(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2024-0107037
(43) 공개일자 2024년07월08일

(51) 국제특허분류(Int. Cl.)
G16B 50/00 (2019.01) *G16B 30/10* (2019.01)
G16B 35/10 (2019.01) *H03M 13/01* (2006.01)
(52) CPC특허분류
G16B 50/00 (2019.02)
G16B 30/10 (2019.02)
(21) 출원번호 10-2023-0196044
(22) 출원일자 2023년12월29일
심사청구일자 2023년12월29일
(30) 우선권주장
1020220189267 2022년12월29일 대한민국(KR)

(71) 출원인
울산과학기술원
울산광역시 울주군 언양읍 유니스트길 50
(72) 발명자
최환혁
울산광역시 울주군 언양읍 유니스트길 50
이찬영
울산광역시 울주군 언양읍 유니스트길 50
(뒷면에 계속)
(74) 대리인
최우성

전체 청구항 수 : 총 13 항

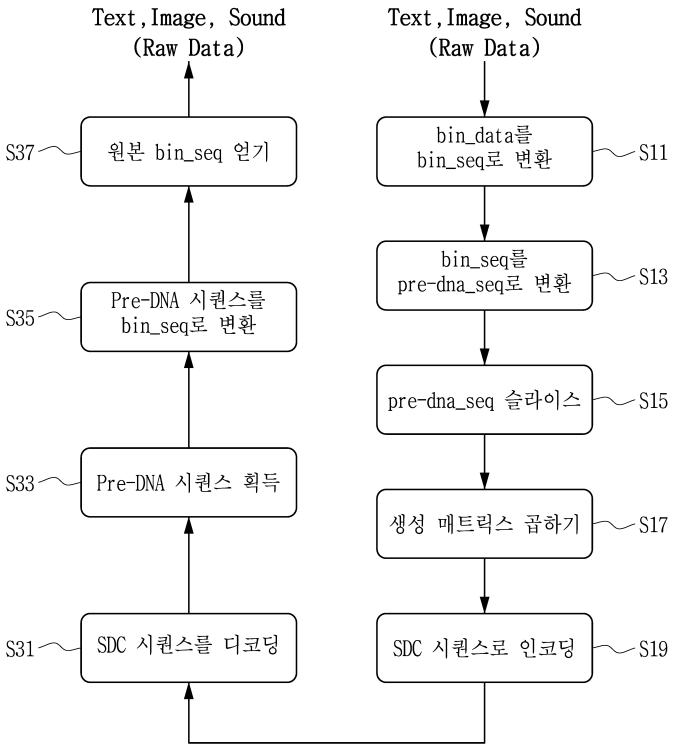
(54) 발명의 명칭 **단일 결실 오류 정정이 가능한 자기 동기화 DNA 코드의 설계 방법**

(57) 요 약

단일 결실 오류 정정이 가능한 자기 동기화 DNA 코드의 설계 방법은, 입력된 이진 데이터의 길이를 나타내는 헤더 데이터, 패딩 데이터 및 입력된 이진 데이터를 차례로 연결하여, 총 길이가 2n(여기서, n은 자연수)의 배수인 이진 시퀀스를 생성하는 단계; 생성된 이진 시퀀스의 각 두 심볼을 DNA 심볼인 A, C, G 및 T로 매핑된 이진 맵을

(뒷면에 계속)

대 표 도 - 도1



사용하여 n의 배수의 길이를 갖는 사전(pre) DNA 시퀀스로 변환하는 단계; 사전 DNA 시퀀스를 n의 길이를 갖는 블록들로 슬라이싱하는 단계; 슬라이싱된 사전 DNA 시퀀스의 각 블록에 미리 설정된 생성 행렬을 곱하여 2n 길이를 갖는 DNA 부호어(codeword)들이 연결된 DNA 부호어 시퀀스를 생성하는 단계; 및 각 DNA 부호어를 2n+2의 길이를 갖는 단일 결실 정정(이하, SDC) 코드로 변환하되, 각 DNA 부호어의 첫 번째 심볼이 이전 부호어의 마지막 심볼과 동일한 경우, 해당 DNA 부호어는 상보적 SDC 코드로 변환하여 구분자가 없는 자기 동기화 SDC 시퀀스로 변환하는 단계;를 포함한다. 이에 따라, 구분자가 없는 연속적인 데이터에 결실 오류가 발생할 경우, 구분자의 위치를 알아내고(self-synchronizing), 결실 오류를 정정(single-deletion-correcting)할 수 있다.

(52) CPC특허분류
G16B 35/10 (2019.02)
H03M 13/01 (2013.01)

(72) 발명자
최한솔
울산광역시 울주군 언양읍 유니스트길 50

권세용
울산광역시 울주군 언양읍 유니스트길 50

이 발명을 지원한 국가연구개발사업

과제고유번호	1711183176
과제번호	2022R1C1C2011689
부처명	과학기술정보통신부
과제관리(전문)기관명	한국연구재단
연구사업명	개인기초연구(과기정통부)
연구과제명	DNA data-storage를 위한 오류정정 DNA 부호 연구
기 여 율	80/100
과제수행기관명	강원대학교
연구기간	2023.03.01 ~ 2024.02.29

이 발명을 지원한 국가연구개발사업

과제고유번호	1345350578
과제번호	2019R1I1A1A01057755
부처명	교육부
과제관리(전문)기관명	한국연구재단
연구사업명	이공학학술연구기반구축
연구과제명	DNA 컴퓨팅 및 DNA 기반 정보 저장을 위한 부호 이론 연구
기 여 율	20/100
과제수행기관명	울산과학기술원
연구기간	2022.03.01 ~ 2022.11.30

명 세 서

청구범위

청구항 1

입력된 이진 데이터의 길이를 나타내는 헤더 데이터, 패딩 데이터 및 입력된 이진 데이터를 차례로 연결하여, 총 길이가 2n(여기서, n은 자연수)의 배수인 이진 시퀀스를 생성하는 단계;

생성된 이진 시퀀스의 각 두 심볼을 DNA 심볼인 A, C, G 및 T로 매핑된 이진 맵을 사용하여 n의 배수의 길이를 갖는 사전(pre) DNA 시퀀스로 변환하는 단계;

사전 DNA 시퀀스를 n의 길이를 갖는 블록들로 슬라이싱하는 단계;

슬라이싱된 사전 DNA 시퀀스의 각 블록에 미리 설정된 생성 행렬을 곱하여 2n 길이를 갖는 DNA 부호어(codeword)들이 연결된 DNA 부호어 시퀀스를 생성하는 단계; 및

각 DNA 부호어를 2n+2의 길이를 갖는 단일 결실 정정(이하, SDC) 코드로 변환하되, 각 DNA 부호어의 첫 번째 심볼이 이전 부호어의 마지막 심볼과 동일한 경우, 해당 DNA 부호어는 상보적 SDC 코드로 변환하여 구분자가 없는 자기 동기화 SDC 시퀀스로 변환하는 단계;를 포함하는, 단일 결실 오류 정정이 가능한 자기 동기화 DNA 코드의 설계 방법.

청구항 2

제1항에 있어서, 자기 동기화 SDC 시퀀스로 변환하는 단계는,

각 DNA 부호어의 n 번째 심볼과 n+1 번째 사이에 n 번째 심볼과 n+1 번째 심볼과 다른 동일 심볼(ϕ) 2개를 삽입하여 SDC 코드로 변환하는 단계;

각 DNA 부호어의 첫 번째 심볼이 이전 부호어의 마지막 심볼과 동일한 경우, 길이가 2n+2이고 모든 성분이 1인 벡터를 추가하여 변환된 SDC 코드의 상보체로 변환하는 단계; 및

각 DNA 부호어가 변환된 각 SDC 코드를 구분자 없이 연결하여 자기 동기화 SDC 시퀀스를 생성하는 단계;를 포함하는, 단일 결실 오류 정정이 가능한 자기 동기화 DNA 코드의 설계 방법.

청구항 3

제1항에 있어서, DNA 부호어 시퀀스를 생성하는 단계는,

사전 DNA 시퀀스의 각 블록을 맵(f)에 의해 행렬 $2 \times 2n$로 변환하는 단계;

변환된 행렬을 생성 행렬과 곱하여 부호어로 인코딩하는 단계;

인코딩된 부호어에 맵(f)의 역맵(f^{-1})를 적용하여 DNA 부호어로 변환하는 단계; 및

사전 DNA 시퀀스의 각 블록에 대해 획득한 DNA 부호어를 연결하여 DNA 부호어 시퀀스를 획득하는 단계;를 포함하는, 단일 결실 오류 정정이 가능한 자기 동기화 DNA 코드의 설계 방법.

청구항 4

제1항에 있어서,

구분자가 없는 자기 동기화 SDC 시퀀스를 입력된 이진 데이터로 디코딩하는 단계;를 더 포함하는, 단일 결실 오류 정정이 가능한 자기 동기화 DNA 코드의 설계 방법.

청구항 5

제4항에 있어서, 입력된 이진 데이터로 디코딩하는 단계는,

DNA 부호어 시퀀스를 획득하기 위해 구분자가 없는 자기 동기화 SDC 시퀀스를 디코딩하는 단계;

각 DNA 부호어로부터 사전(pre) DNA 시퀀스를 획득하는 단계;

사전 DNA 시퀀스에 이진 맵의 역맵을 적용하여 이진 시퀀스를 획득하는 단계; 및

획득한 이진 시퀀스의 끝에서부터 이진 데이터의 해당하는 자릿수의 데이터를 분리하여 입력된 이진 데이터를 획득하는 단계;를 포함하는, 단일 결실 오류 정정이 가능한 자기 동기화 DNA 코드의 설계 방법.

청구항 6

제5항에 있어서, 자기 동기화 SDC 시퀀스를 디코딩하는 단계는,

자기 동기화 SDC 시퀀스의 각 SDC 코드에 단일 결실 오류가 발생한 경우 DNA 부호어의 오류를 정정하는 단계;를 포함하는, 단일 결실 오류 정정이 가능한 자기 동기화 DNA 코드의 설계 방법.

청구항 7

제6항에 있어서, 자기 동기화 SDC 시퀀스를 디코딩하는 단계는,

자기 동기화 SDC 시퀀스 중 남은 심볼수(m, 여기서, m은 자연수)가 0 보다 큰 동안 반복을 수행하는, 단일 결실 오류 정정이 가능한 자기 동기화 DNA 코드의 설계 방법.

청구항 8

제7항에 있어서, 자기 동기화 SDC 시퀀스를 디코딩하는 단계는,

단일 결실 오류가 있는 경우, 남은 심볼수(m)를 m - (2n + 1)로 업데이트하고,

단일 결실 오류가 없는 경우, 남은 심볼수(m)를 m - (2n + 2)로 업데이트하는 단계;를 포함하는, 단일 결실 오류 정정이 가능한 자기 동기화 DNA 코드의 설계 방법.

청구항 9

제8항에 있어서, DNA 부호어의 오류를 정정하는 단계는,

각 SDC 코드의 길이가 2n+1인 경우, SDC 코드를 u, (x_{n+1}), v(u, v는 n의 길이를 가짐)로 분해하는 단계;

$x_{n+1} \neq x_{n+2}$이면 SDC 코드 중 [1..n + 2] 위치에 단일 결실이 있다고 판단하는 단계; 및

SDC 코드 중 [n + 2..2n + 1]로부터 DNA 부호어를 생성하는 단계;를 포함하는, 단일 결실 오류 정정이 가능한 자기 동기화 DNA 코드의 설계 방법.

