(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2022-0037855
(43) 공개일자 2022년03월25일

(51) 국제특허분류(Int. Cl.)
G16B 35/10 (2019.01) *G16B 40/20* (2019.01)
G16B 50/00 (2019.01)
(52) CPC특허분류
G16B 35/10 (2019.02)
G16B 40/20 (2019.02)
(21) 출원번호 10-2020-0120817
(22) 출원일자 2020년09월18일
심사청구일자 2020년09월18일

(71) 출원인
서강대학교산학협력단
서울특별시 마포구 백범로 35 (신수동, 서강대학교)
연세대학교 산학협력단
서울특별시 서대문구 연세로 50 (신촌동, 연세대학교)
이화여자대학교 산학협력단
서울특별시 서대문구 이화여대길 52 (대현동, 이화여자대학교)
(72) 발명자
최환혁
서울특별시 마포구 백범로 35
김현진
서울특별시 서대문구 연세로 50
이윤진
서울특별시 서대문구 이화여대길 52
(74) 대리인
최우성

전체 청구항 수 : 총 10 항

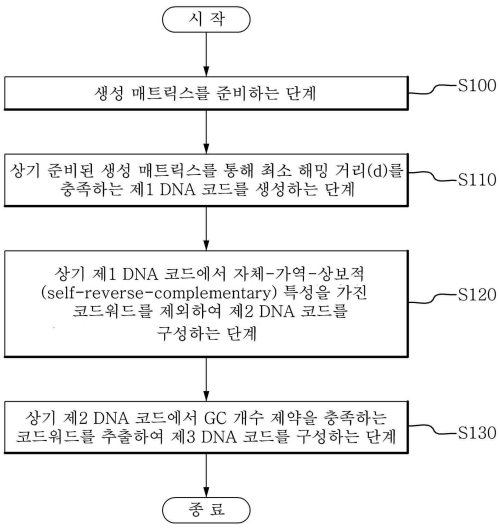
(54) 발명의 명칭 **RSD 코드에 기반한 DNA 코드 생성 방법**

(57) 요 약

본 발명의 실시예에 따른 DNA 코드 생성 방법은 RSD 코드에 기반하여 해밍 거리 제약(HD), 가역-상보 제약(RC) 및 GC 개수 제약을 충족할 수 있는 DNA 코드 생성 방법이 제공된다. 일 실시예에 따른 DNA 코드 생성 방법은 2n의 길이를 가진 DNA 코드를 생성하는 컴퓨터로 구현된 방법으로, 생성 매트릭스 ($I_n \mid M$)를 준비하는 단계로서,

(뒷면에 계속)

대 표 도 - 도2



상기 생성 매트릭스는 n차 단위 매트릭스(I_n) 및 n x n 매트릭스(M)로 구성되며, 상기 n x n 매트릭스(M)는 직교성(orthogonal) 및 반대각성(persymmetric)을 가지는, 단계; 상기 준비된 생성 매트릭스를 통해 최소 해밍 거리(d)를 충족하는 제1 DNA 코드를 생성하는 단계로서, 상기 제1 DNA 코드는 가역적 자기 쌍대(reversible self-dual) 코드인, 단계; 상기 제1 DNA 코드에서 자기-가역-상보적(self-reverse-complementary) 특성을 가진 코드워드를 제외하여 제2 DNA 코드를 구성하는 단계; 및 상기 제2 DNA 코드에서 고정된 GC 개수 제약을 충족하는 코드워드를 추출하여 제3 DNA 코드를 구성하는 단계를 포함한다.

(52) CPC특허분류
G16B 50/00 (2019.02)

이 발명을 지원한 국가연구개발사업
과제고유번호 1345312603
과제번호 2019R1I1A1A01057755
부처명 교육부
과제관리(전문)기관명 한국연구재단
연구사업명 이공학학술연구기반구축(R&D)
연구과제명 DNA 컴퓨팅 및 DNA 기반 정보 저장을 위한 부호 이론 연구
기 여 율 1/3
과제수행기관명 서강대학교
연구기간 2020.03.01 ~ 2021.02.28
이 발명을 지원한 국가연구개발사업
과제고유번호 1345315658
과제번호 2019R1A6A1A11051177
부처명 교육부
과제관리(전문)기관명 한국연구재단
연구사업명 이공학학술연구기반구축(R&D)
연구과제명 수리과학연구소
기 여 율 1/3
과제수행기관명 이화여자대학교
연구기간 2020.03.01 ~ 2021.02.28
이 발명을 지원한 국가연구개발사업
과제고유번호 1711113747
과제번호 2017R1A2B2004574
부처명 과학기술정보통신부
과제관리(전문)기관명 한국연구재단
연구사업명 개인기초연구(과기정통부)(R&D)
연구과제명 순환 함수체 및 드린펠드 모듈의 산술과 대수기하 코드 연구
기 여 율 1/3
과제수행기관명 이화여자대학교
연구기간 2020.03.01 ~ 2021.02.28

명 세 서

청구범위

청구항 1

2n의 길이를 가진 DNA 코드를 생성하는 컴퓨터로 구현된 방법으로,

생성 매트릭스($I_n \mid M$)를 준비하는 단계로서, 상기 생성 매트릭스는 n차 단위 매트릭스(I_n) 및 n x n 매트릭스(M)로 구성되며, 상기 n x n 매트릭스(M)는 직교성(orthogonal) 및 반대각성(persymmetric)을 가지는, 단계;

상기 준비된 생성 매트릭스를 통해 최소 해밍 거리(d)를 충족하는 제1 DNA 코드를 생성하는 단계로서, 상기 제1 DNA 코드는 가역적 자기 쌍대(reversible self-dual) 코드인, 단계;

상기 제1 DNA 코드에서 자기-가역-상보적(self-reverse-complementary) 특성을 가진 코드워드를 제외하여 제2 DNA 코드를 구성하는 단계; 및

상기 제2 DNA 코드에서 GC 개수 제약을 충족하는 코드워드를 추출하여 제3 DNA 코드를 구성하는 단계를 포함하는 DNA 코드 생성 방법.

청구항 2

제1 항에 있어서,

상기 제1 DNA 코드에서 자기-가역-상보적(self-reverse-complementary) 특성을 가진 코드워드를 제외하는 단계는,

상기 제1 DNA 코드를 자기-가역적(self-reversible) 코드워드의 부분 집합인 C_0을 이용하여 복수의 코셋(coset)으로 분해(decompose)하고, 상기 분해된 코셋(coset)을 기준으로 자기-가역-상보적(self-reverse-complementary) 특성을 가진 코드워드를 검출하는 것을 포함하는 DNA 코드 생성 방법.

청구항 3

제1 항에 있어서,

상기 제1 DNA 코드가 코드워드 $\mathbf{x}$를 포함하는 경우, 상기 코드워드 $\mathbf{x}$ 와 상보적인(complement) 코드워드 $\mathbf{x}^c$, 상기 코드워드 $\mathbf{x}$ 와 가역적인(reverse) 코드워드 $\mathbf{x}^r$ 및 상기 코드워드 $\mathbf{x}$ 와 가역-상보적인(reverse-complement) 코드워드 $\mathbf{x}^{rc}$도 상기 제1 DNA 코드에 포함되는 것을 특징으로 하는 DNA 코드 생성 방법.

청구항 4

제1 항에 있어서,

상기 제2 DNA 코드에서 GC 개수가 제약된 코드워드를 추출하여 제3 DNA 코드를 구성하는 단계는,

상기 제2 DNA 코드에서 GC 개수 제약을 충족하는 코드워드의 개수를 산출하는 단계; 및

상기 산출된 코드워드의 개수에 기초하여 GC 개수 제약을 충족하는 코드워드를 추출하고, 추출된 코드워드로 상기 제3 DNA 코드를 생성하는 단계를 포함하는 DNA 코드 생성 방법.

청구항 5

제4 항에 있어서,

상기 제2 DNA 코드에서 GC 개수 제약을 충족하는 코드워드의 개수를 산출하는 과정은 가역-상보적인(reverse-complement) 코드워드가 상기 제1 DNA 코드에 존재하는 지 여부에 따라 달라지는 것을 특징으로 하는 DNA 코드 생성 방법.

청구항 6

2n+2 길이를 가진 DNA 코드를 생성하는 컴퓨터로 구현된 방법으로,

생성 매트릭스($I_n \mid M$)를 준비하는 단계로서, 상기 생성 매트릭스는 n차 단위 매트릭스(I_n) 및 n x n 매트릭스(M)로 구성되며, 상기 n x n 매트릭스(M)는 직교성(orthogonal) 및 반대각성(persymmetric)을 가지는, 단계;

상기 준비된 생성 매트릭스를 통해 최소 해밍 거리(d)를 충족하는 제1 DNA 코드를 생성하는 단계로서, 상기 제1 DNA 코드는 2n 길이의 가역적 자기 쌍대(reversible self-dual) 코드인, 단계;

상기 제1 DNA 코드에서 자기-가역-상보적(self-reverse-complementary) 특성을 가진 코드워드를 제외하여 제2 DNA 코드를 구성하는 단계;

상기 제2 DNA 코드의 코드워드 각각의 n번째 기호와 n+1번째 기호 사이에 2개의 기호를 삽입하여 2n+2 길이를 가진 제3 DNA 코드를 생성하는 단계; 및

상기 제3 DNA 코드에서 GC 개수 제약을 충족하는 코드워드를 추출하여 제4 DNA 코드를 구성하는 단계를 포함하는 DNA 코드 생성 방법.

청구항 7

제6 항에 있어서,

상기 2개의 기호는 동일한 종류의 기호가 반복되어 삽입되되, 상기 2개의 기호는 상기 n번째 기호와 상이한 종류인 것을 특징으로 하는 DNA 코드 생성 방법.

