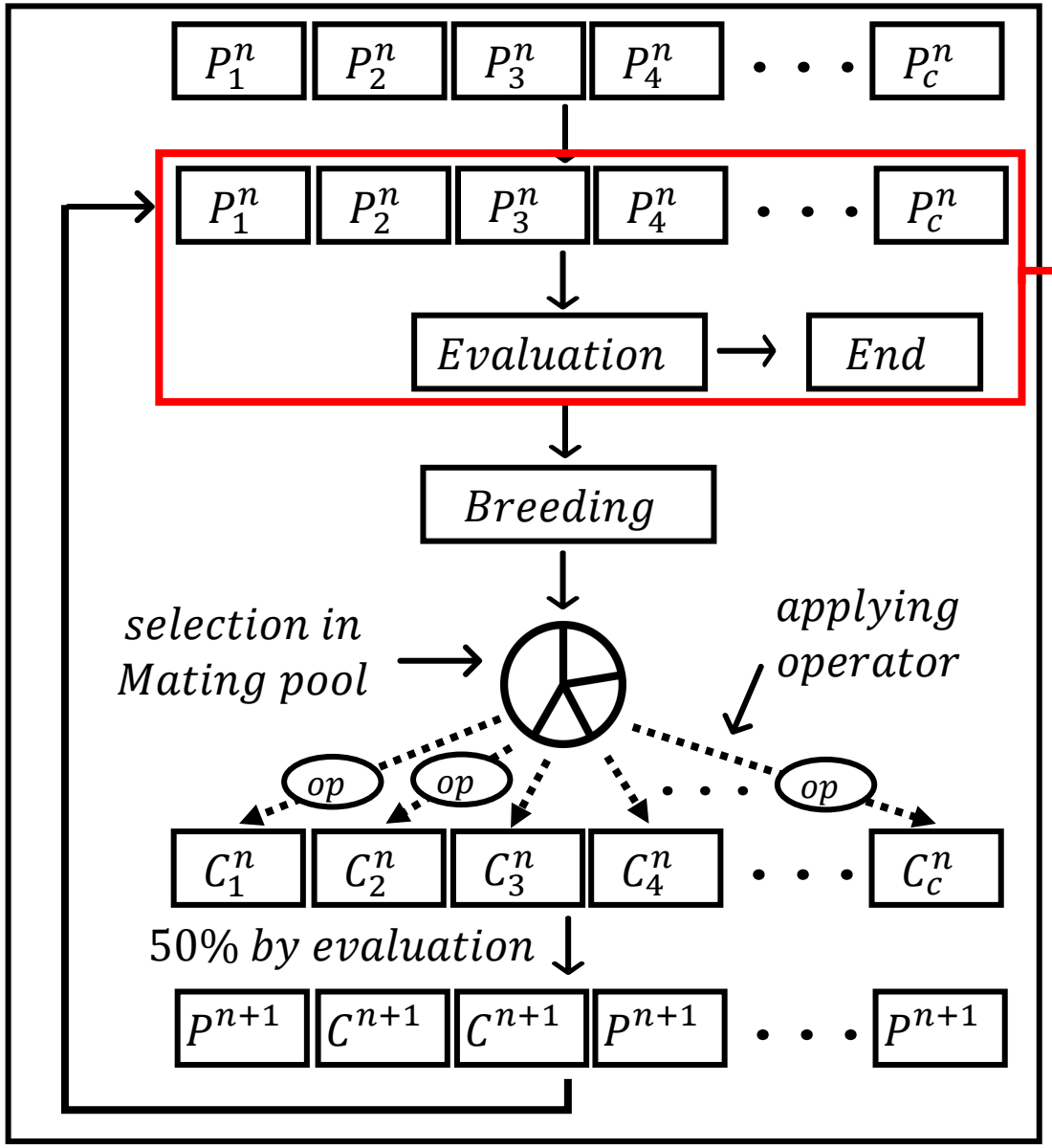
Population with Evaluation



alignment Cost

$$= \sum_{i=2}^{N} \sum_{j=1}^{i-1} \text{교체비용}(A_i, A_j) + gap \text{ 비용}(cost_{open}, cost_{extend})$$

SAGA



2. Evaluation

Input

Population

Evaluation

Output

Population with Evaluation

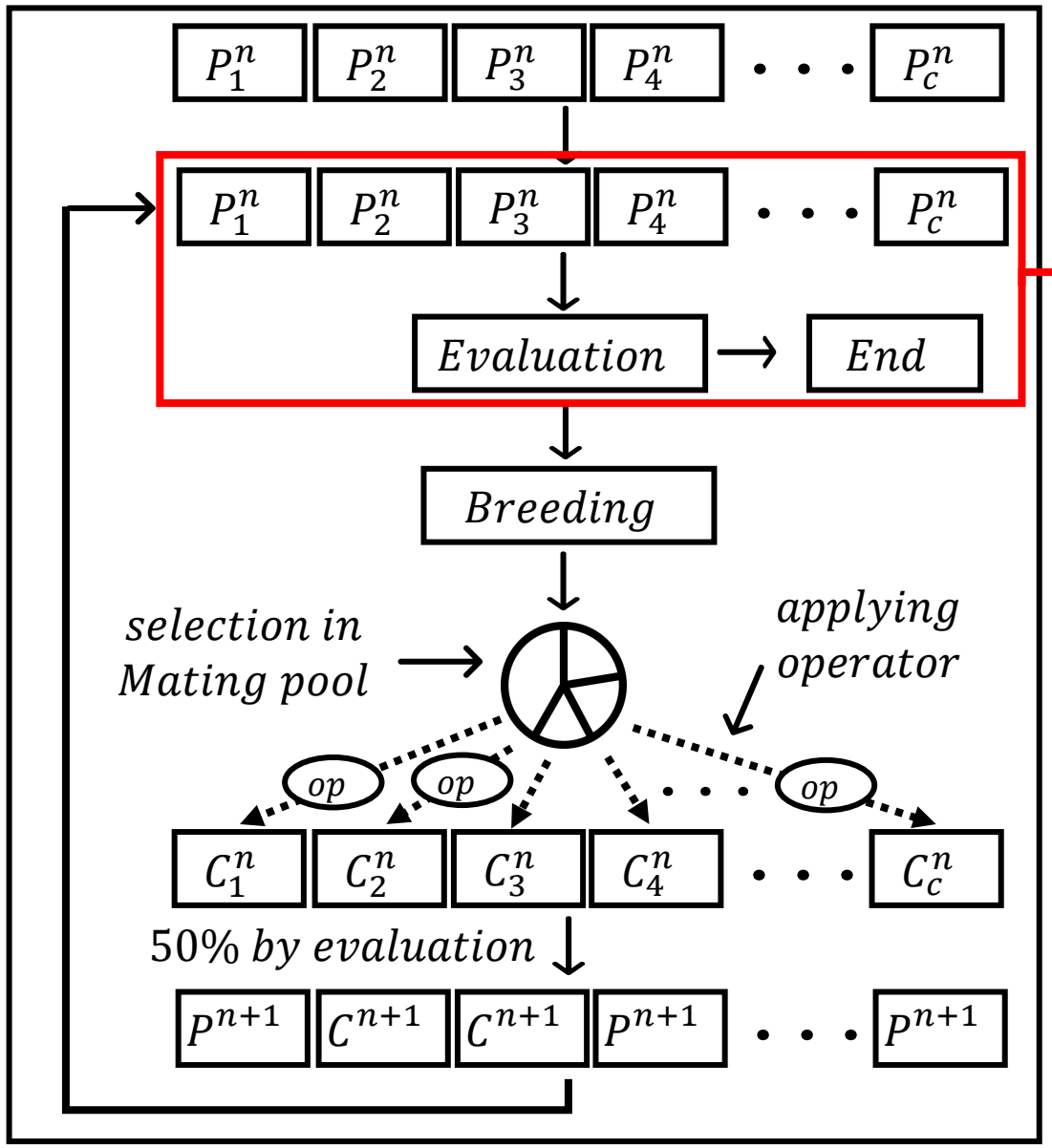
교체 비용

- alignment A가 alignment B로 바뀌기 위해 필요한 비용
- 이 비용은 **BLOSUM 62**테이블을 이용하여 계산 가능

alignment Cost

$$= \sum_{i=2}^{N} \sum_{j=1}^{i-1} \text{교체비용}(A_i, A_j) + gap \text{ 비용}(cost_{open}, cost_{extend})$$

SAGA



2. Evaluation

Input

Population

Evaluation

Output

Population with Evaluation

	C	S	T	A	G	P	D	E	Q	N	H	R	K	M	I	L	V	W	Y	F	
C	9																				C
S	-1	4																			S
T	-1	1	5																		T
A	0	1	0	4																	A
G	-3	0	-2	0	6																G
P	-3	-1	-1	-1	-2	7															P
D	-3	0	-1	-2	-1	-1	6														D
E	-4	0	-1	-1	-2	-1	2	5													E
Q	-3	0	-1	-1	-2	-1	0	2	5												Q
N	-3	1	0	-2	0	-2	1	0	0	6											N
H	-3	-1	-2	-2	-2	-2	-1	0	0	1	8										H
R	-3	-1	-1	-1	-2	-2	-2	0	1	0	0	5									R
K	-3	0	-1	-1	-2	-1	-1	1	1	0	-1	2	5								K
M	-1	-1	-1	-1	-3	-2	-3	-2	0	-2	-2	-1	-1	5							M
I	-1	-2	-1	-1	-4	-3	-3	-3	-3	-3	-3	-3	-3	1	4						I
L	-1	-2	-1	-1	-4	-3	-4	-3	-2	-3	-3	-2	-2	2	2	4					L
V	-1	-2	0	0	-3	-2	-3	-2	-2	-3	-3	-3	-2	1	3	1	4				V
W	-2	-3	-2	-3	-2	-4	-4	-3	-2	-4	-2	-3	-3	-1	-3	-2	-3	11			W
Y	-2	-2	-2	-2	-3	-3	-3	-2	-1	-2	2	-2	-2	-1	-1	-1	-1	2	7		Y
F	-2	-2	-2	-2	-3	-4	-3	-3	-3	-3	-1	-3	-3	0	0	0	-1	1	3	6	F
	C	S	T	A	G	P	D	E	Q	N	H	R	K	M	I	L	V	W	Y	F	

[BLOSUM 62]

SAGA

	C	S	T	A	G	P	D	E	Q	N	H	R	K	M	I	L	V	W	Y	F	
C	9																				C
S	-1	4																			S
T	-1	1	5																		T
A	0	1	0	4																	A
G	-3	0	-2	0	6																G
P	-3	-1	-1	-1	-2	7															P
D	-3	0	-1	-2	-1	-1	6														D
E	-4	0	-1	-1	-2	-1	2	5													E
Q	-3	0	-1	-1	-2	-1	0	2	5												Q
N	-3	1	0	-2	0	-2	1	0	0	6											N
H	-3	-1	-2	-2	-2	-2	-1	0	0	1	8										H
R	-3	-1	-1	-1	-2	-2	-2	0	1	0	0	5									R
K	-3	0	-1	-1	-2	-1	-1	1	1	0	-1	2	5								K
M	-1	-1	-1	-1	-3	-2	-3	-2	0	-2	-2	-1	-1	5							M
I	-1	-2	-1	-1	-4	-3	-3	-3	-3	-3	-3	-3	-3	1	4						I
L	-1	-2	-1	-1	-4	-3	-4	-3	-2	-3	-3	-2	-2	2	2	4					L
V	-1	-2	0	0	-3	-2	-3	-2	-2	-3	-3	-3	-2	1	3	1	4				V
W	-2	-3	-2	-3	-2	-4	-4	-3	-2	-4	-2	-3	-3	-1	-3	-2	-3	11			W
Y	-2	-2	-2	-2	-3	-3	-3	-2	-1	-2	2	-2	-2	-1	-1	-1	-1	2	7		Y
F	-2	-2	-2	-2	-3	-4	-3	-3	-3	-3	-1	-3	-3	0	0	0	-1	1	3	6	F
	C	S	T	A	G	P	D	E	Q	N	H	R	K	M	I	L	V	W	Y	F	

[BLOSUM 62]

Alignment

1: A T G C G G T T

2: A G T C A C G T

SAGA

	C	S	T	A	G	P	D	E	Q	N	H	R	K	M	I	L	V	W	Y	F	
C	9																				C
S	-1	4																			S
T	-1	1	5																		T
A	0	1	0	4																	A
G	-3	0	-2	0	6																G
P	-3	-1	-1	-1	-2	7															P
D	-3	0	-1	-2	-1	-1	6														D
E	-4	0	-1	-1	-2	-1	2	5													E
Q	-3	0	-1	-1	-2	-1	0	2	5												Q
N	-3	1	0	-2	0	-2	1	0	0	6											N
H	-3	-1	-2	-2	-2	-2	-1	0	0	1	8										H
R	-3	-1	-1	-1	-2	-2	-2	0	1	0	0	5									R
K	-3	0	-1	-1	-2	-1	-1	1	1	0	-1	2	5								K
M	-1	-1	-1	-1	-3	-2	-3	-2	0	-2	-2	-1	-1	5							M
I	-1	-2	-1	-1	-4	-3	-3	-3	-3	-3	-3	-3	-3	1	4						I
L	-1	-2	-1	-1	-4	-3	-4	-3	-2	-3	-3	-2	-2	2	2	4					L
V	-1	-2	0	0	-3	-2	-3	-2	-2	-3	-3	-3	-2	1	3	1	4				V
W	-2	-3	-2	-3	-2	-4	-4	-3	-2	-4	-2	-3	-3	-1	-3	-2	-3	11			W
Y	-2	-2	-2	-2	-3	-3	-3	-2	-1	-2	2	-2	-2	-1	-1	-1	-1	2	7		Y
F	-2	-2	-2	-2	-3	-4	-3	-3	-3	-3	-1	-3	-3	0	0	0	-1	1	3	6	F
	C	S	T	A	G	P	D	E	Q	N	H	R	K	M	I	L	V	W	Y	F	

[BLOSUM 62]

Alignment

1: A T G C G G T T

2: A G T C A C G T

4

SAGA

	C	S	T	A	G	P	D	E	Q	N	H	R	K	M	I	L	V	W	Y	F	
C	9																				C
S	-1	4																			S
T	-1	1	5																		T
A	0	1	0	4																	A
G	-3	0	-2	0	6																G
P	-3	-1	-1	-1	-2	7															P
D	-3	0	-1	-2	-1	-1	6														D
E	-4	0	-1	-1	-2	-1	2	5													E
Q	-3	0	-1	-1	-2	-1	0	2	5												Q
N	-3	1	0	-2	0	-2	1	0	0	6											N
H	-3	-1	-2	-2	-2	-2	-1	0	0	1	8										H
R	-3	-1	-1	-1	-2	-2	-2	0	1	0	0	5									R
K	-3	0	-1	-1	-2	-1	-1	1	1	0	-1	2	5								K
M	-1	-1	-1	-1	-3	-2	-3	-2	0	-2	-2	-1	-1	5							M
I	-1	-2	-1	-1	-4	-3	-3	-3	-3	-3	-3	-3	-3	1	4						I
L	-1	-2	-1	-1	-4	-3	-4	-3	-2	-3	-3	-2	-2	2	2	4					L
V	-1	-2	0	0	-3	-2	-3	-2	-2	-3	-3	-3	-2	1	3	1	4				V
W	-2	-3	-2	-3	-2	-4	-4	-3	-2	-4	-2	-3	-3	-1	-3	-2	-3	11			W
Y	-2	-2	-2	-2	-3	-3	-3	-2	-1	-2	2	-2	-2	-1	-1	-1	-1	2	7		Y
F	-2	-2	-2	-2	-3	-4	-3	-3	-3	-3	-1	-3	-3	0	0	0	-1	1	3	6	F
	C	S	T	A	G	P	D	E	Q	N	H	R	K	M	I	L	V	W	Y	F	

[BLOSUM 62]

Alignment

1: A T G C G G T T

2: A G T C A C G T

4 -2

SAGA

	C	S	T	A	G	P	D	E	Q	N	H	R	K	M	I	L	V	W	Y	F	
C	9																				C
S	-1	4																			S
T	-1	1	5																		T
A	0	1	0	4																	A
G	-3	0	-2	0	6																G
P	-3	-1	-1	-1	-2	7															P
D	-3	0	-1	-2	-1	-1	6														D
E	-4	0	-1	-1	-2	-1	2	5													E
Q	-3	0	-1	-1	-2	-1	0	2	5												Q
N	-3	1	0	-2	0	-2	1	0	0	6											N
H	-3	-1	-2	-2	-2	-2	-1	0	0	1	8										H
R	-3	-1	-1	-1	-2	-2	-2	0	1	0	0	5									R
K	-3	0	-1	-1	-2	-1	-1	1	1	0	-1	2	5								K
M	-1	-1	-1	-1	-3	-2	-3	-2	0	-2	-2	-1	-1	5							M
I	-1	-2	-1	-1	-4	-3	-3	-3	-3	-3	-3	-3	-3	1	4						I
L	-1	-2	-1	-1	-4	-3	-4	-3	-2	-3	-3	-2	-2	2	2	4					L
V	-1	-2	0	0	-3	-2	-3	-2	-2	-3	-3	-3	-2	1	3	1	4				V
W	-2	-3	-2	-3	-2	-4	-4	-3	-2	-4	-2	-3	-3	-1	-3	-2	-3	11			W
Y	-2	-2	-2	-2	-3	-3	-3	-2	-1	-2	2	-2	-2	-1	-1	-1	-1	2	7		Y
F	-2	-2	-2	-2	-3	-4	-3	-3	-3	-3	-1	-3	-3	0	0	0	-1	1	3	6	F
	C	S	T	A	G	P	D	E	Q	N	H	R	K	M	I	L	V	W	Y	F	

[BLOSUM 62]

Alignment

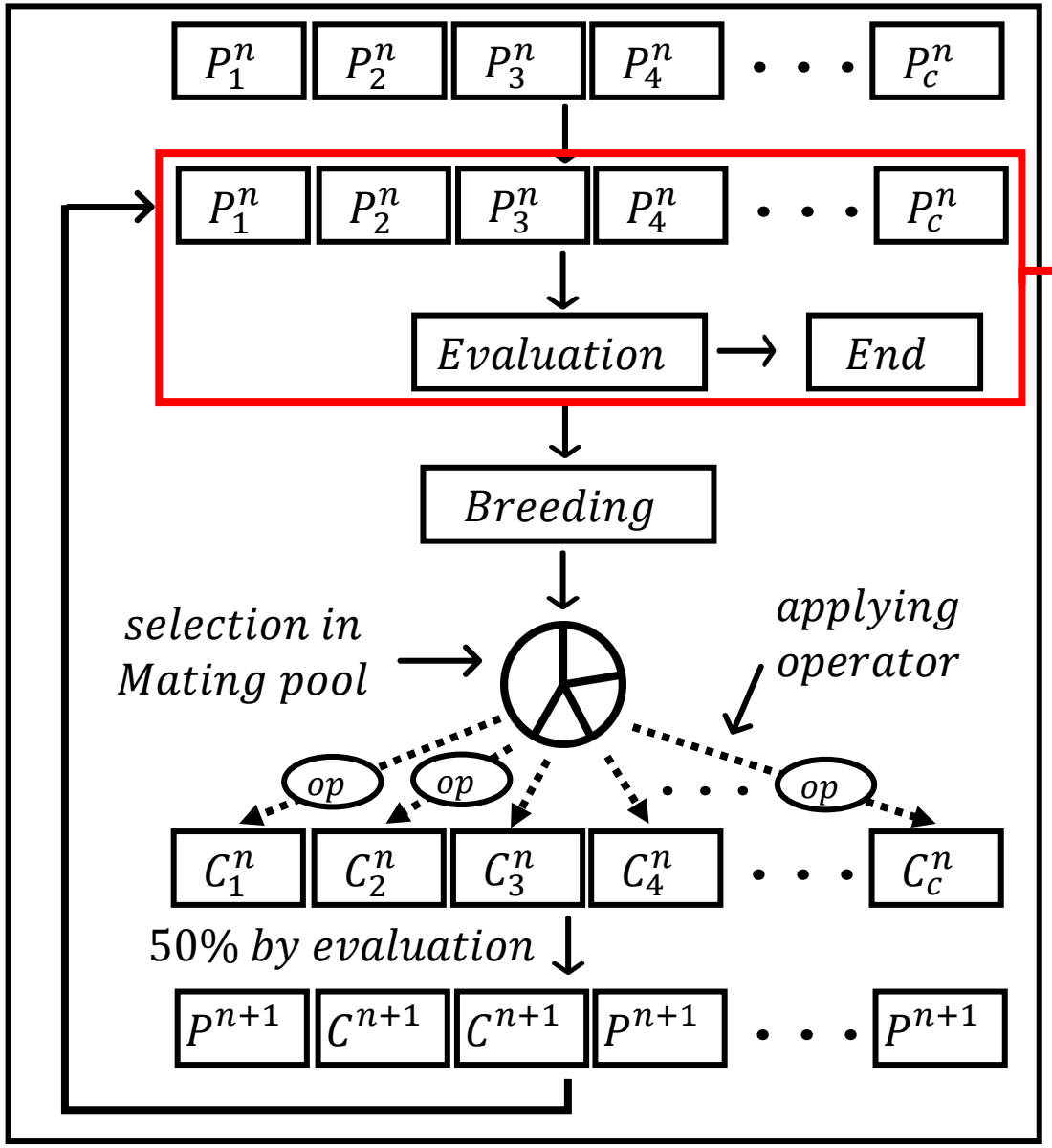
1: A T G C G G T T

2: A G T C A C G T

4-2-2+9+0-3-2+5

= 9

SAGA



2. Evaluation

Input

Population

Evaluation

Output

Population with Evaluation

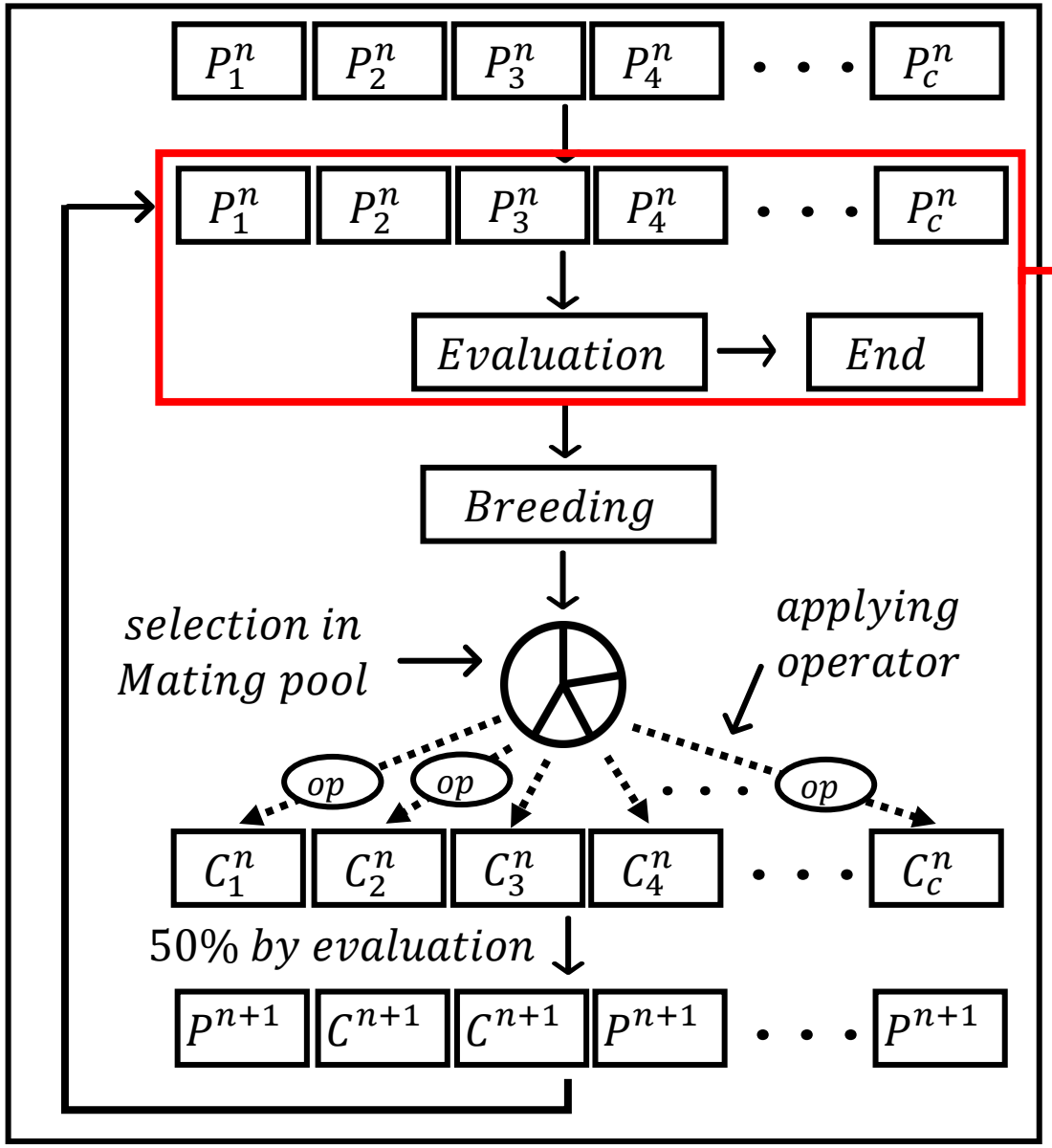
Gap 비용

- Gap 비용은 alignment의 기호 중 '—'에 부과되는 비용이다.
- 여기서 사용된 gap 비용은 **natural affine gap** 비용이다.

alignment Cost

$$= \sum_{i=2}^{N} \sum_{j=1}^{i-1} \text{교체비용}(A_i, A_j) + gap \text{ 비용}(cost_{open}, cost_{extend})$$

SAGA



2. Evaluation

Input

Population

Evaluation

Output

Population with Evaluation

Gap 비용

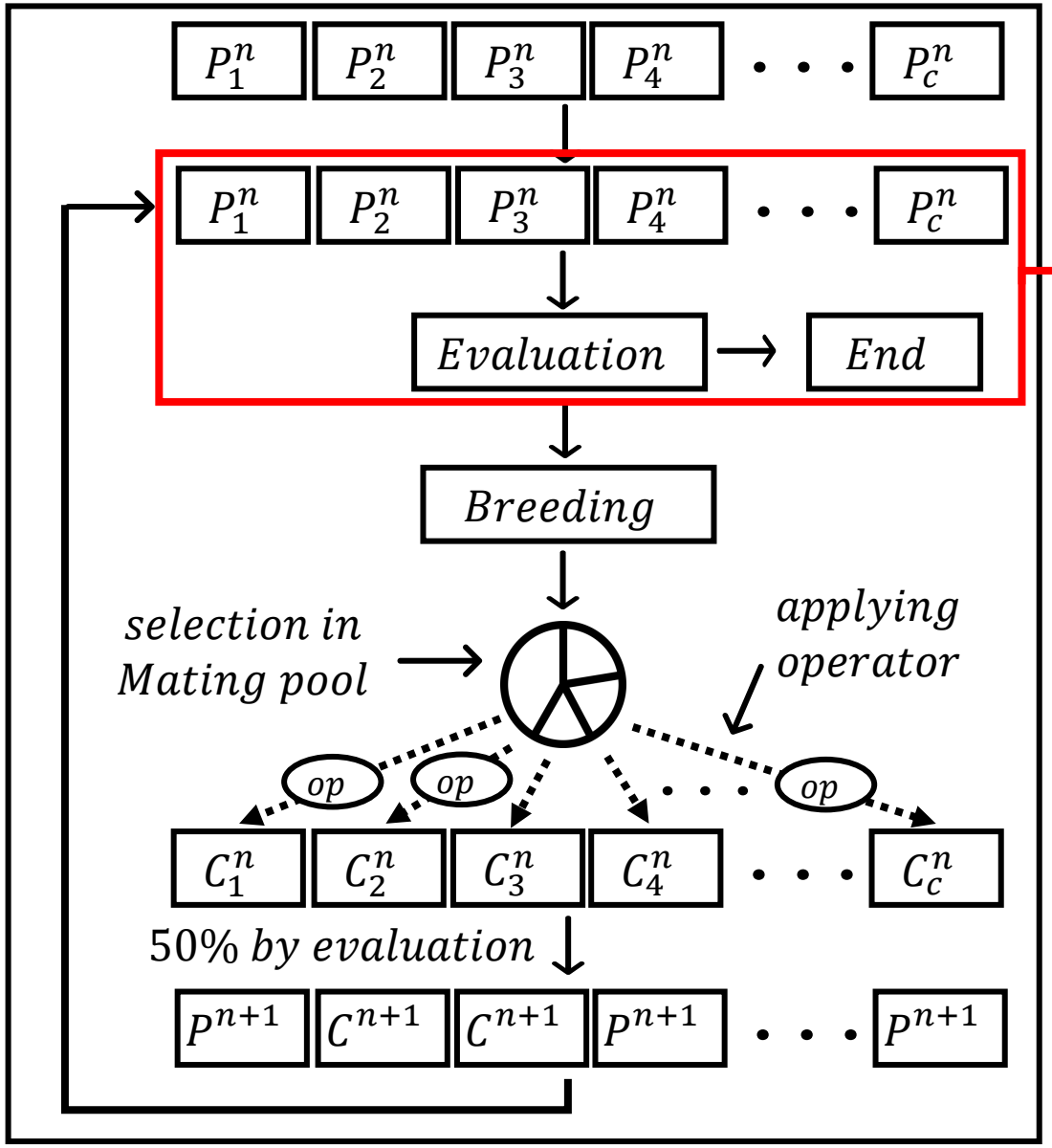
- Gap 비용은 alignment의 기호 중 '—'에 부과되는 비용이다.
- 여기서 사용된 gap 비용은 **natural affine gap** 비용이다.