청구항 10

제9항에 있어서,

$x_{n+1} = x_{n+2}$이면 u에 단일 결실이 없다고 판단하는 단계; 및

SDC 코드 중 [1, . . . , n]로부터 DNA 부호어를 생성하는 단계;를 포함하는, 단일 결실 오류 정정이 가능한 자기 동기화 DNA 코드의 설계 방법.

청구항 11

제8항에 있어서,

각 SDC 코드의 길이가 2n+2인 경우, SDC를 u, (x_{n+1}, x_{n+2}), v(u, v는 n의 길이를 가짐)로 분해하는 단계;

$x_{n+1} \neq x_{n+2}$이면 SDC 코드 중 [1..n + 2] 위치에 단일 결실이 있다고 판단하는 단계; 및

SDC 코드 중 [n + 2..2n + 1]로부터 DNA 부호어를 생성하는 단계;를 포함하는, 단일 결실 오류 정정이 가능한 자기 동기화 DNA 코드의 설계 방법.

청구항 12

제11항에 있어서,

$x_{n+1} = x_{n+2}$이면 u에 단일 결실이 없다고 판단하는 단계;

SDC 코드의 [1, . . . , n]로부터 DNA 부호어를 생성하는 단계;

DNA 부호어 시퀀스의 [n+1..2n]의 심볼과 SDC 코드 중 [n+3..2n+2]의 심볼이 동일하면 단일 결실이 없다고 판단하는 단계; 및

DNA 부호어 시퀀스의 [n+1..2n]의 심볼과 SDC 코드 중 [n+3..2n+2]의 심볼이 동일하지 않으면, SDC 코드 중 [n+3..2n+2] 위치에 단일 결실이 있다고 판단하는 단계;를 포함하는, 단일 결실 오류 정정이 가능한 자기 동기화 DNA 코드의 설계 방법.

청구항 13

제1항 내지 제13항 중 어느 하나의 항에 따른 자기 동기화 가능한 DNA 코드의 결실 오류 정정 방법을 수행하기 위해 컴퓨터로 판독 가능한 저장 매체에 저장된 컴퓨터 프로그램.

발명의 설명

기 술 분 야

[0001] 본 발명은 단일 결실 오류 정정이 가능한 자기 동기화 DNA 코드의 설계 방법에 관한 것으로서, 더욱 상세하게는 컴퓨터로 구현된 방법을 통해 설계된 DNA 코드가 DNA 컴퓨팅 및 DNA 데이터 저장 등에 적절히 사용되기 위해서 DNA 데이터를 DNA 코드로 변환하는 과정을 효율적으로 구현하고, DNA 코드의 결실 오류를 최소화하는 기술에 관한 것이다.

배 경 기 술

[0003] DNA는 A, T, G, C와 같은 특정한 염기서열로 이루어져 있다. 따라서, 기존의 전자회로를 이용한 기억장치가 이진코드 0, 1을 사용하는 것처럼 DNA도 염기서열을 사용하여 저장장치로 활용될 가능성에 대한 연구가 진행되고 있다.

[0004] 이러한 저장장치는 기존 전자 장비와 달리 수명이 반영구적이며, 유지비용도 훨씬 적게 들며 데이터 복제면에서 훨씬 효율적이다. 또한, 데이터를 A, T, G, C로 나누어 저장 가능하므로, 두 개의 표현방식인 0, 1보다 더 많은

저장 공간을 활용 할 수 있다는 장점이 있다.

[0005] 그러나, 시간이 지날수록 여러 곳에서 변이발생의 가능성이 있기에 이러한 오류를 최소화 하는 오류 수정 방법(Error-Correcting method)이 연구되기 시작하였다.

[0006] 고전적 코딩 이론에서는 치환 오류(substitution error)의 영향을 받는 통신 채널에 대한 오류 수정 연구에서 몇 가지 발전이 있었다. 그러나, 실제 전송채널에서는 치환 오류(substitution error)뿐만 아니라 삽입 오류(insertion error), 결실 오류(deletion error)와 같은 동기화 오류도 발생할 수 있다. 이러한 상황에서는 기존 오류 수정 코드가 제대로 작동하지 않을 수 있다. 따라서, 동기화 오류 정정 코드에 대한 연구가 필요하다.

[0007] 기존의 일 연구에서는 VT 코드(Varshamov-Tenengolts 코드)가 동기화 오류를 수정하는 기능이 있다는 것을 발견하였으며, 일부 VT 코드가 완전한 단일 결실 오류 수정 바이너리 코드임을 증명하였다. 다른 연구에서는 이진 단일 결실 오류 수정 코드(SDC 코드)를 연구하였고, 이후 결실 및 삽입 수정 코드에 대한 일부 결과가 발표되었다.

[0008] 최근에는 결실 및 삽입 오류를 수정하는 코드가 DNA 저장 및 DNA 시퀀싱에 적용되어 코딩 커뮤니티에서 많은 주목을 받고 있다. DNA의 생화학적 특성으로 인해 결실/삽입 오류 및 치환 오류를 수정하는 것은 DNA 컴퓨팅 또는 DNA 기반 저장에 필수적이다.

[0009] 일반적으로 실험 데이터에 따르면 결실 및 삽입 오류가 대체 오류보다 더 자주 발생한다. DNA 가닥(strands)의 치환 오류율은 대략 6.0×10^{-4}와 1.4×10^{-3} 사이이고, 삽입 오류와 결실 오류의 비율은 각각 약 1.0×10^{-3}과 5.0×10^{-3}이다.

[0010] 따라서, DNA 컴퓨팅 또는 DNA 기반 저장을 위해 DNA 코드의 결실 오류를 보완할 수 있는 기능이 요구된다.

선행기술문헌

특허문헌

[0012] (특허문헌 0001) KR 10-2022-0037855 A

발명의 내용

해결하려는 과제

[0013] 이에, 본 발명의 기술적 과제는 이러한 점에서 착안된 것으로 본 발명의 목적은 단일 결실 오류 정정이 가능한 자기 동기화 DNA 코드의 설계 방법을 제공하는 것이다.

과제의 해결 수단

[0015] 상기한 본 발명의 목적을 실현하기 위한 일 실시예에 따른 단일 결실 오류 정정이 가능한 자기 동기화 DNA 코드의 설계 방법은, 입력된 이진 데이터의 길이를 나타내는 헤더 데이터, 패딩 데이터 및 입력된 이진 데이터를 차례로 연결하여, 총 길이가 2n(여기서, n은 자연수)의 배수인 이진 시퀀스를 생성하는 단계; 생성된 이진 시퀀스의 각 두 심볼을 DNA 심볼인 A, C, G 및 T로 매핑된 이진 맵을 사용하여 n의 배수의 길이를 갖는 사전(pre) DNA 시퀀스로 변환하는 단계; 사전 DNA 시퀀스를 n의 길이를 갖는 블록들로 슬라이싱하는 단계; 슬라이싱된 사전 DNA 시퀀스의 각 블록에 미리 설정된 생성 행렬을 곱하여 2n 길이를 갖는 DNA 부호어(codeword)들이 연결된 DNA 부호어 시퀀스를 생성하는 단계; 및 각 DNA 부호어를 2n+2의 길이를 갖는 단일 결실 정정(이하, SDC) 코드로 변환하되, 각 DNA 부호어의 첫 번째 심볼이 이전 부호어의 마지막 심볼과 동일한 경우, 해당 DNA 부호어는 상보적 SDC 코드로 변환하여 구분자가 없는 자기 동기화 SDC 시퀀스로 변환하는 단계;를 포함한다.

[0016] 본 발명의 실시예에서, 자기 동기화 SDC 시퀀스로 변환하는 단계는, 각 DNA 부호어의 n 번째 심볼과 n+1 번째 사이에 n 번째 심볼과 n+1 번째 심볼과 다른 동일 심볼(ϕ) 2개를 삽입하여 SDC 코드로 변환하는 단계; 각 DNA

부호어의 첫 번째 심볼이 이전 부호어의 마지막 심볼과 동일한 경우, 길이가 2n+2이고 모든 성분이 1인 벡터를 추가하여 변환된 SDC 코드의 상보체로 변환하는 단계; 및 각 DNA 부호어가 변환된 각 SDC 코드를 구분자 없이 연결하여 자기 동기화 SDC 시퀀스를 생성하는 단계;를 포함할 수 있다.

[0017] 본 발명의 실시예에서, DNA 부호어 시퀀스를 생성하는 단계는, 사전 DNA 시퀀스의 각 블록을 맵(f)에 의해 행렬 $2 \times 2n$로 변환하는 단계; 변환된 행렬을 생성 행렬과 곱하여 부호어로 인코딩하는 단계; 인코딩된 부호어에 맵(f)의 역맵(f^{-1})를 적용하여 DNA 부호어로 변환하는 단계; 및 사전 DNA 시퀀스의 각 블록에 대해 획득한 DNA 부호어를 연결하여 DNA 부호어 시퀀스를 획득하는 단계;를 포함할 수 있다.

[0018] 본 발명의 실시예에서, 단일 결실 오류 정정이 가능한 자기 동기화 DNA 코드의 설계 방법은, 구분자가 없는 자기 동기화 SDC 시퀀스를 입력된 이진 데이터로 디코딩하는 단계;를 더 포함할 수 있다.

[0019] 본 발명의 실시예에서, 입력된 이진 데이터로 디코딩하는 단계는, DNA 부호어 시퀀스를 획득하기 위해 구분자가 없는 자기 동기화 SDC 시퀀스를 디코딩하는 단계; 각 DNA 부호어로부터 사전(pre) DNA 시퀀스를 획득하는 단계; 사전 DNA 시퀀스에 이진 맵의 역맵을 적용하여 이진 시퀀스를 획득하는 단계; 및 획득한 이진 시퀀스의 끝에서부터 이진 데이터의 해당하는 자릿수의 데이터를 분리하여 입력된 이진 데이터를 획득하는 단계;를 포함할 수 있다.

[0020] 본 발명의 실시예에서, 자기 동기화 SDC 시퀀스를 디코딩하는 단계는, 자기 동기화 SDC 시퀀스의 각 SDC 코드에 단일 결실 오류가 발생한 경우 DNA 부호어의 오류를 정정하는 단계;를 포함할 수 있다.

[0021] 본 발명의 실시예에서, 자기 동기화 SDC 시퀀스를 디코딩하는 단계는, 자기 동기화 SDC 시퀀스 중 남은 심볼수(m, 여기서, m은 자연수)가 0 보다 큰 동안 반복을 수행할 수 있다.

[0022] 본 발명의 실시예에서, 자기 동기화 SDC 시퀀스를 디코딩하는 단계는, 단일 결실 오류가 있는 경우, 남은 심볼수(m)를 m - (2n + 1)로 업데이트하고, 단일 결실 오류가 없는 경우, 남은 심볼수(m)를 m - (2n + 2)로 업데이트하는 단계;를 포함할 수 있다.

[0023] 본 발명의 실시예에서, DNA 부호어의 오류를 정정하는 단계는, 각 SDC 코드의 길이가 2n+1인 경우, SDC 코드를 u, (x_{n+1}), v(u, v는 n의 길이를 가짐)로 분해하는 단계; $x_{n+1} \neq x_{n+2}$이면 SDC 코드 중 [1..n + 2] 위치에 단일 결실이 있다고 판단하는 단계; 및 SDC 코드 중 [n + 2..2n + 1]로부터 DNA 부호어를 생성하는 단계;를 포함할 수 있다.