청구항 8

제7 항에 있어서,

상기 2개의 기호의 종류는 상기 제2 DNA 코드의 코드워드 각각의 GC 개수를 고려하여 결정되는 것을 특징으로 하는 DNA 코드 생성 방법.

청구항 9

제8 항에 있어서,

상기 제3 DNA 코드는 적어도 DNA 코드의 단일 변형의 복원이 가능한 것을 특징으로 하는 DNA 코드 생성 방법.

청구항 10

하드웨어와 결합되어 제1 항 내지 제9 항에 따른 DNA 코드 생성 방법을 실행하도록 매체에 저장되는 컴퓨터 프로그램.

발명의 설명

기 술 분 야

[0001] 본 발명은 DNA 코드 생성 방법에 관한 것으로, 더욱 상세하게는 가역적 자기쌍대 코드(RSD code)에 기반한 DNA 코드 생성 방법에 관한 것이다.

배 경 기 술

[0002] 최근, 컴퓨터로-구현된 방법을 통해 생성, 합성된 DNA 코드는 디지털 데이터에 적합한 장기 저장 매체(long-term storage medium)으로 간주되고 있으며, 관련된 연구가 계속하여 진행되고 있다.

[0003] 또한, 컴퓨터로-구현된 방법을 통해 설계된 DNA 코드가 DNA 컴퓨팅 및 DNA 데이터 저장 등에 적절히 사용되기 위해서, DNA 코드의 설계에 대한 적절한 제약이 필요하였다. 이러한 DNA의 생화학적 특성을 충족하기 위한 제약들이 DNA 코드에 요구되었으며, 해밍 거리 제약(Hamming distance constraint, HD), 역-상보 제약(Reverse-complement constraint, RC) 및 고정된 GC 개수 제약(Fixed GC-content constraint, GC)과 같은 제약들이 종래 제안된 바 있으며, 현재 DNA 코드 설계의 기준으로 적용되고 있다. 상술한 제약들을 충족하면서 더 많은 수의

DNA 코드를 용이하게 설계, 생성하거나, 염기서열의 길이가 더 긴 DNA 코드를 상술한 제약 조건을 충족하면서 설계, 생성할 수 있는 방법이 요구되고 있는 실정이다.

[0004] 또한, 컴퓨터로-구현된 방법을 통해 설계된 DNA 코드가 DNA 컴퓨팅 및 DNA 데이터 저장 등에 적절히 사용되기 위해 DNA 코드의 오류, 특히, DNA 코드의 삭제 오류 및 삽입 오류를 보완할 수 있는 기능이 요구된다.

[0005] 이에 본 출원의 발명자는, 해밍 거리 제약(Hamming distance constraint, HD), 역-상보 제약(Reverse-complement constraint, RC) 및 고정된 GC 개수 제약(Fixed GC-content constraint, GC)과 같은 제약들을 충족하면서 더 많은 개수의 DNA 코드를 생성하는 방법과 DNA 코드 내에서의 하나의 삭제 혹은 삽입 오류를 보완할 수 있는 DNA 코드를 생성하는 방법을 제안하는 데 이르렀다.

선행기술문헌

비특허문헌

[0006] (비특허문헌 0001) Gaborit, P., King, O. D.: Linear constructions for DNA codes, Theoretical Computer Science, Volume 334, Issues 1-3, 15 April 2005, Pages 99-113

발명의 내용

해결하려는 과제

[0007] 본 발명은 상술한 종래기술의 문제점을 해결하기 위해 안출된 것으로, RSD 코드에 기반한 DNA 코드 생성 방법을 제공하는 것을 목적으로 한다.

과제의 해결 수단

[0008] 상기 목적을 달성하기 위하여, 본 발명의 일 실시예에 따른 DNA 코드 생성 방법은 2n의 길이를 가진 DNA 코드를 생성하는 컴퓨터로 구현된 방법으로, 생성 매트릭스($I_n \mid M$)를 준비하는 단계로서, 상기 생성 매트릭스는 n차 단위 매트릭스(I_n) 및 n x n 매트릭스(M)로 구성되며, 상기 n x n 매트릭스(M)는 직교성(orthogonal) 및 반대각성(persymmetric)을 가지는, 단계; 상기 준비된 생성 매트릭스를 통해 최소 해밍 거리(d)를 충족하는 제1 DNA 코드를 생성하는 단계로서, 상기 제1 DNA 코드는 가역적 자기 쌍대(reversible self-dual) 코드인, 단계; 상기 제1 DNA 코드에서 자기-가역-상보적(self-reverse-complementary) 특성을 가진 코드워드를 제외하여 제2 DNA 코드를 구성하는 단계; 및 상기 제2 DNA 코드에서 고정된 GC 개수 제약을 충족하는 코드워드를 추출하여 제3 DNA 코드를 구성하는 단계를 포함한다.

[0009] 본 발명의 다른 실시예에 따른 DNA 코드 생성 방법은 2n+2 길이를 가진 DNA 코드를 생성하는 컴퓨터로 구현된 방법으로, 생성 매트릭스($I_n \mid M$)를 준비하는 단계로서, 상기 생성 매트릭스는 n차 단위 매트릭스(In) 및 n x n 매트릭스(M)로 구성되며, 상기 n x n 매트릭스(M)는 직교성(orthogonal) 및 반대각성(persymmetric)을 가지는, 단계; 상기 준비된 생성 매트릭스를 통해 최소 해밍 거리(d)를 충족하는 제1 DNA 코드를 생성하는 단계로서, 상기 제1 DNA 코드는 2n 길이의 가역적 자기 쌍대(reversible self-dual) 코드인, 단계; 상기 제1 DNA 코드에서 자기-가역-상보적(self-reverse-complementary) 특성을 가진 코드워드를 제외하여 제2 DNA 코드를 구성하는 단계; 상기 제2 DNA 코드의 코드워드 각각의 n번째 기호와 n+1번째 기호 사이에 2개의 기호를 삽입하여 2n+2 길이를 가진 제3 DNA 코드를 생성하는 단계; 및 상기 제3 DNA 코드에서 GC 개수 제약을 충족하는 코드워드를 추출하여 제4 DNA 코드를 구성하는 단계를 포함한다.

[0010] 본 발명의 또 다른 실시예에 따른 컴퓨터 프로그램은 하드웨어와 결합되어 상기 DNA 코드 생성 방법을 실행하도록 매체에 저장된다.

발명의 효과

[0011] 본 발명의 실시예에 따른 DNA 코드 생성 방법은 RSD 코드에 기반하여 종래 수학적으로 산출되기 어려웠던 길이 및 조건의 DNA 코드를 해밍 거리 제약(HD), 가역-상보 제약(RC) 및 GC 개수 제약을 충족시키면서 생성할 수 있다.

[0012] 또한, 본 발명의 실시예에 따른 DNA 코드 생성 방법은 해밍 거리 제약(HD), 가역-상보 제약(RC) 및 고정된 GC 개수 제약뿐만 아니라 삭제/삽입 제약(Deletion/Insertion constraint)을 더 충족하는 DNA 코드를 생성할 수 있다. 즉, 하나의 삭제 또는 삽입 오류를 가진 DNA 코드워드를 원형대로 복원하는 것이 지원될 수 있다.

[0013] 이에 따라, DNA 컴퓨팅 및 DNA 데이터 저장 등의 기술이 연구 개발되는 것이 더욱 지원될 수 있다.

[0014] 본 발명에서 얻을 수 있는 효과는 이상에서 언급한 효과들로 제한되지 않으며, 언급하지 않은 또 다른 효과들은 아래의 기재로부터 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자에게 명확하게 이해될 수 있을 것이다.

도면의 간단한 설명

[0015] 도 1은 본 발명의 실시예에 따른 DNA 코드 생성 장치의 구조를 나타내는 예시적인 블록도이다.

도 2는 본 발명의 일 실시예에 따른 DNA 코드 생성 방법의 순서도이다.

도 3은 DNA 코드를 생성하기 위한 생성 매트릭스의 일 예시이다.

도 4는 해밍 거리 제약(HD) 및 가역-상보 제약(RC)을 충족하는 제2 DNA 코드의 일 예시이다.

도 5는 해밍 거리 제약(HD), 가역-상보 제약(RC) 및 고정된 GC 개수 제약을 충족하는 제3 DNA 코드의 일 예시이다.

도 6은 본 발명의 다른 실시예에 따른 DNA 코드 생성 방법의 순서도이다.

도 7은 해밍 거리 제약(HD), 가역-상보 제약(RC) 및 레벤슈타인 거리 제약(DI)를 충족하는 제3 DNA 코드의 일 예시이다.

도 8은 해밍 거리 제약(HD), 가역-상보 제약(RC), 레벤슈타인 거리 제약(DI) 및 고정된 GC 개수 제약을 충족하는 제4 DNA 코드의 일 예시이다.

발명을 실시하기 위한 구체적인 내용

[0016] 본 발명의 이점 및 특징, 그리고 그것들을 달성하는 방법은 첨부되는 도면과 함께 상세하게 후술되어 있는 실시예들을 참조하면 명확해질 것이다. 그러나 본 발명은 이하에서 개시되는 실시 예들에 한정되는 것이 아니라 서로 다른 다양한 형태로 구현될 수 있으며, 단지 본 실시 예들은 본 발명의 개시가 완전하도록 하고, 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자에게 발명의 범주를 완전하게 알려주기 위해 제공되는 것이며, 본 발명은 청구항의 범주에 의해 정의될 뿐이다. 명세서 전체에 걸쳐 동일 참조 부호는 동일 구성 요소를 지칭한다.