$cost_{open}$: gap이 처음 나왔을 때 주는 비용
$cost_{extend}$: gap이 나올 때 마다 주는 비용

alignment Cost

$$= \sum_{i=2}^{N} \sum_{j=1}^{i-1} \text{교체비용}(A_i, A_j) + gap \text{ 비용}(cost_{open}, cost_{extend})$$

SAGA



2. Evaluation

Input

Population

Evaluation →

Output

Population with Evaluation

Gap 비용

- Gap 비용은 alignment의 기호 중 '—'에 부과되는 비용이다.
- 여기서 사용된 gap 비용은 **natural affine gap** 비용이다.

$cost_{open}$: **-1** $cost_{extend}$: **-0.5**라 가정

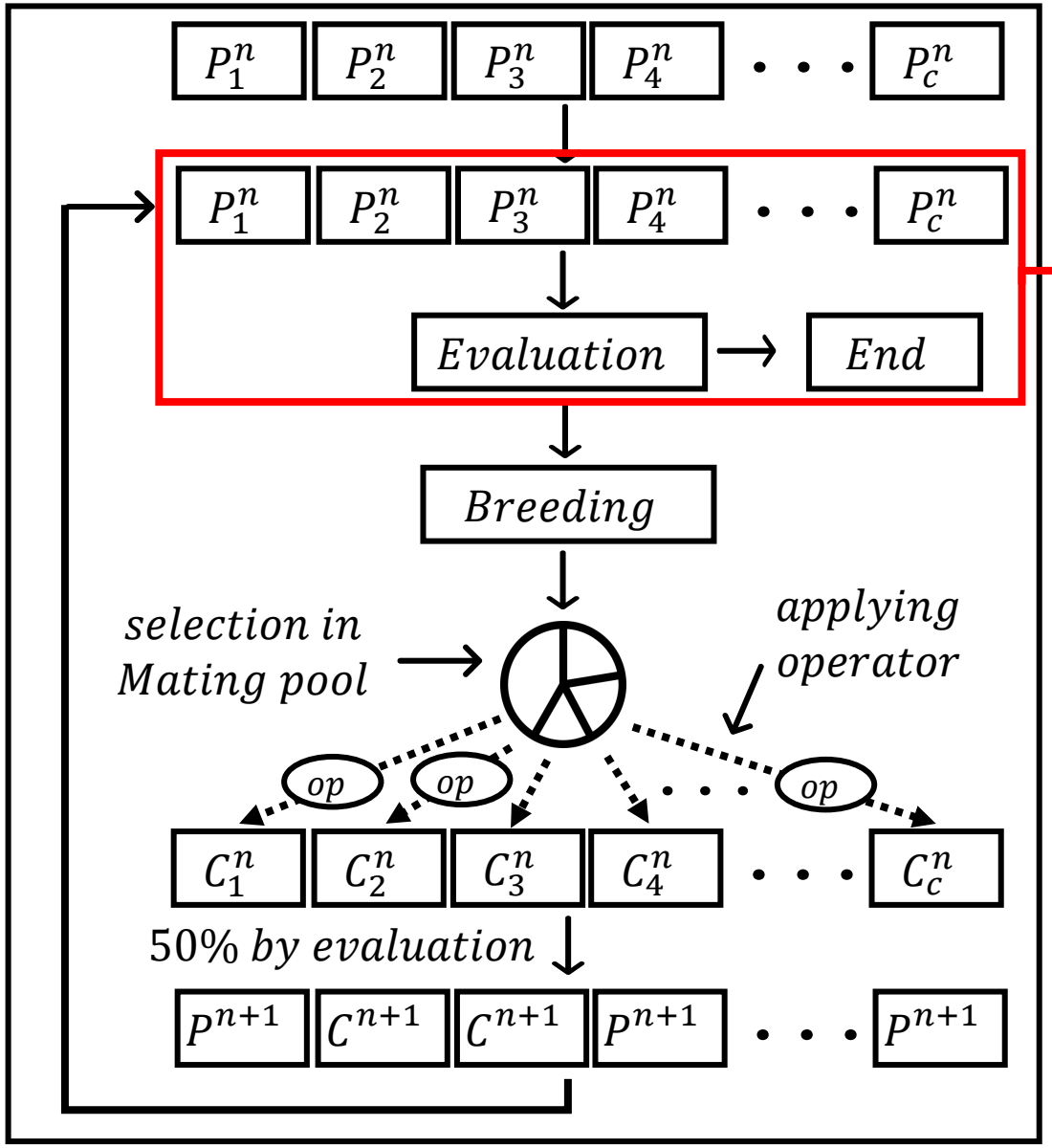
1: A T G C G G T T

2: A — T C — — G T

4-1.5

$cost_{open} + cost_{extend} = -1 - 0.5 = -1.5$

SAGA



2. Evaluation

Input

Population

Evaluation

Output

Population with Evaluation

Gap 비용

- Gap 비용은 alignment의 기호 중 '—'에 부과되는 비용이다.
- 여기서 사용된 gap 비용은 **natural affine gap** 비용이다.

$cost_{open}$: **-1** $cost_{extend}$: **-0.5**라 가정

1: A T G C — — T T

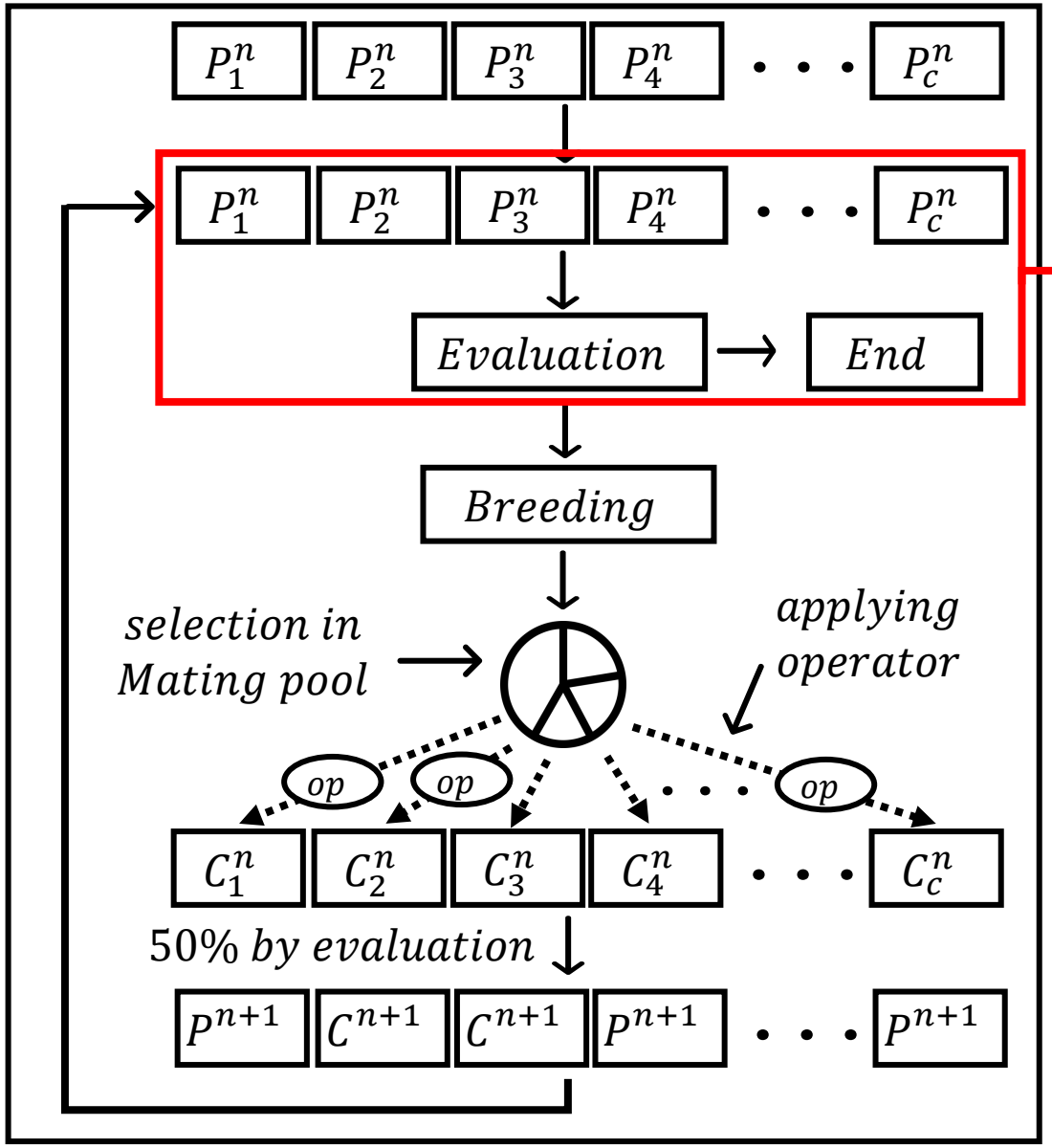
2: A — T C G G G T

4-1.5-2+9-1.5-0.5-2+5 = 10.5

$cost_{open} + cost_{extend} = -1 - 0.5 = \mathbf{-1.5}$

$cost_{extend} = \mathbf{-0.5}$

SAGA



2. Evaluation

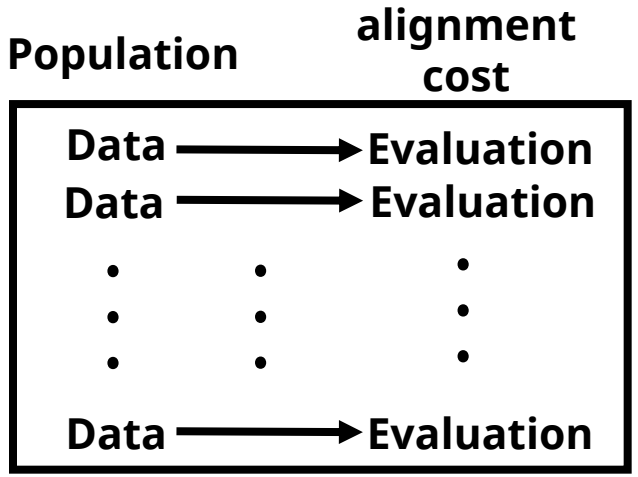
Input

Population

Evaluation →

Output

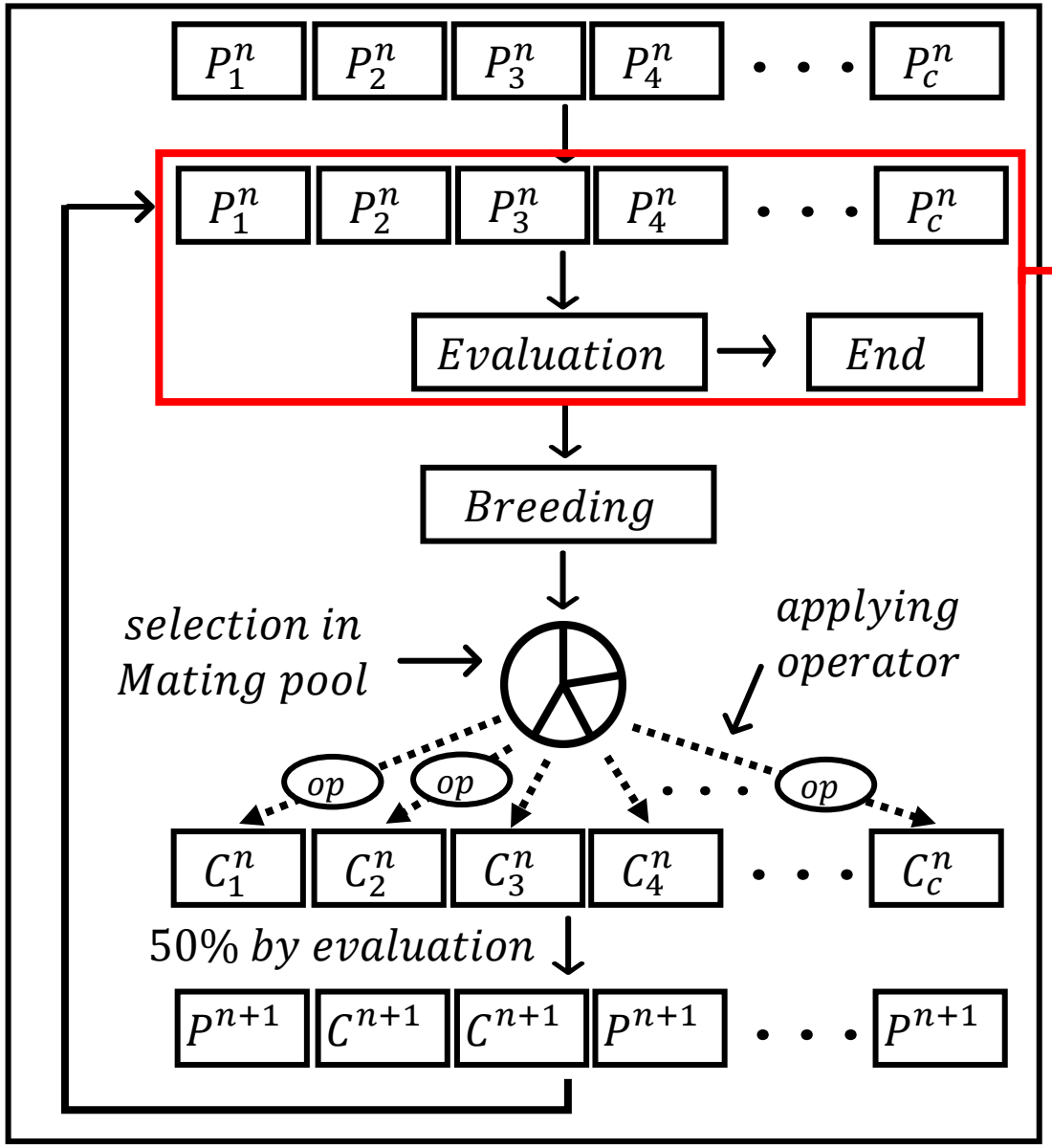
Population with Evaluation



alignment Cost

$$= \sum_{i=2}^{N} \sum_{j=1}^{i-1} \text{교체비용}(A_i, A_j) + gap \text{ 비용}(cost_{open}, cost_{extend})$$

SAGA



2. Evaluation

Input

Population

Evaluation

Output

Population with Evaluation

Population

alignment cost

Data → Evaluation

Data → Evaluation

. . .

Data → Evaluation

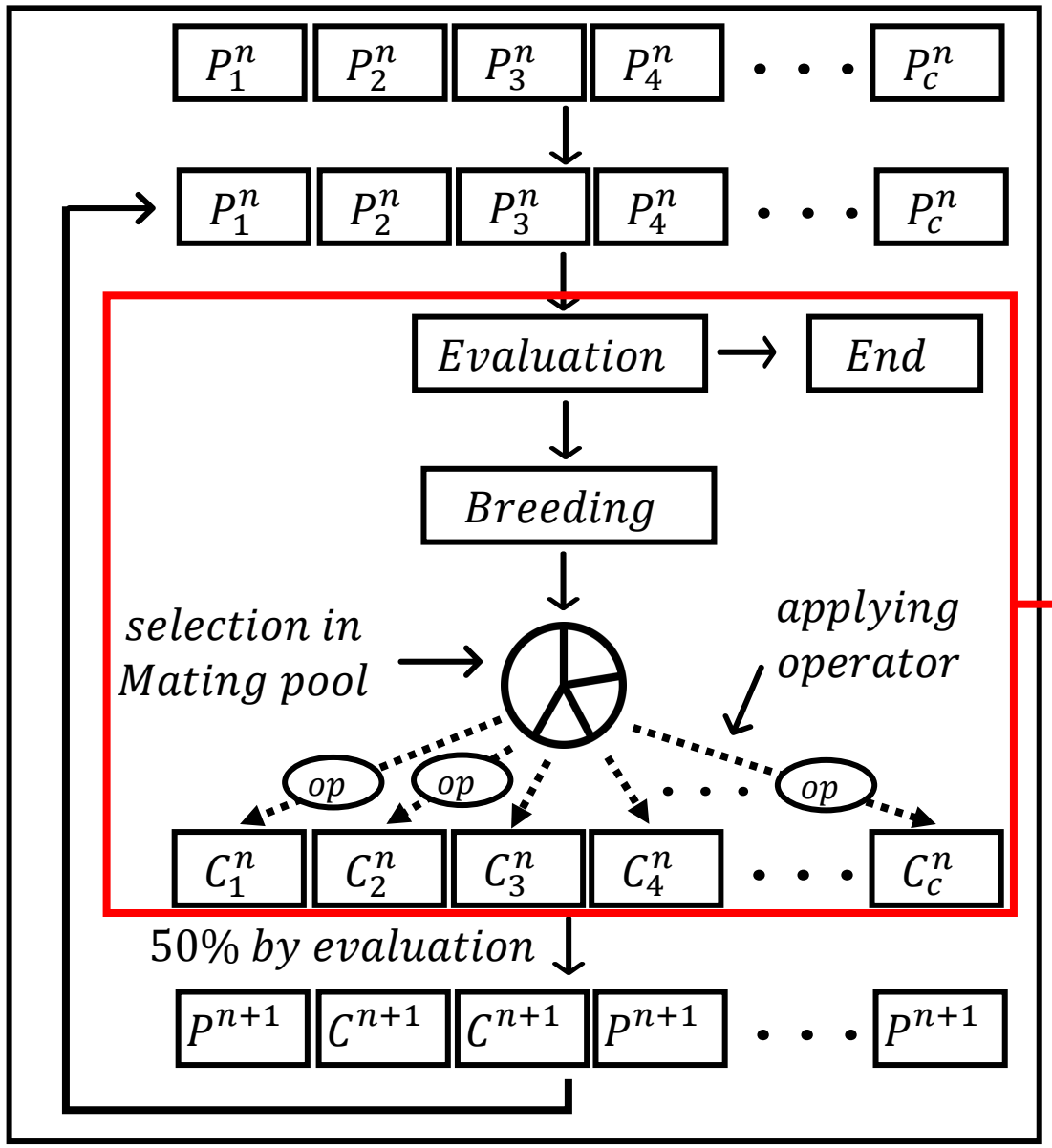
if (max generation < n)
Or
if (20% of best evaluations is nearly unchanged)

End

alignment Cost

$$= \sum_{i=2}^{N} \sum_{j=1}^{i-1} \text{교체비용}(A_i, A_j) + gap \text{ 비용}(cost_{open}, cost_{extend})$$

SAGA



3. Breeding

Input

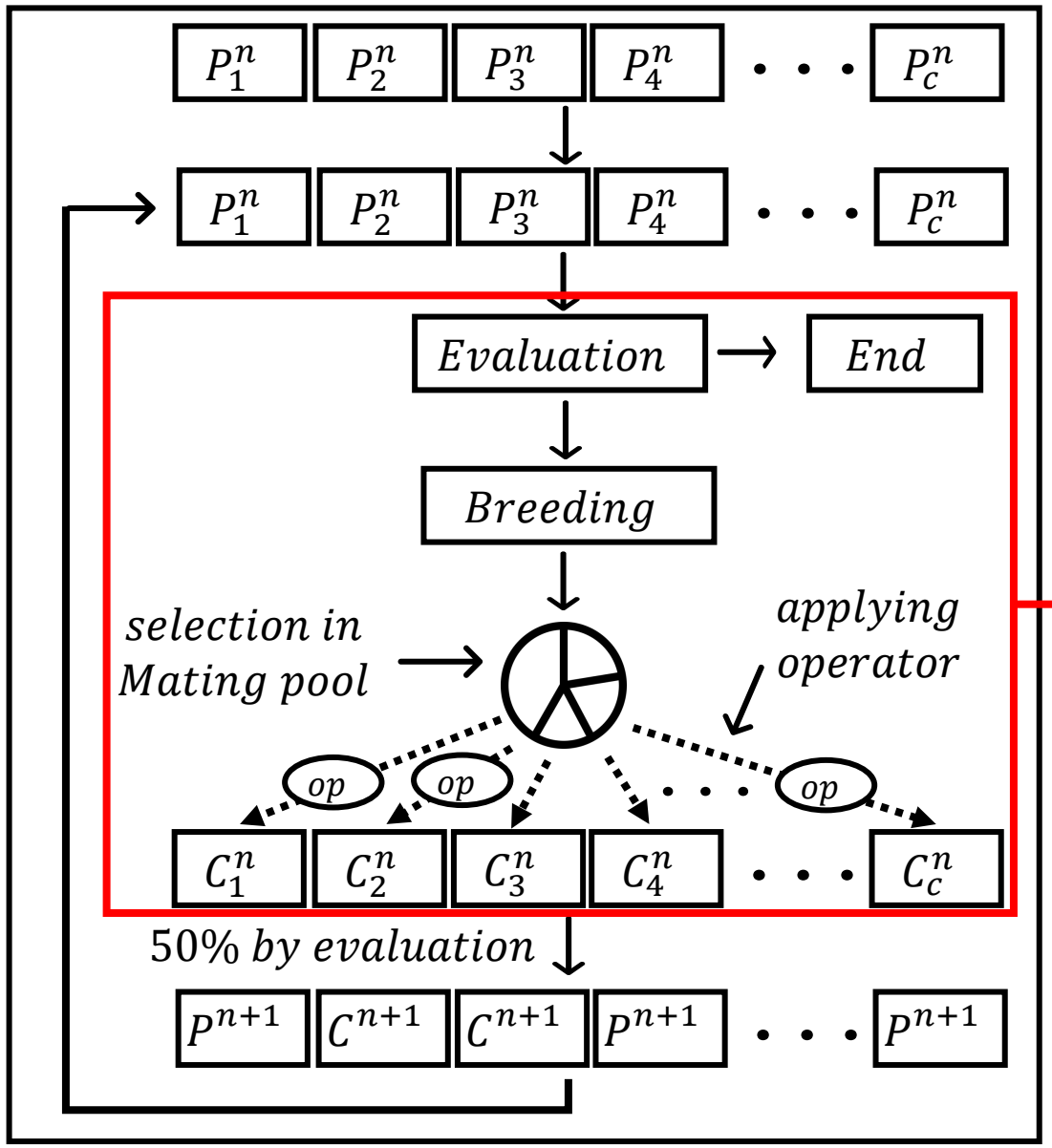
Population

Breeding

Output

Child data set

SAGA



3. Breeding

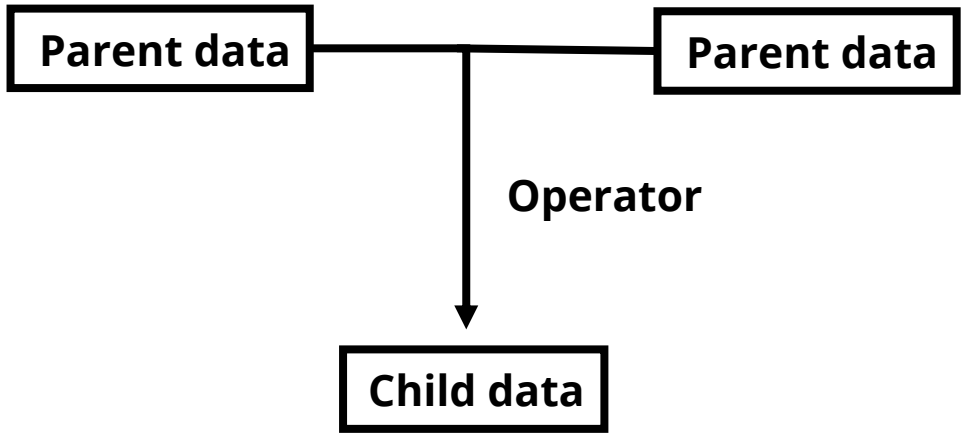
Input

Population

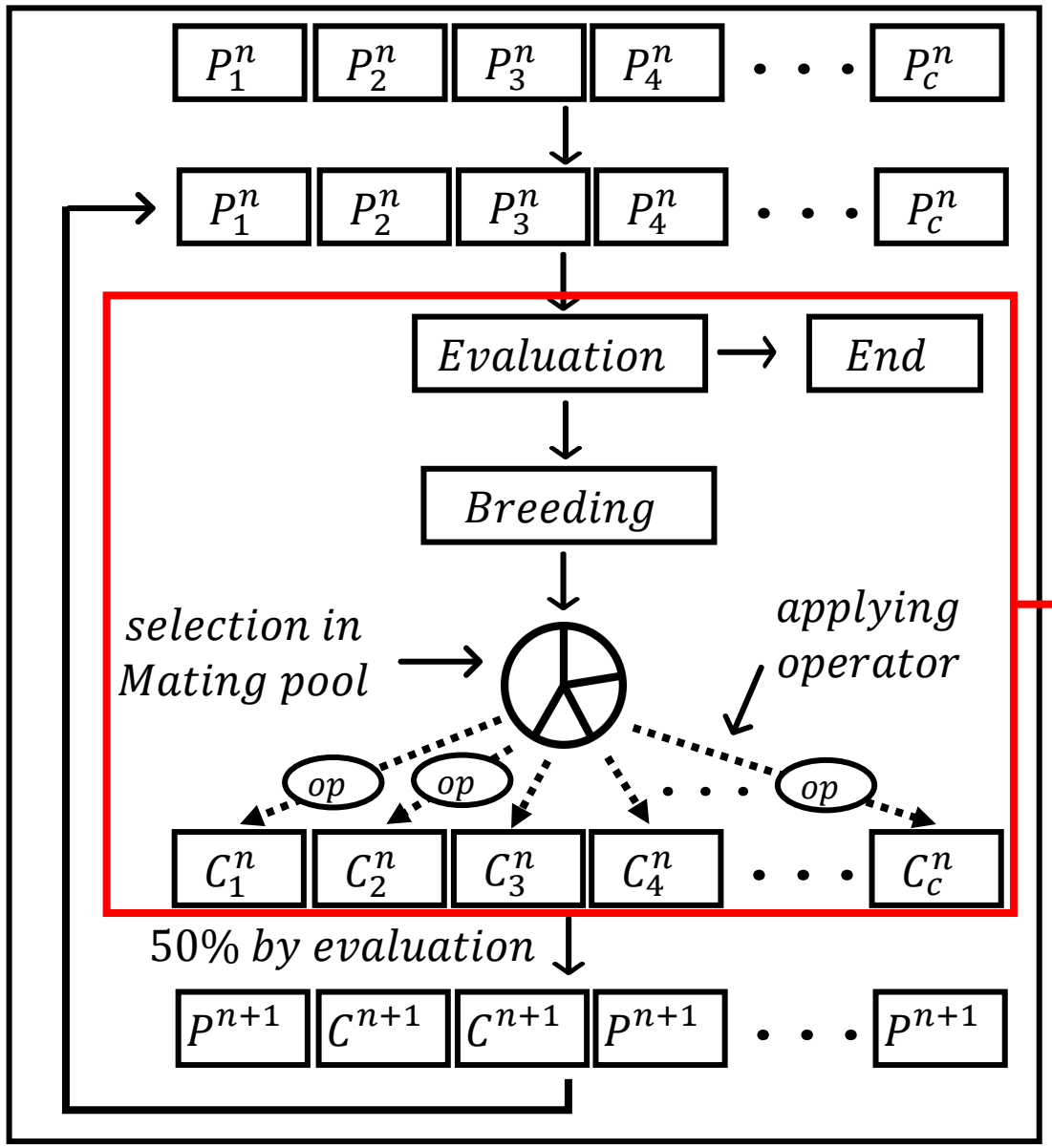
Breeding

Output

Child data set



SAGA



3. Breeding

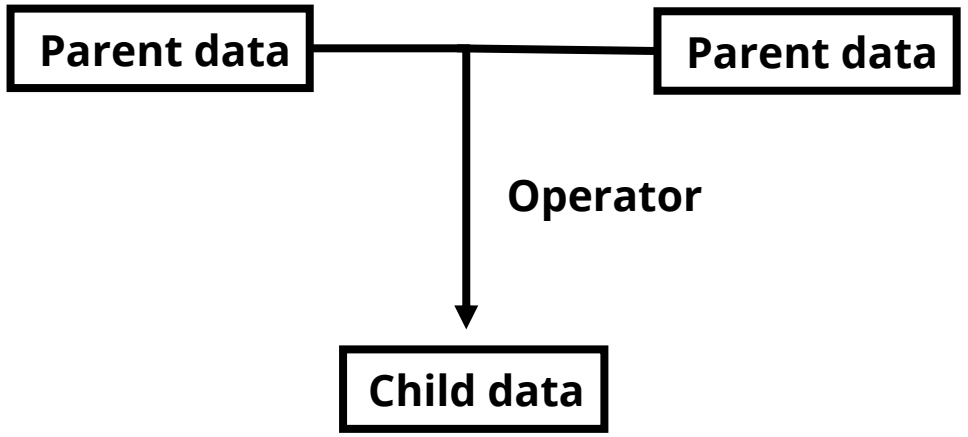
Input

Population

Breeding

Output

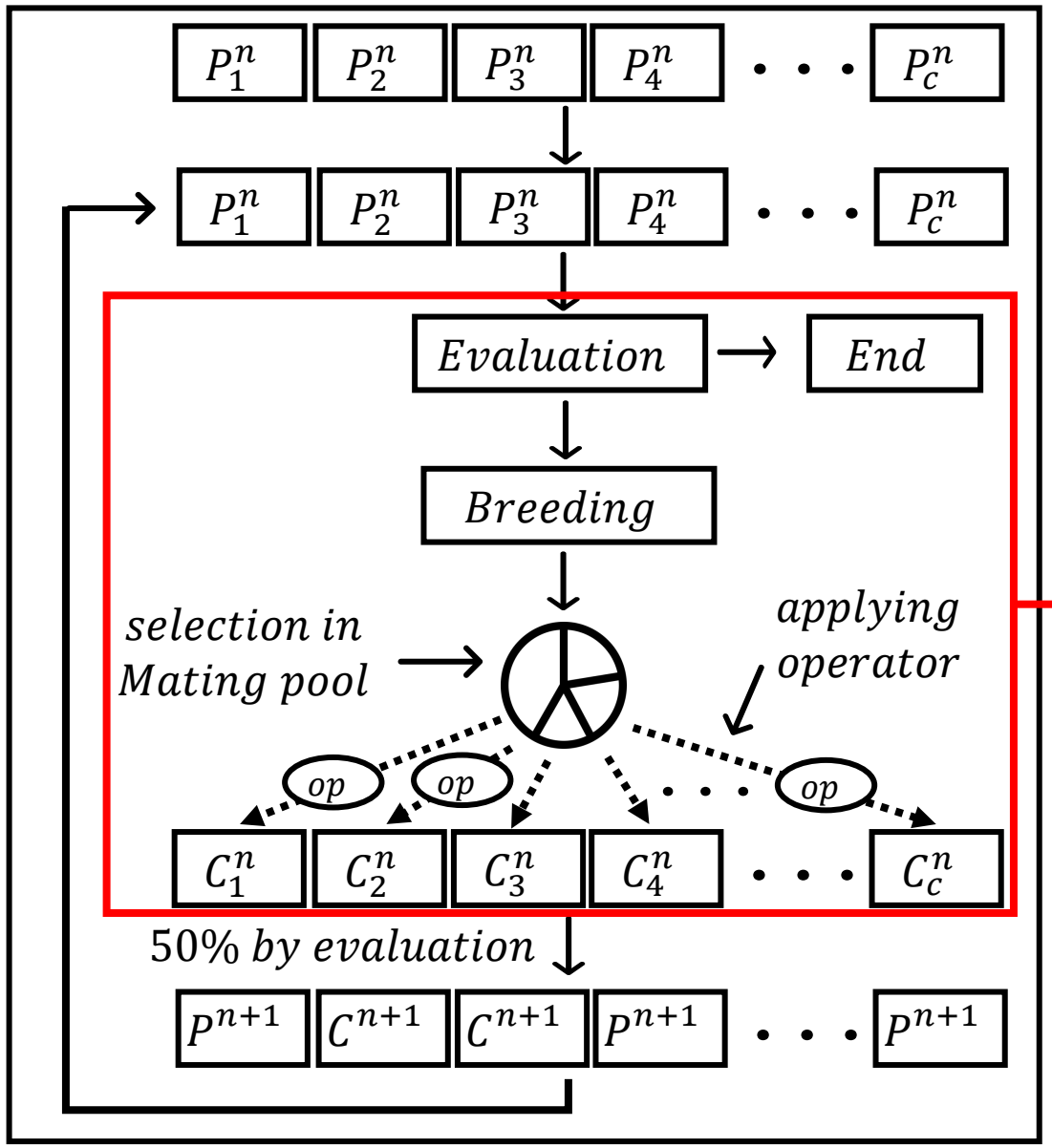
Child data set



이를 위해 고려해야 하는 사항

1. 어떻게 부모 데이터를 선정할 것인가?
2. 어떤 Operator를 적용할 것인가?