[0024] 본 발명의 실시예에서, $x_{n+1} = x_{n+2}$이면 u에 단일 결실이 없다고 판단하는 단계; 및 SDC 코드 중 [1, . . . , n]로부터 DNA 부호어를 생성하는 단계;를 포함할 수 있다.

[0025] 본 발명의 실시예에서, 각 SDC 코드의 길이가 2n+2인 경우, SDC를 u, (x_{n+1}, x_{n+2}), v(u, v는 n의 길이를 가짐)로 분해하는 단계; $x_{n+1} \neq x_{n+2}$이면 SDC 코드 중 [1..n + 2] 위치에 단일 결실이 있다고 판단하는 단계; 및 SDC 코드 중 [n + 2..2n + 1]로부터 DNA 부호어를 생성하는 단계;를 포함할 수 있다.

[0026] 본 발명의 실시예에서, $x_{n+1} = x_{n+2}$이면 u에 단일 결실이 없다고 판단하는 단계; SDC 코드의 [1, . . . , n]로부터 DNA 부호어를 생성하는 단계; DNA 부호어 시퀀스의 [n+1..2n]의 심볼과 SDC 코드 중 [n+3..2n+2]의 심볼이 동일하면 단일 결실이 없다고 판단하는 단계; 및 DNA 부호어 시퀀스의 [n+1..2n]의 심볼과 SDC 코드 중 [n+3..2n+2]의 심볼이 동일하지 않으면, SDC 코드 중 [n+3..2n+2] 위치에 단일 결실이 있다고 판단하는 단계;를 포함할 수 있다.

[0027] 상기한 본 발명의 다른 목적을 실현하기 위한 일 실시예에 따른 상기 단일 결실 오류 정정이 가능한 자기 동기화 DNA 코드의 설계 방법을 수행하기 위한 컴퓨터 프로그램은 컴퓨터로 판독 가능한 저장 매체에 저장된다.

발명의 효과

[0029] 이와 같은 단일 결실 오류 정정이 가능한 자기 동기화 DNA 코드의 설계 방법에 따르면, 구분자가 없는 연속적인 데이터에 결실 오류가 발생할 경우, 구분자의 위치를 알아내고(self-synchronizing), 결실 오류를 정정(single-

deletion-correcting)할 수 있는 코드를 제공한다.

도면의 간단한 설명

[0031] 도 1은 단일 결실 오류 정정이 가능한 자기 동기화 DNA 코드의 설계 방법의 흐름도이다.

발명을 실시하기 위한 구체적인 내용

[0032] 후술하는 본 발명에 대한 상세한 설명은, 본 발명이 실시될 수 있는 특정 실시예를 예시로서 도시하는 첨부 도면을 참조한다. 이들 실시예는 당업자가 본 발명을 실시할 수 있기에 충분하도록 상세히 설명된다. 본 발명의 다양한 실시예는 서로 다르지만 상호 배타적일 필요는 없음이 이해되어야 한다. 예를 들어, 여기에 기재되어 있는 특정 형상, 구조 및 특성은 일 실시예에 관련하여 본 발명의 정신 및 범위를 벗어나지 않으면서 다른 실시예로 구현될 수 있다. 또한, 각각의 개시된 실시예 내의 개별 구성요소의 위치 또는 배치는 본 발명의 정신 및 범위를 벗어나지 않으면서 변경될 수 있음이 이해되어야 한다. 따라서, 후술하는 상세한 설명은 한정적인 의미로서 취하려는 것이 아니며, 본 발명의 범위는, 적절하게 설명된다면, 그 청구항들이 주장하는 것과 균등한 모든 범위와 더불어 첨부된 청구항에 의해서만 한정된다. 도면에서 유사한 참조부호는 여러 측면에 걸쳐서 동일하거나 유사한 기능을 지칭한다.

[0033] 이하, 도면들을 참조하여 본 발명의 바람직한 실시예들을 보다 상세하게 설명하기로 한다.

[0034] 길이 2n인 부호(code of length 2n)가 있을 때, 이 부호의 정보 집합(Information set)이 {1,2, ... n}과 {n+1, n+2, ... 2n}과 같이 서로 여집합 관계에 있는 두 부분집합으로 분리 될 때, 이런 부호를 상보적 정보 집합(Complementary Information Set) 코드라 한다.

[0035] 대표적인 예는 가역적 자기 쌍대 코드(Reversible self-dual code)이며, 이런 부호들을 이용한 단일 결실 정정 코드(single-deletion correcting code)에 관한 알고리즘은 이미 연구된 바 있다. 이전의 연구는 가역적 자기 쌍대 코드와 상보적 정보 집합 코드를 활용하여 한 개의 결실/삽입오류(deletion/insertion error)를 정정할 수 있는 DNA 부호를 설계하는 것에 관한 내용이었다.

[0036] 연속된 부호어(codeword)들의 전달과정에서 생기는 오류는 치환 오류(substitution error), 결실 오류(deletion error), 삽입 오류(insertion error), 삭제 오류(erasure error)등이 있다.

[0037] 보통의 경우에는 각 부호어들 사이마다 구분자(delimiters)를 추가하여 각각의 부호어를 구분하며, 구분자로 구분된 각 부호어마다 최대 한 개의 결실/삽입오류가 발생할 경우 이를 정정할 수 있다.

[0038] 그러나, 구분자 없이 부호어들이 연속되어 있는 일종의 정보의 열(sequence, 또는 시퀀스)을 주고 받을 때에 생기는 결실 오류에 대한 정정 방법은 없으므로 본 발명에서는 이 방법을 제시한다.

[0039] 예를 들어, 다음과 같이 3글자 부호어 5개로 된 정보를 보낸다고 가정한다.

[0040] abc,def,ghi,jkl,mno

[0041] 만약 결실 오류가 발생하여 아래와 같은 정보로 잘못 전달될 경우, 구분자가 있으므로 2,4,5번째 부호어에 결실 오류가 있음을 알 수 있고, 이를 정정할 수 있다.

[0042] abc,df,ghi,kl,no

[0043] 그러나, 구분자가 없는 경우, 위의 문자열은 아래와 같이 전송된다.

[0044] abcdfghiklno

[0045] 이 경우 부호어가 3글자로 이뤄져 있다는 정보만 아는 수신자는 아래와 같이 전송자가 4개의 부호어를 전송한 것으로 착각할 수 있다.

[0046] abc,dfg,hik,lno

[0047] 또는, 한 개의 결실 오류가 발생한 경우, 아래와 같이 전달된다.

[0048] bcdefghijklmno

[0049] 이 경우 결실 오류가 있는 것은 알 수 있으나, 구분자의 위치를 알 수 없기 때문에 몇 번째 부호어에 결실 오류

가 있는지 알 수 없게 된다.

[0050] 이렇게 구분자가 없는 연속적인 데이터에 결실 오류가 발생할 경우, 구분자의 위치를 알아내고(self-synchronizing, 이하, 자기 동기화), 결실 오류 정정(single-deletion-correcting)하는 부호를 설계하고, 그 알고리즘을 제시하는 것이 본 발명의 목적이다.

[0051] 본 발명에서는 DNA 부호의 자기 동기화 문제를 해결한다. 또한, 본 발명은 자기 동기화 기능을 갖춘 단일 결실 정정(Single-Deletion-Correcting, 이하, SDC) DNA 코드를 위한 인코딩 및 디코딩 알고리즘을 제시한다. 이러한 알고리즘은 각 DNA 부호어에서 단일 오류의 위치를 감지하고 오류를 수정하며, 각 DNA 부호어의 시작과 끝을 구별할 수 있게 한다.

[0052] 이하에서는, 본 발명에서 사용할 부호 또는 코드 등을 정의한다.

[0053] 먼저, $\mathbb{F}_q$는 소수 거듭제곱 q의 q차원 유한체이다. 특히, C가 $\mathbb{F}_q^n$ 상의 부분집합이면, C는 길이가 n이고 차원이 k인 q-진 선형 코드로 불린다. 이러한 부호를 $[n,k]_q$ 코드로 나타낸다. 코드의 각 원소를 부호어(codeword, 코드워드)라고 부른다.

[0054] n 길이의 코드 C에 대하여 길이 $|I|$의 모든 가능한 튜플이 C의 정확히 하나의 부호어에서만 발생하는 {1, 2, . . . , n}의 부분 집합 I가 존재하는 경우, 이 부호 C를 체계적(systematic)부호라고 하며, 이 때, 집합 I를 C의 정보집합(information set)이라고 한다. 만약 C가 정보 집합의 크기가 k인 체계적 부호인 경우, C 내의 q^k 부호어와 $\mathbb{F}_q^k$ 간에 일대일 대응이 존재한다.

[0055] 코드 C의 처음 개의 좌표가 정보 집합을 형성하는 경우, 코드C는 $(I_k \mid A)$ 형태의 유일한 생성자 행렬을 갖는다. 여기서, I_k는 $k \times k$ 항등 행렬이고, A는 $k \times (n-k)$ 행렬이다. 이러한 생성자 행렬은 코드C의 표준 생성 행렬이라 불린다.

[0056] $[2n,n]_q$ 체계적 부호 중 두 개의 겹치지 않는 정보 집합을 갖는 코드는 상보적 정보 집합(Complementary Information Set, 이하, CIS) 코드라고 한다. 본 발명에서는 두 개의 겹치지 않는 정보집합 {1,2, ..., n}과 {n+1, n+2, ..., 2n}을 갖는 체계적 CIS 코드를 사용한다.

[0057] 코드 C의 길이가 인 부호어 x를 고려하면, 벡터 $\mathbf{y} \in \mathbb{F}_q^{n-1}$가 x에서 하나의 심볼이 결실되어 얻어진 경우, y는 x의 하위어(subword)이며, x는 y의 상위어(superword)라고 한다. 번째 심볼이 결실된 경우에 x의 하위어 y를 $\mathbf{y} = \mathbf{x}^i$로 나타낸다. $D_1(\mathbf{x})$로 x의 하위어 집합을 나타낸다. 코드 C가 단일 결실 정정(SDC) 코드라고 하는 것은 모든 $\mathbf{x}_1, \mathbf{x}_2 \in C, \mathbf{x}_1 \neq \mathbf{x}_2$에 대해 $D_1(\mathbf{x}_1) \cap D_1(\mathbf{x}_2) = \emptyset$가 성립하는 경우이다.

[0058] 코드 C의 두 부호어 $\mathbf{u}$와 $\mathbf{v}$를 고려하면, 부호어 $\mathbf{u}$와 $\mathbf{v}$ 간의 레벤슈타인(Levenshtein) 거리 $d_l(\mathbf{u},\mathbf{v})$는 $\mathbf{u}$를 $\mathbf{v}$로 변환하는데 필요한 삽입과 결실의 최소 횟수로 정의된다. 레벤슈타인 거리는 실제로 거리함수(metric function)이다. C의 최소 레벤슈타인 거리는 $d_l(C)$로 나타내며, C의 서로 다른 부호어 간의 레벤슈타인 거리 중 최소값이다.