[0017] 본 발명의 실시예들을 설명함에 있어서 공지 기능 또는 구성에 대한 구체적인 설명이 본 발명의 요지를 불필요하게 흐릴 수 있다고 판단되는 경우에는 그 상세한 설명을 생략할 것이며, 후술되는 용어들은 본 발명의 실시예에서의 기능을 고려하여 정의된 용어들로서 이는 사용자, 운용자의 의도 또는 관례 등에 따라 달라질 수 있다. 그러므로 그 정의는 본 명세서 전반에 걸친 내용을 토대로 내려져야 할 것이다.

[0018] 첨부된 블록도의 각 블록과 흐름도의 각 단계의 조합들은 컴퓨터 프로그램 인스트럭션들(실행 엔진)에 의해 수행될 수도 있으며, 이들 컴퓨터 프로그램 인스트럭션들은 범용 컴퓨터, 특수용 컴퓨터 또는 기타 프로그램 가능한 데이터 프로세싱 장비의 프로세서에 탑재될 수 있으므로, 컴퓨터 또는 기타 프로그램 가능한 데이터 프로세싱 장비의 프로세서를 통해 수행되는 그 인스트럭션들이 블록도의 각 블록 또는 흐름도의 각 단계에서 설명된 기능들을 수행하는 수단을 생성하게 된다.

[0019] 이들 컴퓨터 프로그램 인스트럭션들은 특정 방식으로 기능을 구현하기 위해 컴퓨터 또는 기타 프로그램 가능한 데이터 프로세싱 장비를 지향할 수 있는 컴퓨터 이용가능 또는 컴퓨터 판독 가능 메모리에 저장되는 것도 가능하므로, 그 컴퓨터 이용가능 또는 컴퓨터 판독 가능 메모리에 저장된 인스트럭션들은 블록도의 각 블록 또는 흐름도의 각 단계에서 설명된 기능을 수행하는 인스트럭션 수단을 내포하는 제조 품목을 생산하는 것도 가능하다.

[0020] 그리고 컴퓨터 프로그램 인스트럭션들은 컴퓨터 또는 기타 프로그램 가능한 데이터 프로세싱 장비 상에 탑재되는 것도 가능하므로, 컴퓨터 또는 기타 프로그램 가능한 데이터 프로세싱 장비 상에서 일련의 동작 단계들이 수행되어 컴퓨터로 실행되는 프로세스를 생성해서 컴퓨터 또는 기타 프로그램 가능한 데이터 프로세싱 장비를 수행하는 인스트럭션들은 블록도의 각 블록 및 흐름도의 각 단계에서 설명되는 기능들을 실행하기 위한 단계들을

제공하는 것도 가능하다.

[0021] 또한, 본 명세서에서 사용되는 수학 기호들은 하기 표 1과 같은 의미를 가질 수 있다.

[0022] [표 1]

[0023]

Notation	
$\mathbf{x}^r$	the reverse of a codeword $\mathbf{x}$
$\mathbf{x}^c$	the complement of a codeword $\mathbf{x}$
$\mathbf{x}^{rc}$	the reverse-complement of a codeword $\mathbf{x}$
$H(\mathbf{x},\mathbf{y})$	Hamming distance between $\mathbf{x}$ and $\mathbf{y}$
I_n	the identity matrix of degree n
R_n	the column reversed matrix of I_n
M^T	the transpose of a matrix M
M^F	the flip-transpose of a matrix M
M^R	the column reversed matrix of a matrix M
$A_4^{RC}(n,d)$	the maximum size of a DNA code of length n satisfying HD and RC-constraint
$A_4^{RC,GC}(n,d,k)$	the maximum size of a DNA code of length n satisfying HD and RC-constraint with constant GC-weight k
$CWE_{\mathcal{C}}$	the complete weight enumerator of a code $\mathcal{C}$
$GCW_{\mathcal{C}}$	the GC-weight enumerator of a code $\mathcal{C}$

[0025] 본 명세서에서, DNA 코드는 컴퓨터를 이용한 방법을 통해 계산된 부호로, 적어도 하나의 2n의 길이를 가진 순서쌍 [$(x_1,\cdots,x_n)$, where $x_i \in \{A,C,G,T\}$]의 집합으로 구성된다.

[0026] 본 명세서에서, DNA 코드의 각 원소[$(x_1,\cdots,x_n)$]는 코드워드(codeword, 부호어)라고 정의된다.

[0027] 본 명세서에서, 코드워드는 DNA 코드의 길이 2n에 대응하는 개수의 기호를 포함할 수 있으며, 각 기호가 뉴클레오티드에 대응한다.

[0028] 본 명세서에서, DNA 코드워드의 변형은 기호의 삽입오류(코드워드의 기호들 사이에 임의의 기호가 삽입 되는 오류), 삭제오류(코드워드의 기호중 하나가 삭제 되는 오류) 및 치환오류(코드워드의 기호중 하나가 다른 기호로 바뀌는 오류) 중 적어도 하나가 수행된 것을 의미한다.

[0029] 이하, 첨부 도면을 참조하여 본 발명의 실시예를 상세하게 설명한다. 그러나 다음에 예시하는 본 발명의 실시예는 여러 가지 다른 형태로 변형될 수 있으며, 본 발명의 범위가 다음에 상술하는 실시 예에 한정되는 것은 아니다. 본 발명의 실시예는 당업계에서 통상의 지식을 가진 자에게 본 발명을 보다 완전하게 설명하기 위하여 제공된다.

[0031] 도 1은 본 발명의 실시예에 따른 DNA 코드 생성 장치를 도시한다.

[0032] 도 1을 참고하면, 본 발명의 실시예에 따른 DNA 코드 생성 장치(10)는 데스크톱 컴퓨터, 노트북 컴퓨터, 모바일 폰, 태블릿 등일 수 있다. DNA 코드 생성 장치(10)는 시스템 버스를 통해 연결된 프로세서, RAM(Random Access Memory), 비 일시적인 저장 매체, 네트워크 인터페이스, 디스플레이 스크린 및 입력 장치를 포함한다.

[0033] 프로세서는 DNA 코드 생성 장치(10) 전체의 동작을 지원하기 위한 계산 및 제어 능력을 제공하도록 구성된다. 동작 시스템 및 컴퓨터 판독 가능한 명령어들은 DNA 코드 생성 장치(10)의 비 일시적인 저장 매체 내에 저장되고, 컴퓨터 판독 가능한 명령어들은 다음과 같은 실시예에 따른 데이터 표시 방법을 구현하기 위해 프로세서에 의해 실행될 수 있다. DNA 코드 생성 장치(10)의 RAM은 일시적인 저장 매체내의 동작 시스템 및 컴퓨터 판독 가능한 명령어들에 대한 동작 환경을 제공한다. 네트워크 인터페이스는 통신을 위해 네트워크에 연결되도록 구성된다. DNA 코드 생성 장치(10)의 디스플레이 스크린은 액정 디스플레이 스크린 또는 전자 잉크 등 일 수 있으며, 본 실시예에서, 디스플레이 스크린은 다양한 인터페이스들을 표시하기 위한 DNA 코드 생성 장치(10)의 출력 장치일 수 있다. 입력 장치는 디스플레이 스크린 상에 중첩된 터치 층일 수 있으며, 전자 장치의 하우징 상에 제공된 버튼, 트랙볼 또는 터치 패드일 수도 있고, 사용자에 의한 입력을 위한 외부 키보드, 터치 패드 또는 마우스 등 일 수도 있다. 본 실시예에서, 사용자는 입력 장치를 통해 명령을 입력할 수 있다.

[0034] 도 1은 본 발명의 실시예에 따른 DNA 코드 생성 장치(10)의 구조를 나타내는 예시적인 블록도이며, DNA 코드 생성 장치(10)의 실시예가 이에 제한되는 것은 아니다. 다른 실시예에서, DNA 코드 생성 장치(10)는 도 1에 도시된 구조와 비교하여, 특정 단말은 더 많거나 적은 구성 요소를 포함할 수도 있고, 도 1에 도시되지 않은 다른 구성 요소로 구성되거나 다른 구성을 가질 수도 있다.

[0035] 이하, 도 2 내지 도 5를 참조하여 본 발명의 일 실시예에 따른 DNA 코드 생성 방법에 대해 설명하도록 한다.

[0036] 도 2는 본 발명의 일 실시예에 따른 DNA 코드 생성 방법의 순서도이다. 도 3은 DNA 코드를 생성하기 위한 생성 매트릭스의 일 예시이다. 도 4는 해밍 거리 제약(HD) 및 가역-상보 제약(RC)을 충족하는 제2 DNA 코드의 일 예시이다. 도 5는 해밍 거리 제약(HD), 가역-상보 제약(RC) 및 고정된 GC 개수 제약을 충족하는 제3 DNA 코드의 일 예시이다.

[0037] 일 실시예에서, 도 2에 도시된 바와 같이, DNA 코드 생성 방법이 제공되며, 이 방법은 도 1에 도시된 DNA 코드 생성 장치(10)에 의해 수행될 수 있다. 본 발명의 일 실시예에 따른 DNA 코드 생성 방법은 생성 매트릭스($I_n \mid M$)를 준비하는 단계(S100); 상기 준비된 생성 매트릭스를 통해 최소 해밍 거리(d)를 충족하는 제1 DNA 코드를 생성하는 단계(S110); 상기 제1 DNA 코드에서 자기-가역-상보적(self-reverse-complementary) 특성을 가진 코드워드를 제외하여 제2 DNA 코드를 구성하는 단계(S120); 및 상기 제2 DNA 코드에서 고정된 GC 개수 제약을 충족하는 코드워드를 추출하여 제3 DNA 코드를 구성하는 단계(S130)를 포함한다.

[0038] DNA 코드 생성 방법은 일정 길이, 구체적으로 2n의 길이를 가진 DNA 코드를 생성하는 컴퓨터로 구현된 방법일 수 있다. 다음과 같은 조건과 정의가 구현된 상태에서, 본 실시예에 따른 생성 방법이 적용될 수 있다.