SAGA



3. Breeding

Input

Population

Breeding

Output

Child data set

Parent data

Parent data

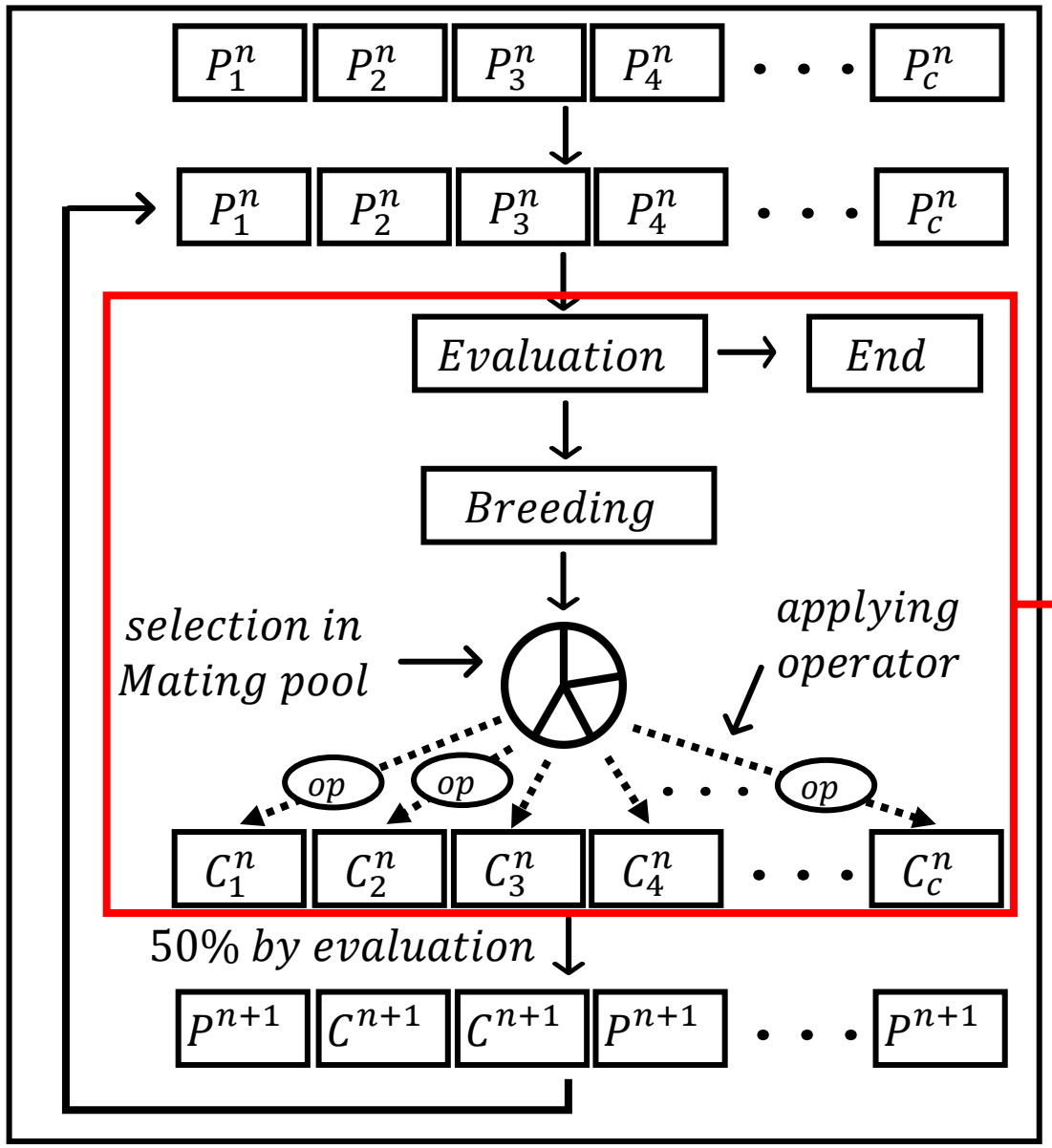
Operator

Child data

이를 위해 고려해야 하는 사항

1. 어떻게 부모 데이터를 선정할 것인가? → mating pool
2. 어떤 Operator를 적용할 것인가? → crossover, mutation

SAGA



3. Breeding

Input

Population

Breeding

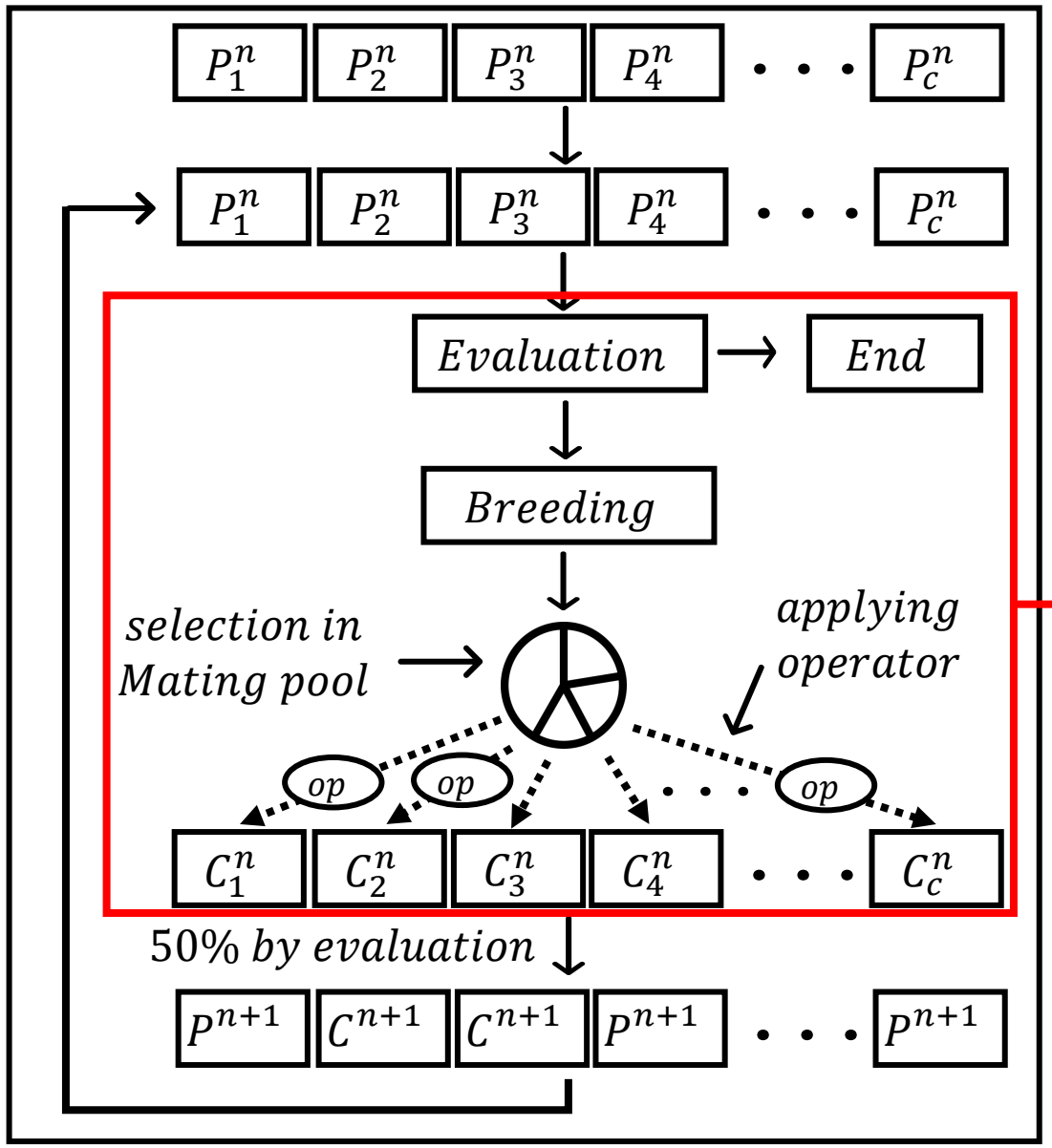
Output

Child data set

Mating Pool

- 한 세대에 있는 데이터의 **evaluation** 값에 따라 **데이터의 분포**가 달라지는 집합

SAGA



3. Breeding

Input → **Breeding** → **Output**

Population → **Child data set**

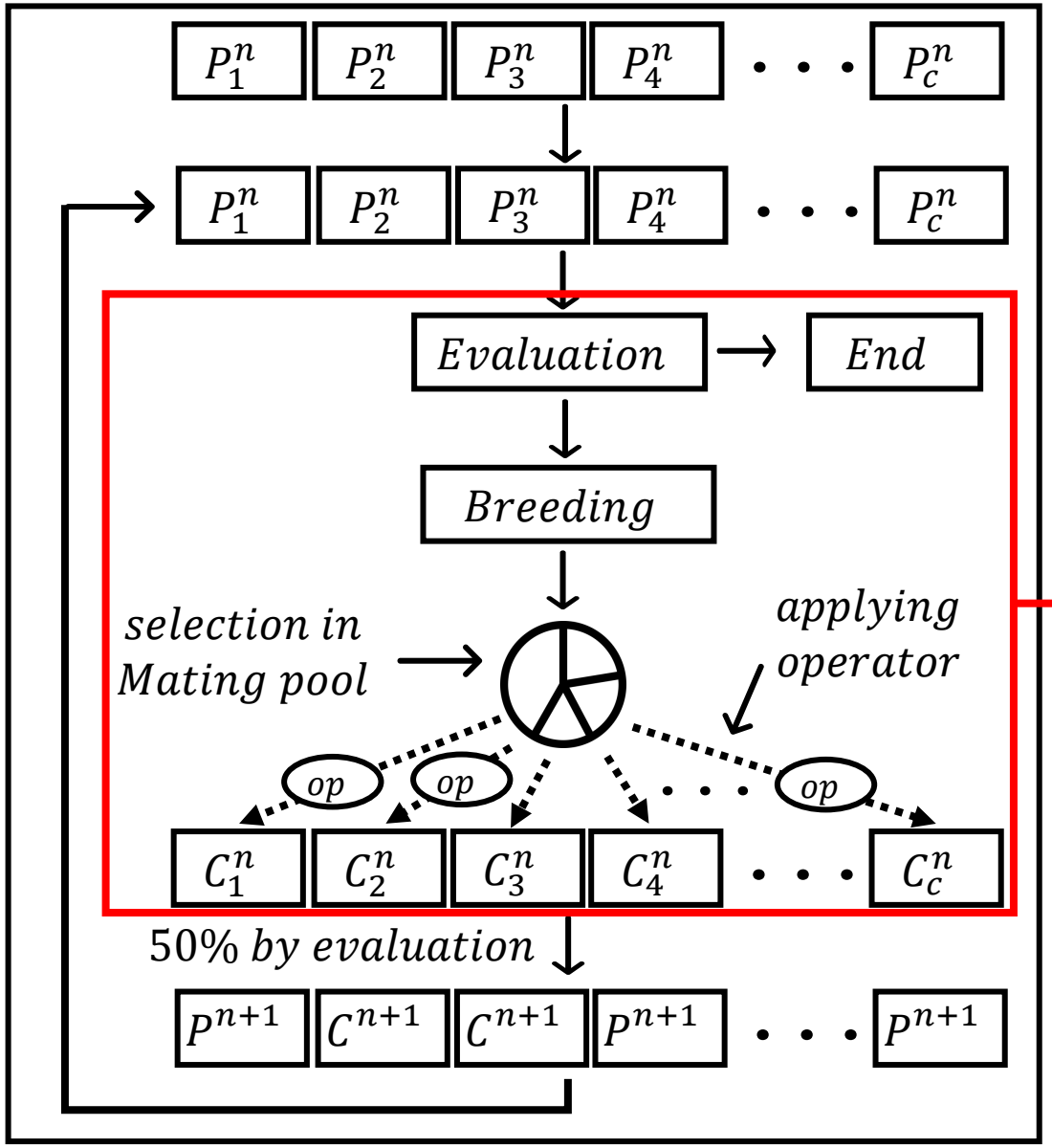
Mating Pool

- 한 세대에 있는 데이터의 **evaluation** 값에 따라 **데이터의 분포**가 달라지는 집합

Evaluation of data ↑ → (비례) → **Proportion of MP** ↑

Evaluation of data ↓ → **Proportion of MP** ↓

SAGA



3. Breeding

Input

Population

Breeding

Output

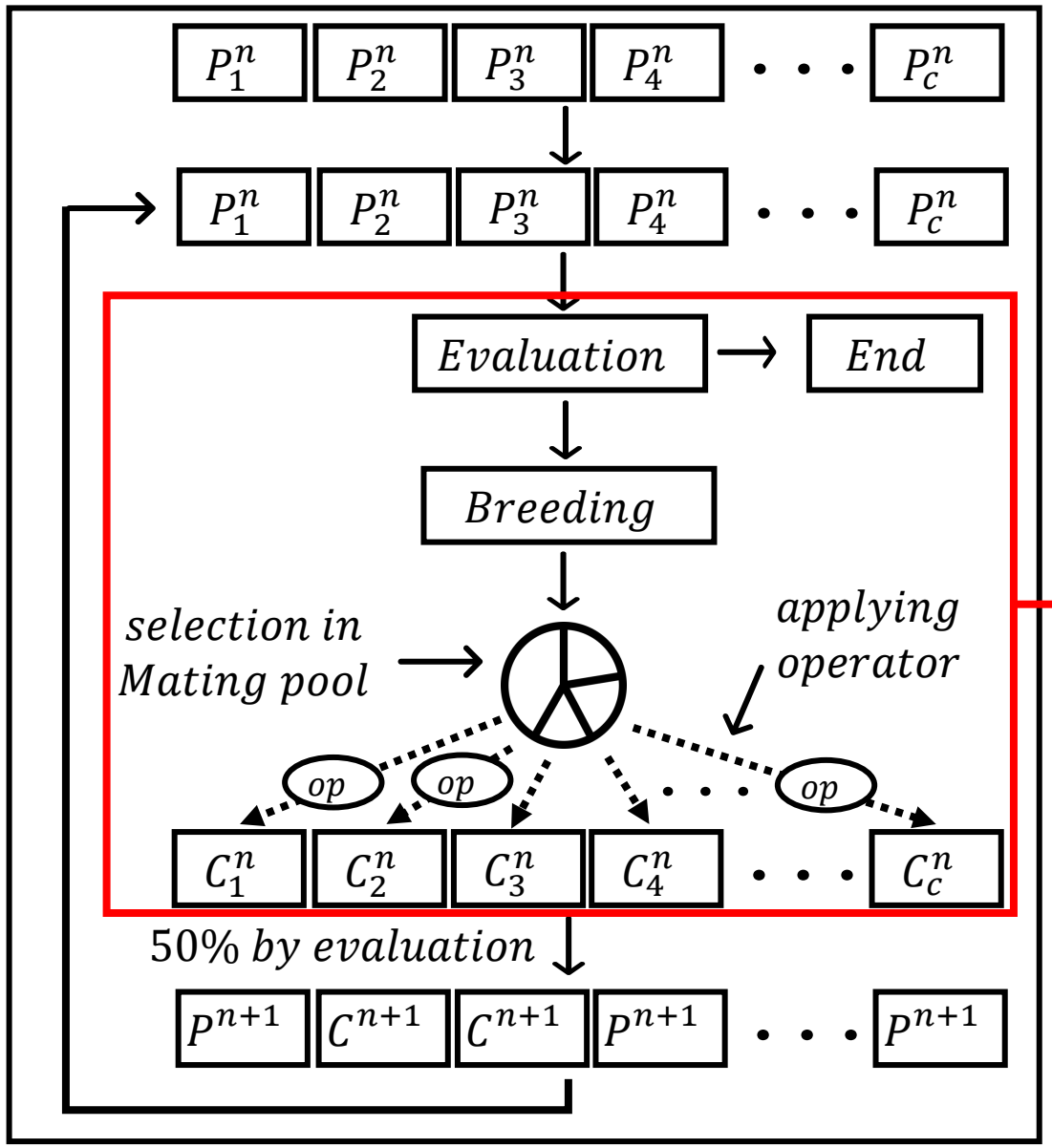
Child data set

Mating Pool

- 한 세대에 있는 데이터의 **evaluation** 값에 따라 **데이터의 분포**가 달라지는 집합

$$\text{MP안의 데이터 수} = \left\lceil \frac{evaluation\ of\ data}{sum\ of\ evaluations\ of\ population} \times 100 \right\rceil$$

SAGA



3. Breeding

Input

Population

Breeding

Output

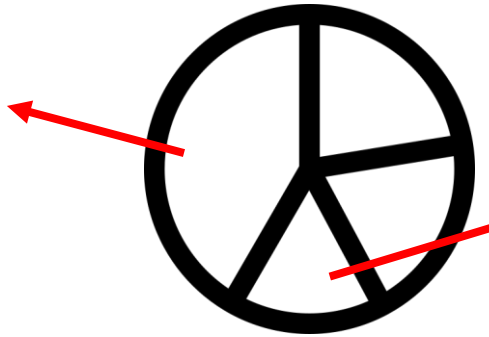
Child data set

Mating Pool

- 한 세대에 있는 데이터의 **evaluation** 값에 따라 **데이터의 분포**가 달라지는 집합

$$\text{MP안의 데이터 수} = \left\lceil \frac{evaluation\ of\ data}{sum\ of\ evaluations\ of\ population} \times 100 \right\rceil$$

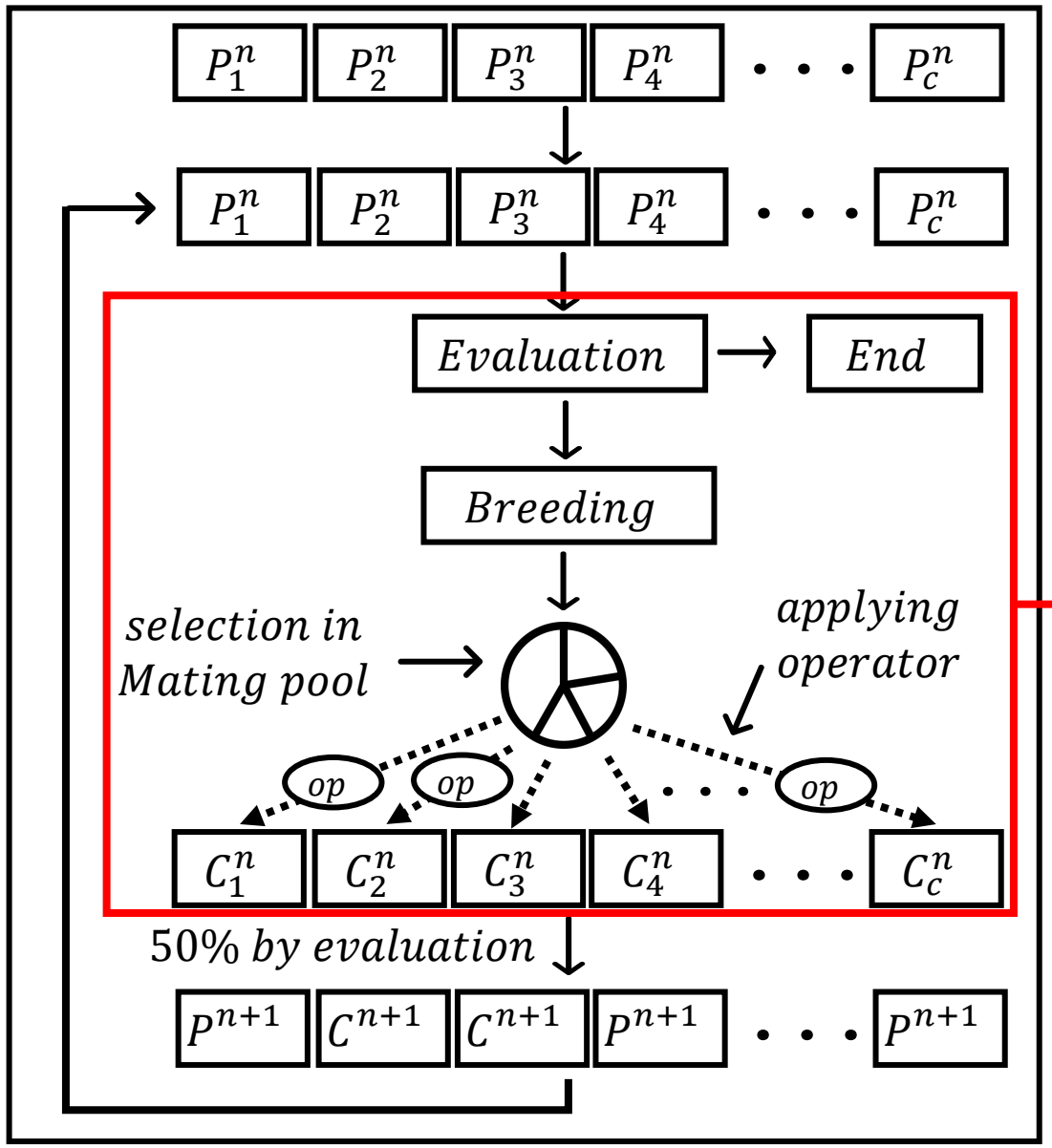
상대적으로 높은 Evaluation



상대적으로 낮은 Evaluation

Mating Pool

SAGA



3. Breeding

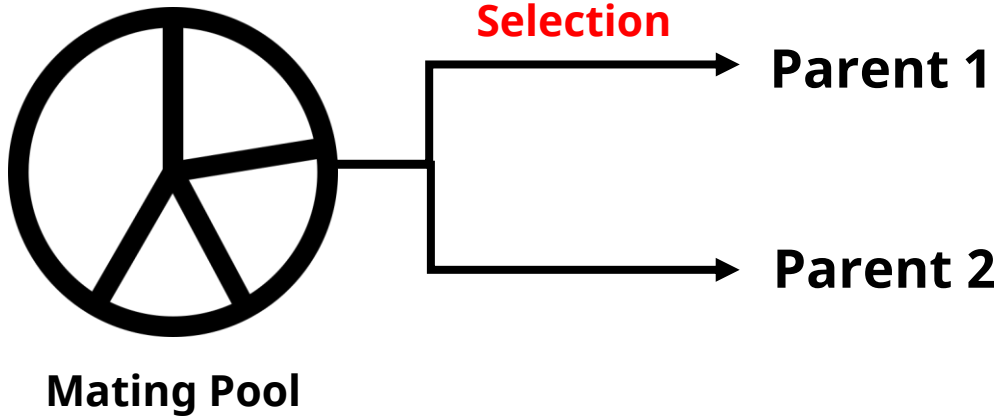
Input

Population

Breeding

Output

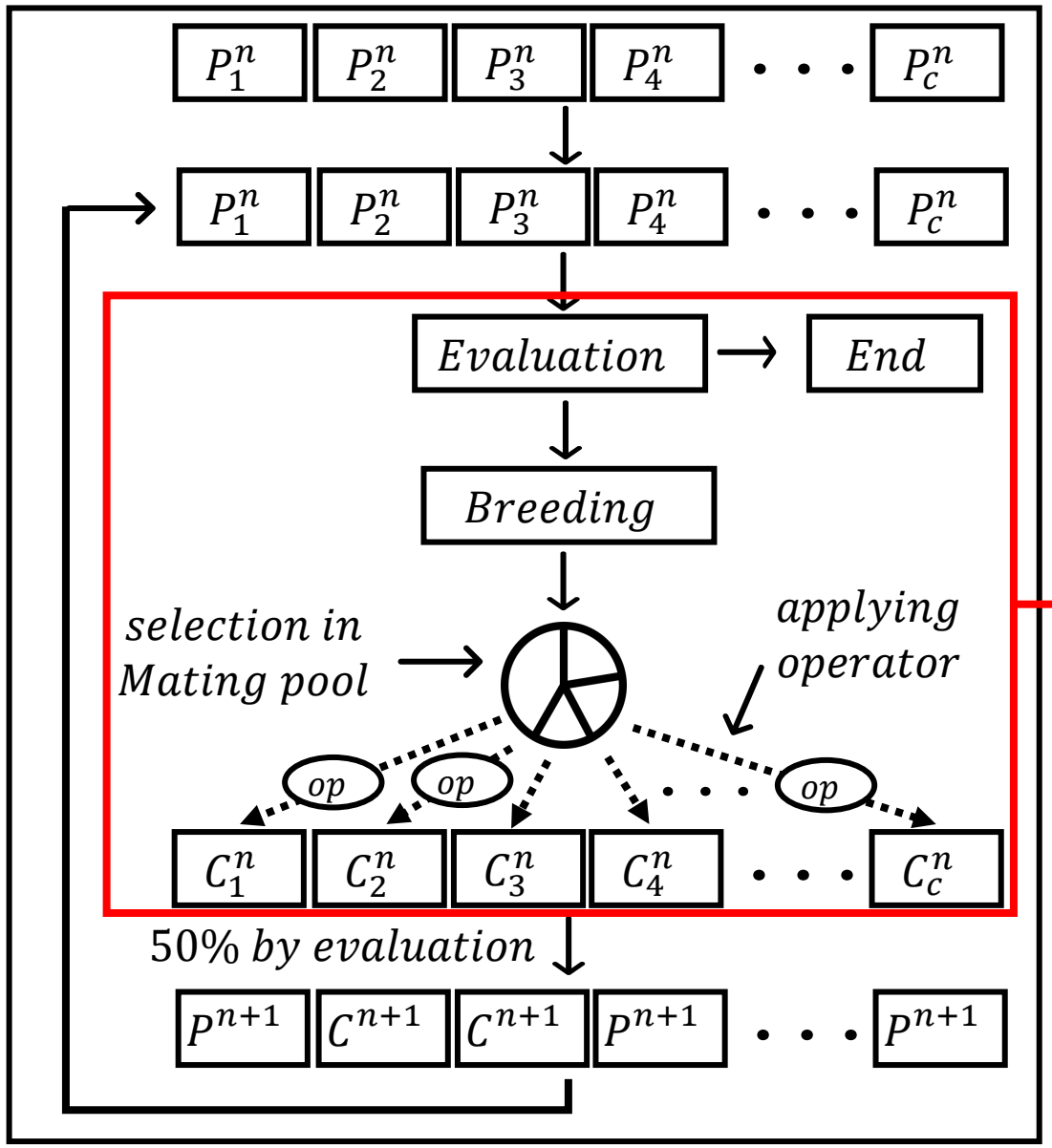
Child data set



- 생성된 Mating Pool에서 랜덤으로 두 부모 데이터를 선택한다.
- 즉, Evaluation 값이 높을 수록 부모가 될 확률이 크다.