[0059] 해밍(Hamming) 거리 $d_h(\mathbf{u},\mathbf{v})$는 $\mathbf{u}$와 $\mathbf{v}$가 다른 좌표의 개수로 정의된다. C의 최소 해밍 거리는 $d_h(\mathbf{u},\mathbf{v})$로 나타내며, C의 서로 다른 부호어 간의 해밍 거리 중 최소값이다. 코드 C는 치환 오류를 수정할 수 있는 경우에는 $d_h(C) > 2t$이고, 코드 C는 e 결실/삽입 오류를 수정할 수 있는 경우에는 $d_l(C) > 2e$임이 알려져 있다.

[0060] CIS 코드에서 특정 규칙으로 두 개의 동일한 심볼을 삽입하여 SDC 코드를 생성할 수 있다.

[0061] C를 $\mathbb{F}_q$ 상의 길이가 2n인 아래와 같은 CIS 코드라고 하고,

[0062] $\phi: \mathbb{F}_q^{2n} \rightarrow \mathbb{F}_q$

[0063] 임의의 벡터 $\mathbf{x} = (x_i) \in \mathbb{F}_q^{2n}$ 에 대하여 아래와 같이 맵을 정의하면,

[0064] $\phi(\mathbf{x}) = x_n + 1,$

[0065] 아래와 같은 새로운 벡터가 정의된다.

[0066] $\mathbf{x}_\phi = (x_1, \cdots, x_n, \phi(\mathbf{x}), \phi(\mathbf{x}), x_{n+1}, \cdots, x_{2n}) \in \mathbb{F}_q^{2n+2}$

[0067] 위와 같이 C의 모든 부호어 x에 대해 n번째 위치와 (n + 1)번째 위치 사이에 두 개의 $\phi(\mathbf{x})$를 추가하여 얻은 벡터 $\mathbf{x}_\phi$들의 집합, 즉 아래와 같은 집합은 SDC 코드가 된다.

[0068] $C_\phi = \{\mathbf{x}_\phi \mid \mathbf{x} \in C\}$

[0069] 만약 $\mathbb{F}_q$의 고정된 0이 아닌 원소 에 대해 ϕ가 $\phi(\mathbf{x}) = x_n + a$로 정의된다면, 집합 C_ϕ도 SDC 코드이다. 사실, CIS 코드 C의 각 부호어 x에 대해 맵 $\phi(\mathbf{x})$가 x_n과 다른 값을 가지도록 정의하는 한, SDC 코드 C_ϕ을 얻을 수 있다.

[0070] 길이가 2인 CIS 코드에서 SDC 코드를 구성하는 것은 각 CIS 코드의 부호어 x의 번째 심볼 x_n과 다른 심볼 $\phi(\mathbf{x})$를 선택하고 해당 심볼을 부호어의 중간에 두 번 삽입하는 것이다.

[0071] 아래 알고리즘 1은 선행기술에서 제시되었던, C가 길이가 2n인 CIS 코드이고, C_ϕ에서 수신된 벡터에서 구분자를 사용한 단일 결실 오류 정정의 예시이다.

[0072] [알고리즘 1]

Algorithm 1 decoding algorithm for single-deletion-error correction in [2]

Require: a received vector x through a single-deletion channel from C_ϕ
Ensure: the codeword c in C decoded from x

```
1: 2n ← the length of C
2: L ← the length of x
3: if L = 2n + 1 then                          ▷ the case in which a single-deletion occurs in x
4:     decompose x = u ⊖ (x_{n+1}) ⊕ v, where u and v are vectors of length n.
5:     φ ← x_{n+1}
6:     u_n ←the last symbol of u
7:     if u_n = x_{n+1} then
8:         there is no deletion in v
9:         take y = v with the information set {n + 1, . . . , 2n}.
10:    else if u_n ≠ x_{n+1} then
11:        there is no deletion in u
12:        take y = u with the information set {1, . . . , n}.
13:    end if
14:    obtain c generated by y with the corresponding information set.
15: else if L = 2n + 2 then                    ▷ the case in which there are no single-deletion errors
16:    decompose x = u ⊖ (x_{n+1}, x_{n+2}) ⊕ v, where u and v are vectors of length n.
17:    φ ← x_{n+1}
18:    c ← u ⊕ v
19: else
20: end if
21: Return c, φ
```

[0073]

[0074] 본 발명에서는 전송된 데이터에 결실 오류가 있더라도 각 부호어 사이에 구분자가 없을 때 각 부호어의 시작과 끝을 어떻게 식별할 수 있는지에 대한 방법을 제안한다. 이하에서는, 이를 위해 몇 가지를 추가로 정의한다.

[0075] C가 코드이고 $(\mathbf{c}_1,\mathbf{c}_2,\cdots,\mathbf{c}_m)$이 C의 m개의 부호어로 이루어진 시퀀스인 경우, 부호어 간의 모든 구분자를 삭제하면 시퀀스를 간단히 구분자 없이 $\mathbf{c}_1\mathbf{c}_2\cdots\mathbf{c}_m$으로 나타낸다. 만약 구분자 없는 시퀀스 $\mathbf{c}_1\mathbf{c}_2\cdots\mathbf{c}_m$를 원래의 시퀀스 $(\mathbf{c}_1,\mathbf{c}_2,\cdots,\mathbf{c}_m)$로 변환하는데 사용되는 적절한 알고리즘이 있다면, 해당 시퀀스는 자기 동기화 시퀀스(self-synchronizing sequence)로 정의한다.

[0076] C가 코드이고, $\mathbf{c}_1\mathbf{c}_2\cdots\mathbf{c}_m$가 구분자 없이 C의 m개의 부호어로 이루어진 시퀀스인 경우, $\mathbf{c}_1\mathbf{c}_2\cdots\mathbf{c}_m$에서 각 부호어마다 최대 한 번의 결실 오류가 발생하며 결실 오류들이 연속적으로 발생하지 않는 경우, 이러한 오류를 단일 결실 오류(single-deletion-errors)라고 한다.

[0077] 시퀀스 $\mathbf{c}_1\mathbf{c}_2\cdots\mathbf{c}_m$ 에서 단일 결실 오류를 올바르게 수정할 수 있고 동시에 구분자 없는 시퀀스 $\mathbf{c}_1\mathbf{c}_2\cdots\mathbf{c}_m$을 원래의 시퀀스 $(\mathbf{c}_1,\mathbf{c}_2,\cdots,\mathbf{c}_m)$로 변환할 수 있는 적절한 알고리즘이 있다면, 해당 시퀀스는 자기 동기화 단일 결실 오류 정정 시퀀스(self-synchronizing single-deletion-correcting sequence)로 정의한다.

[0078] 만약 C의 부호어로 만든 모든 시퀀스가 적절한 알고리즘을 통해 자기 동기화 단일 결실 오류 정정 시퀀스라면, 코드 C는 자기 동기화 SDC 코드라고 정의한다.

[0079] DNA는 생명의 유전 정보를 아데닌(A), 사이토신(C), 구아닌(G) 및 티민(T)이라고 불리는 네 가지 기본 단위로 인코딩한다. DNA 나선은 네 개의 뉴클레오타이드를 결합하고 상보적인 베이스 페어링(complementary base pairing)을 통해 구성된 이중 나선으로, 왓슨-크릭(Watson Crick) 상보체인 $A^c=T, T^c=A, C^c=G$ 및 $G^c=C$에 의해 연결된다.

[0080] 길이가 n인 DNA 코드는, $(x_1,\cdots,x_n)$ 형식의 튜플 집합이며, 여기서 $x_i\in\{A,C,G,T\}$이며, DNA 부호어는 DNA 코드의 원소를 지칭한다. DNA 부호어의 CG-가중치는 부호어에서 C 및 G의 발생 횟수이다.

[0081] DNA 심볼은 $\delta:\mathbb{F}_2\times\mathbb{F}_2\rightarrow\{A,C,G,T\}$로 정의된 맵에 따라 두 자리 이진 숫자로 아래와 같이 식별할 수 있다.

[0082] $$\delta(00)=A,\delta(11)=T,\delta(10)=C,\ \delta(01)=G.$$

[0083] 이 맵 δ를 사용하면 길이가 2n인 이진 시퀀스를 길이가 n인 DNA 시퀀스로 인코딩할 수 있다. 예를 들어, 길이가 20인 이진 시퀀스 10011011011100100111은 DNA 시퀀스 CGCTGTACGT로 인코딩된다.

[0084] 다른 방법은 $\mathbb{F}_4$의 네 원소와 {A, C, G, T}을 매핑하는 전사(bijection) $\mu:\mathbb{F}_4\rightarrow\{A,C,G,T\}$를 아래와 같이 정의하여 DNA 심볼을 식별할 수 있다.

[0085] $$\mu(0)=A,\mu(1)=T,\mu(\omega)=C \text{ and } \mu(\bar{\omega})=G$$

[0086] 맵 μ는 $\mathbb{F}_4$ 원소의 상보성(complement)을 정의하기 위해 선택되었으며, 왓슨-크릭 상보체와 호환된다. $\mathbb{F}_4$에서 x의 상보성은 $x^c=x+1$로 나타낸다. 이러한 맵 μ를 벡터 및 코드로 확장한다. 따라서, DNA 코드는 맵 μ에 대한 전사 하에서 $\mathbb{F}_4$ 상의 코드와 동일한 것으로 간주 할 수 있다.

[0087] 벡터 $\mathbf{x}=(x_1,\cdots,x_n)\in\mathbb{F}_4^n$에 대해 x의 상보적(complement) 벡터는 $\mathbf{x}^c=(x_1^c,\cdots,x_n^c)$로 표시되며, x의 역(reverse)은 $\mathbf{x}^r=(x_n,\cdots,x_1)$로 표시된다. x의 역-상보(reverse-complement) 벡터는 $\mathbf{x}^{rc}=(x_n^c,\cdots,x_1^c)$로 나타낸다. 벡터 x가 $\mathbf{x}=\mathbf{x}^r$를 만족하면 자체반전가능(self-reversible)하다고 말하며 $\mathbf{x}=\mathbf{x}^{rc}$를 만족하면 자체상보가능(self-reversible-complementary)하다고 한다.

[0088] $\mathcal{M}_2(F_2)$를 2×2 이진 행렬의 집합이라고 하면, 계산 편의를 위하여 또 하나의 맵 $\tau: F_4 \to \mathcal{M}_2(F_2)$를 아래와 같이 정의할 수 있다.

[0089] $$\tau(0) = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}, \tau(1) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \tau(\omega) = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}, \tau(\bar{\omega}) = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix},$$

[0090] DNA 심볼은 합성 맵 $\tau \circ \mu^{-1}$ 에 의해 $\mathcal{M}_2(F_2)$의 행렬로 나타낼 수 있다. 이 맵 $\tau \circ \mu^{-1}$을 f로 나타내고, 제한 조건 하에서의 그 역 맵을 f^{-1}로 나타낼 수 있다. 이러한 맵을 벡터, 시퀀스 및 코드로 확장한 예는 다음과 같다.

[0091] $$f(ACCTG) = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \end{pmatrix}$$

[0093] 이하에서는 구분자 없이 SCD 코드의 각 부호어의 시작과 끝을 구별할 수 있는 알고리즘을 설명한다. 특정 추가 조건 하에서 기호 (x)를 선택하는 것을 포함하며, 이로써 자기 동기화 단일 결실 오류 정정 코드를 구성한다.

[0094] 도 1은 자기 동기화 가능한 DNA 코드의 결실 오류 정정 방법의 흐름도이다.