[0039] DNA은 생명체의 유전 정보를 뉴클레오티드라고 하는 4가지 기본 단위(아데닌(A), 시토신(C), 구아닌(G) 및 티민(T)로 암호화한다. 또한, DNA의 4개의 뉴클레오티드는 $A^c = T$, $T^c = A$, $C^c = G$ 및 $G^c = C$와 같은 상보적인 관계를 가진다. 이러한, DNA의 고유한 특징에 기반하여 DNA 코드(DNA 부호)는 2n의 길이를 가진 순서쌍 [$(x_1, \cdots, x_n)$, where $x_i \in \{A, C, G, T\}$]의 집합으로 구성될 수 있다. DNA 코드의 각 원소[$(x_1, \cdots, x_n)$]은 코드워드(codeword, 부호어)라고 정의될 수 있다. 4개의 뉴클레오티드(A, C, G, T)는 하기 수학식 1과 같은 바이젝션(μ, bijection)을 통해 유한체 GF(4) (Finite field GF(4), $\{0, 1, \omega, \bar{\omega}\}$)로 정의될 수 있다. 이러한, 유한체 위에서 DNA 코드의 생성이 수행될 수 있다.

[0041] [수학식 1]

[0042] $$\mu : GF(4) = \{0, 1, \omega, \bar{\omega}\} \rightarrow \mathfrak{N} = \{A, C, G, T\} \ \ by$$
$$\mu(0) = A, \mu(1) = T, \mu(\omega) = C \ and \ \mu(\bar{\omega}) = G.$$

[0044] 여기서, 유한체의 각 원소들과 DNA 뉴클레오티드와 대응되는 상보적인 관계($1^c = 0$, $0^c = 1$, $\bar{\omega}^c = \omega$, $\omega^c = \bar{\omega}$)가 더 정의될 수 있다. 또한, 유한체 GF(4)를 통해 구성된 DNA 코드는 복수의 코드워드로 구성될 수 있다. 복수의 코드워드는 DNA 코드의 길이 2n에 대응하는 개수의 기호를 포함할 수 있으며, 각 기호가 뉴클레오티드에 대응할 수 있다.

[0045] 즉, DNA 코드에 포함된 특정 코드워드($\mathbf{x}$)는 $(x_1, \cdots, x_n)$로 구성된 복수의 기호를 포함한다. 여기서, $\mathbf{x}$와 상보적인(complement) 코드워드($\mathbf{x}^c$)은 $(x_1^c, \cdots, x_n^c)$, $\mathbf{x}$와 가역적인(reverse) 코드워드($\mathbf{x}^r$)은 $(x_n, \cdots, x_1)$로 나타날 수 있다. 또한, $\mathbf{x}$와 가역-상보적인(reverse-complement) 코드워드($\mathbf{x}^{rc}$)은 $(x_n^c, \cdots, x_1^c)$로 나타날 수 있다. 또한, $\mathbf{x} = \mathbf{x}^r$인 경우, $\mathbf{x}$를 자기-가역적(self-reversible)이라고 정의하고, $\mathbf{x} = \mathbf{x}^{rc}$인 경우, $\mathbf{x}$를 자기-가역-상보적(self-reverse-complementary)라고 정의한다.

[0047] 실시예에 따른 DNA 코드 생성 방법은 상술한 유한체 환경을 기초로, 다음과 같은 제약을 충족하도록 DNA 코드($\mathcal{D}$)를 생성할 수 있다.

[0049]
- Hamming distance constraint (HD): $d(\mathbf{x},\mathbf{y}) \geq t$ for all $\mathbf{x},\mathbf{y} \in \mathcal{D}$ with $\mathbf{x} \neq \mathbf{y}$.
- Reverse-complement constraint (RC): $d(\mathbf{x},\mathbf{y}^{rc}) \geq t$ for all $\mathbf{x},\mathbf{y} \in \mathcal{D}$.
- Fixed GC-content constraint (GC): GC-weights of all codewords of $\mathcal{D}$ are constant.

[0051] 여기서, 해밍 거리 제약(HD), 가역-상보 제약(RC)는 DNA 코드의 치환 오류 수정 능력과 관련된 것으로 해당 능력이 일정 수준 이상이 되어야 되는 제약을 설정하는 것을 의미한다. 즉, DNA 코드 내에 포함된 코드워드 사이의 해밍 거리(d)에 대한 제한을 설정할 수 있다. 해밍 거리란 같은 비트 수를 갖는 코드(부호) 사이에 대응되는 비트 값이 일치하지 않는 개수를 의미하며, 해밍 거리 d가 $d \geq 2a+1$ 이면, a개의 오류를 정정할 수 있다. GC 개수의 고정(GC)은 생성된 DNA 코드가 주로 활용되는 DNA 컴퓨팅 및 DNA 데이터 저장에 필수적인 DNA 가닥 증폭의 효율성 및 정확성을 위해 제약된다.

[0052] 이하, 해밍 거리 제약(HD), 가역-상보 제약(RC) 및 GC 개수 고정 제약(GC)을 충족하는 복수의 DNA 코드를 용이하게 생성할 수 있는 실시예에 따른 DNA 코드 생성 방법의 각 단계에 대해 설명하도록 한다.

[0054] 먼저, 생성 매트릭스를 준비한다(S100).

[0055] DNA 코드를 생성하기 위한 생성 매트릭스를 준비한다. 생성하려는 DNA 코드의 조건(길이)에 따라 생성 매트릭스가 준비될 수 있다. DNA 코드 길이(2n)에 대응하여, n차 단위 매트릭스(I_n)와 n x n 매트릭스(M)로 구성된 생성 매트릭스($I_n \mid M$)가 준비될 수 있다. 여기서, n은 하나의 뉴클레오티드에 대응하는 하나의 기호를 의미할 수 있다. n x n 매트릭스(M)가 직교 매트릭스(orthogonal matrix)인 경우, 생성 매트릭스에서 생성되는 DNA 코드는 자기 쌍대 코드(self-dual code)에 해당한다.

[0056] 쌍대 코드(dual code, $\mathcal{C}^{\perp}$)란 유한체 상의 선형 코드($\mathcal{C}$)에 대하여 그 검사 행렬을 생성 행렬로 하는 선형 코드를 의미하며, 본래의 선형 코드에 대하여 직교보 공간을 이룬다. 즉, 쌍대 코드(dual code, $\mathcal{C}^{\perp}$)는 $\mathcal{C}^{\perp} = \{\mathbf{v} \in GF(4)^n \mid \mathbf{v} \cdot \mathbf{w} = 0 \text{ for all } \mathbf{w} \in \mathcal{C}\}$와 같이 정의될 수 있다.

[0057] 여기서, $\mathcal{C} \subset \mathcal{C}^{\perp}$인 경우, 선형 코드($\mathcal{C}$)를 자기 직교(self-orthogonal)라고 정의하며, $\mathcal{C} = \mathcal{C}^{\perp}$인 경우, 선형 코드($\mathcal{C}$)를 자기 쌍대(self-dual)이라고 정의한다. 즉, 듀얼 코드와 동일한 선형 코드를 자기 쌍대 코드라 한다.

[0058] 여기서, n x n 매트릭스[$M = (m_{ij})_{n\times n}$]이 transpose된 매트릭스를 $M^T = (m_{ji})_{n\times n}$, n x n 매트릭스[$M = (m_{ij})_{n\times n}$]이 flip-transpose된 매트릭스를 $M^F = (m_{n-j+1,n-i+1})_{n\times n}$, n x n 매트릭스[$M = (m_{ij})_{n\times n}$]가 column reversed 매트릭스를 $M^R = (m_{i,n-j+1})_{n\times n}$라고 정의한다. 또한, 단위 매트릭스($I_n$)의 column reversed 매트릭스를 R_n이라 정의하며, $MM^T = I$인 매트릭스(M)을 직교성(orthogonal)을 가진다고 정의하고, $M = M^F$인 매트릭스(M)을 부대칭 행렬 또는 반대각성(persymmetric)을 가진다고 정의한다.

[0059] n x n 매트릭스(M)가 직교 매트릭스(orthogonal matrix)로 구성된 생성 매트릭스($I_n \mid M$)는 이러한 자기 쌍대(self-dual) 특성을 가지도록 코드를 생성할 수 있다.

[0060] 또한, n x n 매트릭스(M)가 직교성(orthogonal) 및 반대각성(persymmetric)을 가진 경우, 생성 매트릭스($I_n \mid M$)에서 생성되는 자기 쌍대 코드는 가역적(reversible) 특성을 더 포함할 수 있다. 어떤 코드의 모든 코드워드를 그 기호의 앞뒤 순서를 반전시켜도 집합(set)으로서 변하지 않는 코드를 가역적(reversible)이라고 할 수 있다. 자기 쌍대 코드가 가역적인 특성을 가진 경우, 해당 자기 쌍대 코드를 가역성 자기 쌍대 코드라 정의할 수 있다. n x n 매트릭스(M)가 직교성(orthogonal) 및 반대각성(persymmetric)을 가진 경우, 생성 매트릭스($I_n \mid M$)에서 생성되는 코드는 가역적 자기 쌍대 코드(reversible self-dual code, RSD code)에 해당한다.

[0061] n x n 매트릭스(M)가 직교성(orthogonal)및 반대각성(persymmetric)을 가질 때, 생성되는 코드가 가역성을 가지

게 되는 이론적인 근거는 다음과 같으며, 더욱 상세한 설명은 "Construction of reversible self-dual codes, Finite Fields and Their Applications 67 (2020) 101714"이 참조될 수 있다.

[0062] 여기서, n x n 매트릭스(M)는 하기와 같은 특징 중 2가지를 만족하는 경우, 나머지 한가지도 충족하게 된다.