= 좀 더 **우월한 Evaluation을 가진 부모**를 이용하여 **새로운 세대를 구성**

SAGA



3. Breeding

Input

Population

Breeding

Output

Child data set

Parent data

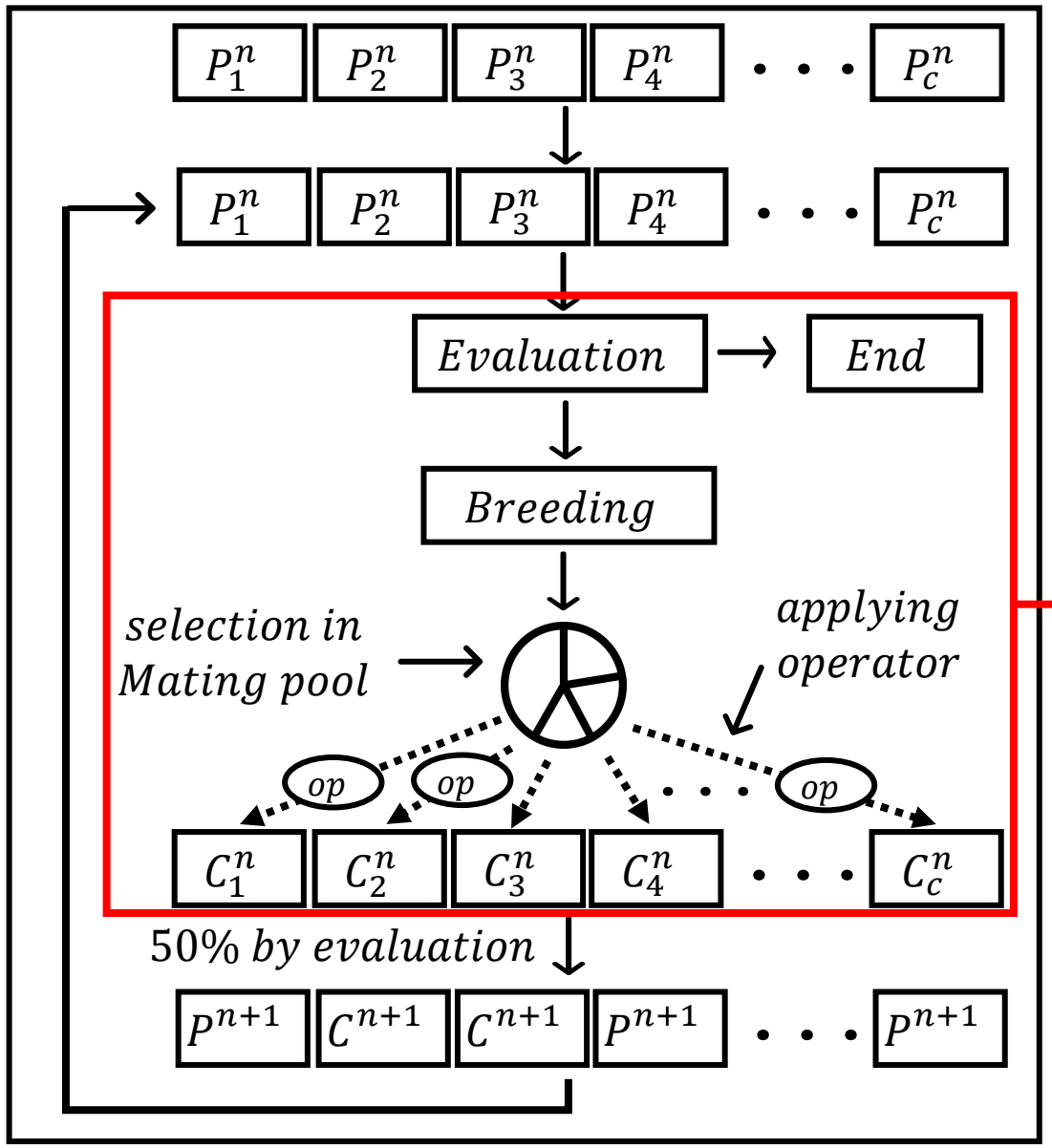
Parent data

Child data

Operator

- Crossover
- Mutation
- Block Shuffling
- Block Searching
- Local Optimal Rearrangement
- Sub-optimal rearrangement
- etc …

SAGA



3. Breeding

Input

Population

Breeding

Output

Child data set

Parent data

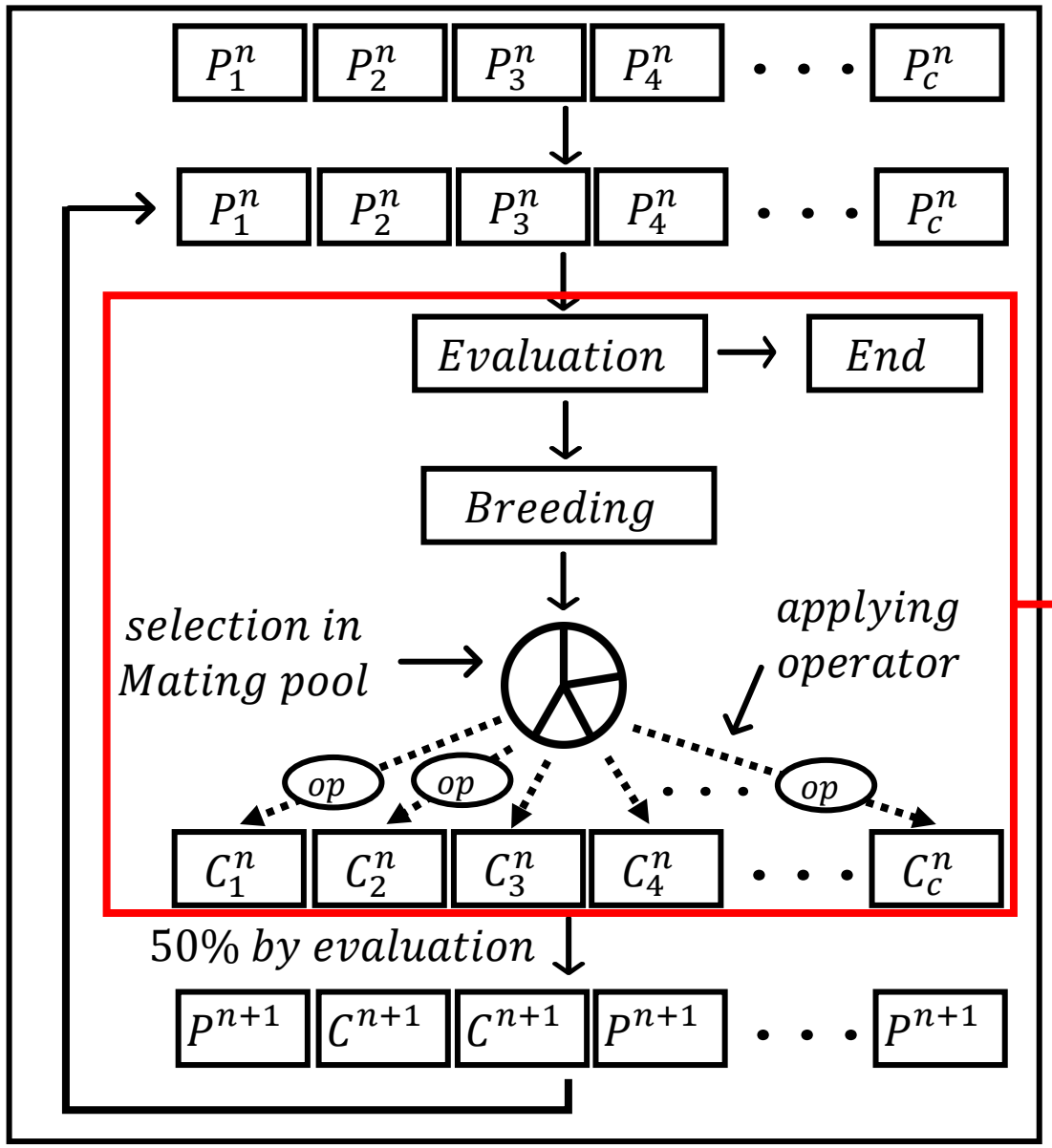
Parent data

Child data

Operator

- Crossover
- Mutation
- Block Shuffling
- Block Searching
- Local Optimal Rearrangement
- Sub-optimal rearrangement
- etc ...

SAGA



3. Breeding

Input

Population

Breeding

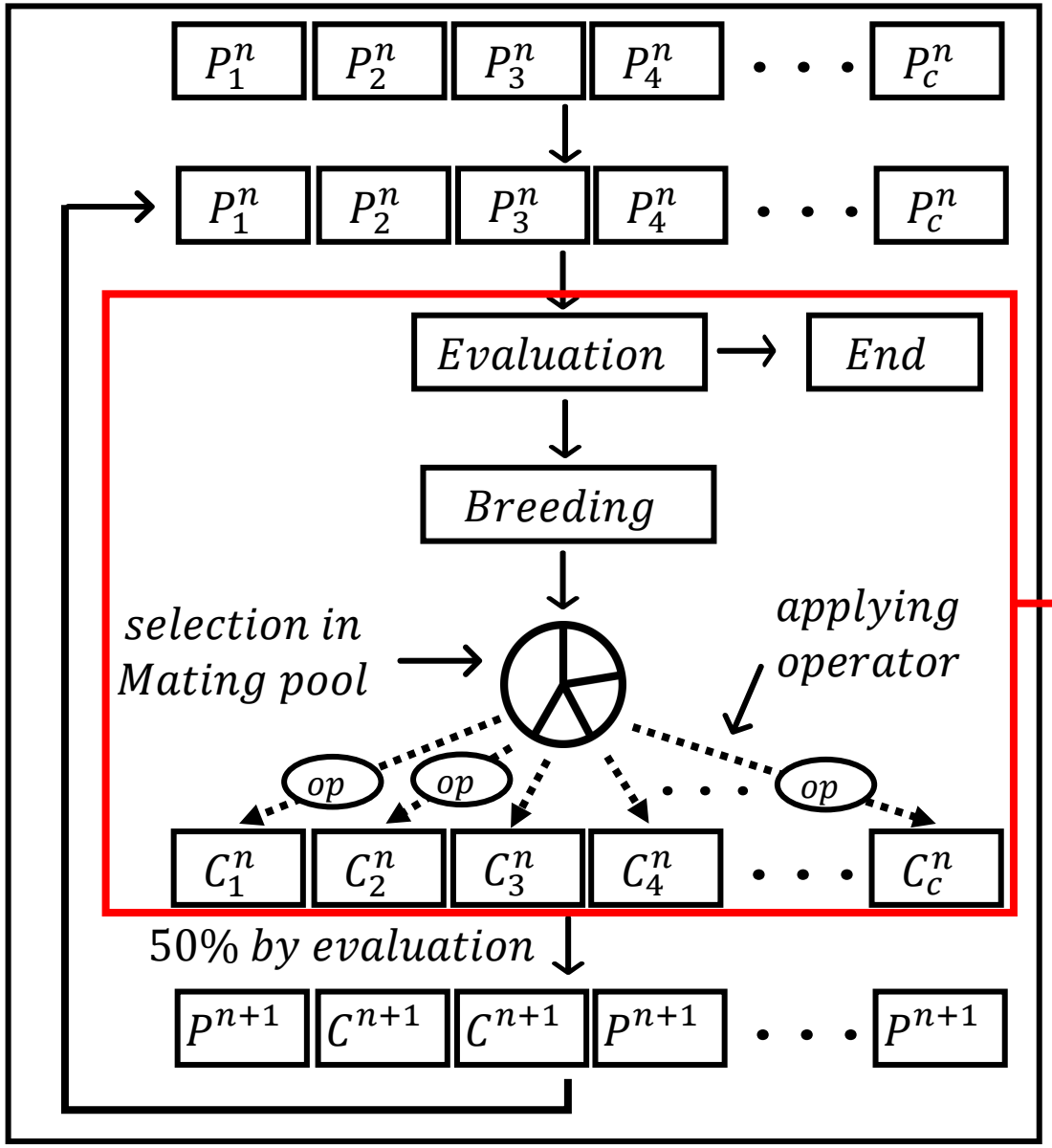
Output

Child data set

Crossover

- 두 부모 데이터를 이용하여 **새로운 자식 데이터를 생성**하는 연산자
- 각 데이터의 **sequence들을 랜덤으로 섞는다**.

SAGA



3. Breeding

Input

Population

Breeding

Output

Child data set

Crossover

- 두 부모 데이터를 이용하여 **새로운 자식 데이터를 생성**하는 연산자
- 각 데이터의 **sequence들을 랜덤으로 섞는다**.

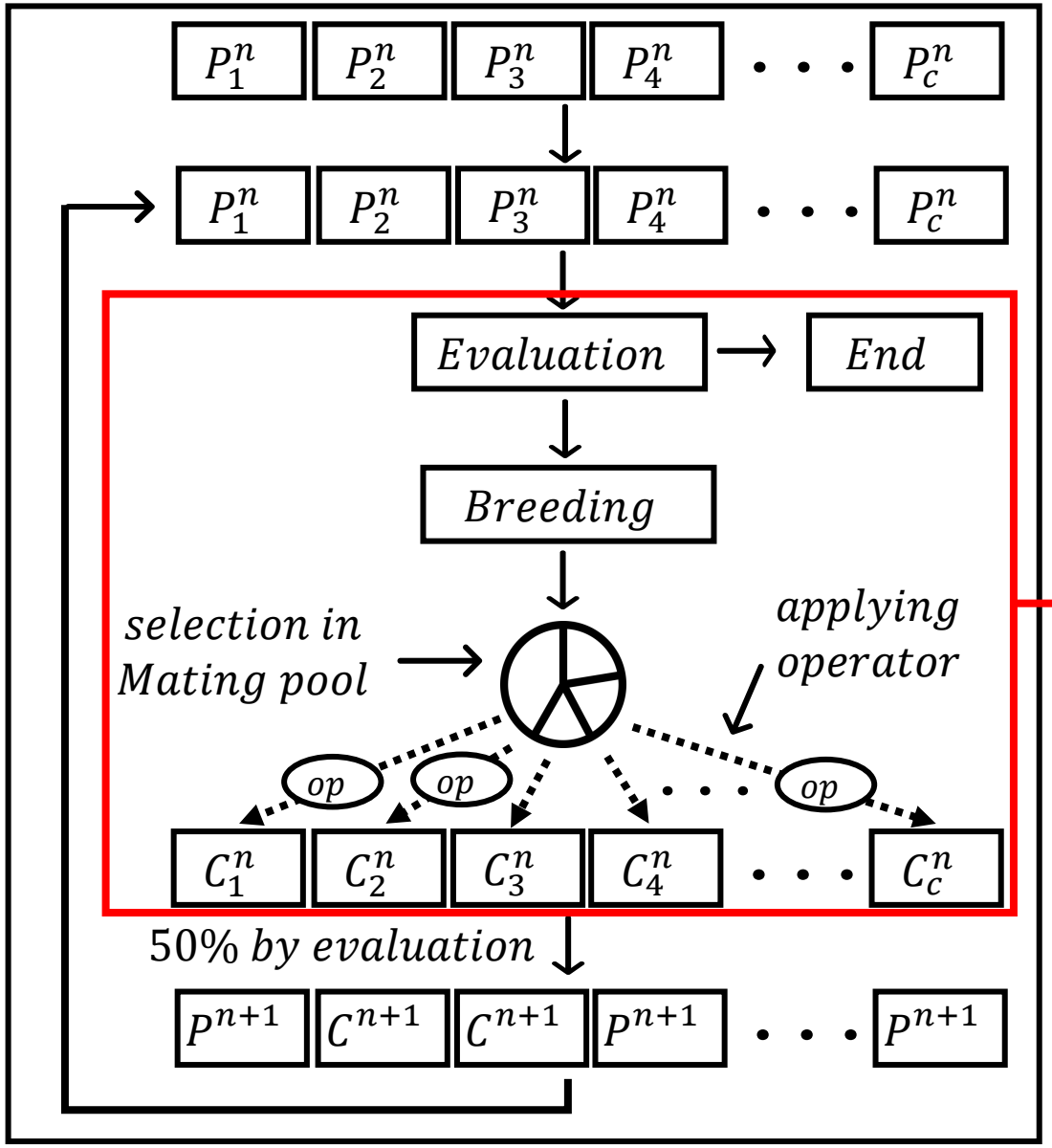
Parent 1

alignment
alignment
alignment
alignment
alignment

Parent 2

alignment
alignment
alignment
alignment
alignment

SAGA



3. Breeding

Input

Population

Breeding

Output

Child data set

Crossover

- 두 부모 데이터를 이용하여 **새로운 자식 데이터를 생성**하는 연산자
- 각 데이터의 **sequence들을 랜덤으로 섞는다**.

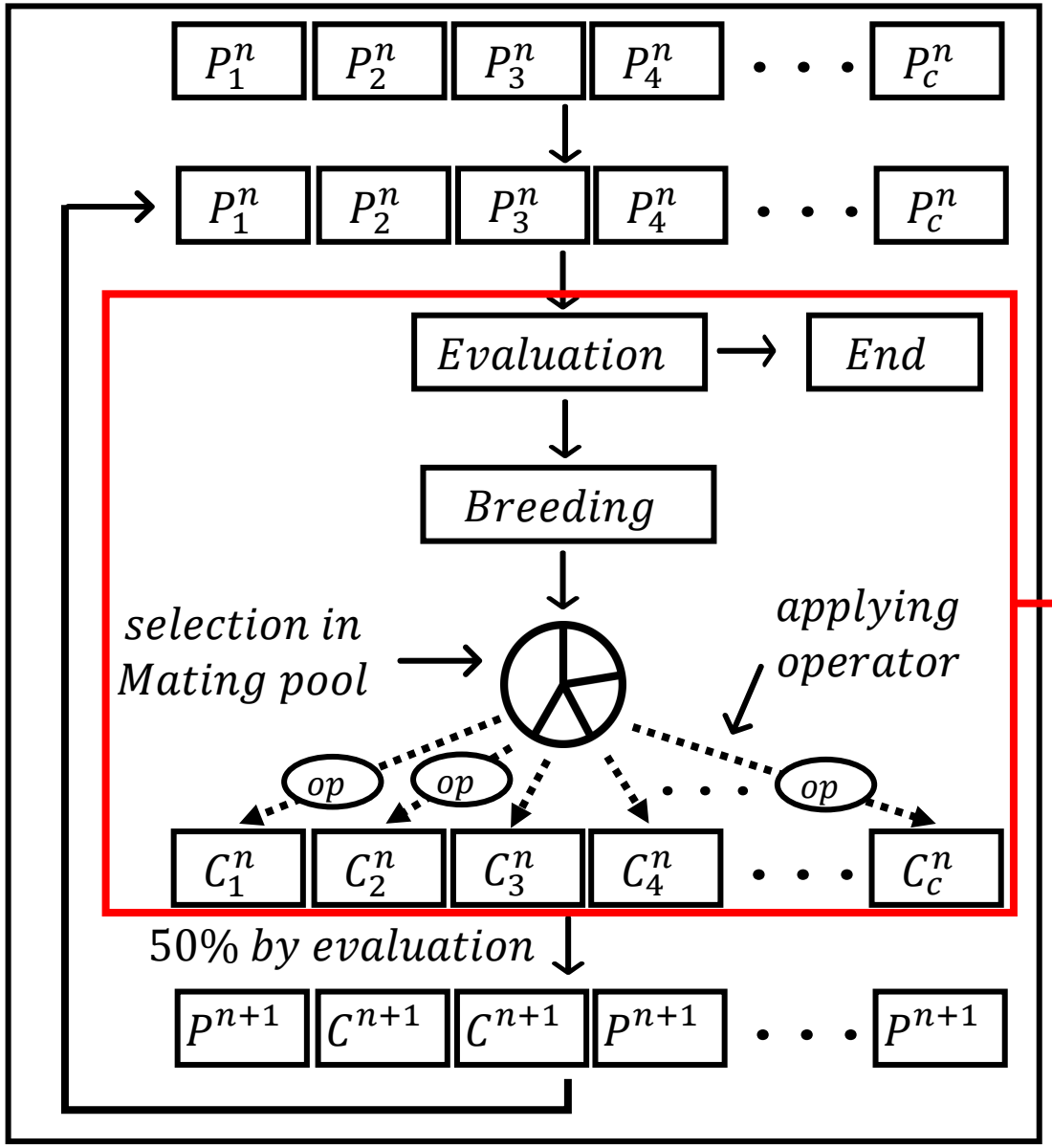
Parent 1

alignment
alignment
alignment
alignment
alignment

Parent 2

alignment
alignment
alignment
alignment
alignment

SAGA



3. Breeding

Input

Population

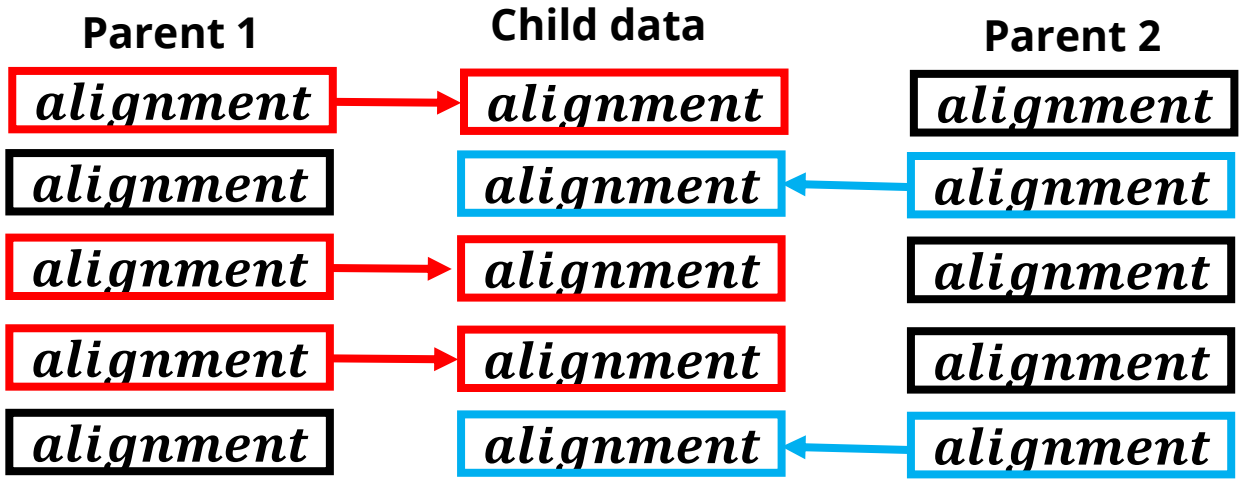
Breeding

Output

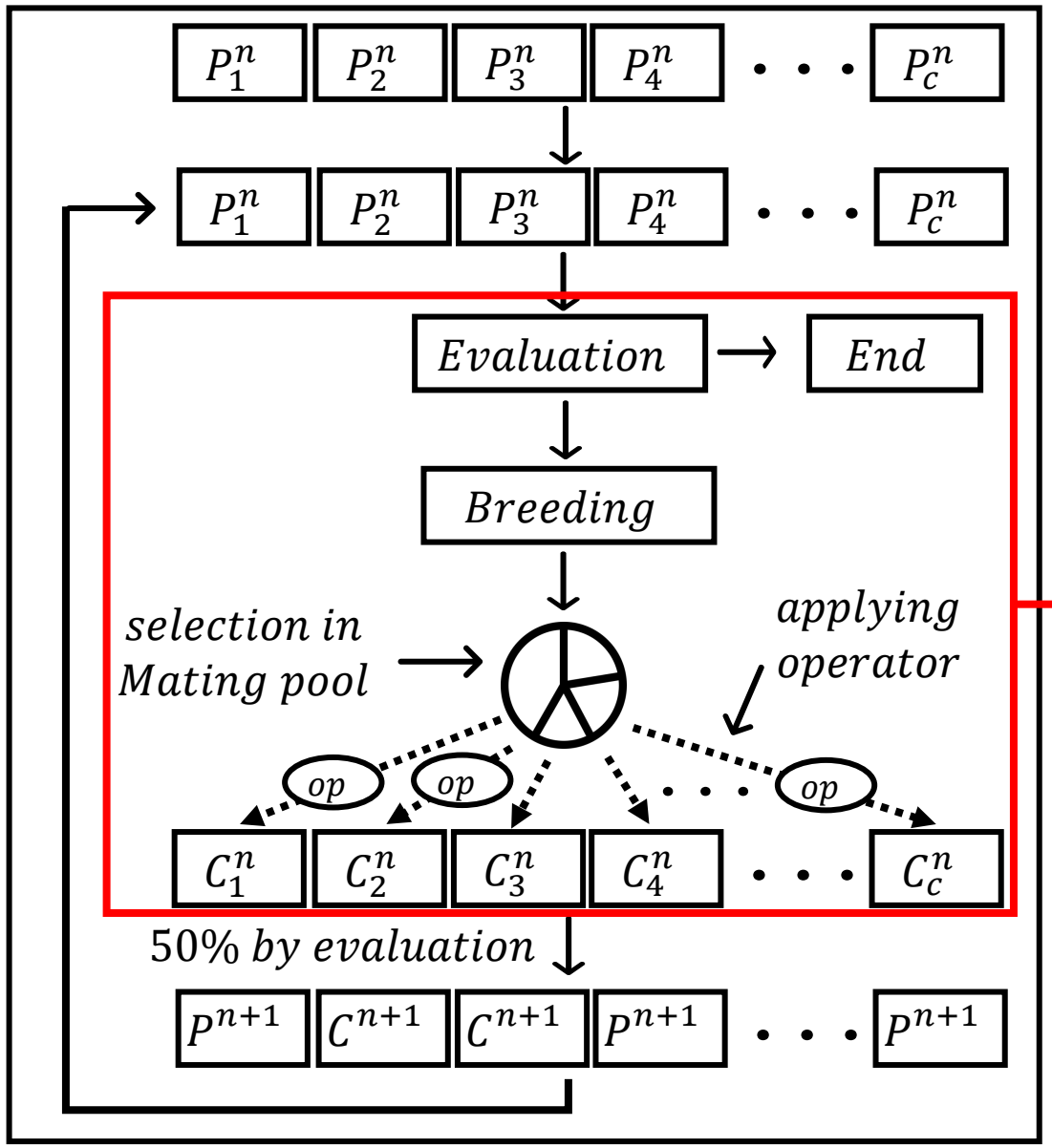
Child data set

Crossover

- 두 부모 데이터를 이용하여 **새로운 자식 데이터를 생성**하는 연산자
- 각 데이터의 **sequence들을 랜덤으로 섞는다**.