[0095] 이하에서는 DNA 코드를 설명하며, C가 $\mathbb{F}_4$ 상의 2n 길이의 CIS 코드라고 가정한다. 코드 C에서 유도된 DNA 부호는 $\mathcal{D} = \mu(C)$로 표시되지만, 간결함을 위해 C와 $\mathcal{D}$를 구분하지 않고 사용한다.

[0096] [정리 1]

[0097] C가 $\mathbb{F}_q$ 상의 2n 길이의 CIS 코드이고, 맵 ψ가 아래와 같이 정의되고

[0098] $$\psi: \mathbb{F}_q \times \mathbb{F}_q \to \mathbb{F}_q$$

[0099] 다음 조건을 만족한다고 가정한다.

[0100] i) $\psi(a,b) \neq a$ and $\psi(a,b) \neq b$

[0101] ii) $\psi(a+1, b+1) \neq \psi(a,b) + 1$

[0102] 아래와 같이 벡터를 정의한다.

[0103] $$\mathbf{x}_\psi = (x_1, \cdots, x_n, \psi(x_n, x_{n+1}), \psi(x_n, x_{n+1}), x_{n+1}, \cdots, x_{2n}) \in \mathbb{F}_q^{2n+2}$$

[0104] 즉, $\mathbf{x}_\psi$는 $\mathbf{x}$의 대해 n번째 위치와 (n+1)번째 위치 사이에 심볼 $\psi(x_n, x_{n+1})$을 두 번 더한 벡터이다. 이때, 모든 부호어 $\mathbf{x}$에 대한 벡터 집합인 $\mathbf{x}_\psi$, 즉,

[0105] $$C_\psi = \{\mathbf{x}_\psi \mid \mathbf{x} \in C\}$$

[0106] 는 SDC 코드이다.

[0108] 맵 ψ의 조건에 의해 심볼 $\psi(x_n, x_{n+1})$은 C의 모든 부호어 $\mathbf{x}$에 대해 x_n과 항상 다르다.

[0109] 다음 정의는 ψ : $\mathbb{F}_4 \times \mathbb{F}_4 \to \mathbb{F}_4$로서 정리 1을 만족하는 맵 ψ중 하나를 제공한다.

[0110]
$$\psi(x,y) = \begin{cases} x+y+\omega & \text{if } x+y \in \{0,1\} \text{ and } x,y \in \{0,1\} \\ x+y & \text{if } x+y \in \{0,1\} \text{ and } x,y \in \{\omega,\bar{\omega}\} \\ x+y & \text{if } x+y \in \{\omega,\bar{\omega}\} \text{ and } xy \neq 0 \\ 1 & \text{if } x+y \in \{\omega,\bar{\omega}\} \text{ and } xy = 0 \end{cases}$$

[0111] 아래의 표 1은 맵 ψ의 값들을 설명한다.

표 1

[0112]

$x \setminus y$	0	1	ω	$\bar{\omega}$
0	ω	$\bar{\omega}$	1	1
1	$\bar{\omega}$	ω	$\bar{\omega}$	ω
ω	1	$\bar{\omega}$	0	1
$\bar{\omega}$	1	ω	1	0

[0113] 위에서 정의한 맵 ψ에 대하여 $\mathbb{F}_4$에서 각 a와 b에 대하여 $\psi(a,b) \neq a, \psi(a,b) \neq b$ 및 $\psi(a+1,b+1) \neq \psi(a,b)+1$를 확인하는 것은 간단하다(정리 1의 조건).

[0115] 예를 들어, 길이가 6인 4개의 부호어를 가진 간단한 DNA 코드 D_1을 고려해 본다.

[0116] $D_1 = \{AACCAA, TATTCG, CATTGT, GAGGTC\}$.

[0117] 실제로, $D_1 = \mu(C_{\psi_1})$ 이며, 여기서 C는 맵 ψ 에 따른 $\mathbb{F}_4$에서 길이가 4인의 CIS 코드이다. 어떤 데이터가 D_1에서 만든 DNA 염기서열에 인코딩되어 있다고 가정해 본다.

[0118] $CATTGT, GAGGTC, CATTGT, TATTCG, AACCAA.$

[0119] 여기서, 각 부호어는 구분자인 쉼표로 구분되어 있다. 이제 쉼표가 없고 각 부호어가 하나의 결실 오류를 허용한다고 가정하면, 붉은 표시된 심볼이 결실되고 DNA 염기서열은 다음과 같다.

[0120] $CTTGTGAGTCCATTGTATCGACCAA.$

[0121] 이 DNA 염기서열을 복원하려면 먼저, 처음 6개 심볼 $CTTGTG$를 검토하고, 이를 알고리즘 1을 사용하여 부호어 $CATTGT$로 디코딩할 수 있다.

[0122] $CATTGT, GAGTCCATTGTATCGACCAA.$

[0123] 결과적으로, 첫 번째 $CTTGTG$의 마지막 심볼이었던 심볼 G가 사실은 두 번째 가능한 부호어인 $GAGTCC$의 첫 번째 심볼임을 알 수 있다. 그 다음에, $GAGTCC$는 알고리즘 1을 사용하여 D_1에서 부호어인 $GAGGTC$로 디코딩된다.

[0124] $CATTGT, GAGGTC, CATTGTATCGACCAA.$

[0125] 세 번째 부호어 $CATTGT$에 대해서는 문제가 있다. 네 번째 부호어도 심볼 T로 시작하기 때문에 세 번째 부호어 $CATTGT$의 마지막 심볼 T가 결실되었다는 것을 알아차릴 수 없다. 세 번째 부호어를 $CATTGT$으로 인정한

다면, 수정된 DNA 염기서열은 다음과 같다.

[0126] $CATTGT, GAGGTC, CATTGT, ATCGACCAA.$

[0127] 결과적으로, 나머지 염기서열 $ATCGACCAA$는 실제로는 두 부호어 시퀀스인 $TATTCG$, $AACCAA$에서 두 개의 단일 결실이 발생했음에도 불구하고, 두 부호어 시퀀스에서 세 번의 결실로 간주되어 해독할 수 없다고 결론지어진다.

[0128] 앞의 예에서 다음과 같은 관찰 사항이 있다.

[0129] - 관찰 1: 부호어별로 첫 번째에서 마지막까지 차례대로 DNA 서열을 해독하는 것이 가능하다면 전체 DNA 서열을 해독할 수 있다.

[0130] - 관찰 2: 이전 부호어의 마지막 기호가 다음 부호어의 첫 번째 기호와 동일한 경우 부호어의 시작 위치에 혼란이 생길 수 있다.

[0131] 이러한 관찰을 기반으로 본 발명은 새로운 알고리즘, 알고리즘 2, 3 및 4를 제안한다. 이 알고리즘의 핵심은 이전 부호어의 마지막 기호가 다음 부호어의 첫 번째 기호와 동일한 경우 DNA 부호어를 보완적으로 만들기 위해 모든 성분이 1인 벡터($\mathbf{1}$ 로 표기)를 추가하는 것이며, 이는 이전 관찰 2에서의 혼란을 방지한다.

[0132] 다음은 알고리즘 2, 3 및 4의 의사(pseudo) 알고리즘이다. 여기서, C는 길이가 2n인 $\mathbb{F}_4$ 상의 CIS 부호이며, 모든 모든 성분이 1인 벡터 $\mathbf{1}$이 부호어로 사용되며, $1 \le i \le m$에 대해 c_1가 C의 부호어이며, ψ는 정리 1의 조건을 만족하는 맵이다. 또한, $\Sigma[i]$는 Σ의 i번째 요소를 나타내며 $\Sigma[i..k]$는 Σ의 i번째에서 j번째 연속된 기호로 이루어진 부분 서열을 나타낸다.

[0133] 본 발명에서는 구분자가 없다고 가정하므로 길이가 2n+2인 모든 부분 서열이 가능한 부호어라고 가정해야 한다. 따라서, 이전 알고리즘인 알고리즘 1을 수정하고 길이가 2n+2인 부분 서열에서의 단일 결실 오류의 감지 및 정정을 위한 다음 알고리즘 2를 제안한다.

[0134] 정리 1을 사용하여 다음을 증명한다.

[0135] ψ가 정리 1의 조건을 만족하는 맵이고, C가 $\mathbb{F}_4$ 상의 길이가 2n인 CIS 부호로서 모든 성분이 1인 벡터 $\mathbf{1}$을 부호어로 갖는 경우, 그리고 C_ψ가 C의 모든 부호어 $\mathbf{x}$에 대한 벡터 $\mathbf{x}_\psi$의 집합인 경우, C의 m개 부호어로 이루어진 시퀀스 $(\mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_m)$에서 구분자 없는 C_ψ의 연속된 부호어 시퀀스 $\mathbf{d}_1\mathbf{d}_2\cdots\mathbf{d}_m$으로 인코딩되었다고 가정할 때 다음의 인코딩 규칙을 따른다.

[0136] i) $\mathbf{d}_1 = \mathbf{c}_{1\psi}$.

[0137] ii) $i \ge 2, \mathbf{d}_i = \mathbf{c}_{i\psi}$, 만약 $\mathbf{c}_{i-1}$의 마지막 심볼이 $\mathbf{c}_i$의 첫 번째 심볼과 같지 않은 경우

[0138] iii) $i \ge 2, \mathbf{d}_i = \mathbf{c}_{i\psi} + \mathbf{1}$, 만약 $\mathbf{c}_{i-1}$의 마지막 심볼이 $\mathbf{c}_i$의 첫 번째 심볼과 같은 경우, 여기서 $\mathbf{1}$은 길이가 $2n+2$인 모든 성분이 1인 벡터.

[0139] 그러면 구분자 없이 연속된 시퀀스인 인코딩된 시퀀스 $\mathbf{d}_1\mathbf{d}_2\cdots\mathbf{d}_m$는 자기 동기화가 가능한 단일 결실 정정 시퀀스이다.