[0063] (i) M *is orthogonal.*
(ii) $(M^R)^2 = I_n$.
(iii) M *is persymmetric.*

[0064] 즉, $M^F = R_n M^T R_n$ 이고, M이 직교성(orthogonal)을 가지는 것으로 추정할 때, 상기 두 번째 특성으로부터 다음과 같은 관계가 도출될 수 있다.

[0065]
$$\begin{aligned}(M^R)^2 = I_n &\Leftrightarrow (MR_n)^{-1} = MR_n \\ &\Leftrightarrow R_n M^{-1} = MR_n \\ &\Leftrightarrow R_n M^T R_n = M \\ &\Leftrightarrow M^F = M.\end{aligned}$$

[0067] 여기서, 자기 쌍대 코드($\mathcal{C}$)은 $(M^R)^2 = I_n$ 인 경우 가역적인 특성을 나타낼 수 있다. 즉, 상술한 관계에 따라, M이 반대각성(persymmetric)을 가지는 경우, 자기 쌍대 코드($\mathcal{C}$)은 가역성(reversible)을 가질 수 있다.

[0068] $\mathcal{C}$ 가 가역적이라고 가정할 때, 생성 매트릭스($I_n \mid M$)의 column reversed 매트릭스(G)는 다음과 같이 정의된다: $G = (\ M^R \mid R_n\ )$. 또한, 매트릭스(G) 또한 자기 쌍대 코드($\mathcal{C}$)를 생성할 수 있다. 여기서, $\mathcal{C}$가 자기 쌍대이므로 M은 직교성(orthogonal)을 가질 수 있으며, M 및 M^R 의 역변환이 가능하다. $(M^R)^{-1}G$ 또한 자기 쌍대 코드($\mathcal{C}$)를 생성하는 생성 매트릭스의 기본형으로 $(M^R)^{-1}G = (\ (M^R)^{-1}M^R \mid (M^R)^{-1}R_n\ ) = (\ I_n \mid (M^R)^{-1}R_n\ )$ 와 같이 구성될 수 있다. 이에 따라, $(M^R)^2 = I_n$ 이 하기와 같이 충족되는 것을 알 수 있다.

[0069]
$$\begin{aligned}(I_n \mid (M^R)^{-1}R_n) = (I_n \mid M) &\Leftrightarrow (M^R)^{-1}R_n = M \\ &\Leftrightarrow (MR_n)^{-1} = MR_n \\ &\Leftrightarrow I_n = (M^R)^2.\end{aligned}$$

[0071] 즉, DNA 코드 길이(2n)에 대응하여, n x n 매트릭스(M)가 직교성(orthogonal)및 반대각성(persymmetric)을 가져 가역적 자기 쌍대(reversible self-dual, RSD) 코드를 생성할 수 있는 생성 매트릭스($I_n \mid M$)가 본 단계(S100)에서 준비될 수 있다.

[0072] 이러한, 생성 매트릭스($I_n \mid M$)는 복수 개일 수 있으며, 생성하려는 DNA 코드 길이(2n)에 대응하고, 상술한 특징을 가진 구성을 가진 복수의 생성 매트릭스가 준비될 수 있다. 도 3은 DNA 코드 길이(6)에 대응하여, I_3 단위 행렬과 3 X 3 매트릭스로 구성된 생성 매트릭스의 일 예시를 나타낸다.

[0074] 다음으로, 최소 해밍 거리(d)를 충족하는 제1 DNA 코드를 생성한다(S110).

[0075] 생성된 코드들의 최소 해밍 거리가 설정된 거리(d) 이상이 되도록 이전 단계(S100)에서 생성된 복수의 생성 매트릭스 중에서 특정 생성 매트릭스가 선택될 수 있다. 즉, 생성 매트릭스에서 해밍 거리에 영향을 미치는 부분은 n x n 매트릭스(M)일 수 있다. 따라서, 생성된 코드들의 최소 해밍 거리가 설정된 거리(d) 이상이 되도록 하는 n x n 매트릭스(M)을 포함하는 생성 매트릭스가 선택될 수 있다.

[0076] 선택된 생성 매트릭스를 이용하여, 2n 길이의 제1 DNA 코드($\mathcal{C}$)를 생성한다. 생성 매트릭스는 DNA 코드를 효율적으로 생성하는 도구에 해당하며, 행렬 곱셈에 의해 코드의 효율적인 생성이 가능하다. 즉, 생성 매트릭스와

연산을 위한 메시지 벡터가 제공될 수 있으며, 생성 매트릭스의 행에 따라 제1 DNA 코드가 생성될 수 있다.

[0077] 본 단계(S110)에서, 수행한 상술한 선택 과정에 따라 생성 매트릭스를 통해 생성되는 제1 DNA 코드($\mathcal{C}$)에 포함된 모든 코드워드들은 상호간의 거리가 최소 해밍 거리(d) 이상일 수 있다. 즉, 생성 매트릭스를 통해 생성되는 DNA 코드($\mathcal{C}$)는 적어도 해밍 거리 제약(HD)을 충족할 수 있다.

[0078] 또한, 이전 단계(S100)에서 설명한 바와 같이, 본 실시예에서 생성 매트릭스의 n x n 매트릭스(M)는 직교성(orthogonal)및 반대각성(persymmetric)을 가지기에, 제1 DNA 코드($\mathcal{C}$)은 가역적 자기 쌍대(reversible self-dual, RSD) 코드에 해당한다.

[0079] 가역적 자기 쌍대(RSD) 코드는 다음과 같은 특성을 가진다. 먼저, 본 실시예에 따른 유한체 환경에서 생성되는 DNA 코드는 유클리드 자기 쌍대 코드인 바, 모두 올-원(all-one) 코드워드(**1**)을 포함할 수 있다.

[0080] 여기서, 제1 DNA 코드($\mathcal{C}$)에 포함된 코드워드를 $\mathbf{x}$라고 가정할 때, $\mathbf{x}$와 상보적인(complement) 코드워드 $\mathbf{x}^c$, $\mathbf{x}$와 가역적인(reverse) 코드워드 $\mathbf{x}^r$ 및 $\mathbf{x}$와 가역-상보적인(reverse-complement) 코드워드 x^{rc} 또한 제1 DNA 코드($\mathcal{C}$)에 포함될 수 있다. 이에 대한 이론적 증명은 다음과 같다.

[0081] 제1 DNA 코드($\mathcal{C}$)는 가역적인(reversible) 특성을 가지므로, 제1 DNA 코드($\mathcal{C}$)가 코드워드 $\mathbf{x}$를 포함하는 경우, $\mathbf{x}^r$ 또한 제1 DNA 코드($\mathcal{C}$)에 포함될 수 있다. 또한, 올 원 코드워드(1)이 제1 DNA 코드($\mathcal{C}$)에 포함되므로, $\mathbf{x}^{rc} = \mathbf{x}^r + \mathbf{1}$ 및 $\mathbf{x}^c = \mathbf{x} + \mathbf{1}$ 에 따라, $\mathbf{x}^c$ 및$\mathbf{x}^{rc}$ 또한 제1 DNA 코드($\mathcal{C}$)에 포함될 수 있다. 제1 DNA 코드($\mathcal{C}$)의 특성에 따라 $\mathbf{x}$, $\mathbf{x}^c, \mathbf{x}^r$ 및$\mathbf{x}^{rc}$를 모두 포함할 수 있다.

[0082] 다만, 코드워드 $\mathbf{x}$와 가역-상보적인(reverse-complement) 코드워드 x^{rc}가 동일한 자기-가역-상보적(self-reverse-complementary) 특성을 가진 코드워드 $\mathbf{x}$도 제1 DNA 코드($\mathcal{C}$)에 포함될 수 있다.

[0083] 이러한, 자기-가역-상보적(self-reverse-complementary) 특성을 가진 코드워드는 오리지널 코드워드 $\mathbf{x}$와 가역-상보적인(reverse-complement) 코드워드 x^{rc}가 동일하므로 $\mathbf{x}$ - x^{rc} 사이의 최소 해밍 거리(d)가 0일 수 있다. 예를 들어, ACGT로 구성된 코드워드 X는 가역-상보 변환을 하면 본래의 형태인 ACGT에 해당하기에, $\mathbf{x}$- x^{rc} 사이의 최소 해밍 거리(d)가 0일 수 있다. 이러한 자기-가역-상보적(self-reverse-complementary) 코드는 가역-상보 제약(RC)에 따른 제약을 충족하지 못하는 코드에 해당한다.

[0084] 제1 DNA 코드($\mathcal{C}$)를 생성 매트릭스로 생성할 시에는, 매트릭스 연산된 결과에 해당하는 코드워드 사이의 거리만이 고려되므로, 실제 가역-상보적으로 변환된 코드워드와 오리지널 코드워드의 동일 여부가 검출되지 않는다. 즉, 제1 DNA 코드($\mathcal{C}$)를 생성 매트릭스로 생성할 시에는 자기-가역-상보적(self-reverse-complementary) 코드워드의 검출이 불가능하다. 따라서, 후술하는 단계에서, 자기-가역-상보적(self-reverse-complementary) 코드워드를 제외하는 과정이 더 수행될 수 있다.

[0086] 다음으로, 생성된 DNA 코드에서 자기-가역-상보적 코드워드를 제외하여 제2 DNA 코드를 구성한다(S120).

[0087] 제1 DNA 코드($\mathcal{C}$)에서 자기-가역-상보적(self-reverse-complementary) 특성을 가진 코드워드를 삭제하는 과정이 본 단계(S120)에서 수행될 수 있다. 즉, 본 단계(S120)를 통해 RC 제약까지 더 충족하는 제2 DNA 코드가 생성될 수 있다.