SAGA



3. Breeding

Input

Population

Breeding

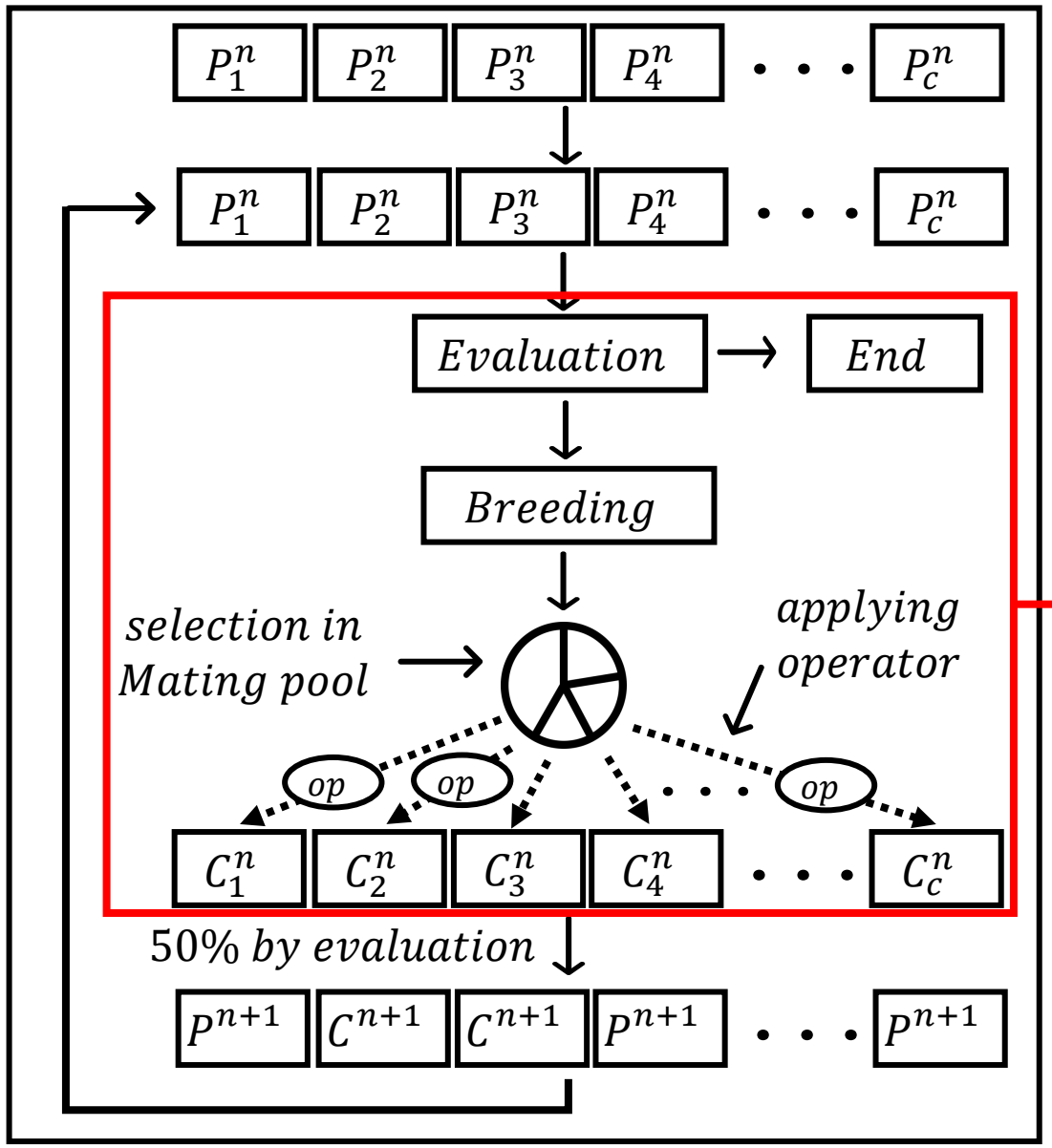
Output

Child data set

Mutation

- 하나의 데이터의 일정 부분을 변형하는 연산자
- 실제 변이가 일어날 확률은 낮기에 낮은 확률(5%)로 발생하도록 설정

SAGA



3. Breeding

Input

Population

Breeding

Output

Child data set

Mutation

- 하나의 데이터의 일정 부분을 변형하는 연산자
- 실제 변이가 일어날 확률은 낮기에 낮은 확률(5%)로 발생하도록 설정

1. Gaps Removal

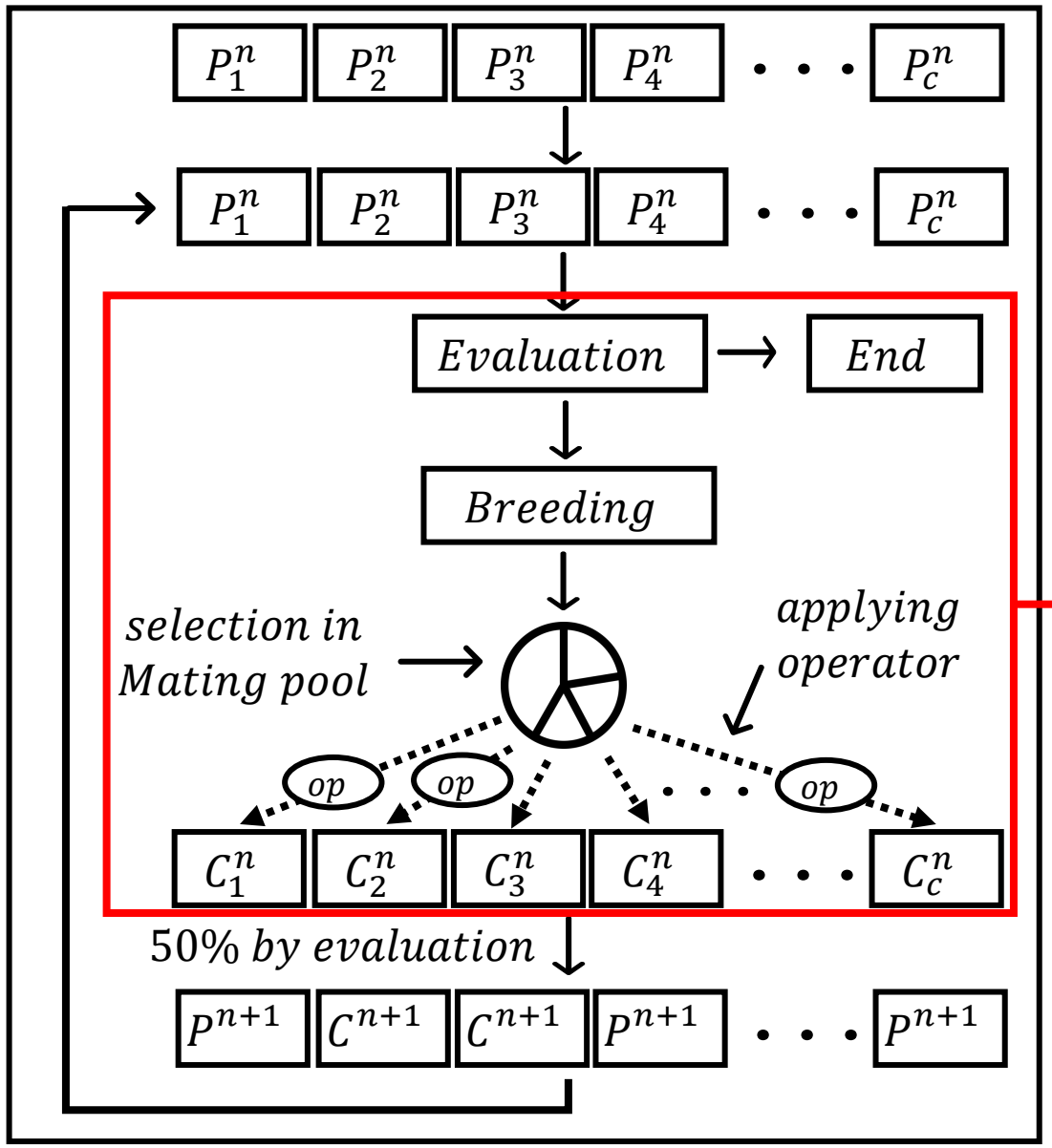
데이터의 sequence에서 **연속된 – 기호**를 찾아낸 뒤 **없애는 mutation**

A T – – G – T T C G

Remove

A T G – T T C G

SAGA



3. Breeding

Input

Population

Breeding

Output

Child data set

Mutation

- 하나의 데이터의 일정 부분을 변형하는 연산자
- 실제 변이가 일어날 확률은 낮기에 낮은 확률(5%)로 발생하도록 설정

2. Gaps Addition

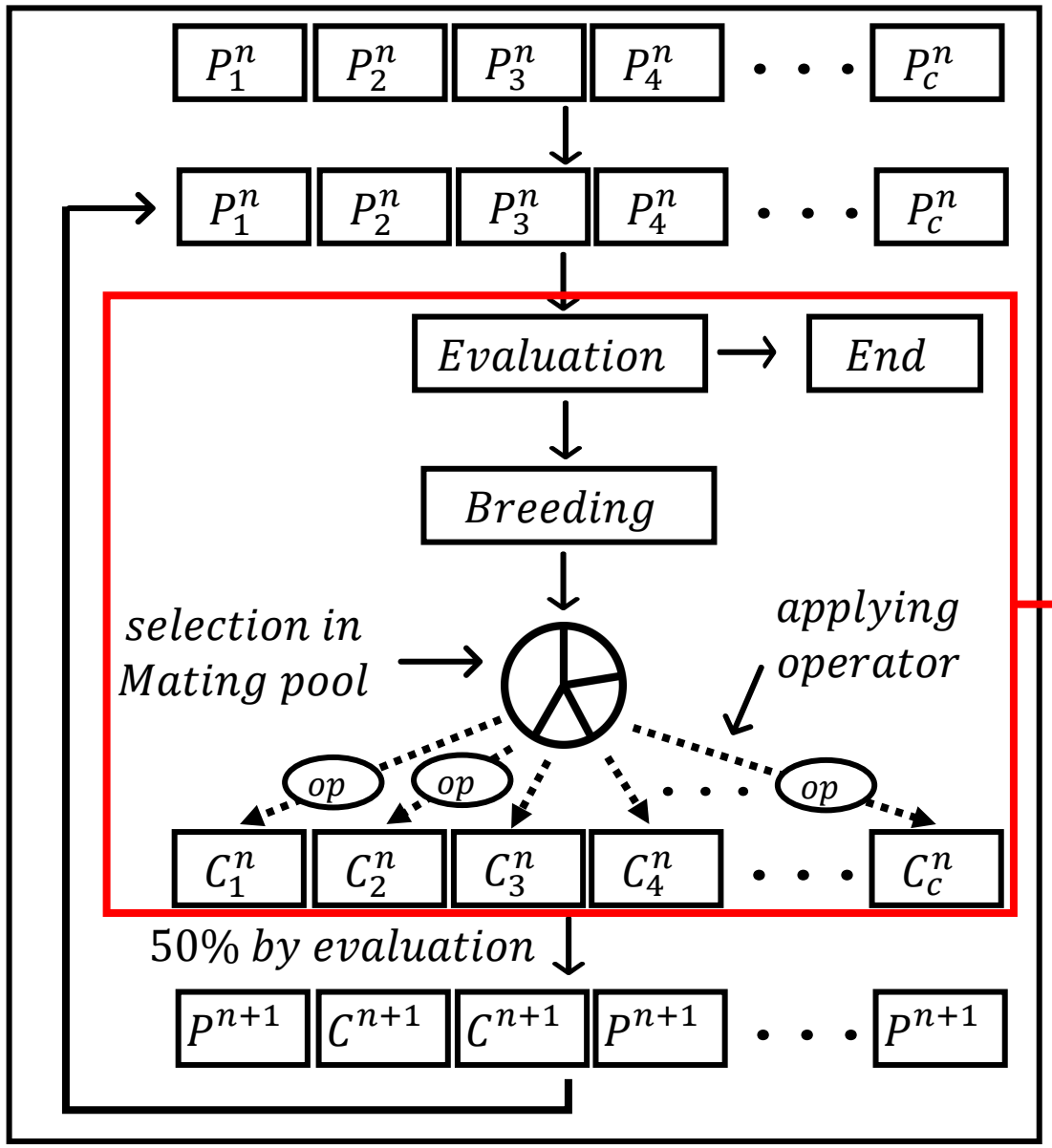
데이터의 sequence에 **랜덤 위치**에 **무작위 개수의 – 기호**를 **추가하는 mutation**

A T G – T T C G

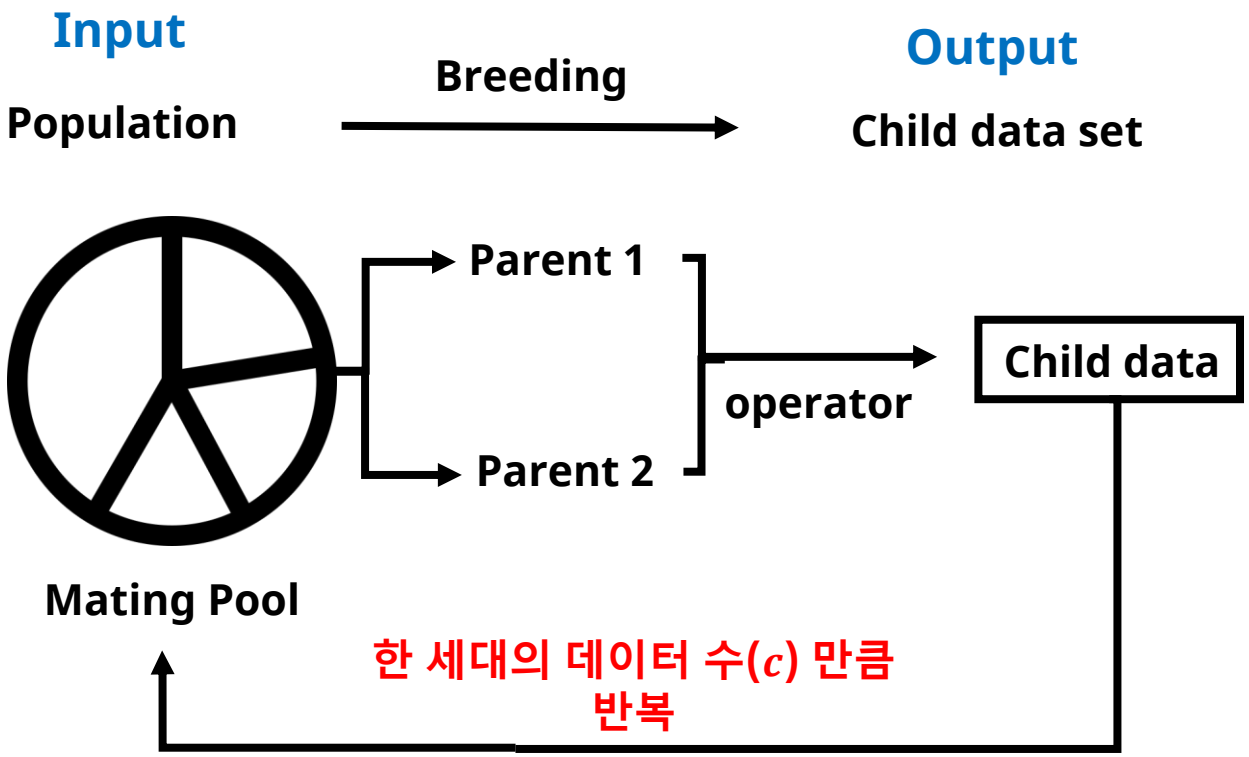
Addition

A T – – G – T T C G

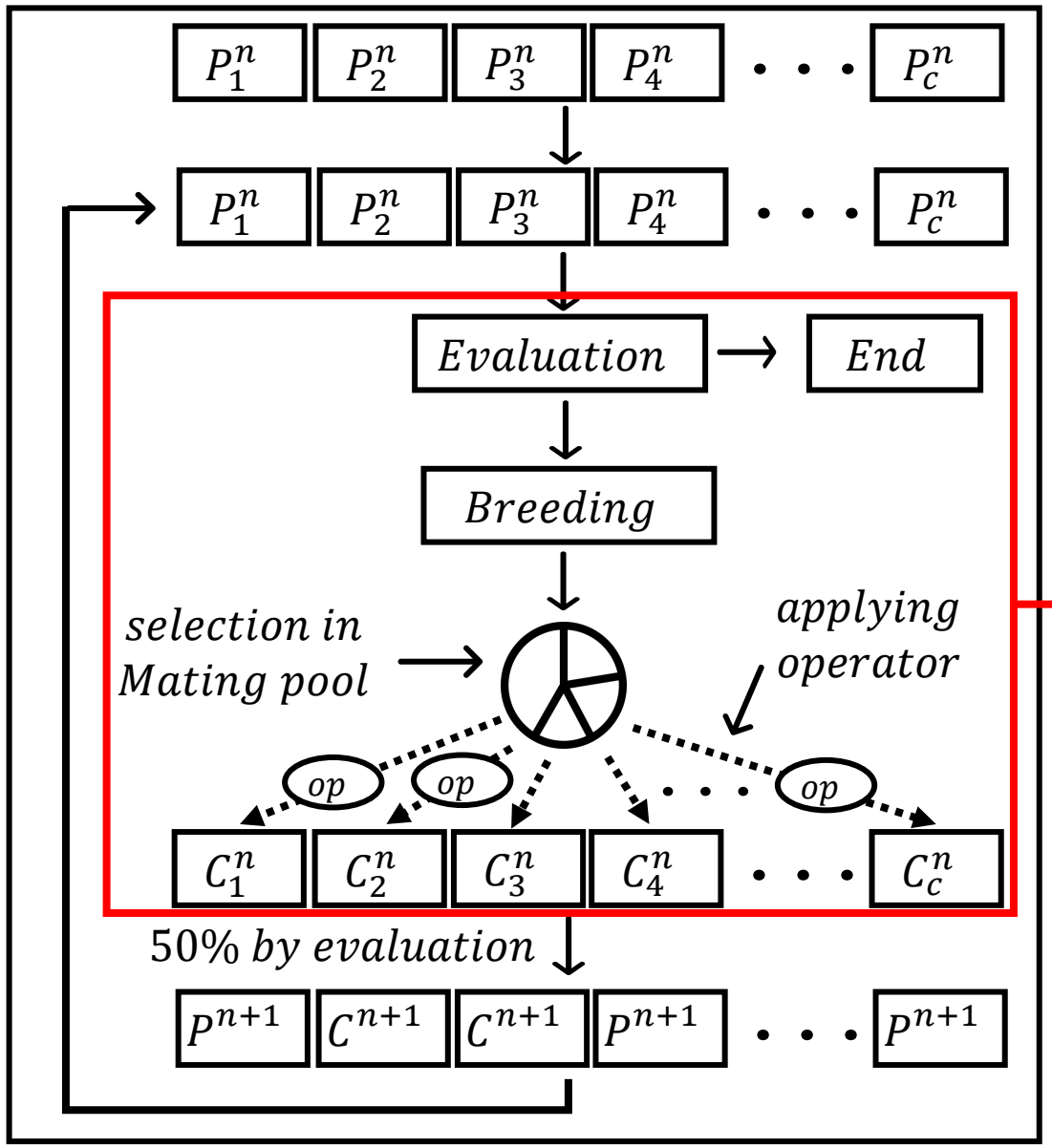
SAGA



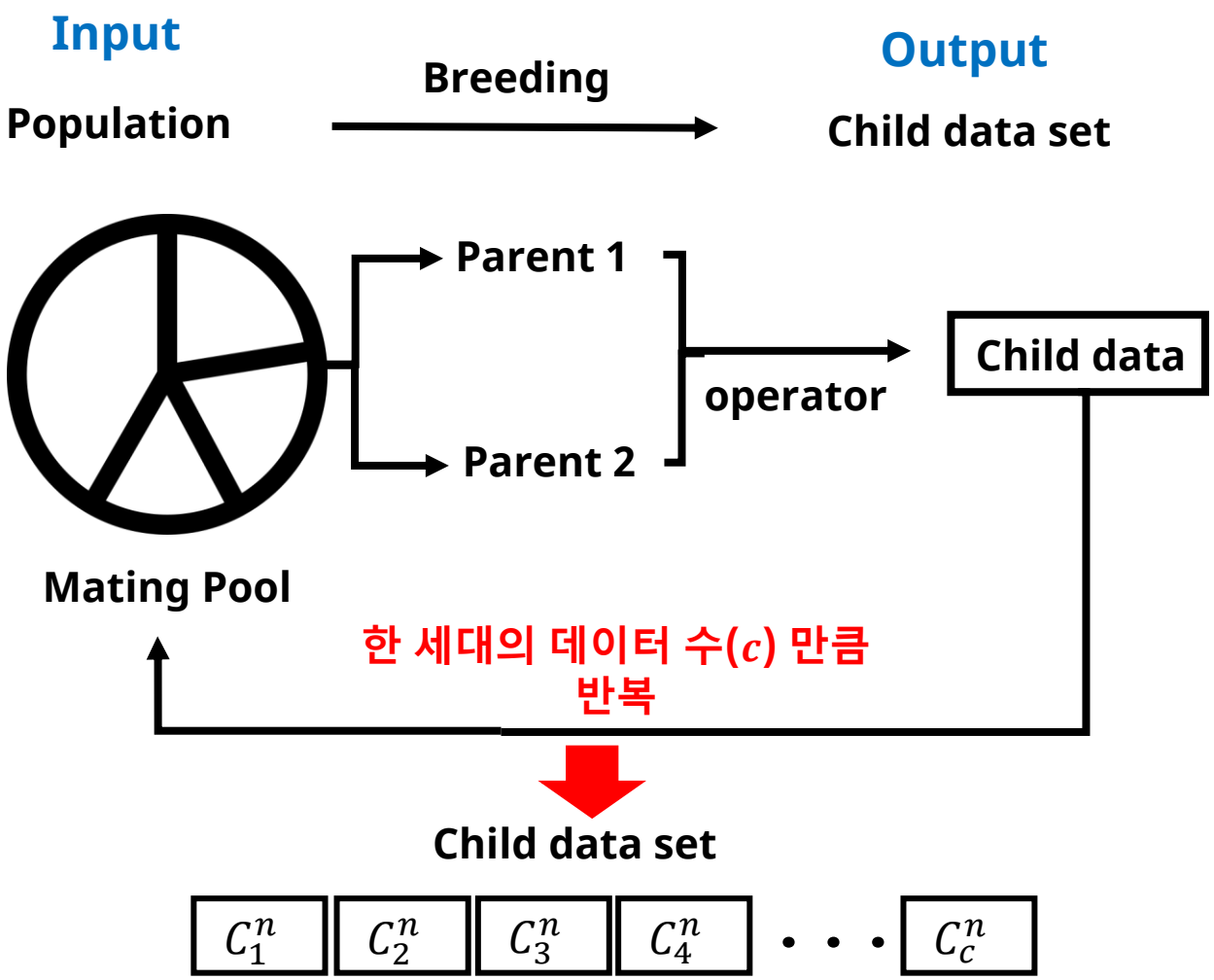
3. Breeding



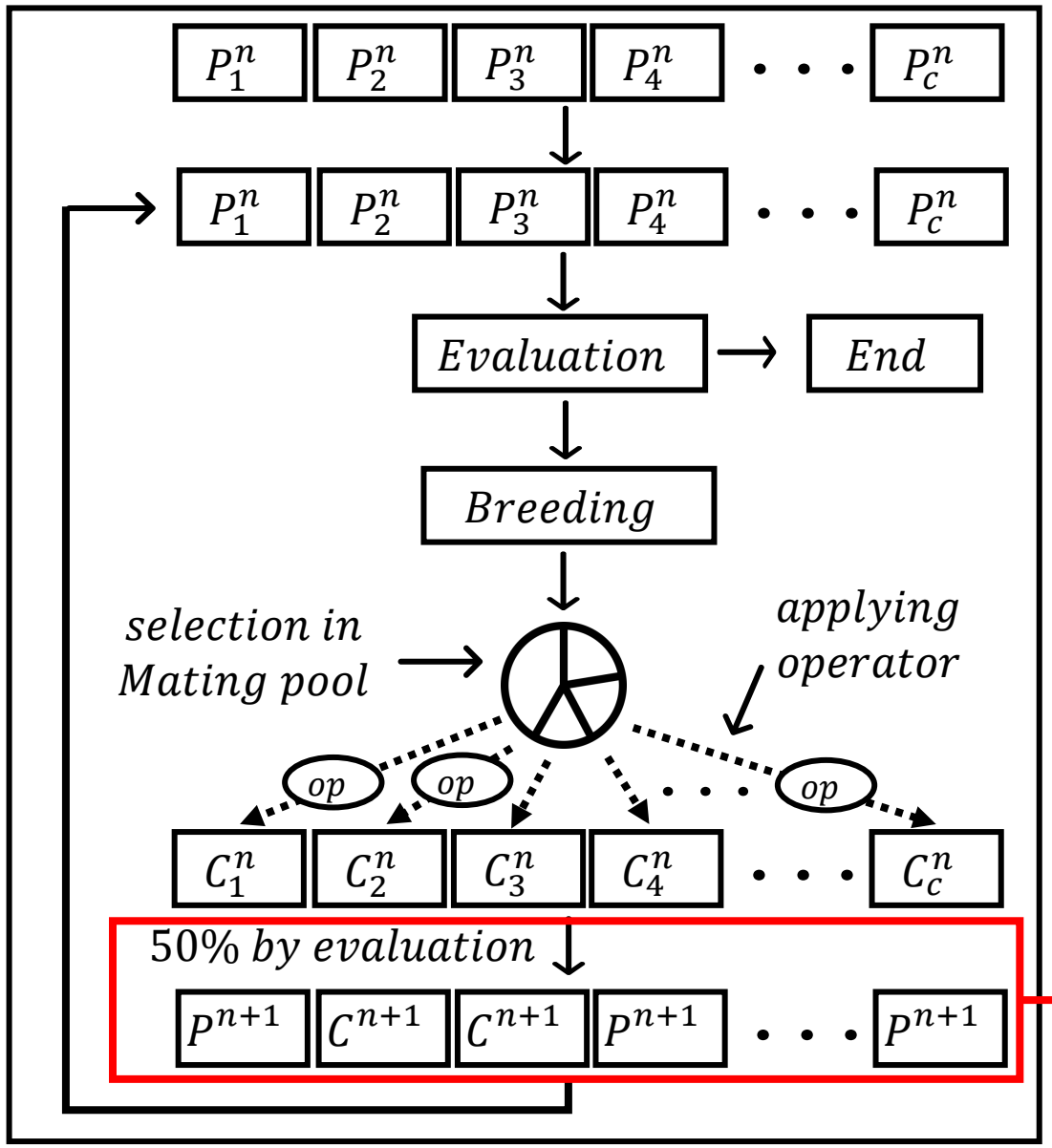
SAGA



3. Breeding



SAGA



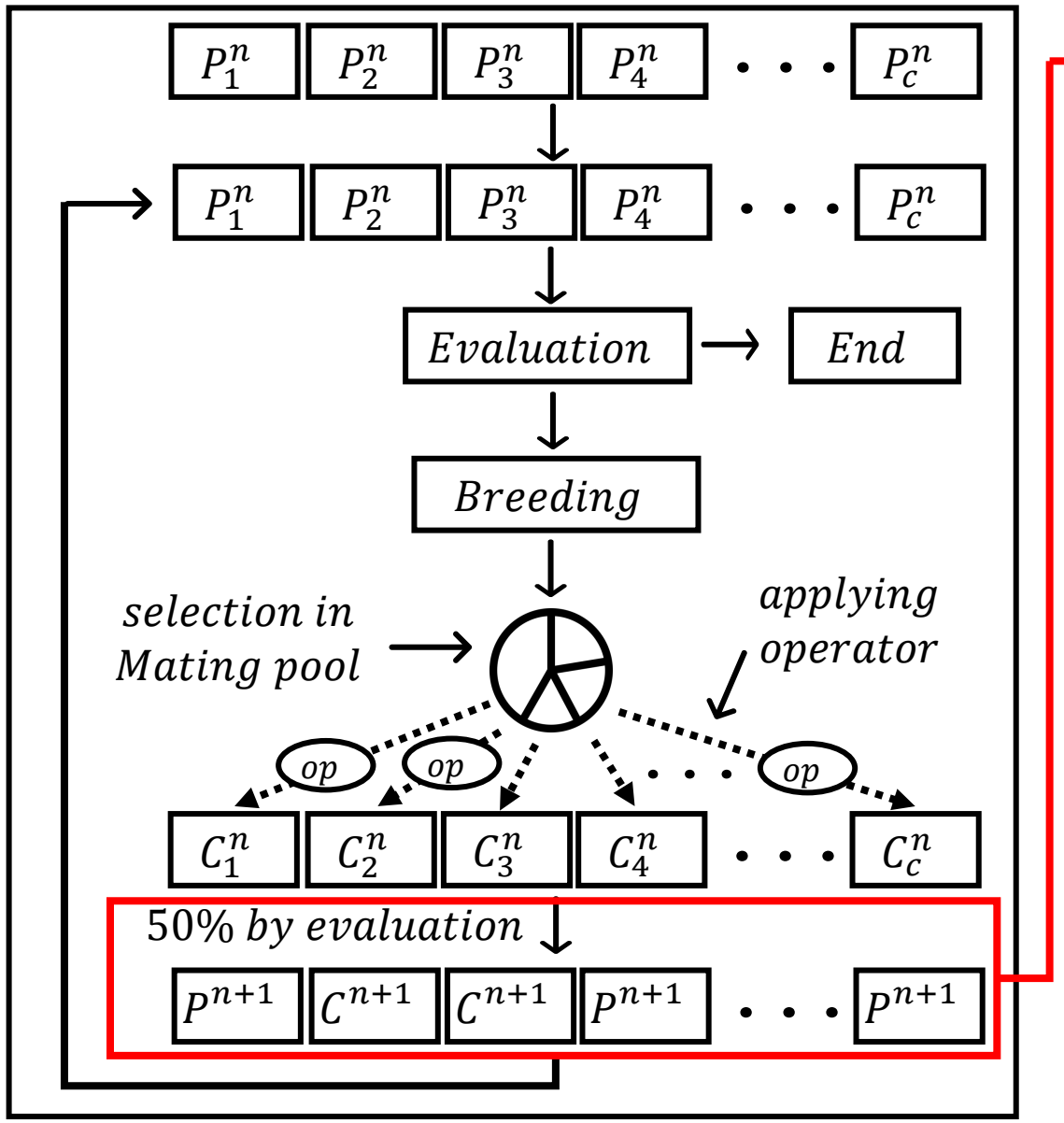
4. Generation

Input
Population(n)
+
Child data set

Generation

Output
Population($n+1$)

SAGA



4. Generation

Input
Population(n)
+
Child data set

Generation

Output
Population($n+1$)

Sort by Evaluation

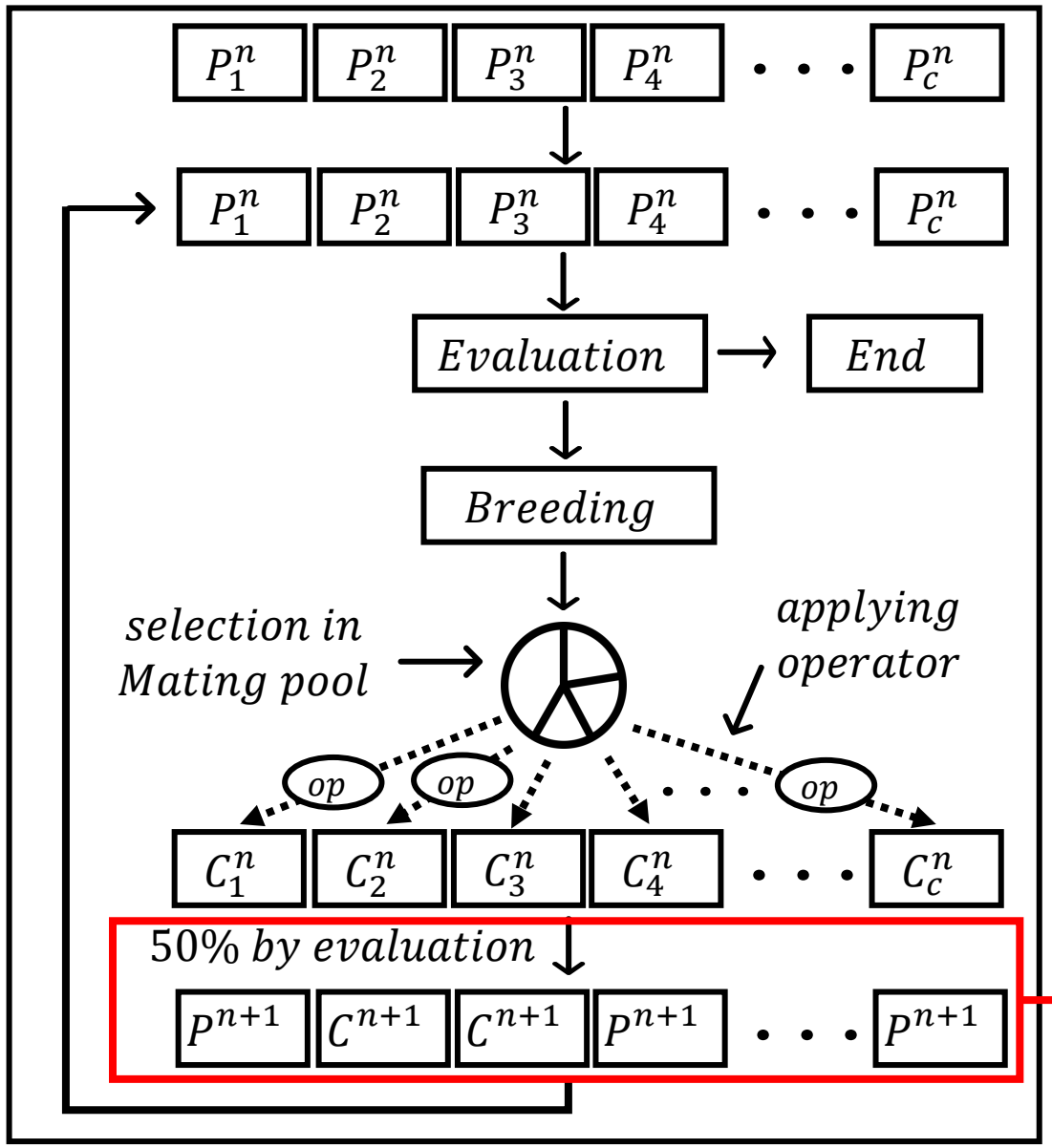
Population

Data
Data
.
.
.
Data

Child data set

Data
Data
.
.
.
Data

SAGA



4. Generation

Input
Population(n)
+
Child data set

Generation

Output
Population($n+1$)