[0140] [알고리즘 2]

Algorithm 2 Decoding algorithm for a received vector

Require: a received vector $\mathbf{x}$ of length $2n+2$ through a single-deletion channel from C_ψ
Ensure: the codeword $\mathbf{c}$ in C decoded from $\mathbf{x}$

$L \leftarrow$ the length of $\mathbf{x}$
if $L = 2n+1$ **then**; $is_del \leftarrow true$
  decompose $\mathbf{x} = \mathbf{u} \oplus (x_{n+1}) \oplus \mathbf{v}$, where $\mathbf{u}$ and $\mathbf{v}$ are vectors of length n.
  $\phi \leftarrow x_{n+1}$
  if $x_{n+1} \neq x_{n+2}$ **then**
    there is a single deletion in the position of $[1..n+2]$
    obtain $\mathbf{c}$ generated by $\mathbf{x}[n+2..2n+1]$ with the information set $\{n+1, \ldots, 2n\}$.
  else if $x_{n+1} = x_{n+2}$ **then**
    there is no single-deletion in $\mathbf{u}$
    obtain $\mathbf{c}$ generated by $\mathbf{u}$ with the information set $\{1, \ldots, n\}$.
  end if
else if $L = 2n+1$ **then**
  decompose $\mathbf{x} = \mathbf{u} \oplus (x_{n+1}, x_{n+2}) \oplus \mathbf{v}$, where $\mathbf{u}$ and $\mathbf{v}$ are vectors of length n.
  $\phi \leftarrow x_{n+1}$
  if $x_{n+1} \neq x_{n+2}$ **then**
    there is a single deletion in the position of $[1..n+2]$; $is_del \leftarrow true$
    obtain $\mathbf{c}$ generated by $\mathbf{x}[n+2..2n+1]$ with the information set $\{n+1, \ldots, 2n\}$.
  else if $x_{n+1} = x_{n+2}$ **then**
    there is no single-deletion in $\mathbf{u}$
    obtain $\mathbf{c}$ generated by $\mathbf{u}$ with the information set $\{1, \ldots, n\}$.
    if $\mathbf{c}[n+1..2n] = \mathbf{x}[n+3..2n+2]$ **then**
      there is no single-deletion; $is_del \leftarrow false$
    else
      there is a single-deletion in the position of $[n+3..2n]$; $is_del \leftarrow true$
    end if
  end if
end if
Return $\mathbf{c}, \psi = x_{n+1}, is_del$

[0141]

[0143] 이하에서는 m, 즉 부호어의 수에 대한 귀납법을 사용하여 정리를 증명한다. $m = 1$일 때, 첫 번째 부호어 $\mathbf{d}_1$에서 발생한 어떤 단일 결실 오류도 알고리즘 1을 사용하여 복호화할 수 있다. 다시 말해, 하나의 부호어의 열은 자체 동기화되며 단일 결실 오류를 수정할 수 있는 열이다. 귀납 가정으로, 양의 정수 k에 대해 구분자 없이 k개의 부호어로 이루어진 자체 단일 결실 수정 시퀀스라 가정하고, $\mathbf{d}_1\mathbf{d}_2 \cdots \mathbf{d}_k\mathbf{d}_{k+1}$은 구분자가 없는 $k+1$ 부호어 시퀀스라 한다. 귀납법의 가정에 의해 $\mathbf{d}_1\mathbf{d}_2 \cdots \mathbf{d}_k$는 자체 동기화되고, 따라서 $k+1$ 번째 부호어에서 발생한 단일 결실 오류를 수정해야 한다는 것만 증명하면 된다.

[0144] 만약 $\mathbf{d}_k$의 마지막 심볼이 $\mathbf{d}_{k+1}$의 첫 번째 심볼과 같지 않다면, $\mathbf{d}_{k+1}$의 첫 번째 심볼의 위치에 대한 혼동은 없다. 따라서, 알고리즘 2를 사용하여 $\mathbf{d}_{k+1}$을 수정할 수 있다.

[0145] 만약, $\mathbf{d}_k$의 마지막 심볼이 $\mathbf{d}_{k+1}$의 첫 번째 심볼과 같고, $\mathbf{d}_k$의 마지막 심볼이 결실된 경우, 알고리즘 2는 $\mathbf{d}_k$에서 단일 결실을 감지할 수 없으며 $\mathbf{d}_{k+1}$의 첫 번째 심볼을 $\mathbf{d}_k$의 마지막 심볼로 간주한다. 그러나, $\mathbf{d}_{k+1}$의 어떤 위치에서 단일 결실이 발생하면 알고리즘 2는 두 개의 심볼이 결실된 부분열 $\mathbf{d}_{k+1}$을 얻게 된다. 따라서 알고리즘 2는 $\mathbf{d}_{k+1}$을 복호화하지 못할 것이다.

[0146] 그러나, 인코딩 규칙 iii)은 $\mathbf{d}_k$의 마지막 심볼을 항상 $\mathbf{d}_{k+1}$의 첫 번째 심볼과 다르게 만든다. 따라서, $\mathbf{d}_{k+1}$의

첫 번째 심볼을 결정하는데 혼동이 없으며 알고리즘 2를 사용하여 $\mathbf{d}_{k+1}$을 복호화할 수 있다. 이로써 귀납 및 증명이 완료된다.

[0147] 다음으로, 알고리즘 3과 4를 제안한다. 이러한 알고리즘을 기반으로 하여 자체 동기화 단일 결실 정정 DNA 코드의 구성 방법을 제안한다.

[0148] [알고리즘 3]

[0149]

Algorithm 3 Encoding algorithm for self-synchronizing single-deletion-correcting codes

Require: A sequence of m codewords $(\mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_r)$
Ensure: The sequence $\Sigma = \mathbf{d}_1\mathbf{d}_2\cdots\mathbf{d}_r$ with no delimeters.

$\mathbf{x} \leftarrow \mathbf{c}_1$
convert $\mathbf{x}$ to $\mathbf{x}_\psi$.
$\Sigma \leftarrow \mathbf{x}_\psi$
for $n = 2, \ldots, r$ **do**
  $\mathbf{x} \leftarrow$ the i-th codeword $\mathbf{c}_i$
  convert $\mathbf{x}$ to $\mathbf{x}_\psi$.
  $\sigma \leftarrow$ the first symbol of $\mathbf{x}_\psi$
  if $\sigma = \lambda$ **then**
    $\mathbf{x}_\psi \leftarrow \mathbf{x}_\psi + 1$
  end if
  concatenate $\mathbf{x}_\psi$ to Σ without delimiter
  $\lambda \leftarrow$ the last symbol of Σ.
end for

[0151] 상기에서 정의된 맵 ψ를 고려해 보면, C는 길이 2n+2인 $\mathbb{F}_4$ 상의 CIS 코드로, 모든 원 벡터가 부호어로 있는 경우를 가정하고, C_ψ가 C의 모든 부호어 $\mathbf{x}$에 대한 벡터 $\mathbf{x}_\psi$의 집합인 것으로 가정한다. 그러면, $\mathcal{D} = \mu(C_\psi)$는 자기 동기화 단일 결실 정정 DNA 코드이며, 이를 인코딩하고 디코딩하는 알고리즘은 알고리즘 3과 알고리즘 4에 의해 달성된다.

[0152] 이하에서는 자기 동기화 단일 결실 정정 DNA 코드를 사용하여 이진 데이터를 인코딩 및 디코딩하는 알고리즘을 제안한다. 이러한 알고리즘은 컴퓨터 프로그래밍 언어인 파이톤(Python)으로 구현한다. 여기서, C는 모든 성분이 1인 벡터를 갖는 길이 2n인 $\mathbb{F}_4$ 상의 CIS 코드로, ψ는 상기에서 정의된 맵이며, n_h는 이진서열 bin_data의 길이의 길이보다 충분히 큰 고정된 양의 정수로 가정한다.

[0153] [알고리즘 4]

Algorithm 4 Decoding algorithm for self-synchronizing single-deletion-correcting codes

Require: A sequence $\Sigma = \mathbf{d}_1\mathbf{d}_2\cdots\mathbf{d}_r$ with no delimiters, with possible single-deletion errors, encoded using Algorithm 3.
Ensure: The sequence $\Lambda = (\mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_r)$

```
Λ ← the empty sequence
m ← the number of symbols in the sequence Σ
while m > 0 do
    if m ≥ 2n + 2 then
        d ← Σ[1..(2n + 2)]
    else if m = 2n + 1 then
        d ← Σ[1..(2n + 1)]
    else
        Σ is undecodable; Terminate.
    end if
    Apply Algorithm 2 on d and obtain c and φ
    if d is proved to have a single-deletion then
        Σ ← Λ[(2n + 2)..m]
        m ← m − (2n + 1)
    else
        Σ ← Λ[(2n + 3)..m]
        m ← m − (2n + 2)
    end if
    a ← c[n]; b ← c[n + 3]
    if ψ(a, b) = φ + 1 then
        c ← c + 1
    else
        Pass
    end if
    Append c to Λ
end while
Return Λ
```

[0154]

[0155] [알고리즘 5] 자체 동기화 단일 결실 정정 DNA 코드를 위한 인코딩 알고리즘

[0156] 입력: bin_data $:=$ txt 파일, 이미지 파일 등과 같은 이진 데이터

[0157] 출력: $\Sigma:=$ bin_data의 인코딩된 DNA 시퀀스

[0159] 먼저, 첫 번째 단계로, bin_data를 bin_seq로 변환한다(도 1의 S11).

[0160] bin_data에 대한 메타데이터로 헤더를 추가하고, 변환된 이진 시퀀스인 bin_seq의 길이가 2n의 배수여야 한다.

[0161] - (길이 확인) ℓ 을 bin_data의 길이로 하고,

[0162] - (헤더) ℓ 을 n_h 자릿수의 이진 숫자 ℓ_2 로 변환한다.

[0163] - (패딩) ℓ_p 을 $2n-(\ell+n_h)$ 모듈로 2n의 길이의 모든 0으로 이루어진 시퀀스로 한다.

[0164] - bin_seq는 ℓ_2, ℓ_p 와 bin_data의 순서로 연결된 시퀀스이다.

[0165] 두 번째 단계로, bin_seq를 사전(pre) DNA 시퀀스로 변환한다(도 1의 S13).

[0166] bin_data의 각 두 심볼을 DNA 심볼인 A, C, G 및 T로 매핑된 맵 δ 를 사용하여 pre-DNA_seq를 얻는다.

[0167] $$\mathbf{pre}-DNA_seq=\delta(bin_seq).$$

[0168] pre-DNA_seq의 길이는 n의 배수이다.

[0169] 세 번째 단계로, pre-dna_seq로 슬라이싱한다(도 1의 S15).

[0170] pre-dna_seq의 각 n 심볼을 나누고 나눈 횟수를 r이라고 한다. 그런 다음 pre-DNA_seq는 다음 형식으로 한다.

[0171] $(\mathbf{m}_1, \mathbf{m}_2, \cdots, \mathbf{m}_r)$

[0172] 여기서 $\mathbf{m}_i$는 $1 \le i \le r$에 대한 n DNA 심볼의 시퀀스이다.

[0173] 네 번째 단계로, 생성 행렬을 이용하여 pre-dna_seq의 각 블록을 인코딩한다(도 1의 S17).

[0174] pre-dna_seq의 각 $\mathbf{m}_i$를 다음과 같이 C의 DNA 부호어로 인코딩한다.

[0175] - 각 $\mathbf{m}_i$를 맵 f에 의해 $\mathbb{F}_2$ 상의 $2 \times 2n$ 행렬 $f(\mathbf{m}_i)$로 변환한다.

[0176] - 각 $f(\mathbf{m}_i)$를 맵 τ 상에서 C의 생성 행렬과 곱하여 부호어 $f(\mathbf{c}_i)$로 인코딩한다.

[0177] - 맵 f^{-1}를 $f(\mathbf{c}_i)$에 적용하여 C의 DNA 부호어 $\mathbf{c}_i$를 얻는다.

[0178] 결과적으로, DNA 부호어 $(\mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_r)$의 시퀀스를 얻는다.

[0179] 다섯 번째로, 자기 동기화 SDC 시퀀스로 인코딩한다(도 1의 S19).

[0180] 알고리즘 3을 $(\mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_r)$의 DNA 부호어 시퀀스에 적용한다. 마지막으로 구분자가 없는 DNA 시퀀스 $\Sigma = \mathbf{d}_1\mathbf{d}_2\cdots\mathbf{d}_r$을 얻는다.

[0182] [알고리즘 6] 디코딩 알고리즘

[0183] 입력: 구분자가 없는 경우 가능한 단일 결실 오류를 포함한 DNA 시퀀스 $\Sigma = \mathbf{d}_1\mathbf{d}_2\cdots\mathbf{d}_r$

[0184] 출력: bin_data $:=$ 원래의 이진 데이터.

[0186] 먼저, Σ 디코딩을 수행한다(도 1의 S31).