[0088] 여기서, 본 단계(S120)는 제1 DNA 코드($\mathcal{C}$)를 특정 조건에 따라 복수의 코셋(coset)으로 분해하고, 분해된 코셋(coset)을 기준으로 자기-가역 상보적 코드를 검출 및 제외할 수 있다. 또한, 본 단계(S120)는 자기-가역 상보적 코드워드가 존재하는 지 여부를 먼저 판단하고, 자기-가역 상보적 코드가 존재하는 경우, 해당 코드워드를 제외할 수 있다. 큰 카디널리티(large cardinality)를 가진 제1 DNA 코드($\mathcal{C}$)가 아닌 분해된 코셋(coset)을 기준으로 연산을 수행하기에, 데이터 연산량이 감소될 수 있다.

[0089] 본 단계(S120)는 다음과 같은 알고리즘 1을 통해 수행될 수 있으나, 이에 한정되는 것은 아니다.

[0091]
Input:
$\mathcal{C}$:= *an RSD code of length* $2n$ *with generator matrix* $[I_n \mid M]$ *and minimum distance* d

Output:
$\mathcal{D}$:= *a DNA code with Hamming distance* $\geq d$ *and reverse-complement distance* $\geq d$

step 1. *[Obtain* $\phi(\mathcal{C})$ *and* $\mathcal{C}_0$*]*
Obtain $\phi(\mathcal{C})$ *generated by the matrix* $(I_n + MR_n \mid M + R_n)$ *and obtain* $\mathcal{C}_0$ *from the dual code of* $\pi(\psi(\mathcal{C}))$.
step 2. *[Decompose* $\mathcal{C}$ *into Cosets]*
Decompose $\mathcal{C}$ *into cosets of* $\mathcal{C}_0$*, and let* $\mathscr{S}$ *be the set of cosets.*
step 3. *[Remove self-reverse-complementary codwords]*
If there exists $\mathbf{u}$ *in* $\mathcal{C}$ *such that* $\phi(\mathbf{u}) = \mathbf{1}$*, then the coset* $\mathbf{u} + \mathcal{C}_0$ *is the set of all self-reverse-complementary codewords in* $\mathcal{C}$*. Remove the coset from* $\mathscr{S}$.
step 4. *[Extract codewords from a coset]*
For a coset $\mathcal{C}_i \in \mathscr{S}$*, set* $\mathcal{C}_i' = \emptyset$*. Take a codeword* $\mathbf{x} \in \mathcal{C}_i$*. Set* $\mathcal{C}_i' := \mathcal{C}_i' \cup \{\mathbf{x}, \mathbf{x}^r\}$ *and* $\mathcal{C}_i := \mathcal{C}_i - \{\mathbf{x}, \mathbf{x}^r, \mathbf{x}+\mathbf{1}, \mathbf{x}^r+1\}$*. Repeat this until* $\mathcal{C}_i$ *becomes empty.*
step 5. *[Iteration]*
Set $\mathcal{D} = \mathcal{D} \cup \mathcal{C}_i'$ *and repeat the step 4 for all the cosets in* $\mathscr{S}$.
step 6. *[End]*
$\mathcal{D}$ *is a DNA code with Hamming distance and reverse-complement distance* $\geq d$.

[0092] (알고리즘 1)

[0094] 구체적으로, 가역적 자기 쌍대(RSD) 코드, 제1 DNA 코드($\mathcal{C}$) 내의 모든 자기-가역적(self-reversible) 코드워드의 부분 집합을 $\mathcal{C}_0$라고 정의할 때, $\phi : \mathcal{C} \to \mathcal{C}$ by $\phi(\mathbf{x}) = \mathbf{x} + \mathbf{x}^r$와 같은 선형 맵이 정의될 수 있다.

[0095] 여기서, $\phi(\mathcal{C}) \subset \mathcal{C}_0$ 및 $\mathcal{C}_0 = ker(\phi)$에 따라, $\mathcal{C}_0$는 $\phi(\mathcal{C})$를 그것의 서브코드로 포함하는 $\mathcal{C}$의 선형 서브코드에 해당할 수 있으며, $|\mathcal{C}| = |\mathcal{C}_0||\phi(\mathcal{C})|$와 같은 관계가 정의될 수 있다.

[0096] 또한, 선형 맵(ϕ)는 $\mathcal{C}$의 서브코드 $\phi(\mathcal{C})$를 생성하는 매트릭스($I_{2n} + R_{2n}$)로 $G(I_{2n} + R_{2n}) = (I_n + MR_n \mid M + R_n)$과 같이 표현될 수 있다. 또한, $\mathcal{C}_0$를 생성하기 위한 projection map[$\pi(\psi(\mathcal{C}))$]이 정의될 수 있으며, 생성된 $\pi(\psi(\mathcal{C}))$을 기초로 $\mathcal{C}_0$을 획득할 수 있다(알고리즘 1의 step 1).

[0097] 다음으로, 제1 DNA코드를 자기-가역적 특성을 가진 코드워드들의 집합을 이용하여 코셋(coset)으로 분리한 후, 그 중 제1 DNA코드에서 자기-가역-상보적 특성을 가진 코드워드의 코셋이 있다면 제거하는 단계가 수행된다(알고리즘 1의 step 2,3).

[0098] 모든 자기-가역적(self-reversible) 코드워드의 부분 집합인 $\mathcal{C}_0$을 이용하여 가역적 자기 쌍대(RSD) 코드($\mathcal{C}$)를 복수의 코셋(coset, 잉여류)으로 분해(decompose)할 수 있다.

[0099] 분해된 복수의 코셋(coset)의 집합을 $\mathscr{S}$이라 정의할 수 있다(알고리즘 1의 step 2). 자기 가역적(self-reversible) 코드워드의 부분 집합인 $\mathcal{C}_0$를 이용하여 복수의 코셋을 형성함에 따라, 코셋(coset)은 자기 쌍대(RSD) 코드($\mathcal{C}$)의 특성을 동일하게 가질 수 있다. 즉, 코셋(coset)에 임의의 코드워드 x가 존재하는 경우, 코셋(coset)에는 $\mathbf{x}^c, \mathbf{x}^r$ 및 $\mathbf{x}^{rc}$가 모두 포함될 수 있다. 따라서, 코셋(coset)을 통한 분산 처리가 가능할 수 있다. 분해된 복수의 코셋에는 여러 코드워드들이 포함되어 있으며, 몇몇의 코드워드는 자기-가역-상보적(self-reverse-complementary) 코드워드들에 해당하며, 이러한 코드워드들을 제외하는 과정이 다음 스탭에서 수행되게 된다.

[0100] 여기서, 자기-가역-상보적(self-reverse-complementary) 코드워드의 존재 여부는 $\mathbf{1} \in \phi(\mathcal{C})$을 통해 확인될 수 있다. 즉, $\mathbf{1} \in \phi(\mathcal{C})$이 만족되는 경우, 생성된 가역적 자기 쌍대(RSD) 코드($\mathcal{C}$)에 자기-가역-상보적(self-reverse-

complementary) 코드워드가 존재할 수 있다. 구체적으로, $\mathbf{x} \in \mathcal{C}$를 자기-가역-상보적(self-reverse-complementary) 코드워드, $\mathbf{x} = \mathbf{x}^{rc} = \mathbf{x}^{r} + \mathbf{1}$에 해당하는 것으로 가정한다면, $\phi(\mathbf{x}) = \mathbf{x} + \mathbf{x}^{r} = \mathbf{1}$에 해당하고, $\mathbf{1} \in \phi(\mathcal{C})$이 성립하는 것을 알 수 있다. 또한, $\mathbf{1} \in \phi(\mathcal{C})$이라면, $\mathbf{x} + \mathbf{x}^{r} = \mathbf{1}$을 충족하는 코드워드 $\mathbf{x}$가 존재하며, $\mathbf{x} = \mathbf{x}^{r} + \mathbf{1} = \mathbf{x}^{rc}$이 충족되게 된다. 즉, $\mathbf{1} \in \phi(\mathcal{C})$인 조건을 통해 생성된 가역적 자기 쌍대(RSD) 코드($\mathcal{C}$)에 자기-가역-상보적(self-reverse-complementary) 코드워드가 존재 여부를 판단할 수 있다.

[0101] 또한, $\mathcal{C}$에 자기-가역-상보적(self-reverse-complementary) 특성을 가진 코드워드 x가 존재하는 경우, 코셋(coset) $\mathbf{x} + \mathcal{C}_0$은 $\mathcal{C}$의 모든 자기-가역-상보적 코드워드의 집합일 수 있다. 따라서, 코셋(coset) $\mathbf{x} + \mathcal{C}_0$을 $\mathscr{S}$에서 제거함으로써 자기-가역-상보적 코드를 가진 코셋(coset)이 제외될 수 있다(알고리즘 1의 step 3). 여기서, $\mathbf{1} \notin \phi(\mathcal{C})$라면, 상술한 자기-가역-상보적(self-reverse-complementary) 특성을 가진 코드워드 x를 제외하는 과정이 진행되지 않을 수 있다.

[0102] 또한, 자기-가역-상보적(self-reverse-complementary) 코드워드를 포함하는 코셋이 제외된 상태에서 코드워드들을 추출하는 과정이 수행되어 제2 DNA 코드($\mathcal{D}$)가 구성된다. 즉, 상기 DNA코드의 각각의 코셋에서 임의의 코드워드($\mathbf{x}$)와 해당 코드워드의 가역적(reverse) 코드워드($\mathbf{x}^{r}$)를 선택하여 모음과 동시에 상보적(complement) 코드워드($\mathbf{x}^{c}$, $(x_1^c, \cdots, x_n^c)$)와 가역-상보적(reverse-complement) 코드워드($\mathbf{x}^{rc}$, $(x_n^c, \cdots, x_1^c)$)은 제거함으로써 역-상보 제약(RC constraint)를 만족하는 제2 DNA 코드를 구성한다(알고리즘 1의 step 4, 5). 이에 따라, 제2 DNA 코드($\mathcal{D}$)에 포함된 DNA 코드의 개수는 제1 DNA 코드($\mathcal{C}$)보다 절반으로 줄어들 수 있다.