Sort by Evaluation

Population

Data
Data
.
.
.
Data

Combine 50% of high evaluation

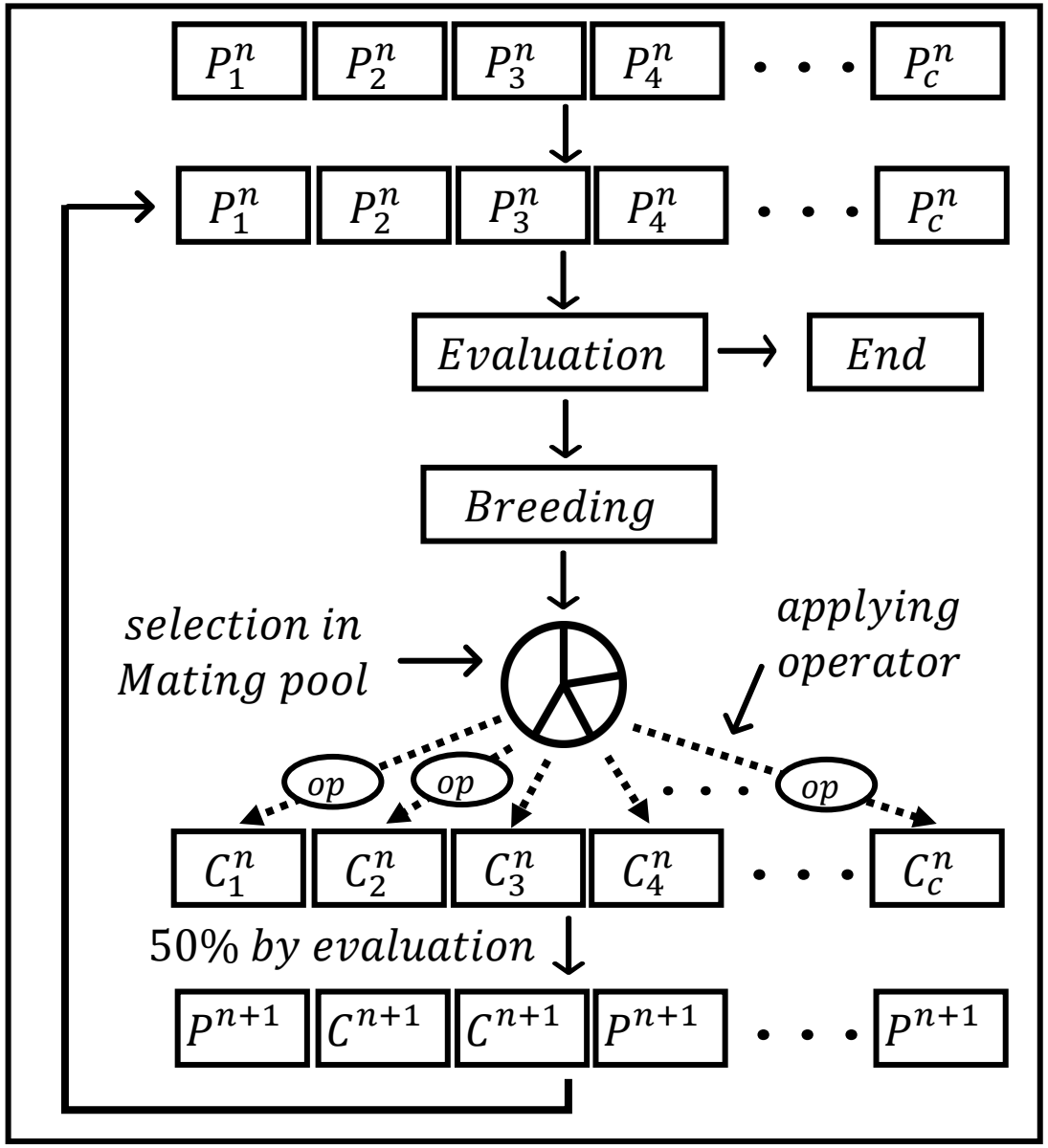
Child data set

Data
Data
.
.
.
Data

New Population

Data
Data
.
.
.
Data

SAGA



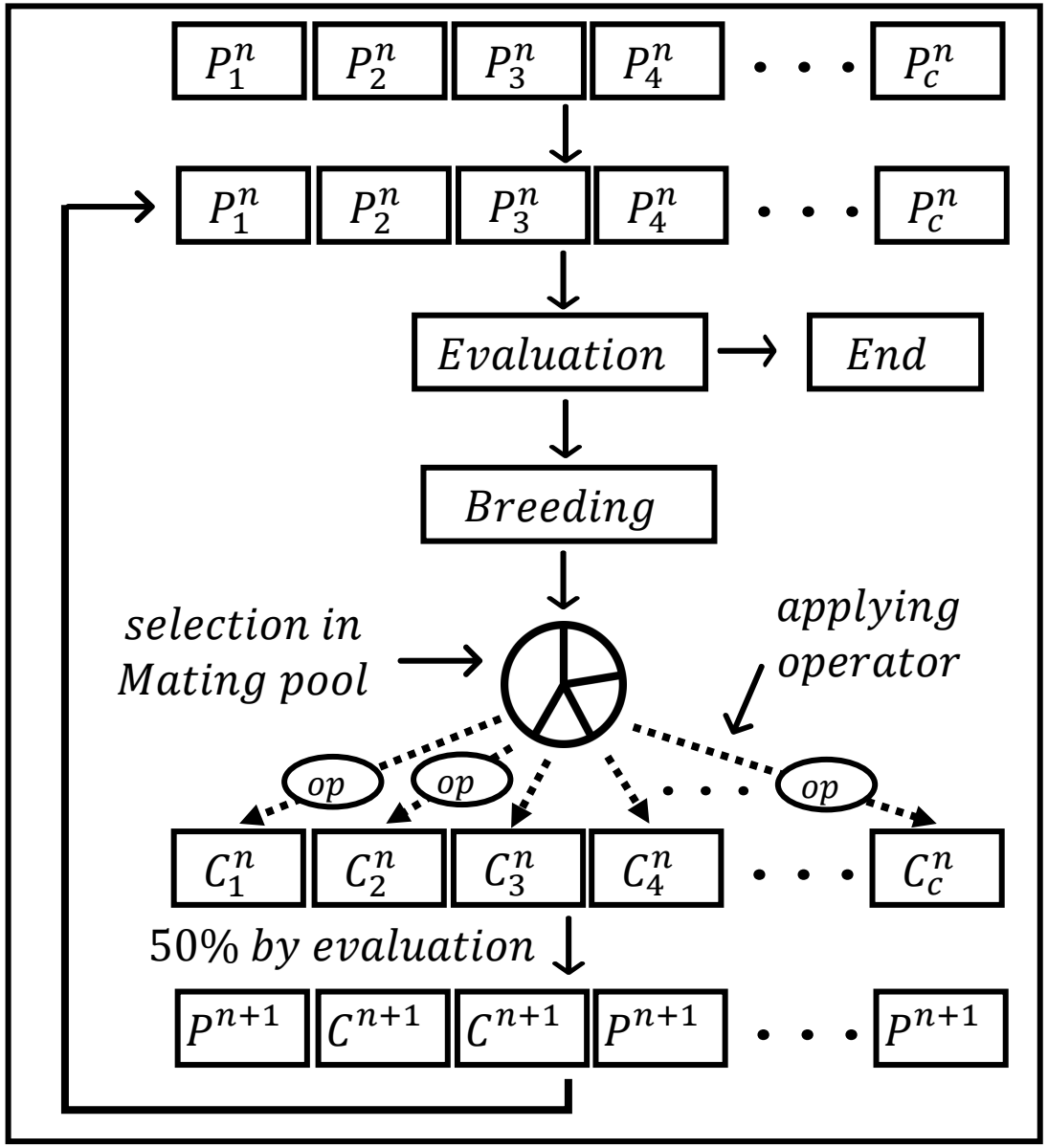
5. Iteration and End

Input
Multiple Sequence → **SAGA** → **Output**
높은 score를 가진 Multiple Alignment

Iteration

- 이후 2~4 과정이 반복된다.
- 반복하는 동안 각 세대 별 **최대 Evaluation**과 그에 대한 **data 저장**
- **Evaluation 값**이 변하지 않거나 **최대 반복 수**에 도달하면 종료

SAGA



5. Iteration and End

Input
Multiple Sequence → **SAGA** → **Output**
높은 score를 가진 Multiple Alignment

Iteration

- 이후 2~4 과정이 반복된다.
- 반복하는 동안 각 세대 별 **최대 Evaluation**과 그에 대한 **data 저장**
- **Evaluation 값**이 변하지 않거나 **최대 반복 수**에 도달하면 종료

End

- 최대 Evaluation 값과 그에 대한 **Multiple Alignment** 출력
- 동일한 Input을 다른 MSA 알고리즘인 **clustal w**를 통해 얻은 **Multiple Alignment** 와 **Evaluation** 값 출력

Experiment & Result

Experiment & Result

Default Setting

- 한 번의 시행을 할 때 사용되는 변수들의 기본 값은 다음과 같다.

Setting	Explanation	Default
Max generation	최대 반복 횟수	**400**
Min generation	최소 반복 횟수	**200**
Chromosome	한 population의 총 데이터 수	**100**
Opening gap	gap cost에서 처음에만 주는 비용	**-1**
Extension gap	gap cost에서 지속적으로 주는 비용	**-0.5**
Mutation rate	데이터에 변이가 발생할 확률	**5%**

Experiment & Result

Input Format

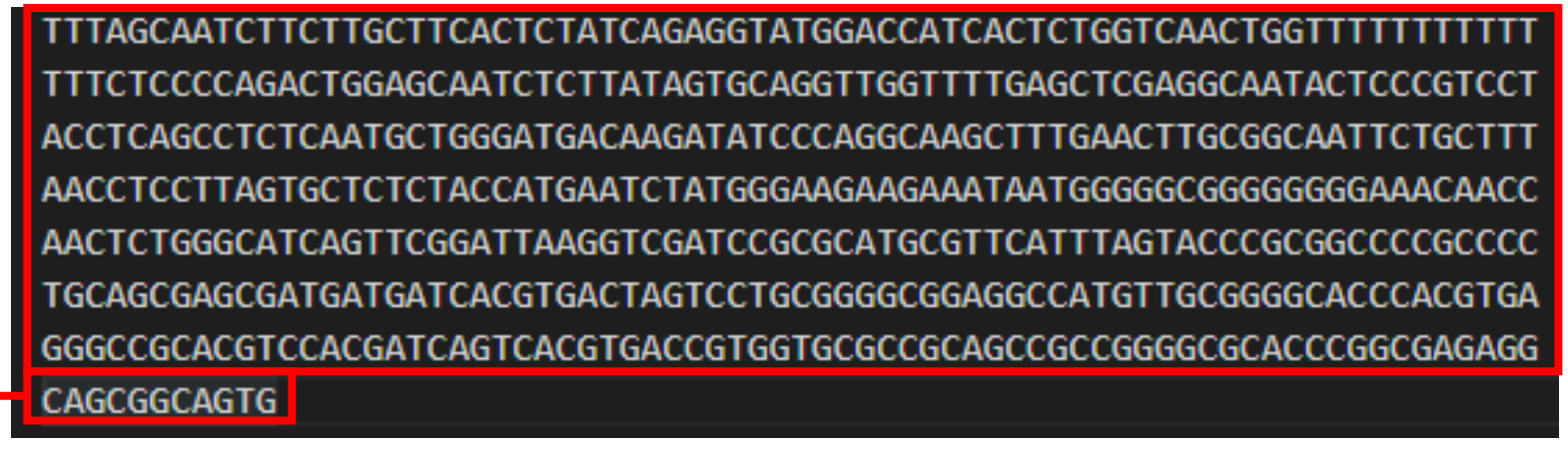
- 입력에 사용된 예시는 다음과 같다.

Example of Input 1

Experiment & Result

Input Format

- 입력에 사용된 예시는 다음과 같다.



Different Length

Example of Input 2

Experiment & Result

Result of Input 1

- 위의 예시 Input 1를 **SAGA**로 **Multiple Alignment**와 그에 대한 **Evaluation** 결과

최대 Evaluation: 1212.0

```
SAGA Result : 1212.0
C-T-A-TCG-A-GTCTTC-C-C-TC-CCT-C-C-TT-CTCTGCC-C-C-C-TCC-GCTCCC--G-CT-GG-AG---
C-C-C-TC-CACCCTA-C-AAG-TGGCCT-AC-AG-G-G-CACAG-GTG----AG-G-C-G-GG-ACTGGAC----
-AGCT-CCT-GCT-T-TG---ATC-GC-C-GGAGA-T--CTG-CA-A--AT-T-CT-GCCCATG-TCGGGGC----
T-GCA-GAGCACTCC-GACGT-G-TCCCA-T-AGTGT-T--TCCA-AA-CT-TGGA-A-A--------GGGCG---
G-GG-GAGG-GCG-G-G-A-G-GATGCGG-A---GG-G-CG-G--AGGTA-TGCAG-ACAACGAGTCAG-------
A-G-T-TT-C-CCCTT-G-AAA---GCCTCAA-AAGTGTC-CAC-GTC-CTCAA-A-A-A---GAA-TGGA-----
A----CCA-A-TTT-AAGAAGCCA-GCCC-C--GTGG-CCACGTCC-CT-T-C-C-C-C----C-ATT-CGC-TC-
C--CT----C-CTCT-G-CGC-C--CCCG-C-AG-GCTCCTCCC-----AG-CTGTGGC-T-G-CCCGGGCC-CC-
C----AGC-C-CC---AGC-C-CTC-CC--AT-TGGT-GGAGGCCCTT-TTGG-AGG-CA--CCC-TAGGGC----
CA-G-G-GAAACTT-TTGCCGT-A-T-----A-A-A-T-AGGGCA-G-A-TCCG-G-GCTT-TA-TTA-T---TTT
```

최종 Multiple Alignment

Experiment & Result

Result of Input 1

- 위의 예시를 **Clustal W**를 이용하여 **Multiple Alignment**와 그에 대한 **Evaluation** 결과

최대 Evaluation: 1856.5

```
clustal_w Result : 1856.5
-AGTTTCCCCTTGAAAG-----------CCTCAAAAGTGTCCACGTCCTCA----AAAAG
---ACCAATTTAAGAAG-----------CCAGCCCCGTGGCCACGTCCCTT----CCCCC
-------TGC--AGAGCACTCCG---ACGTGTCCCATAGTGTTTCCAAACTTGGAAAGGG
CTATCGAGTCTTCCCTCCCTCCT---TCTCTGCCCCCT---------CCG-----CTCCC
-AGCTCCTGCTTTGATCGCCGGA---GATCTGCAAATT---------CTG-----CCCAT
---------CCCTCCACCCTACAAGTGGCCTACAGGGC---------ACAGGTGAGGCGGG
-----CCTCCTCTGCGCCCCCGCAGGCTCCTCCCAGCTGTGGCTGCCCGGG-----CCCC
------------CAGCCCCA-----GCCCTCCCATTGGTGGAGGCCCTTTTGGAGGCAC
---GGGGAGGGCGGGAG-----------GATGCGGAGGGCGGAGGTATGCAGACAACGAG
CAGGGAAACTTTTGCCG-----------TATAAATAGGGCAGATCCGGGCTT--TATTAT
```

최종 Multiple Alignment

Experiment & Result

Result of Total Input cases

- 준비한 각 Input case에 대하여 **10번** 실행한 결과 **평균 Evaluation**과 **Clustal W의 Evaluation** 비교

Test Case	SAGA	Clustal W
1	1047.95	1856.5
2	475.7	786.5
3	602.2	883.0
4	444.3	1185.5
5	375.3	1125.5
6	695.45	1328.0
7	848.9	1179.0
8	1114.2	2229.5
9	871.1	1934.0

Experiment & Result

Result of Total Input cases

- 준비한 각 Input case에 대하여 10번 실행한 결과 **평균 Evaluation**과 **Clustal W의 Evaluation** 비교

Test Case	SAGA		Clustal W
1	1047.95		1856.5
2	475.7		786.5
3	602.2		883.0
4	444.3	<	1185.5
5	375.3		1125.5
6	695.45		1328.0
7	848.9		1179.0
8	1114.2		2229.5
9	871.1		1934.0

Experiment & Result

Result of Total Input cases (Different Gap)

- Input case 1에 대하여 **5번** 실행한 결과 **평균 Evaluation**과 **Clustal W의 Evaluation** 비교
- 단, **Gap 비용**을 다르게 설정

Gap_{open}	Gap_{extend}	SAGA	Clustal W
1	0.5	1,138.1	1856.5
3	0.5	-332.8	1538.5
6	0.5	-2392.9	1061.5
1	0.05	1899.77	2342.94
1	0.2	1448.83	2180.8
1	0.7	633.93	1640.3

Gap_{open}이 커질 수록 **Evaluation** 또한 현저히 낮아지는 것을 확인 가능

Experiment & Result

Result of Total Input cases (Different Gap)

- Input case 1에 대하여 **5번** 실행한 결과 **평균 Evaluation**과 **Clustal W의 Evaluation** 비교
- 단, **Gap 비용**을 다르게 설정

Gap_{open}	Gap_{extend}	SAGA		Clustal W
1	0.5	1,138.1		1856.5
3	0.5	-332.8		1538.5
6	0.5	-2392.9	<	1061.5
1	0.05	1899.77		2342.94
1	0.2	1448.83		2180.8
1	0.7	633.93		1640.3

이전과 같이 **Clustal W**를 뛰어넘지 못함

Experiment & Result

Result of Total Input cases (Different Mutation Prob)

- Input case 1에 대하여 **5번** 실행한 결과 **평균 Evaluation**과 **Clustal W의 Evaluation** 비교
- 단, **Mutation 확률**을 다르게 설정

Mutation Prob	SAGA 평균	SAGA 최대값	Clustal W
5%	1173.7	1418.0	1856.5
10%	1333.3	1491.0	1856.5
20%	1140.8	1443.0	1856.5
2.5%	942.5	1068.0	1856.5
1%	883.2	1072.5	1856.5
0.5%	677.4	812.0	1856.5

Mutation 10% 확률로 설정했을 때 상대적 좋은 성능을 보임

Experiment & Result

Result of Total Input cases (Different Mutation Prob)

- Input case 1에 대하여 **5번** 실행한 결과 **평균 Evaluation**과 **Clustal W의 Evaluation** 비교
- 단, **Mutation 확률**을 다르게 설정

Mutation Prob	SAGA 평균	SAGA 최대값		Clustal W
5%	1173.7	1418.0		1856.5
10%	1333.3	1491.0		1856.5
20%	1140.8	1443.0	<	1856.5
2.5%	942.5	1068.0		1856.5
1%	883.2	1072.5		1856.5
0.5%	677.4	812.0		1856.5

그러나 마찬가지로 **Clustal W**보다는 성능이 좋지 않음

Experiment & Result

Result of Total Input cases (Different Chromosome)

- Input case 1에 대하여 **5번** 실행한 결과 **평균 Evaluation**과 **Clustal W의 Evaluation** 비교
- 단, **Chromosome(데이터의 수)**을 다르게 설정

Chromosome	SAGA 평균	SAGA 최대값	Clustal W
100	966.5	1085.0	1856.5
200	1,151	1233.5	1856.5
300	1,250	1613.5	1856.5
400	1246.8	1457.5	1856.5

대략 **300개의 데이터**를 가질 경우 **높은 Evaluation**을 가지는 경향을 보임

Experiment & Result

Result of Total Input cases (Different Chromosome)

- Input case 1에 대하여 **5번** 실행한 결과 **평균 Evaluation**과 **Clustal W의 Evaluation** 비교
- 단, **Chromosome(데이터의 수)**을 다르게 설정

Chromosome	SAGA 평균	SAGA 최대값		Clustal W
100	966.5	1085.0	<	1856.5
200	1,151	1233.5		1856.5
300	1,250	1613.5		1856.5
400	1246.8	1457.5		1856.5

그러나 마찬가지로 **Clustal W**보다는 성능이 좋지 않음

Experiment & Result

Discussion about Result

- 대부분의 실행에 대하여 좋지 않은 결과가 나온 것을 확인할 수 있었다.
- 여러 방면에서 이에 대한 이유를 고찰해보았을 때 추측된 이유는 다음과 같다.

Experiment & Result

Discussion about Result

- 대부분의 실행에 대하여 좋지 않은 결과가 나온 것을 확인할 수 있었다.
- 여러 방면에서 이에 대한 이유를 고찰해보았을 때 추측된 이유는 다음과 같다.

1. 모델의 단순화

- 현재 제작된 모델은 참고한 논문에 비해 **간단화 된 모델**이다.
- 사용한 연산자의 수가 논문에 비하여 GA에서 의미적인 요소들만 사용됨

  개발 : 사용한 연산자를 **crossover**와 **mutation**으로 한정

  논문 : 총 **22개의 연산자**를 사용
- 모델이 반복하는 동안 연산자를 동일한 순서로 사용함

  개발 : crossover를 적용한 후 mutation을 확률적으로 진행함

  논문 : **Dynamic Scheduling**을 사용하여 효율적으로 연산자 사용

Experiment & Result

Discussion about Result

- 대부분의 실행에 대하여 좋지 않은 결과가 나온 것을 확인할 수 있었다.
- 여러 방면에서 이에 대한 이유를 고찰해보았을 때 추측된 이유는 다음과 같다.

2. 1990년대 논문 참조

- SAGA를 만들기 위해 참조한 논문들은 각각 **1996년, 1997년대**에 발표된 논문들이다.
- 현재는 더 좋은 MSA 기법들이 개발되고 있다.

→ 따라서, **Genetic Algorithm**을 MSA에 사용하는 것은 **적합하지 않을 수 있다**.

Experiment & Result

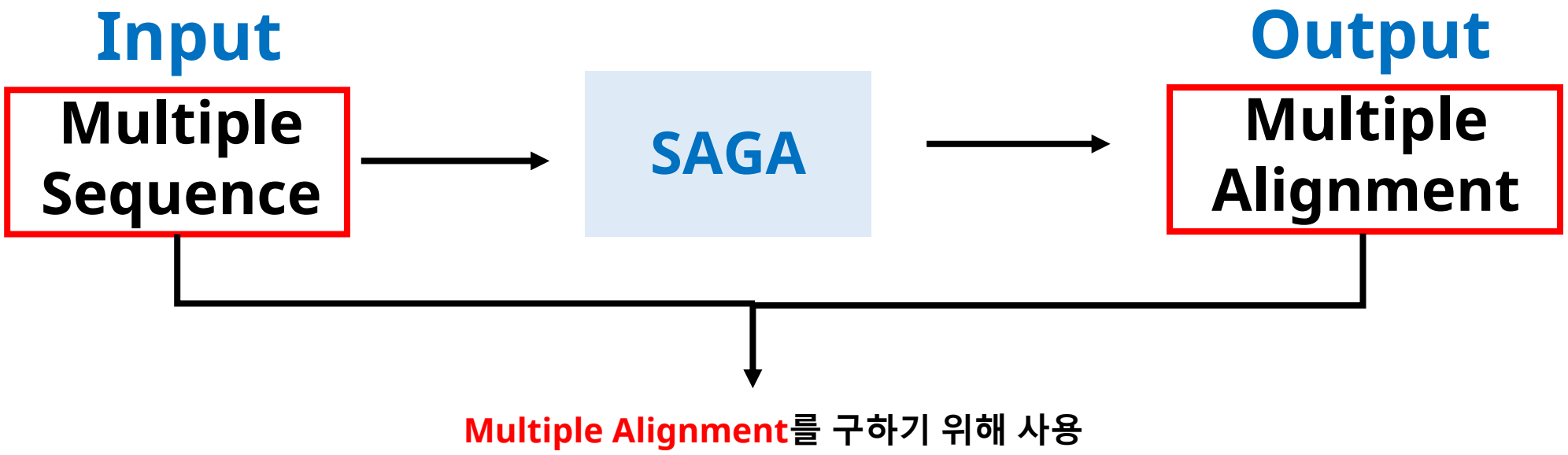
Conclusion

- 결국 위의 실험 결과와 고찰을 통해 개발한 **SAGA는 별로 효과적이지 못한 것**을 알 수 있다.
- 그러나, 앞서 설명했듯 Genetic Algorithm을 MSA에 **다르게 활용할 수 있는 방법**은 존재한다.

Experiment & Result

Conclusion

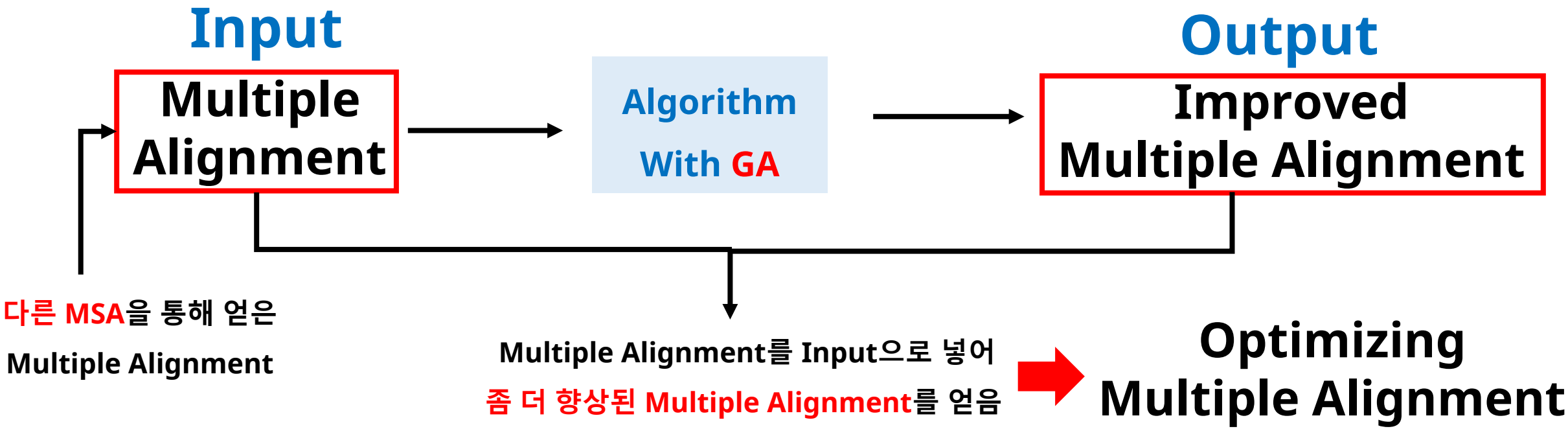
- 결국 위의 실험 결과와 고찰을 통해 개발한 **SAGA는 별로 효과적이지 못한 것**을 알 수 있다.
- 그러나, 앞서 설명했듯 Genetic Algorithm을 MSA에 **다르게 활용할 수 있는 방법**은 존재한다.



Experiment & Result

Conclusion

- 결국 위의 실험 결과와 고찰을 통해 개발한 **SAGA는 별로 효과적이지 못한 것**을 알 수 있다.
- 그러나, 앞서 설명했듯 Genetic Algorithm을 MSA에 **다르게 활용할 수 있는 방법**은 존재한다.

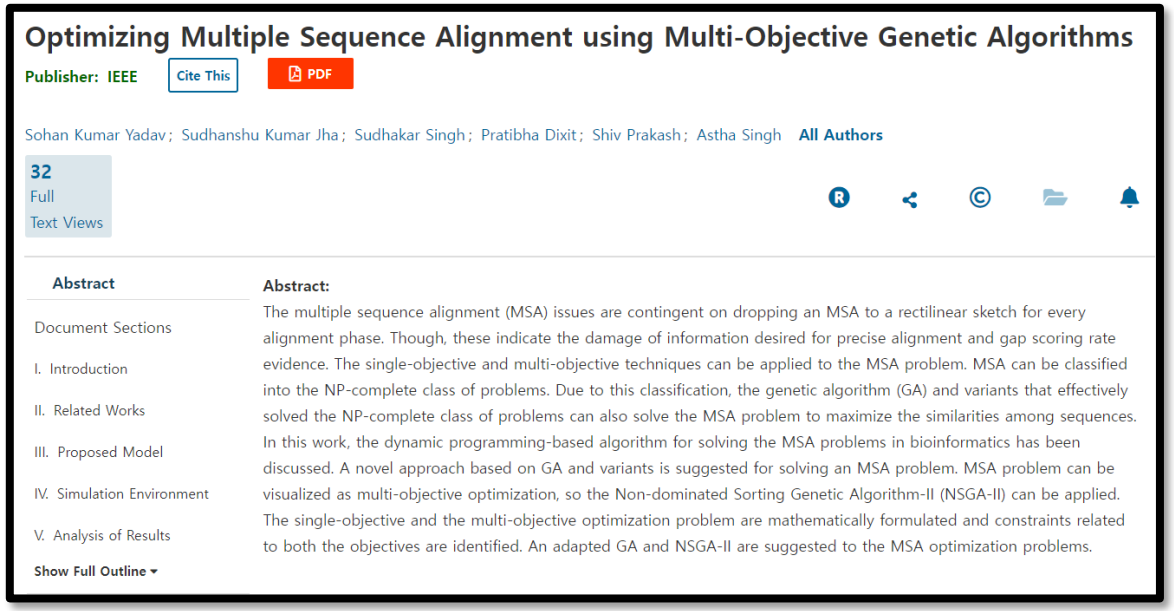


Experiment & Result

Conclusion

- 결국 위의 실험 결과와 고찰을 통해 개발한 **SAGA는 별로 효과적이지 못한 것**을 알 수 있다.
- 그러나, 앞서 설명했듯 Genetic Algorithm을 MSA에 **다르게 활용할 수 있는 방법**은 존재한다.