[0187] 구분자가 없는 DNA 시퀀스 $\Sigma = \mathbf{d}_1\mathbf{d}_2\cdots\mathbf{d}_r$에 알고리즘 4를 적용한다. 이 단계에서, SDC 코드에 발생한 단일 결실 오류를 찾아내고 정정한다.

[0188] 그런 다음 DNA 부호어의 시퀀스 $\Lambda = (\mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_r)$을 얻는다.

[0189] 두 번째 단계로, pre-dna_seq 획득한다(도 1의 S33).

[0190] Λ의 각 DNA 부호어 $\mathbf{c}_i$에 대해 $\mathbf{m}_i$를 $\mathbf{c}_i[1..n]$로 한다. 모든 i에 대해 $\mathbf{m}_i$를 연결하면 pre-dna_seq를 얻는다.

[0191] 세 번째 단계로, pre-dna_seq를 bin_seq로 변환한다(도 1의 S35).

[0192] δ의 역매핑을 사용하여 bin_seq를 얻는다.

[0193] $\text{bin_seq} = \delta^{-1}(\text{pre} - DNA_seq)$.

[0194] 네 번째 단계로, 원래의 이진 데이터 획득한다(도 1의 S37).

[0195] bin_seq의 처음 n_h 자릿수로 만든 이진 숫자는 원래의 이진 데이터의 길이이다. bin_seq의 끝에서부터 해당하는 자릿수의 양만큼을 취하여 원래의 이진 데이터 bin_data를 얻는다.

[0196] 알고리즘 5와 6을 설명하고 이해하는데 도움이 되는 구체적인 예시를 제공한다. 각 복구 가능한 자기 쌍대(self-dual) 코드가 CIS 코드임을 고려한다. 다음 예시에서는 부호어의 가역성(reversibility)을 활용하여 인코딩 및 디코딩 프로세스를 설명한다.

[0198] 예시 1.

[0199] C를 $\mathbb{F}_4$ 상에서 길이 6의 가역적 자기 쌍대 코드로 하되, 아래와 같은 생성 행렬을 갖는다고 가정한다.

[0200] $$G = \begin{pmatrix} 1 & 0 & 0 & \omega & 1 & \omega \\ 0 & 1 & 0 & \omega^2 & \omega^2 & 1 \\ 0 & 0 & 1 & 0 & \omega^2 & \omega \end{pmatrix}$$

[0201] 헤더의 길이를 $n_h = 6$ 설정하면 DNA 심볼을 사용한 생성 행렬 변환은 아래와 같다.

[0202] $$\mu^{-1}(G) = \begin{pmatrix} T & A & A & C & T & C \\ A & T & A & G & G & T \\ A & A & T & A & G & C \end{pmatrix}$$

[0203] 그리고 $\mathrm{GF}(2)$ 상의 생성 행렬 변환은 다음과 같다.

[0204] $$\tau(G) = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

[0205] 19 비트의 이진 데이터를 가지고 있다고 가정한다.

[0206] bin_data=1010100101010100111

[0208] 단계 1. [bin_data를 bin_seq로 변환]

[0209] 헤더 길이를 $n_h = 6$로 설정했으므로, 헤더는 19의 길이에 대한 여섯 자리 이진 표현인 010011이 된다. 따라서, 5 개의 0이 패딩으로 필요하며 연결된 이진 시퀀스의 길이는 6의 배수인 30이어야 한다.

[0210] $$bin_seq = 010011/00000/1010100101010100111.$$

[0211] 이 시퀀스에는 $m = 5$ 개의 블록이 있다.

[0212] 단계 2. [bin_seq를 pre-dna_seq로 변환]

[0213] 연결된 데이터는 길이가 30/2=15인 pre-dna_seq로 변환된다.

[0214] $$pre\text{-}dna_seq = \delta(bin_seq) = GATAAGGGACCCCGT.$$

[0215] 단계 3. [pre-dna_seq를 나누기]

[0216] 시퀀스 $GATAAGGGACCCCGT$를 3개의 DNA 심볼로 이루어진 5개의 하위 시퀀스로 나눈다.

[0217] $$(GAT, AAG, GGA, CCC, CGT)$$

[0218] 단계 4. [pre-dna_seq의 각 블록을 인코딩]

[0219] 각 pre-dna_seq 하위 시퀀스는 맵 f에 의해 2×6 행렬로 변환된다.

[0220] $$\left(\begin{pmatrix}110010\\100001\end{pmatrix},\begin{pmatrix}000011\\000010\end{pmatrix},\begin{pmatrix}111100\\101000\end{pmatrix},\begin{pmatrix}010101\\111111\end{pmatrix},\begin{pmatrix}011110\\111001\end{pmatrix}\right)$$

[0221] 각 블록에 대해 생성 행렬 $\tau(G)$를 곱하여 2×12 행렬의 다섯 개의 부호어를 얻는다. 예를 들어, 첫 번째 블록은 다음과 같이 인코딩된다.

[0222] $$\begin{pmatrix}110010\\100001\end{pmatrix}\begin{pmatrix}1&0&0&0&0&0&0&1&1&0&0&1\\0&1&0&0&0&0&1&1&0&1&1&1\\0&0&1&0&0&0&1&1&1&1&1&0\\0&0&0&1&0&0&1&0&1&0&0&1\\0&0&0&0&1&0&0&0&1&1&0&1\\0&0&0&0&0&1&0&0&1&0&1&1\end{pmatrix}=\begin{pmatrix}110010100011\\100001010010\end{pmatrix}$$

[0223] f^{-1} 를 적용하면 $f^{-1}\begin{pmatrix}110010100011\\100001010010\end{pmatrix}=$ GATTAG가 된다.

[0224] 이 프로세스를 모든 부호어에 대해 반복하고, 이들을 연결하면 DNA 부호어 시퀀스가 된다.

[0225] $$(GATTAG, AAGACT, GGAGTC, CCCCCC, CGTTGC)$$

[0226] 단계 5. [자기 동기화 SDC 시퀀스로 인코딩]

[0227] 알고리즘 3을 적용하면 DNA 부호어 시퀀스가 다음으로 변환된다.

[0228] $$(GATCCTAG, AAGTTACT, GGATTGTC, GGGTTGGG, CGTCCTGC)$$

[0229] 네 번째 블록 CCCCCC은 처음에 CCCAACCC로 변환되지만 첫 번째 심볼 C가 이전 부호어 GGATTGTC의 마지막 심볼과 동일하므로, CCCAACCC의 여분인 GGGTTGGG를 사용한다. 따라서, 자기 동기화 SDC DNA 시퀀스로 최종적으로 얻는 것은 다음과 같다.

[0230] $$\Sigma = GATCCTAGAAGTTACTGGATTGTCGGGTTGGGCGTCCTGC$$

[0232] 예시 2.

[0233] 예를 들어, 발신자가 예시 1의 원래 DNA 시퀀스를 보내고 있다고 가정한다.

[0234] $$\Sigma = GATCCTAGAAGTTACTGGATTGTCGGGTTGGGCGTCCTGC$$

[0235] 발신자가 사용한 코드 C 및 $n_h = 6$이라는 정보가 있다고 가정하고, 전송 중에 네 번의 단일 결실 오류가 발생하고 다음과 같은 DNA 시퀀스를 받았다고 가정한다.

[0236] $$\begin{aligned}\Sigma' &= GATCCTAGAAGTTACTGGATTGTCGGGTTGGGCGTCCTGC\\ &= GATCCTAGAGTTACTGGATGTCGGGTTGGCTCCTGC\end{aligned}$$

[0237] 단계 1. [Σ' 디코딩]

[0238] 알고리즘 4를 Σ'에 적용한다.

[0239] - 집합 $\Lambda = ()$,(빈 시퀀스) 및 $m = 36$ (Σ'의 심볼 수)로 설정한다.

[0240] - $m = 36 > 0$ 이므로, 첫 번째 반복(iteration)이 시작된다.

[0241] (1) $m > 8$이므로 $\mathbf{d} = GATCCTAG$로 설정한다.

[0242] (2) $\mathbf{d}$ 에 알고리즘 2를 적용한다.

[0243] - $\mathbf{d} = GATCCTAG$ 은 GAT, CC, TAG 로 분해되고 $\phi = C$ 로 설정한다.

[0244] - $x_{n+1} = C = x_{n+2}$ 이므로 위치 $\mathbf{d}[1..3] = GAT$ 에서 결실이 없다.

[0245] - $f(GAT) = \begin{pmatrix}110010\\100001\end{pmatrix}$ 이고, $\tau(G)$ 를 곱하여 얻은 $f(\mathbf{c}) = \begin{pmatrix}110010100011\\100001010010\end{pmatrix}$ 를 획득하므로, $\mathbf{c} =$ $GATTAG$이다.

[0246] (3) $\mathbf{c}[4..6] = \mathbf{d}[6..8]$ 이므로 결실이 없다고 결론짓는다. 따라서, $m = m - 8 =$ 28이 되고 Σ' 은 다음과 같이 된다.

[0247] $\Sigma' = AGTTACTGGATGTCGGGTTGGCTCCTGC$

[0248] (4) $\psi(a, b) = \psi(T, T) = C = \phi$ 이므로, $\mathbf{c}$ 를 Λ 에 추가한다. 즉, $\Lambda = (\mathrm{GATTAG})$ 이고 첫 번째 반복이 종료된다.

[0249] - $m = 28 > 0$ 이므로 두 번째 반복이 시작된다.

[0250] (1) $m > 8$, 따라서 $\mathbf{d} = AGTTACTG$ 로 설정한다.

[0251] (2) $\mathbf{d}$ 를 기반으로 알고리즘 2를 적용한다:

[0252] - $\mathbf{d} = AGTTACTG$ 는 AGT, TA, CTG 로 분해되며 $\phi = T$ 로 설정된다.

[0253] $x_{n+1} = T \neq A = x_{n+2}$ 이므로 위치 [1..5]에 단일 결실이 있다.

[0254] 정보 집합 [4..6]을 사용하여 $f(\mathbf{d}[5..7]) = f(ACT)$ 를 얻는다. $\mathbf{c}$ 를 얻기 위해 $f(TCA^r) = f(TCA) = \begin{pmatrix}100100\\011100\end{pmatrix}$ 를 사용한다.

[0255] $\tau(G)$ 와의 곱셈으로 아래와 같이 얻게 된다.

[0256]
$$\begin{pmatrix}100100\\011100\end{pmatrix}\begin{pmatrix}1&0&0&0&0&0&0&1&1&0&0&1\\0&1&0&0&0&0&1&1&0&1&1&1\\0&0&1&0&0&0&1&1&1&1&1&0\\0&0&0&1&0&0&1&0&1&0&0&1\\0&0&0&0&1&0&0&0&1&1&0&1\\0&0&0&0&0&1&0&0&1&0&1&1\end{pmatrix} = \begin{pmatrix}100100110000\\011100100000\end{pmatrix}$$

[0257] 이로써 $TCAGAA$ 가 C 의 부호어임을 알 수 있으며, C 의 가역성은 ACT 가 정보 집합 [4..6]에 있을 때 $AAGACT$ 도 또한 C 의 부호어임을 보장한다. 따라서, $\mathbf{c} = AAGACT$ 라는 결론을 얻는다.

[0258] (3) $\mathbf{d}$ 가 단일 결실을 가지고 있으므로, $m = m - 7 = 21$ 이 되며, Σ' 은 아래와 같다.