[0103] 본 단계(S120)의 상술한 과정을 통해, 제1 DNA 코드($\mathcal{C}$)에서 $\mathbf{x} - \mathbf{x}^{rc}$ 사이의 거리가 0에 해당하는 자기-가역-상보적(self-reverse-complementary) 코드워드 가 모두 제외될 수 있으며, 해밍 거리 제약(HD)뿐만 아니라, 가역-상보 제약(RC)에 따른 제약을 충족할 수 있는 코드워드로 구성된 제2 DNA 코드($\mathcal{D}$)가 생성될 수 있다. 즉, 코드워드들간의 최소 거리가 d 이상이고, 가역-상보적인 코드 사이의 최소 거리가 d 이상인 코드워드만으로 구성된 제2 DNA 코드($\mathcal{D}$)가 도 4의 예시와 같이 생성될 수 있다.

[0104] 도 4는 길이 6, 해밍 제약 거리(d)가 3인 조건 하에서, 해밍 거리 제약(HD) 및 가역 상보 제약(RC)을 충족하도록 생성된 제2 DNA 코드($\mathcal{D}$)를 예시적으로 도시한 것으로, 32개의 코드워드가 생성된 것을 알 수 있다. 32개의 코드워드는 6개의 기호(symbol)로 구성될 수 있으며, 각 기호는 뉴클레오티드에 대응한다.

[0105] 다음으로, 제2 DNA 코드에서 GC 개수 제약을 충족하는 코드워드를 추출하여 제3 DNA 코드를 구성한다(S130).

[0106] GC 개수의 제약은 DNA 가닥 증폭의 효율성 및 정확성을 위해 하나의 코드워드 내에 포함된 GC 개수를 제약하는 것으로, DNA 컴퓨팅 및 DNA 데이터 저장에 효율적으로 활용되기 위해서 필요한 제약일 수 있다. 즉, 하나의 코드워드 내에 포함된 기호가 G 또는 C에 해당하는 개수를 제한하는 것으로, GC 개수 k는 DNA의 길이(2n)을 고려하여 결정될 수 있다.

[0107] 본 단계(S130)를 통해 GC 개수 제약을 충족하는 코드워드로 구성된 제3 DNA 코드가 생성될 수 있다. 본 단계(S130)는 제2 DNA 코드에서 GC 개수 제약을 충족하는 코드워드 개수를 산출하는 단계; 상기 산출된 코드워드 개수를 고려하여 GC 개수 제약을 충족하는 코드워드를 추출하고, 추출된 코드워드로 상기 제3 DNA 코드를 생성하는 단계를 포함한다.

[0108] 제2 DNA 코드에서 고정된 GC 개수 제약을 충족하는 코드워드 개수를 산출하는 단계는, 제1 DNA 코드($\mathcal{C}$)를 기준으로 GC 개수 제약을 충족하는 코드워드 개수($CWE_{\mathcal{C}}(a,a,b,b)$)를 먼저 산출할 수 있다. 제2 DNA 코드($\mathcal{D}$)에서 고정된 GC 개수 제약을 충족하는 코드워드 개수($GCW_{\mathcal{D}}(a,b)$)는 먼저 산출된 $CWE_{\mathcal{C}}(a,a,b,b)$ 개수를 이용하여 산출될 수 있다.

[0109] 제1 DNA 코드($\mathcal{C}$)에 포함된 각 코드워드(A, T, C, G)의 비중은 하기 수학식 1과 같이 계산될 수 있다.

[0111] [수학식 1]

[0112] $$CWE_{\mathcal{C}}(a,b,c,d)=\sum_{c\in\mathcal{C}}a^{n_0(c)}b^{n_1(c)}c^{n_\omega(c)}d^{n_{\bar{\omega}}(c)}$$

[0114] 여기서, $n_t(c)$는 제1 DNA 코드($\mathcal{C}$)에 포함된 각 코드워드(c)의 개수를 의미하며, $\mu(\omega)=C$ 및 $\mu(\bar{\omega})=G$로 설정되었기에, 제1 DNA 코드($\mathcal{C}$)의 GC 개수 산출식은 하기 수학식 2와 같이 정리될 수 있다.

[0116] [수학식 2]

[0117] $$GCW_{\mu(\mathcal{C})}(a,b):=CWE_{\mathcal{C}}(a,a,b,b)$$

[0119] 다음으로, 산출된 $CWE_{\mathcal{C}}(a,a,b,b)$ 개수를 이용하여 제2 DNA 코드($\mathcal{D}$)에서 GC 개수 제약을 충족하는 코드워드 개수($GCW_{\mathcal{D}}(a,b)$)를 산출한다. 여기서, 제2 DNA 코드($\mathcal{D}$)를 재구성하는 단계에서, 코드워드 개수가 제1 DNA 코드($\mathcal{C}$) 대비 절반으로 감소되었기에, $GCW_{\mathcal{D}}(a,b)$는 $CWE_{\mathcal{C}}(a,a,b,b)$의 1/2의 개수에 해당할 수 있다. 다만, 상기 제2 DNA 코드에서 GC 개수 제약을 충족하는 코드워드의 개수를 산출하는 과정은 제1 DNA 코드($\mathcal{C}$)에 자기-가역-상보적(self-reverse-complementary) 코드가 존재하는 지 여부에 따라 달라질 수 있다. 구체적으로, 제1 DNA 코드($\mathcal{C}$)에 자기-가역-상보적(self-reverse-complementary) 코드가 존재하는 지 여부에 따라 $GCW_{\mathcal{D}}(a,b)$를 구하는 수식은 하기 수학식 3과 같이 달라질 수 있다.

[0121] [수학식 3]

(i) *If* $\mathbf{1}\notin\phi(\mathcal{C})$, *then*

$$GCW_{\mathcal{D}}(a,b)=\frac{CWE_{\mathcal{C}}(a,a,b,b)}{2}.$$

(ii) *If* $\mathbf{1}\in\phi(\mathcal{C})$, *then*

$$GCW_{\mathcal{D}}(a,b)=\frac{CWE_{\mathcal{C}}(a,a,b,b)-CWE_{\mathbf{x}+\mathcal{C}_0}(a,a,b,b)}{2}$$

where $\phi(\mathbf{x})=\mathbf{1}$

[0122]

[0124] 즉, 제1 DNA 코드($\mathcal{C}$)에 자기-가역-상보적(self-reverse-complementary) 코드워드가 포함되지 않는 경우($\mathbf{1}\notin\phi(\mathcal{C})$), 계산된 $CWE_{\mathcal{C}}(a,a,b,b)$에 1/2이 적용되어 $GCW_{\mathcal{D}}(a,b)$이 계산될 수 있다.

[0125] 이와 달리, 제1 DNA 코드($\mathcal{C}$)에 자기-가역-상보적(self-reverse-complementary) 코드워드가 포함되는 경우($\mathbf{1}\in\phi(\mathcal{C})$), 상기 알고리즘 1의 step 3과 같이 코셋(coset) $\mathbf{x}+\mathcal{C}_0$이 $\mathcal{S}$에서 제거되는 점이 더 고려될 수 있다. 즉, $CWE_{\mathcal{C}}(a,a,b,b)$에서 $CWE_{\mathbf{x}+\mathcal{C}_0}(a,a,b,b)$을 차감한 개수에 1/2이 적용되어 $GCW_{\mathcal{D}}(a,b)$이 계산될 수 있

다.

[0126] 도 3 및 도 4의 예시에서, 제1 DNA 코드($\mathcal{C}$)는 자기-가역-상보적(self-reverse-complementary) 코드가 포함되지 않는 경우($\mathbf{1} \notin \phi(\mathcal{C})$) 상태이다. 제1 DNA 코드($\mathcal{C}$)의 GC 개수 4(한 코드 내에서 G 또는 C로 구성된 기호의 개수가 4개)를 충족하는 코드워드의 개수는 $CWE_{\mathcal{C}}(a,a,b,b) = 2a^6+30a^4b^2+30a^2b^4+2b^6$ 와 같이 계산될 수 있다. 따라서, 제2 DNA 코드($\mathcal{D}$)에 포함된 GC 개수 4를 충족하는 코드워드의 개수($GCW_{\mathcal{D}}(a,b)$)는 15개의 코드워드인 것이 산출될 수 있다.

[0127] 다음으로, 제2 DNA 코드($\mathcal{D}$)에서 GC 개수를 충족시키는 코드워드가 상기 계산된 개수만큼 추출되어 제3 DNA 코드를 구성할 수 있다. 프로세서는 해밍 거리 제약(HD), 가역-상보 제약(RC)에 따른 제약을 충족할 수 있는 코드로 구성된 DNA 코드($\mathcal{D}$)에서 제한된 GC 개수를 충족하는 코드워드를 추출하며, 추출된 코드워드로 제3 DNA 코드를 도 5와 같이 구성할 수 있다. 도 5에 도시된 제3 DNA 코드는, 길이 6, 해밍 제약 거리(d)가 3인 조건, GC 개수 4를 충족하는 코드에 해당하며, 15개의 코드워드로 구성되는 것을 알 수 있다. 여기서, 추출해야 되는 목표값이 상술한 바와 같이 먼저 계산되는 바 제한된 GC 개수를 충족하는 코드워드를 추출하는 과정에서 소요되는 연산 부하가 감소하고 추출 결과의 정확성이 더욱 높아질 수 있다.