1. Optimizing Multiple Sequence Alignment using Multi-Objective Genetic Algorithms



Optimizing Multiple Sequence Alignment using Multi-Objective Genetic Algorithms

Publisher: IEEE | Cite This | PDF

Sohan Kumar Yadav; Sudhanshu Kumar Jha; Sudhakar Singh; Pratibha Dixit; Shiv Prakash; Astha Singh All Authors

32 Full Text Views

Abstract

Document Sections

I. Introduction

II. Related Works

III. Proposed Model

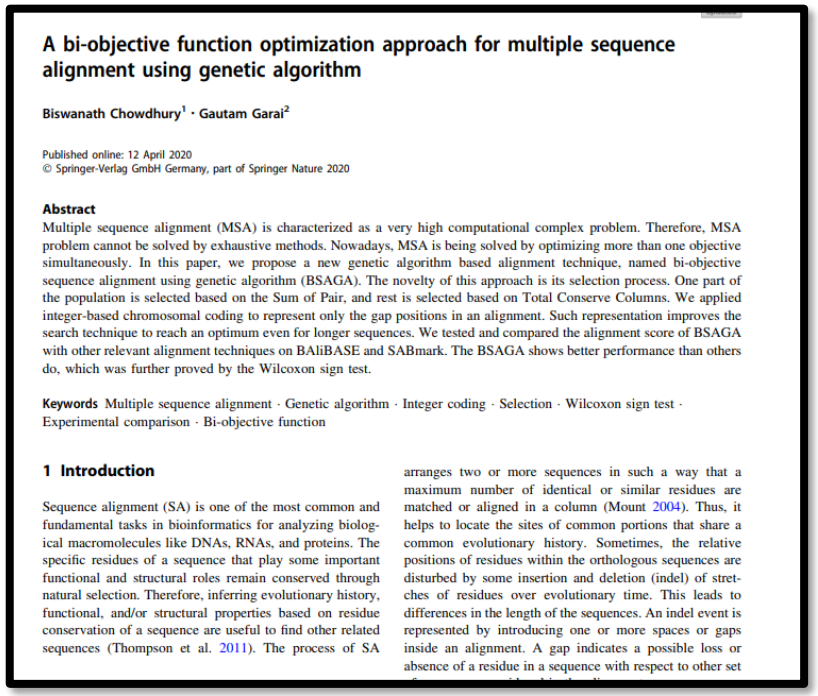
IV. Simulation Environment

V. Analysis of Results

Show Full Outline

Abstract:
The multiple sequence alignment (MSA) issues are contingent on dropping an MSA to a rectilinear sketch for every alignment phase. Though, these indicate the damage of information desired for precise alignment and gap scoring rate evidence. The single-objective and multi-objective techniques can be applied to the MSA problem. MSA can be classified into the NP-complete class of problems. Due to this classification, the genetic algorithm (GA) and variants that effectively solved the NP-complete class of problems can also solve the MSA problem to maximize the similarities among sequences. In this work, the dynamic programming-based algorithm for solving the MSA problems in bioinformatics has been discussed. A novel approach based on GA and variants is suggested for solving an MSA problem. MSA problem can be visualized as multi-objective optimization, so the Non-dominated Sorting Genetic Algorithm-II (NSGA-II) can be applied. The single-objective and the multi-objective optimization problem are mathematically formulated and constraints related to both the objectives are identified. An adapted GA and NSGA-II are suggested to the MSA optimization problems.

2. A bi-objective function optimization approach for multiple sequence alignment using genetic algorithm



A bi-objective function optimization approach for multiple sequence alignment using genetic algorithm

Biswanath Chowdhury[1] · Gautam Garai[2]

Published online: 12 April 2020
© Springer-Verlag GmbH Germany, part of Springer Nature 2020

Abstract
Multiple sequence alignment (MSA) is characterized as a very high computational complex problem. Therefore, MSA problem cannot be solved by exhaustive methods. Nowadays, MSA is being solved by optimizing more than one objective simultaneously. In this paper, we propose a new genetic algorithm based alignment technique, named bi-objective sequence alignment using genetic algorithm (BSAGA). The novelty of this approach is its selection process. One part of the population is selected based on the Sum of Pair, and rest is selected based on Total Conserve Columns. We applied integer-based chromosomal coding to represent only the gap positions in an alignment. Such representation improves the search technique to reach an optimum even for longer sequences. We tested and compared the alignment score of BSAGA with other relevant alignment techniques on BAliBASE and SABmark. The BSAGA shows better performance than others do, which was further proved by the Wilcoxon sign test.

Keywords Multiple sequence alignment · Genetic algorithm · Integer coding · Selection · Wilcoxon sign test · Experimental comparison · Bi-objective function

1 Introduction

Sequence alignment (SA) is one of the most common and fundamental tasks in bioinformatics for analyzing biological macromolecules like DNAs, RNAs, and proteins. The specific residues of a sequence that play some important functional and structural roles remain conserved through natural selection. Therefore, inferring evolutionary history, functional, and/or structural properties based on residue conservation of a sequence are useful to find other related sequences (Thompson et al. 2011). The process of SA arranges two or more sequences in such a way that a maximum number of identical or similar residues are matched or aligned in a column (Mount 2004). Thus, it helps to locate the sites of common portions that share a common evolutionary history. Sometimes, the relative positions of residues within the orthologous sequences are disturbed by some insertion and deletion (indel) of stretches of residues over evolutionary time. This leads to differences in the length of the sequences. An indel event is represented by introducing one or more spaces or gaps inside an alignment. A gap indicates a possible loss or absence of a residue in a sequence with respect to other set

dongguk UNIVERSITY

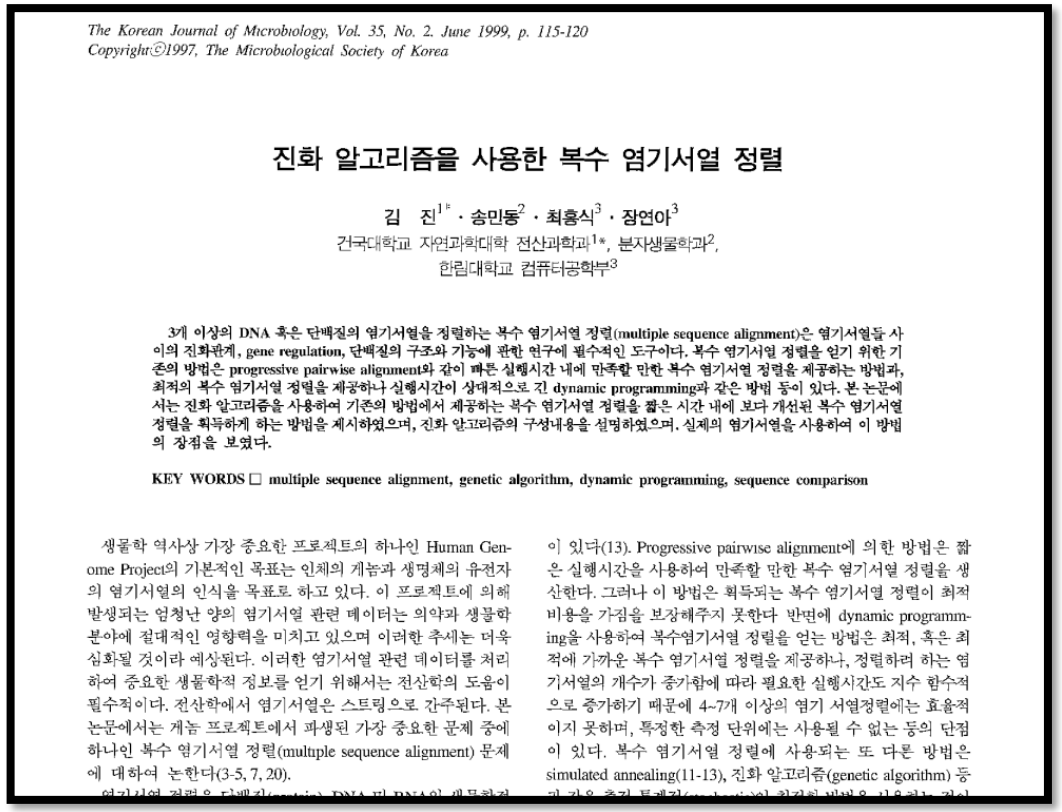
Thank you.


Reference & etc

Reference

1. 웹으로 multi sequence 할 수 있는 곳: https://www.ebi.ac.uk/Tools/msa/clustalo/

2. 진화 알고리즘을 사용한 복수 염기서열 정렬



The Korean Journal of Microbiology, Vol. 35, No. 2. June 1999, p. 115-120

진화 알고리즘을 사용한 복수 염기서열 정렬

김 진[1*] · 송민동[2] · 최흥식[3] · 장연아[3]
건국대학교 자연과학대학 전산과학과[1*], 분자생물학과[2],
한림대학교 컴퓨터공학부[3]

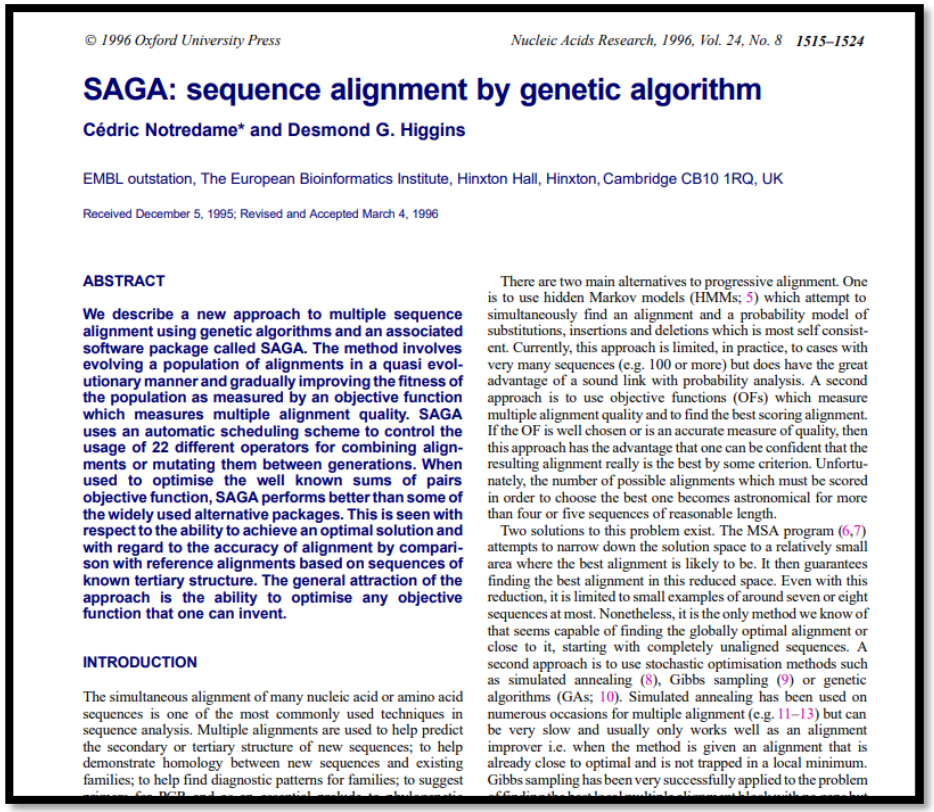
3개 이상의 DNA 혹은 단백질의 염기서열을 정렬하는 복수 염기서열 정렬(multiple sequence alignment)은 염기서열들 사이의 진화관계, gene regulation, 단백질의 구조와 기능에 관한 연구에 필수적인 도구이다. 복수 염기서열 정렬을 얻기 위한 기존의 방법은 progressive pairwise alignment와 같이 빠른 실행시간 내에 만족할 만한 복수 염기서열 정렬을 제공하는 방법과, 최적의 복수 염기서열 정렬을 제공하나 실행시간이 상대적으로 긴 dynamic programming과 같은 방법 등이 있다. 본 논문에서는 진화 알고리즘을 사용하여 기존의 방법에서 제공하는 복수 염기서열 정렬을 짧은 시간 내에 보다 개선된 복수 염기서열 정렬을 획득하게 하는 방법을 제시하였으며, 진화 알고리즘의 구성내용을 설명하였으며, 실제의 염기서열을 사용하여 이 방법의 장점을 보였다.

KEY WORDS □ multiple sequence alignment, genetic algorithm, dynamic programming, sequence comparison

생물학 역사상 가장 중요한 프로젝트의 하나인 Human Genome Project의 기본적인 목표는 인체의 게놈과 생명체의 유전자의 염기서열의 인식을 목표로 하고 있다. 이 프로젝트에 의해 발생되는 엄청난 양의 염기서열 관련 데이터는 의약과 생물학 분야에 절대적인 영향력을 미치고 있으며 이러한 추세는 더욱 심화될 것이라 예상된다. 이러한 염기서열 관련 데이터를 처리하여 중요한 생물학적 정보를 얻기 위해서는 전산학의 도움이 필수적이다. 전산학에서 염기서열은 스트링으로 간주된다. 본 논문에서는 게놈 프로젝트에서 파생된 가장 중요한 문제 중에 하나인 복수 염기서열 정렬(multiple sequence alignment) 문제에 대하여 논한다(3-5, 7, 20).

이 있다(13). Progressive pairwise alignment에 의한 방법은 짧은 실행시간을 사용하여 만족할 만한 복수 염기서열 정렬을 생산한다. 그러나 이 방법은 획득되는 복수 염기서열 정렬이 최적 비용을 가짐을 보장해주지 못한다. 반면에 dynamic programming을 사용하여 복수염기서열 정렬을 얻는 방법은 최적, 혹은 최적에 가까운 복수 염기서열 정렬을 제공하나, 정렬하려 하는 염기서열의 개수가 증가함에 따라 필요한 실행시간도 지수 함수적으로 증가하기 때문에 4~7개 이상의 염기 서열정렬에는 효율적이지 못하며, 특정한 측정 단위에는 사용될 수 없는 등의 단점이 있다. 복수 염기서열 정렬에 사용되는 또 다른 방법은 simulated annealing(11-13), 진화 알고리즘(genetic algorithm) 등

3. SAGA: Sequence Alignment by Genetic Algorithm



Nucleic Acids Research, 1996, Vol. 24, No. 8 **1515–1524**

SAGA: sequence alignment by genetic algorithm

Cédric Notredame* and Desmond G. Higgins

EMBL outstation, The European Bioinformatics Institute, Hinxton Hall, Hinxton, Cambridge CB10 1RQ, UK

Received December 5, 1995; Revised and Accepted March 4, 1996

ABSTRACT

We describe a new approach to multiple sequence alignment using genetic algorithms and an associated software package called SAGA. The method involves evolving a population of alignments in a quasi evolutionary manner and gradually improving the fitness of the population as measured by an objective function which measures multiple alignment quality. SAGA uses an automatic scheduling scheme to control the usage of 22 different operators for combining alignments or mutating them between generations. When used to optimise the well known sums of pairs objective function, SAGA performs better than some of the widely used alternative packages. This is seen with respect to the ability to achieve an optimal solution and with regard to the accuracy of alignment by comparison with reference alignments based on sequences of known tertiary structure. The general attraction of the approach is the ability to optimise any objective function that one can invent.

INTRODUCTION

The simultaneous alignment of many nucleic acid or amino acid sequences is one of the most commonly used techniques in sequence analysis. Multiple alignments are used to help predict the secondary or tertiary structure of new sequences; to help demonstrate homology between new sequences and existing families; to help find diagnostic patterns for families; to suggest

There are two main alternatives to progressive alignment. One is to use hidden Markov models (HMMs; 5) which attempt to simultaneously find an alignment and a probability model of substitutions, insertions and deletions which is most self consistent. Currently, this approach is limited, in practice, to cases with very many sequences (e.g. 100 or more) but does have the great advantage of a sound link with probability analysis. A second approach is to use objective functions (OFs) which measure multiple alignment quality and to find the best scoring alignment. If the OF is well chosen or is an accurate measure of quality, then this approach has the advantage that one can be confident that the resulting alignment really is the best by some criterion. Unfortunately, the number of possible alignments which must be scored in order to choose the best one becomes astronomical for more than four or five sequences of reasonable length.

Two solutions to this problem exist. The MSA program (6,7) attempts to narrow down the solution space to a relatively small area where the best alignment is likely to be. It then guarantees finding the best alignment in this reduced space. Even with this reduction, it is limited to small examples of around seven or eight sequences at most. Nonetheless, it is the only method we know of that seems capable of finding the globally optimal alignment or close to it, starting with completely unaligned sequences. A second approach is to use stochastic optimisation methods such as simulated annealing (8), Gibbs sampling (9) or genetic algorithms (GAs; 10). Simulated annealing has been used on numerous occasions for multiple alignment (e.g. 11–13) but can be very slow and usually only works well as an alignment improver i.e. when the method is given an alignment that is already close to optimal and is not trapped in a local minimum. Gibbs sampling has been very successfully applied to the problem

Data 1


CTATCGAGTCTTCCCTCCCTCCTTCTCTGCCCCCTCCGCTCCCGCTGGAG
CCCTCCACCCTACAAGTGGCCTACAGGGCACAGGTGAGGCGGGACTGGAC
AGCTCCTGCTTTGATCGCCGGAGATCTGCAAATTCTGCCCATGTCGGGGC
TGCAGAGCACTCCGACGTGTCCCATAGTGTTTCCAAACTTGGAAAGGGCG
GGGGAGGGCGGGAGGATGCGGAGGGCGGAGGTATGCAGACAACGAGTCAG
AGTTTCCCCTTGAAAGCCTCAAAAGTGTCCACGTCCTCAAAAAGAATGGA
ACCAATTTAAGAAGCCAGCCCCGTGGCCACGTCCCTTCCCCCATTCGCTC
CCTCCTCTGCGCCCCCGCAGGCTCCTCCCAGCTGTGGCTGCCCGGGCCCC
CAGCCCCAGCCCTCCCATTGGTGGAGGCCCTTTTGGAGGCACCCTAGGGC
CAGGGAAACTTTTGCCGTATAAATAGGGCAGATCCGGGCTTTATTATTTTT

Data 1

```
SAGA Result : 1578.5
CTAT-CGA-GTCT-T-CCCTCCC-T-CC-T-TC-T-C-T-GCCCC-C-TC-C--GC-TC-CCGCT-GGAG--
-----AG--CTCCTG-CT-TTGATCGCCG-G-AGA--TC-TG-CAAATTCTGC-CC-ATGTCGGG-GC----
T-GCA-GAGCACTCC-GACGT-G-TCCCA-T-AGTGT-T--TCCA-AA-CT-TGGA-A-A----GGGCG---
G-G-G-G-A-G-G-G-CG-G-GAG-GAT-GCG-GAGGGCG-G-AGGT-AT-GCAGACA-A-CGAGTCAG---
-AGT-T-T-C-C-CC--TTGA-A-AGC-C-TCAAA-AGT-GTCCACG-TCCT---CA-AA-AA-GAA-TGGA
--AC-CA-A-T-TTA-A-G-A-A-GCCA-GCC--C-CGTGG-CCA-CGTCC-CTTCCCC-CATT-CGC-TC-
C--CT----C-CTCT-G-CGC-C--CCCG-C-AG-GC-TCCTCCC-AG-CTGTGGC-T-G-CCCGGGCCCC-
C----AGC-C-CC---AGC-C-CTCCC--AT-TGGT-GGAGGCCCTT-TTGGAGG-CAC-CCTAGGGC----
CAG-G-GAAACTTTT--GC-C-GT--ATA-A-A-T-AG-GG-CA-GATCCGG-G-CT-T-TATTA-T-T-TT
clustal_w Result : 1856.5
-AGTTTCCCCTTGAAAG----------CCTCAAAAGTGTCCACGTCCTCA----AAAAG
---ACCAATTTAAGAAG----------CCAGCCCCGTGGCCACGTCCCTT----CCCCC
-------TGC--AGAGCACTCCG---ACGTGTCCCATAGTGTTTCCAAACTTGGAAAGGG
CTATCGAGTCTTCCCTCCCTCCT---TCTCTGCCCCCT---------CCG-----CTCCC
-AGCTCCTGCTTTGATCGCCGGA---GATCTGCAAATT---------CTG-----CCCAT
---------CCCTCCACCCTACAAGTGGCCTACAGGGC---------ACAGGTGAGGCGGG
-----CCTCCTCTGCGCCCCCGCAGGCTCCTCCCAGCTGTGGCTGCCCGGG-----CCCC
------------CAGCCCCA-----GCCCTCCCATTGGTGGAGGCCCTTTTGGAGGCAC
---GGGGAGGGCGGGAG----------GATGCGGAGGGCGGAGGTATGCAGACAACGAG
CAGGGAAACTTTTGCCG----------TATAAATAGGGCAGATCCGGGCTT--TATTAT
```

Data 2

```
TTTAGCAATCTTCTTGCTTCACTCTATCAGAGGTATGGACCATCACTCTGGTCAACTGGTTTTTTTTTTTT
TTTCTCCCCAGACTGGAGCAATCTCTTATAGTGCAGGTTGGTTTTGAGCTCGAGGCAATACTCCCGTCCT
ACCTCAGCCTCTCAATGCTGGGATGACAAGATATCCCAGGCAAGCTTTGAACTTGCGGCAATTCTGCTTT
AACCTCCTTAGTGCTCTCTACCATGAATCTATGGGAAGAAGAAATAATGGGGGCGGGGGGGGAAACAACC
AACTCTGGGCATCAGTTCGGATTAAGGTCGATCCGCGCATGCGTTCATTTAGTACCCGCGGCCCCGCCCC
TGCAGCGAGCGATGATGATCACGTGACTAGTCCTGCGGGGCGGAGGCCATGTTGCGGGGCACCCACGTGA
GGGCCGCACGTCCACGATCAGTCACGTGACCGTGGTGCGCCGCAGCCGCCGGGGCGCACCCGGCGAGAGG
CAGCGGCAGTG
```

Data 2

```
SAGA Result : 772.5
TT-T-AGCA-AT-CT--TC-T-TGC-TTCAC-T--CTA-TCA--GA-GGT-A-T-GGA--CCATC-A-CT-CT-GGT-CAA-C
T-GGTTT-TTTTTTTT
T-T--TCT--CC-C-CAGA--CTGGAGCAATC-T-CT-T--ATAGTG--CAGGT-T-G-GT-TTTGAG-CTC-GA-GGCAATA-CTC-CC-GTCCT---
A-C-CTCA-GCCTCTCA-ATGCTGG-G--ATG-A-CA-A-GATA-TC-CCAGGC-A-A-GC-TTTGAA-CTT-GC-GGCAATT-CTG-CT-----TT--
AA--CCT-C-CT-T--AG-TGC-TC--TCTACC-ATGAA-TCT-ATGG-GA-AGA-A-GA-AA-T-A-A-T-G-GGGGCGGG-G-G-GGGA-AACAACC
A-A-CTCTG-G-G-CAT-CAG-TTCGGAT-T-AAG-G-TCG--ATC-CGC-GCATGCGTTC-A-TTTA-GTAC-C-CGCGGCCCCGCCCC---------
T--GCA-GCGA-G-CG-AT-GATG-A-TCACGTGACT-AGTC-C-TGCGG-G-GCGGAGGCCAT--------GT-T-GCGGG-GCA-CCCA--CGT-GA
G-GGCC-GC-ACGTCC-AC-GATC-AGTCACGTGACC--GTG-G-TGCGC-C-GC--A-GCC------GC-C-G-GGGCGCACCCG-GCGA-GAGG---
C-A-G-C-G--------GCAG-TG----------------------------------------------------------------------
clustal_w Result : 786.5
AACCTCCTTAGTGCTCTCTACCATGAATCTATGGGAAGAAGAAATAATGGGGGCG-----
----------TTTCTCCCCAGACTGGAGCAATCTCTTATAGT-GC----AGGTTGGTTTT
----TTT-------------------AGCAATCTTCTTGCTTCACTCTATCAGAGGTATG
--GGGCCGCACGTCCACGATCAGTCACGTGACCGTGGTGCGCCGCA--GCCGCCGGGGCG
----TGCAGCGAGCGATGATGATCACGTGACTAGTCCTGCGGGGC---GGAGGCCATGT-
-AACTCTGGGCATCAGTTCGG-----ATTAAGGTCGA-TCCGCGCA--TGCGTTCATTTA
-ACCTCAGCCTCTCAATGCTG---GGATGACAAGATATCCCAGGCA--AGCTTT------
--------------------------------------------------
```

Data 3

```
GCAGGCTGCCTTTGGTGACTCACCGGGTGAACGGGGGCATTGCGAGGCATCCCCTCCCTGGGTTTGGCTC
CTGCCCACGGGGCTGACAGTAGAAATCACAGGCTGTGAGACAGCTGGAGCCCAGCTCTGCTTGAACCTAT
TTTAGGTCTCTGATCCCCGCTTCCTCTTTAGACTCCCCTAGAGCTCAGCCAGTGCTCAACCTGAGGCTGG
GGGTCTCTGAGGAAGAGTGAGTTGGAGCTGAGGGGTCTGGGGCTGTCCCCTGAGAGAGGGGCCAGAGGCA
GTGTCAAGAGCCGGGCAGTCTGATTGTGGCTCACCCTCCATCACTCCCAGGGCCCCTGGCCCAGCAGCCG
CAGCTCCCAACCACAATATCCTTTGGGGTTTGGCCTACGGAGCTGGGGCGGATGACCCCCAAATAGCCCT
GGCAGATTCCCCCTAGACCCGCCCGCACCATGGTCAGGCATGCCCCTCCTCATCGCTGGCACAGCCCAGA
GGGTATAAACA
```

Data 3

```
SAGA Result : 598.0
--G--CA---GGCTGC-CT-T-TG-GT-G-ACTCACC-GGG-TGA-ACGG-GG-GCATTGCG-AGGCAT-CCCC-T-CC-CTGG-GTTTGG---CTC--
CTGCCCACGGGGCTGA-CA-G-TA-GA-A-A-TCACA-GGC-TGT-GA-GA----CA--GCT--GG-AG-CCCA-G-CT-CTGC---TTGA-ACCTA-T
T-T-T-AGG--TCT---CT-GAT--C-CCCGCT-TCC-TC-T-TT-AGA-C-T-CCCCTA-GAGCTCAG-C-CAGTGCT-CAA-CCT-GAGG-CTGG--
G-G-GTCTCT-G-AG-G-A-AG-AG-TGAG----TTGGA-GC-TG-AGGG-G-T-CTGGG-GCTGT-CC-CCTGAG-AGAGGG-GCCAG-AGGCA----
G-T-G-TCA--AGAGCCG-GGCAG-TC-TG-A--TTGT-GGC-TCA-C-CCTC-CA-T-CA-CTC-CCAGGGCC-C-C-TGG---CCCAG-C--AGCCG
C-A-GCTCC-CA-A-CCA---CAA-TA-TC-C-TTTGG-GGT-TTG-G-CCTA-CG-G--AGCTG----GGGCG-G-A-TGA-CCCCCAAA-TAGCCCT
G-G-C-AGA-TTCC-CCCTAGAC-CCGCCCGCA--CCATG-G-TC-AGG-CATGCCCCTC----CTCAT-CGCT-GGCA-CAGCCC--AGA--------
G-G-G-TA-T-AAA-C---A-----------------------------------------------------------------------------
clustal_w Result : 883.0
GGCAGATTCC----------CCCTAGACCCGCCCGCACCATGGTCAGGCATGCCCCTCCT
---CAG---------CT--CCCAACCA----CAATATCCTTTGGGGTTTGGCCTACGGAG
--------------GT--GTCAAGAGCCGGGCAGTCTGATTGTGGCTCACCCTCCATCA
-------------GGGT--CTCTGAGGAAGAGTGAGTTGGAGCTGAGGGGTCT-GGGGCT
----------TTTAGGTCTCTGATCCC----CGCTTCCTCTTTAGACTCCCCT-AGAGCT
-GCAGGCTGCCTTTGGTGACTCACCGGGTGAACGGGGGCATTGCGAGGCATCC-------
------CTGCCCACGGGG-CTGACAGTAGAAAT-CACAGGCTGTGAGACAGCT-GGAGCC
----------------------GGGTATAAAC-A----------------------
```

Data 4

```
ACTTCTTTTCTGGTGACAGGAAAAGGACTGGGAAGAGGGTTTTCTTCGGGGTGCTTCAGGAAGGCAGAAT
TCTGCCTTAGCGGCTGAGGGATGGGCAGAAAGAAAAAAAAAAGAAAGAAAGAAAAGAAAAACCCAGGAAG
ACTGTGCCTGAGCCAGCTGAGCCAGCTGAGCCAGTTAGCAAGTGCATGGACTGCCGGATTCGATTGGATT
GGATTGCGGATTGCCAAAACAAGGACAGCCACTGGGACAGGGACAGCGGGACAGGGACAGTGACGGCGGA
GGGCAGCAGGGTGGGGGTGGGGTGTCTTTCTCGACTGGTGAGGAAAAGGGAGGGGAGAAGACTTTTTAAA
TGAGTGCAAAATTCTACGCACAAAGAGAGGGGAGGGAGGGACCCAGAGGGGGAAAGGGGTGGAGAAAAGA
AAGAGGAGGAAGAAGTGGGAGGAGGGAGGGCGATGGGATACCGGTTAAAAAGAGGGTGGGCGGGAGCAAC
TAACAGGCTCA
```