[0259] $\Sigma' = GGATGTCGGGTTGGCTCCTGC$

[0260] $\mathbf{d}$ 의 마지막 기호 G 를 반환한다.

[0261] (4) $\psi(a, b) = \psi(G, A) = T = \phi$ 이므로 $\mathbf{c}$ 를 Λ 에 추가하고, 두 번째 반복이 종료된다.

[0262] - $m = 21 > 0$ 이므로, 세 번째 반복이 시작된다.

[0263] (1) $m > 8$ 이므로, $\mathbf{d} = GGATGTCG$ 로 설정한다.

[0264] (2) $\mathbf{d}$ 를 기반으로 알고리즘 2를 적용한다.

[0265] - $\mathbf{d} = GGATGTCG$ 는 GGA, TG, TCG 로 분해되며 $\phi = T$ 로 설정된다.

[0266] - $x_{n+1} = T \neq G = x_{n+2}$ 이므로, 위치 [1..5]에 단일 결실이 있다.

[0267] - 정보 집합 [4..6]을 사용하여 $x_{n+1} = T \neq G = x_{n+2}$ 를 얻는다.

[0268] - 두 번째 반복과 유사한 과정을 거치고 $\mathbf{c} = GGAGTC$ 를 얻는다.

[0269] (3) $\mathbf{d}$ 가 단일 결실을 가지고 있으므로 $m = m - 7 = 14$ 가 되고, Σ' 은 아래와 같다.

[0270] $\Sigma' = GGGTTGGCTCCTGC$

[0271] $\mathbf{d}$ 의 마지막 기호 G 를 반환한다.

[0272] (4) $\psi(a, b) = \psi(A, G) = T = \phi$ 이므로 $\mathbf{c}$ 를 Λ 에 추가하고, 세 번째 반복이 종료된다.

[0273] - $m = 14 > 0$ 이므로, 네 번째 반복이 시작된다.

[0274] (1) $m > 8$ 이므로, $\mathbf{d} = GGGTTGGC$ 로 설정한다.

[0275] (2) $\mathbf{d}$ 를 기반으로 알고리즘 2를 적용한다.

[0276] - $\mathbf{d} = GGGTTGGC$ 는 GGG, TT, GGC 로 분해되며 $\phi = T$ 로 설정한다.

[0277] - $x_{n+1} = T = x_{n+2}$ 이므로, 위치 [1..5]에 단일 결실이 없다.

[0278] - 정보 집합 [1..3]을 사용하여 $f(\mathbf{d}[1..3]) = f(GGG)$ 를 얻는다.

[0279] - 첫 번째 반복과 유사한 과정을 거치고 $\mathbf{c} = GGGGGG$ 를 얻는다.

[0280] (3) $\mathbf{c}[4..6] = GGG \neq= GGC = \mathbf{d}[6..8]$ 이므로 $\mathbf{d}$ 가 단일 결실을 가지고 있다. 따라서, $m =$ $m - 7 = 7$ 이 되고, Σ' 은 아래와 같다.

[0281] $\Sigma' = CTCCTGC$

[0282] $\mathbf{d}$ 의 마지막 기호 C를 반환한다.

[0283] (4) $\psi(G, G) = A \neq T = \phi$ 이므로 $\mathbf{c} + \mathbf{1} = CCCCCC$ 를 Λ 에 추가하고, 네 번째 반복이 종료된다.

[0284] - $m = 7 > 0$ 이므로, 다섯 번째 반복이 시작된다.

[0285] (1) $m = 7 = 2n + 1$ 이므로, $\mathbf{d} = CTCCTGC$ 로 설정한다.

[0286] (2) $\mathbf{d}$ 를 기반으로 알고리즘 2를 적용한다.

[0287] - $\mathbf{d} = CTCCTGC$ 는 CTC, C, TGC 로 분해되며 $\phi = C$ 로 설정한다.

[0288] - $x_{n+1} = C \neq T = x_{n+2}$ 이므로, 위치 [1..5]에 단일 결실이 있다.

[0289] - 정보 집합 [4..6]을 사용하여 $f(\mathbf{d}[5..7]) = f(TGC)$ 를 얻는다.

[0290] - 두 번째 반복과 유사한 과정을 거치고 $\mathbf{c} = CGTTGC$ 를 얻는다.

[0291] (3) $\mathbf{d}$ 에 단일 결실이 있으므로, $m = m - 7 = 0$ 이 되고 Σ' 는 비어있게 된다.

[0292] (4) $\psi(a, b) = \psi(T, T) = C = \phi$ 이므로 $\mathbf{c}$ 를 Λ 에 추가하고, 다섯 번째 반복이 종료된다.

[0293] - $m = 0$ 이므로, 종료된다.

[0294] - $\Lambda = (GATTAG, AAGACT, GGAGTC, CCCCCC, CGTTGC)$ 를 반환한다.

[0295] 단계 2. [Pre-DNA 시퀀스 획득]

[0296] $\Lambda = (GATTAG, AAGACT, GGAGTC, CCCCCC, CGTTGC)$ 로부터 pre-DNA 시퀀스를 아래와 같이 획득한다.

[0297] $GATAAGGGACCCCGT$

[0298] 단계 3. [Pre-DNA 시퀀스를 이진 시퀀스로 변환]

[0299] $GATAAGGGACCCCGT$ 에 대해 δ 의 역맵(inverse map)을 사용하여 아래와 같이 얻게된다.

[0300] $\text{bin_seq} = 010011000001010100101010100111.$

[0301] 단계 4. [원본 이진 데이터 얻기]

[0302] $n_h = 6$ 이므로 원본 이진 데이터의 길이는 이진 010011, 즉 19이다. 따라서, bin_seq의 끝에서 19자리를 취하면 원본 이진 데이터 bin_data는 1010100101010100111이 된다.

[0303] 이와 같은, 단일 결실 오류 정정이 가능한 자기 동기화 DNA 코드의 설계 방법은 애플리케이션으로 구현되거나 다양한 컴퓨터 구성요소를 통하여 수행될 수 있는 프로그램 명령어의 형태로 구현되어 컴퓨터 판독 가능한 기록 매체에 기록될 수 있다. 상기 컴퓨터 판독 가능한 기록 매체는 프로그램 명령어, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다.

[0304] 상기 컴퓨터 판독 가능한 기록 매체에 기록되는 프로그램 명령어는 본 발명을 위하여 특별히 설계되고 구성된 것들이거니와 컴퓨터 소프트웨어 분야의 당업자에게 공지되어 사용 가능한 것일 수도 있다.

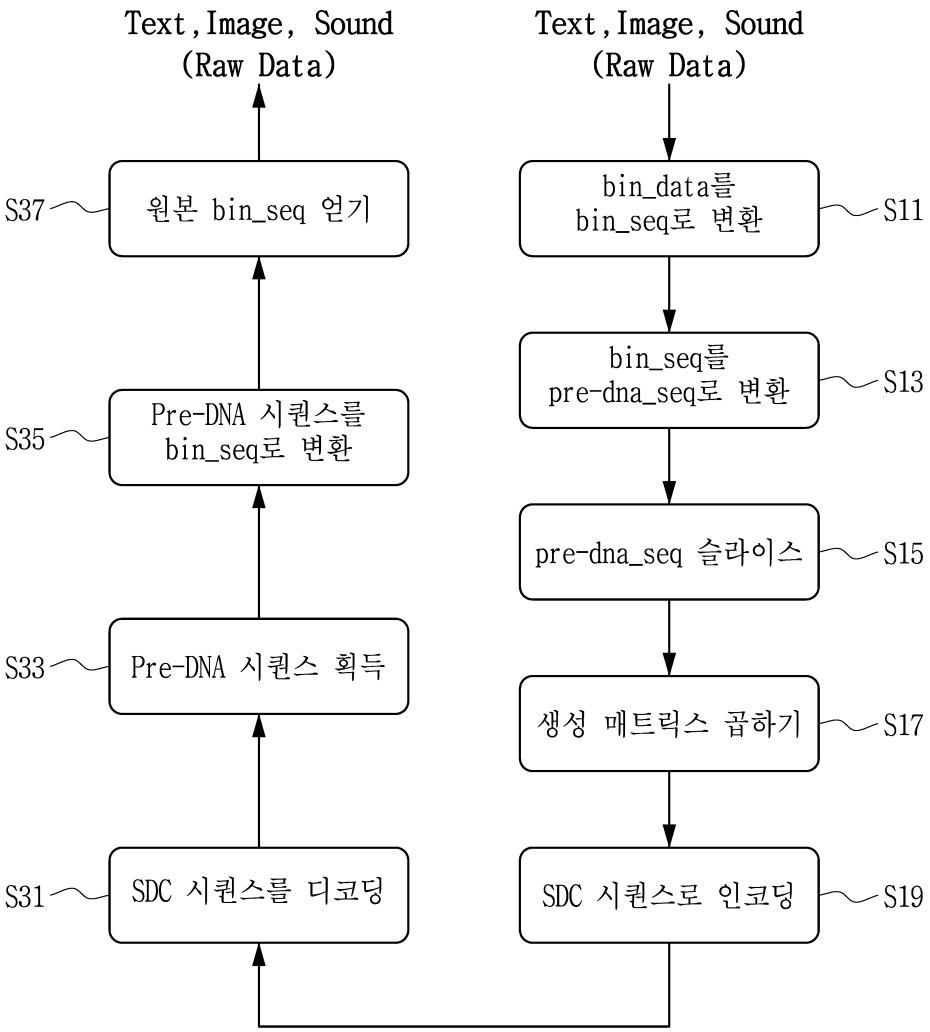
[0305] 컴퓨터 판독 가능한 기록 매체의 예에는, 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체, CD-ROM, DVD와 같은 광기록 매체, 플롭티컬 디스크(floptical disk)와 같은 자기-광 매체(magneto-optical media), 및 ROM, RAM, 플래시 메모리 등과 같은 프로그램 명령어를 저장하고 수행하도록 특별히 구성된 하드웨어 장치가 포함된다.

[0306] 프로그램 명령어의 예에는, 컴파일러에 의해 만들어지는 것과 같은 기계어 코드뿐만 아니라 인터프리터 등을 사용해서 컴퓨터에 의해서 실행될 수 있는 고급 언어 코드도 포함된다. 상기 하드웨어 장치는 본 발명에 따른 처리를 수행하기 위해 하나 이상의 소프트웨어 모듈로서 작동하도록 구성될 수 있으며, 그 역도 마찬가지이다.

[0307] 이상에서는 실시예들을 참조하여 설명하였지만, 해당 기술 분야의 숙련된 당업자는 하기의 특허 청구의 범위에 기재된 본 발명의 사상 및 영역으로부터 벗어나지 않는 범위 내에서 본 발명을 다양하게 수정 및 변경시킬 수 있음을 이해할 수 있을 것이다.

도면

도면1


Text,Image, Sound
(Raw Data)
Text,Image, Sound
(Raw Data)
S37
원본 bin_seq 얻기
bin_data를
bin_seq로 변환
S11
bin_seq를
pre-dna_seq로 변환
S13
S35
Pre-DNA 시퀀스를
bin_seq로 변환
pre-dna_seq 슬라이스
S15
S33
Pre-DNA 시퀀스 획득
생성 매트릭스 곱하기
S17
S31
SDC 시퀀스를 디코딩
SDC 시퀀스로 인코딩
S19