Data 4

```
SAGA Result : 491.5
--AC-TT-CT-T---TTCTG-G-TGACAG-GAAA-AGGACTGGGAAGAGGGT-T-T-TCT-TC-GGGGTGC-T-TC-A-GGA-A-G-GC-AG-AAT-
TC--TGCCTT-AGCG--GCTGAGG-G-A--TGG-G-CAGAA-AGA-AAA-A-A-AAA-A-GAA-A-GA-A-A-GAA-AA-GAA-AAACC-CAGGAAG
ACTGTGCCTG-AGCC-AGCTGAGC-C-AGCTGA-GCCAGTT-AGC-AAG-T-G-CAT-G-GAC-T-GC-C-G-GAT-TC-GAT-T-------GGATT
G-GAT-TGCG-G--ATTGCC-AA-A--ACAAG-GAC-AGCC--ACTGGGACAGGGA--CAGC---GGGAC-AGGGA-CA-GT-GAC-G-G-CGGA--
--G-GGCAGC-AG-G--GTGGG-GGT-GGG-GT-GTCT-TTC-T-CG--ACT-GGTGAG-GAAA-AGGGA-GGGGA-GAAGA------CTTTTTAAA
TGAG-TG-CA-A-AATTCTA-C--G-CAC--AAAGA-GAG-GGGAG-GGAGGGA-C-C--CA-GA-GGGG-GA-AAGGG-G-TGG-AGAAA-AGA--
AAGAGG-AGG-AA-GAAGTGGGAGGA-GGGAGG-G-CG-ATGG--GATACC-GGTT-AA--AAAGAGGGT-GGGCGGG-AGC-----------AAC
TAACAGGCTCA-----------------------------------------------------------------------------------
clustal_w Result : 1185.5
-TGAGTGCAAAATTCTACGCACAAAGAGAGGGGAGGGAGGGACCCAGAGGGG------GA
------TCTGCCTTAGCGGCTGAGGGATGGGCAGA------AAGAAAAAAAAAA----GA
AAGAGGAGGAAGAAGTGGGAGGAGGGAGGGCGATGGGATACCGGTTAAAAAG--------
--------------GGGCAGCAGGGTGGGGGTGGGGTGTCTTTCTCGACTGGTGAGGAA
ACTTCTTTTCTGGTGACAGGAAAAGGACTGGGAAGAGGGT-TTTCTTCGGGGTGC---TT
---------ACTGTGCCTGAGCCAGCTGAGCCAGCTGAGC-CAGTTAGCAAGTGCATGGA
--GGATTGCGGATTGCCAAAACAAGGACAGCCACTGGGAC-AGGGACAGCGGGACAGGGA
-----------------TAAC-------------------AGGCTCA------------
```

Data 5

```
CTAATGGCTGTGGCTTGGGAGAATTACCTTTTCCTGGGTTCCCTGGAGGGTTTGTTGTGTGGATAGACTC
CTTTTCTGGTGACAGAAAAAGGACTGGAAAAAGAGATTTTCCTCAGGGTGCTTCAGGAAAGCAGAATTCT
GTAGCCTCAGCAGCAGAGAGATGGGCAGAAAAAGACCCAGGAAGGCTGTGCCTGAGTCAGCTGAGACAGC
TAGCAAGTGCATGGACTGCCAGATTGCCGATTGTGGATCGCCACGACAAGAACAGCCACTTGGACAGGGA
CAGCGGAGGGCAGCAGGGTGGGGGGTGGGGTGTCTTTCTCGACTGGTGAGAAGAAGGGAGGAGAGAGGAC
TTTTATAATGAGTACAAAATTCTAAACACAAAAGAGAGGGGAGGGAGGGACCCAGAGGGGGAAAGGGGTG
GAGAGAGAAAGAGGAGGAAGAAGTGGGAGGAGGGAGGGCGATGGGATACCGGTTAAAAAGAGGGTGGGCG
GGAGCAACCAG
```

Data 5

SAGA Result : 330.5

```
C-TAATGG-CTG-T-GGCT-TGG--GAGA-ATT-A-C-CT-T-T-T-C-C-T-GG-GTTCCCT-GGAG-GGT-T-TGT--TGT-GTG-GA--T-AGACT-C-
CTT-T-T--CTGGTGA-CA--GAA-A-AAG-G-A-CTG-G-A-A-A-A-A-GAGATTT-TCCTC--AGGG--TGC-T-TC-A-GGAA-AGCAGAATTCT---
G-T-A-GC-CTC-A--GCA--GC-AGAGAGATG-G-G-CA-G-A-A-A-A-A-GA----CCCA-GGAA-GGC-TGTGC-CTGA-GTC-AGC-TGAGACAGC-
-TA-G--CA-A-GTG-CAT------GGACTGC-C-AGAT-TGCCGATTG-TGG-ATC-GC-C-A--CG-A--CAA-GAACAGCCACTTGGACAG-GG-A---
C-AGC-G-G--AG-GGCAGC-AG-G--GTGGGG-GGT-GGG-GTGTCT-T-TC-T-CG--ACT-GGTG-AGAAGAAGGGAGGAGA-GAGGA------C----
T-TTTAT-AATGAGTACAAA-ATT-C-T-A-AAC-ACAA-A-A--GA-GAGGGGA-GG-------GAGGG------ACCCAGAGG-G-G-G-AAAG-GGGTG
G-AGA-GAGAAAGAGG-AGG-AA-GAAGTGGGA-GGA-GGGAGGG-CG-A-TG-G--GATACC-GGTT-A-AA-AA--GAGG-GT-G-GG-CG---------
G-G-A-G--CA-A-C-C-A---------G-----------------------------------------------------------------------
```

clustal_w Result : 1112.5

```
CTTTTCTGGTGACAGAAAAAGGACTGGAAAAAGAGATTTTCCTCAGGGTGCTTCAGGA--
CTAAT--------GGCTGTGGCTTGGGAGAATTACCTTTTCCTGGGTTCCCTGGAGGGTT
----CAGCGG--AGGGCAGCAGGGTGGGGGGTGGGGTGTCTTTCTCGACTGGTGAGAAG-
---GTAGCCTCAGCAGCAGAGAGATGGGCAGAAAAAGACCCAGGAAGGCTGTGCCTGAG-
-TTTTATAAT---GAGTACAAAATTCTAAACACAAAAGAGAGGGGAGGGAGGGACC----
----TAGCAA---GTGCATGGACTGCCAGATTGCCGATTGTGGATCGCCACGACAAGAAC
GAGAGAGAAAGAGGAGGAAGAAGTGGGAGGAGGGAGGGCGATGGGATACCGGTTAAAAA-
-----------GGAGCAACCAG----------------------------------
```

Data 6

```
TGGGCGCGGGTCGGCGGCCGCGACCCGGGAGCGGGTTTGCTCAGGAAAAGGCCCGTCGCCCCCCAAACCC
CTCCCTGGCCGGCTCCCTGCCGGGCCTCCCGGGCCTGGTGCTAGGGCACCGCGGGGAGCGCCGAATGGGA
GCGACGACCCGGCCAGCCCGGCACCCGCGGGCGGCAGCCAGGGCGACGCGGAGGCGGCGGGCAGGGGGTG
CGCACTGCAGGGGCGCCAGATTTGGCGGGAGGGGGAGTGTCCAAAGCTCTTTGTTTGATGGCATCTCTGT
TTACAGAGTTTACACTTTAATATCAACCTGTTTCCTCCTCCTCCTTCTCCTCCTCCTCCGTGACCTCCTC
CTACCTCTTTCTCCTGAGAAACTTCGCCCCAGCGGTGCGGAGCGCCCTGCGCAGCCGGGGAGGTACGCAC
GCAGGCAGGCGGCGGGCAGCGGGAGGCGGCAGCCCGGTCGGTCCCCGCGGCTCTCGCGGAGCCCCGCCGC
CCGCCGCGCCA
```

Data 6

SAGA Result : 559.0

```
TGG-G-CG-CG--GGT-CG-GCGGCC-GCGACCCG-GGAGCGGG-T-TTGCTCAGGA-A-A-AG-GCC-C--GT-CGC-CC-C-C-CAAA-C-C-C---------
C-TC-CC-TG-G----C-CG-G-C---TC-C-CT-G-CCG-GG---CCTCC-CG-G-GCCTG-G-TG-CTA-GGGCA-CCG-CG-GG-GAG-CG-CCGAATGGGA
-GCGAC-GACC-C-GGC-CAGCCC-GG-C-A-C-C-CGCGGGC-GGC-A-GCCAGG-G-CG-AC-GCG-G-A-GGC-GGC-GGGCAG-G-G-G--GTG-------
-CGCACTG-CA-G-GGG-C-GCCA--G-AT-T-T-GGCGGGA-GGG-G-G--AGT-GTCC-AA-AGCTC-T-T-TGT-T-TGA-T-GGCAT-CT-CTG--T----
T-TA-CA--GAGTTTACACT-T-T-AATA-T-CA-A-CCT-GT-TTCCTCCTCC-T--CCTT-C-TC-CTC----CT-CCT-CC-GT-GAC-CT-CC---TC---
-CTAC-C-TCT-T-T-CT-C-CT-GAGA-AA-CT-T-C--GC-CC--CAG--CGGTGCGGA-GCGCCCTGC-GCA-GC-CGG-GGAGG-T-A-CGCA--C-----
GC-AG-G--CA-G-G-CG-G-CG-G-GC-AG-CG-G-G-AGG-CG-GCAGCCCGGT-CGGT--C-CCC-GCGGCT--CTCGC-GGAGC-C-C-CGCC-GC-----
C-CGC-C----------G-CGC----------CA----------------------------------------------------------------------
```

clustal_w Result : 1328.0

```
-TTACAGAGTTTACA--------------CTTTAATATCAACCTGTTTCCTCCTCCTCCT
-------------------------CGCACTGCAGGGGCGCCAGATTTGGCGGGAGGGG
GCGACGACCCGGCCAGCC----CGGCACCCGCGGGCGGCAGCC----AGGGCGACGCGGA
--------GCAGGCAGGCGGC-GGGCAGCGGGAGGCGGCAGCCCGGTCGGTCCCCGCGGC
-CTACCTCTTTCTCCTGAGAAACTTCGCCCCAGCGGTGCGGAGCGCCCTGCGCAGCCGGG
-----CTCCCTGGCCGGCTCCCTGCCGGGCCTCCCGGGCCTGGTGCTAGGGCACCGCGGG
----------TGGGCGCGGGTCGGCGGCCGCGACCCGGGAGCGGGTTTGCTCAGGAAAAG
-----------------------CCGCCGCGCCA-------------------------
```

Data 7

```
CCCACTGGCAAGCTTGAGGAGCCAGGCTGCCAGTCGGGAGATTCGGCCCAGTGTTCCCACTGGAGAGGGC
GGCAAGTGCCCGGGCGATCACCTCGCCTGCGTTCGGGAGATATACCTCCGCCCCCGCCCCGCCAGGAGGG
TGAAAAGATGGCCCCAGGAGCCAGCCGGCTGGGACAAGGCGGAGTGAGAGGACAGGCTGGGGCCGGGGGC
GCTGGGCTGTCCCGGGCAGCCCTCCTCCGGGCAAGCCGGAGCAGGGGTGGATTGGGAGCGCTCGGGGCGG
GCCCGCGGTGGCCCGGGGCGGTGGCGCCCGGCCGGAGAGGGTGGGGCGGAGACGCCGCCCTGTACTTCCC
CTTCGCCGCTAGCTCTACAACAGCCTGATTTCCCCGAAATGACGGCACGCAGCCGGCCAATGGGCGCCGC
GCGGCTGTCCGGGGGCGGGGCCGGCCAGGGCTGGGGAATCCCGCTAAGTGTTTGGATTGCTCGGTGGCGC
CGCGCCCTGGC
```

Data 7

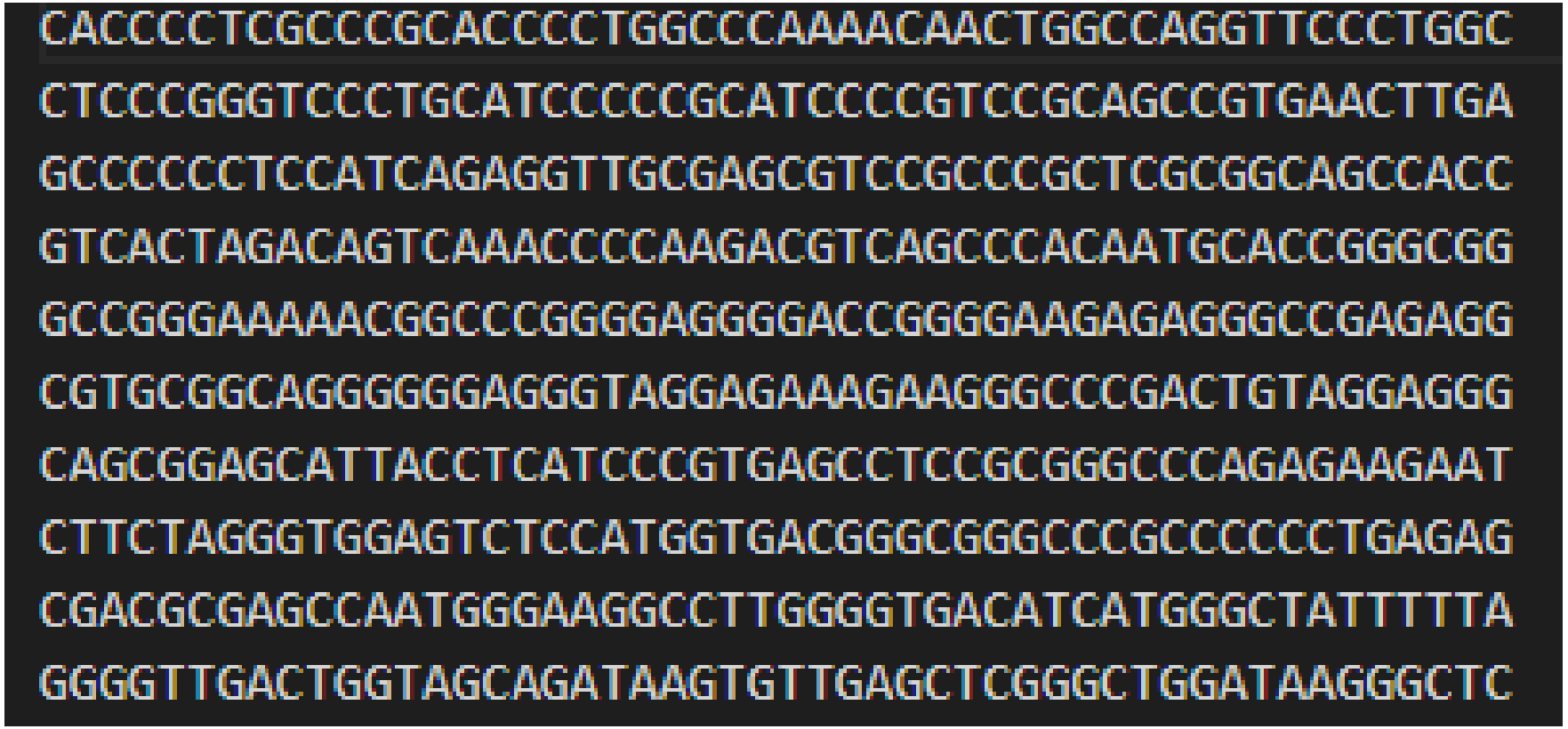
SAGA Result : 863.5

```
C-C-C-AC-T-GG-C-A-AGCT-T-G-AGGAGCCAGG-CT-GCCAGTCGGGAG-AT-TCGG------CC-C-AGTGT-T-C-C---CA--CTGGA-GAGGGC-
GGCA-A--G-TG-CCCGGGCGAT-CACCTCG--CCT--GCG-T-T-CG-GGAGA-TAT-A-C-C-TC-CGC-C-C-C-C-GC-C-CCGC-CA-GGAG-GG---
T-GA-AAAGATGGCCCC-A-G-GAG-CCAG-C-CGG-CTGG-GA--CA-AGGC-GGAGTGAG-AG-G-A-CA-GG-CT-G-G-G----GCC-GGGGGC-----
-GCT-G-GGCTGTCCCGGGC-AG-C-CCTCC-TCCG-GGCA-A-G-CC-GGAGC--AG-G-G-G-TG--GA-T-T-G-G-GA-G--CGCTCG-GG-GCGG---
GCC-CGCG-GTGGCCC-GGGGCGGTGG-CGC-CC-GGC-C-GGA-GAG-GGTG-G-G-GCG-G-AGA-CGC-C--GC-C-C-TG-TACT--TC--C-C-----
-CT-T-C-G-CCG-C-T-A-GCT-C-T-A-CAACA-GCCTGAT-T-T-CC-C-C-GA-AATG-A-CGGC-ACGC-AGC-C--GGCCAA-TGG-G-CGC--CGC
GCG--GCT-GT--CCG-GGGGCGG-GGCCGG-CCAGGG-CTGGG-GAA---TC-C-C-GCT-A-AGT--GT-T-TGG-A-T-TGCT-CG-GTG-GCGC-----
CG--C-------GCCCT-G-GC----------------------------------------------------------------------------
```

clustal_w Result : 1179.0

```
---------------CTTCGCCGCTAGCTCTACAACAGCCTGATTTCCCCGAAATGACG
---CCCAC---------TGGCAAGCTTGAGGAGCCA-GGCTGCCAGTCGGGAGATTCGGC
---GCGGCTGTCCGG----------GGGCGGGG-CC-GGCCAGGGCTGGGGAATCCC-GC
-GCTGGGCTGTCCCGGGCAGCCCTCCTCCGGGC-AA-GCCGGAGCAGGGGTGGATTGGGA
TGAAAAGATGGCCCCAGGAGCCAGCCGGCTGGG-AC-AAGGCGGAGTGAGAGGACAGGCT
GCCCGCGGTGGCCCG--GGGCGGTGGCGCCCGG-CC-GGAGAGGGTGGGGCGGAGACGCC
----------GGCAA--GTGCCCGGGCGATCAC-CT-CGCCTGCGTTCGGGAGATATACC
-------------C--GC-------------------------------------
```

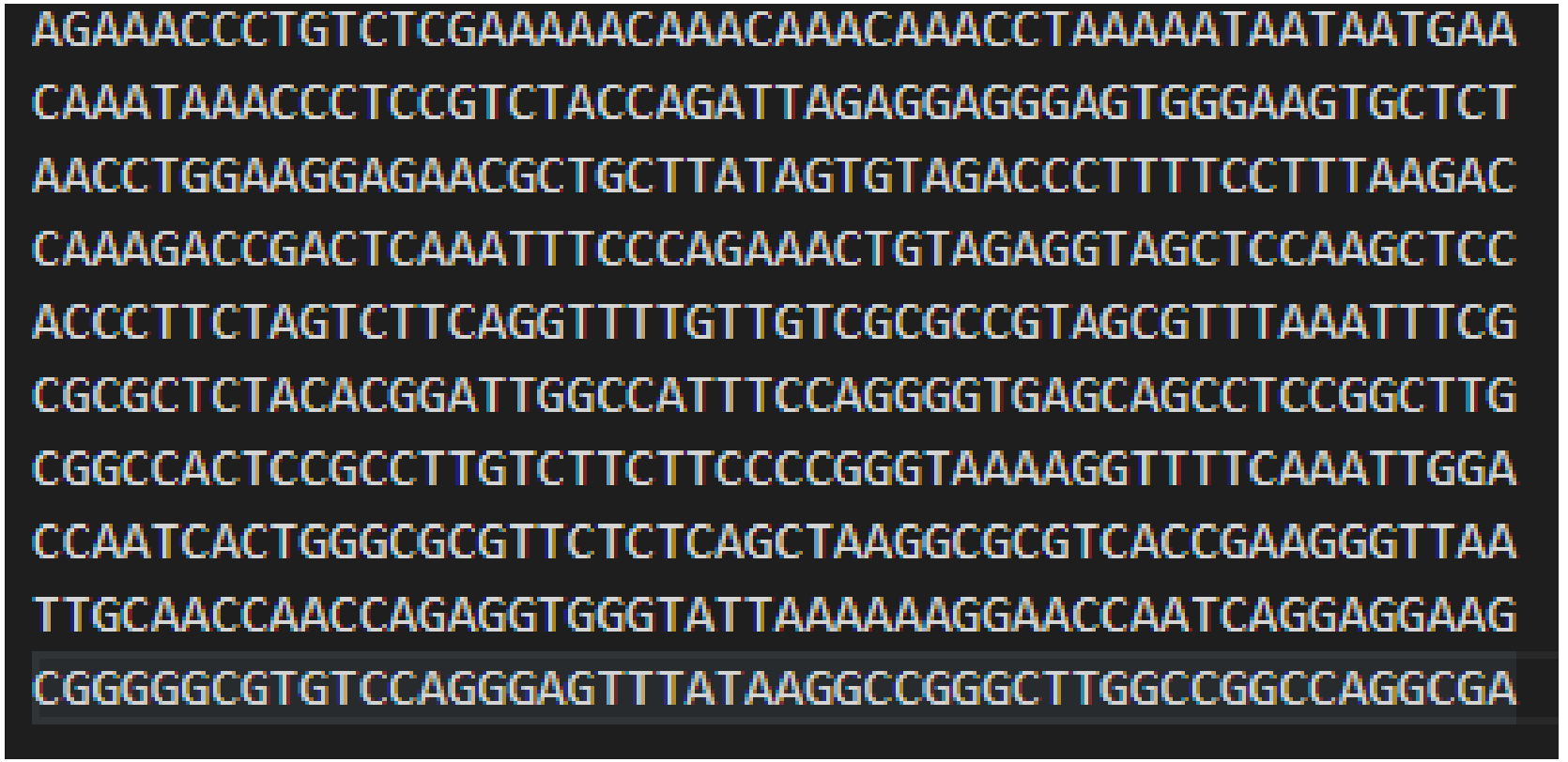
Data 8


CACCCCTCGCCCGCACCCCTGGCCCAAAACAACTGGCCAGGTTCCCTGGC
CTCCCGGGTCCCTGCATCCCCCGCATCCCCGTCCGCAGCCGTGAACTTGA
GCCCCCCCTCCATCAGAGGTTGCGAGCGTCCGCCCGCTCGCGGCAGCCACC
GTCACTAGACAGTCAAACCCCAAGACGTCAGCCCACAATGCACCGGGCGG
GCCGGGAAAAACGGCCCGGGGAGGGGACCGGGGAAGAGAGGGCCGAGAGG
CGTGCGGCAGGGGGGAGGGTAGGAGAAAGAAGGGCCCGACTGTAGGAGGG
CAGCGGAGCATTACCTCATCCCGTGAGCCTCCGCGGGCCCAGAGAAGAAT
CTTCTAGGGTGGAGTCTCCATGGTGACGGGCGGGCCCGCCCCCCTGAGAG
CGACGCGAGCCAATGGGAAGGCCTTGGGGTGACATCATGGGCTATTTTTA
GGGGTTGACTGGTAGCAGATAAGTGTTGAGCTCGGGCTGGATAAGGGCTC

Data 8

```
SAGA Result : 1013.5
C-A-CC-CCTC-GC-CC-G-CAC-C-CCTGGCCCAAAACA-A-CT-GG-CCAG-G-T-T-C-CCT-GGC-------
C-TCCCGGGT-C-C-C-T-GCA-T-CCCC-C-GC-AT-C-C-CCGT-CCGC-A-GC-C-GT-GAA---CT-TGA--
GCC-CCCCTC-CA-T--CAG-A-G-G-TT-GCG--AGC-G-TCC-G-CC-C-GCTCG-CGGC-A-G-CCA-C-C--
G-T-CA--CTA-GA-CA-GTCAA-A--C---CCCAAGACG-T-CA-GC-CCAC-A-A-TGCACCG-GGCGG-----
G---CCGGGA-A-A-A-A--CG-G-CCCG-G-GG-AG-G-G---GA-CCGG-G-GA-A-GAGAG-GGCC-GAGAGG
CGT--GCG-G-CAG-G-GGGG-AGGG--TA-GGA-GA-A-AGA--A-GGGCC-CGAC-TGTAGGAGGG--------
CA-G-CG-GAGC-A-TT--AC--CT-CATC-CCGTGA-GC--CT---CCGCGG-G-CCCA-G-AGAAGAA-T----
C-TTCTAGG---GTG-G-AG-T-CTCCAT-GGTGACGG-GCG-GGCCCG-CC-C---CCCT-GAGAG---------
CGA-CGCGAGCCAA-T-GGGA-AGGC-CTT-GGG-GTG-AC-ATCATGGGCT---ATTT-TT-------A------
G-----GGGTT-G-ACTGG--TA-G-CA--G-ATA-AG-T-GTT--G-AGC-TCGGGCTGGATAAG-G-GC-TC--
clustal_w Result : 2229.5
-------CTTCTAGGG-T-GGAGTCTCCATGGTGACGGGCGGGCCCGCCCCCCTGAGAG-
GCCGGGAAAAACGGCC-C-GGGGAGGGGACCGGGGAAGAGAGGGCCGAGAGG--------
-------CGTGCGGCA-G-GGGGGAGGGTAGGAGAAAGAAGGGCCCGACTGTAGGAGGG-
---GGGGTTGACTGGTAGCAGATAAGTGTTGAGCTCGGGCTGGATAAGGGCTC-------
------CGACGCGA-----GCCAATGGGAAGGCCTTGG--------GGTGACATCATGGG
------CACCCCTCGC-CCGCACCCCTGGCCCAAAACAACTGGCCAGGTTCCCTGGC---
-------GCCCCCCTC-CATCAGAGGTTGCGAGCGTCCGCCCGCTCGCGGCAGCCACC--
-------CTCCCGGGT---CCCTGCATCCCCCGCATCCCCGTCC--GCAGCCGTGAACTT
-------GTCACTAGA-CAGTCAAACCCCAAGACGTCAGCCCAC--AATGCACCGGGCGG
-----------CAGCG-GAGCATTACCTCATCCCGTGAGCCTCC--GCGGGCCCAGAGAA
```

Data 9


AGAAACCCTGTCTCGAAAAACAAACAAACAAACCTAAAAATAATAATGAA
CAAATAAACCCTCCGTCTACCAGATTAGAGGAGGGAGTGGGAAGTGCTCT
AACCTGGAAGGAGAACGCTGCTTATAGTGTAGACCCTTTTCCTTTAAGAC
CAAAGACCGACTCAAATTTCCCAGAAACTGTAGAGGTAGCTCCAAGCTCC
ACCCTTCTAGTCTTCAGGTTTTGTTGTCGCGCCGTAGCGTTTAAATTTCG
CGCGCTCTACACGGATTGGCCATTTCCAGGGGTGAGCAGCCTCCGGCTTG
CGGCCACTCCGCCTTGTCTTCTTCCCCGGGTAAAAGGTTTTCAAATTGGA
CCAATCACTGGGCGCGTTCTCTCAGCTAAGGCGCGTCACCGAAGGGTTAA
TTGCAACCAACCAGAGGTGGGTATTAAAAAAGGAACCAATCAGGAGGAAG
CGGGGGCGTGTCCAGGGAGTTTATAAGGCCGGGCTTGGCCGGCCAGGCGA

Data 9

```
SAGA Result : 870.5
A--GA-AACC--CT--GT-C--TC-GA-A-AA-ACAAACAAAC-A-AACCTAAAAATA-AT--A-ATGAA--
CAAAT-AAACC--CTCC-G--TC-TA-CCAGAT-TAG-A-GG-AG-GG-AG-TG-G-GAAGTGCTCT-----
AA-CCTG-G-AAGG-A-G-A--AC-GC-TGCTTA-TA-------GTGTA-GAC-CCT-TTT-CCTTTAAGAC
CAAAG---ACCGACTCA-A-AT------T-TC-CCAGAAAC-T-GT-AGAGGTAGCTC-C--AAGCTCC---
AC-C-CTTCTAGTCT-TCAGGT-TTT-G---TTGTCGC-GC-C-GTAGC-GT----T-TA-A-A-TTTCG--
CGCG-C-TCTACA-C-GGA-T-T-GG-C-CATTTC--C-AG-GGGT-GA-GCAGCCTC-CGGCTTG------
CGGC-C-A-C---TCC-GC-C-T-TGTCTTC-TT-C-C--CCGGG-TA-AAAGGT-T-T-TCAAA-TTGGA-
C---CAAT-C-ACT--G-GGC-GC-GTT-CTCTC-AG-C-TAAGGC-G-CG---TCAC-CGAAGGGTTAA--
T-TGCAAC-CAACC-AGAGGT-GG-GTA--T-TA-AA-A--AAGGA-A-C--CAA-TC-AG-GAGG--AAG-
CGGG-G-G-CGTGTCCAGG-G-A--GT-TTA-TA-A-G-GCCGGGCTT-----GGCC-G-G-CC-AGGCGA-
clustal_w Result : 1934.0
--------AC--CCTTCTAGTCTTCAGGTTTTG-TTGTCGCGCCGTAGCGTTTAAATTTC
---------A--ACCTGGAAGGAGAACGCTGCT--TATAGTGTAGACCCTTTTCCTTTAA
--CAAATAAA--CCCTCCGTCTACCAGATTAGA-GGAGGGAGTGGGAAGTGCTCT-----
--CAAAGACC--GACTCAAATTTCCCAGAAACT-GTAGAGGTAGCTCCAAGCTCC-----
CGGGGGCGTGTCCAGGGAGTTTATAAGGCCGGGCTTGGCCGGCCAGGCG---------A-
-----CCAAT--CACTGGGCGCGTTCTCTCAG--CTAAGGCGCGTCACCGAAGGGTT-AA
-CGCGCTCTA--CACGGATTGGCCATTTCCAGGGGTGAGCAGCCTCCGGCTTG-------
---CGGCCAC--TCCGCCTTGTCTTCTTCCCCGGGTAAAAGGTTTTCAAATTGGA-----
-----AGAAA--CCCTGTCTCGAAAAACAAAC--AAA-CAAACCTAAAAATAATAATGAA
TTGCAACCAA--CCAGAGGTGGGTATTAA---------AAAAGGAACCAATCAGGAGGAA
```