[0128] 하기 표 2는 본 발명의 일 실시예에 따른 DNA 코드 생성 방법을 통해 생성될 수 있는 DNA 코드의 개수를 종래 알려진 방법과 비교하여 도시한 것이다.

[0130] [표 2]

n	d	$A_4^{RC}(n,d)$	best-known	$A_4^{RC,GC}(n,d,k)$	best-known
4	3	**6**	6 [30]	**6**	6 [11]
6	3	32	62 [30]	15	44 [7]
8	4	120	128 [30]	108	128 [7]
10	4	512	2016 [30]	216	871 [16]
12	6	**2016**	2016 [30]	904	924 [11]
14	6	**8192**	8192 [30]	3003	3712 [11]
16	6	32640	130560 [30]	13284	55424 [1]
18	6	131072	2095104 [30]	43856	699624 [1]
20	8	**523776**	523776 [30]	184244	184756 [11]
22	8	**2097152**	-	**647192**	353496 [1]
24	8	**8386560**	-	3338240	5406464 [1]
26	8	**33554432**	-	9674752	83204800 [1]
28	9	**134209536**	-	**40136128**	40114884 [1]
30	9	**536870912**	-	**145427724**	40114884 [1]
32	10	**2147450880**	-	**601060492**	300533760 [31]
34	10	**8589934592**	-	**2203953584**	1166803110 [31]
36	10	**34359607296**	-	**9076376320**	4537543340 [31]
38	11	**137438953472**	-	**33578000610**	-
40	12	**549755289600**	-	**137846189288**	-
42	12	**2199023255552**	-	**513790477200**	-

[0131]

[0133] 여기서, $A_4^{RC}(n,d)$ 는 길이(n)과 제한 거리(d)에 따른 해밍 거리 제약(HD), 가역-상보 제약(RC)을 충족하는 DNA 코드(제2 DNA 코드)의 개수를 의미하며, $A_4^{RC,GC}(n,d,k)$ 는 길이(n), 제한 거리(d), GC 개수(k)를 충족하는 DNA 코드(제3 DNA 코드)의 개수를 의미한다.

[0134] 종래의 방법으론 DNA 코드가 일정 길이 이상 길어졌을 때 DNA 코드를 생성할 수 없었으나, 본 발명의 실시예에 따른 DNA 코드 생성 방법을 통해서는 종래 계산되기 어려웠던 길이 및 조건의 DNA 코드를 해밍 거리 제약(HD), 가역-상보 제약(RC)을 충족시키면서 산출할 수 있다. 또한, 종래 계산될 수 없었던 길이(n), 제한 거리(d), GC 개수(k)를 충족하는 DNA 코드의 추출이 가능할 수 있다. 이에 따라, DNA 컴퓨팅 및 DNA 데이터 저장 등의 기술이 연구 개발되는 것이 지원될 수 있다.

[0136] 다음으로, 본 발명의 다른 실시예에 따른 DNA 코드 생성 방법에 대해 설명하도록 한다.

[0137] 도 6은 본 발명의 다른 실시예에 따른 DNA 코드 생성 방법의 순서도이다. 도 7은 해밍 거리 제약(HD), 가역-상보 제약(RC) 및 레벤슈타인 거리 제약(DI)를 충족하는 제3 DNA 코드의 일 예시이다. 도 8은 해밍 거리 제약(HD), 가역-상보 제약(RC), 레벤슈타인 거리 제약(DI) 및 고정된 GC 개수 제약을 충족하는 제4 DNA 코드의 일 예시이다.

[0138] 다른 실시예에서, 도 6에 도시된 바와 같이, DNA 코드 생성 방법이 제공되며, 이 방법은 도 1에 도시된 DNA 코드 생성 장치(10)에 의해 수행될 수 있다.

[0139] 본 발명의 다른 실시예에 따른 DNA 코드 생성 방법은 생성 매트릭스 $I_n \mid M$를 준비하는 단계(S200); 상기 준비된 매트릭스를 통해 최소 해밍 거리(d)를 충족하는 제1 DNA 코드를 생성하는 단계(S210); 상기 제1 DNA 코드에서 자기-가역-상보적(self-reverse-complementary) 특성을 가진 코드워드 특성을 가진 코드워드를 제외하여 제2 DNA 코드를 구성하는 단계(S220); 상기 제2 DNA 코드의 코드워드 각각의 n번째 기호와 n+1번째 기호 사이에 2개의 기호를 삽입하여 2n+2 길이를 가진 제3 DNA 코드를 생성하는 단계(S230); 상기 제3 DNA 코드에서 GC 개수 제약을 충족하는 코드워드를 추출하여 제4 DNA 코드를 구성하는 단계(S240)를 포함한다.

[0140] 본 실시예에 따른 DNA 코드 생성 방법은 일정 길이, 구체적으로 2n+2의 길이를 가진 DNA 코드를 생성하는 컴퓨터로 구현된 방법일 수 있다. 상술한 일 실시예에 따른 생성 방법과 동일한 조건과 정의가 구현된 상태에서 본 실시예에 따른 생성 방법이 적용될 수 있다.

[0142] 본 실시예에 따른 DNA 코드 생성 방법은 다음과 같은 제약을 충족하도록 DNA 코드($\mathcal{D}$)를 생성할 수 있다.

[0144]
- Hamming distance constraint (HD): $d(\mathbf{x},\mathbf{y}) \geq t$ for all $\mathbf{x},\mathbf{y} \in \mathcal{D}$ with $\mathbf{x} \neq \mathbf{y}$.
- Reverse-complement constraint (RC): $d(\mathbf{x},\mathbf{y}^{rc}) \geq t$ for all $\mathbf{x},\mathbf{y} \in \mathcal{D}$.
- Fixed GC-content constraint (GC): GC-weights of all codewords of $\mathcal{D}$ are constant.
- Deletion/insertion constraint (DI) : $dd(\mathbf{x},\mathbf{y}) \geq e$ for all $\mathbf{x},\mathbf{y} \in \mathcal{D}$ with $\mathbf{x} \neq \mathbf{y}$.

[0146] 즉, 상술한 일 실시예에 따른 DNA 코드 생성 방법과 비교하여, 다른 실시예에 따른 DNA 코드 생성 방법은 DI 제약을 더 충족하는 DNA 코드를 생성할 수 있다. dd는 레벤슈타인 거리(Levenshtein distance)를 의미하고, 두 문자열이 얼마나 다른지를 나타내는 거리를 의미하며, 문자를 삽입, 삭제하여 다른 문자열로 변형하는데 필요한 최소 횟수(e)를 의미할 수 있다.

[0147] 본 실시예에 따라 생성된 DNA 코드 생성 방법은 적어도 단일 오류 (삭제/삽입)를 검출할 수 있는 DNA 코드를 생성할 수 있다.

[0148] 본 실시예에 따른 DNA 코드 생성 방법의 단계(S200, S210, S220)은 상술한 일 실시예에 따른 DNA 코드 생성 방법의 단계(S100, S110, S120)와 실질적으로 동일하고, 본 실시예에 따른 DNA 코드 생성 방법의 단계(S240)는 상술한 일 실시예에 따른 DNA 코드 생성 방법의 단계(S130)과 실질적으로 동일할 수 있다. 본 실시예에 따른 DNA 코드 생성 방법의 설명과 이해를 위해 상술한 일 실시예에 따른 설명이 참조될 수 있다.

[0149] 먼저, 생성 매트릭스를 준비한다(S200).

[0150] DNA 코드 길이(2n+2)에 대응하여, n차 단위 매트릭스(I_n)와 n x n 매트릭스(M)로 구성된 생성 매트릭스($I_n \mid M$)가 준비될 수 있다. 즉, DNA 코드 길이(2n)에 대응하여, n x n 매트릭스(M)가 직교성(orthogonal)및 반대각성(persymmetric)을 가져 가역적 자기 쌍대(reversible self-dual, RSD) 코드를 생성할 수 있는 생성 매트릭스($I_n \mid M$)가 본 단계(S200)에서 준비될 수 있다.

[0151] 다음으로, 최소 해밍 거리(d)를 충족하는 제1 DNA 코드를 생성한다(S210).

[0152] 생성된 코드들의 최소 해밍 거리가 설정된 거리(d) 이상이 되도록 하는 n x n 매트릭스(M)을 포함하는 생성 매

트릭스가 선택될 수 있다. 선택된 생성 매트릭스를 이용하여, 2n 길이의 제1 DNA 코드($\mathcal{C}$)를 생성한다. 본 단계(S210)에서, 수행한 상술한 선택 과정에 따라 생성 매트릭스를 통해 생성되는 제1 DNA 코드($\mathcal{C}$)에 포함된 모든 코드워드들은 상호간의 거리가 최소 해밍 거리(d) 이상일 수 있다. 즉, 생성 매트릭스를 통해 생성되는 제1 DNA 코드($\mathcal{C}$)는 적어도 해밍 거리 제약(HD)을 충족할 수 있다.

[0153] 다음으로, 제1 DNA 코드에서 자기-가역-상보적 코드워드를 제외하여 제2 DNA 코드를 구성한다(S220).

[0154] 이전 단계(S210)에서 생성된 제1 DNA 코드, 가역적 자기 쌍대(RSD) 코드($\mathcal{C}$) 중에서 자기-가역-상보적(self-reverse-complementary) 특성을 가진 코드워드를 삭제하는 과정이 본 단계(S220)에서 수행될 수 있다. 즉, 본 단계(S220)를 통해 해밍 거리 제약(HD)및 RC 제약을 충족하는 제2 DNA 코드($\mathcal{D}'$)가 생성될 수 있다. 길이 6, 해밍 제약 거리(d)가 3인 조건 하에서, 해밍 거리 제약(HD) 및 가역 상보 제약(RC)을 충족하도록 생성된 제2 DNA 코드가 도 4의 예시와 같이 생성될 수 있다.본 단계(S220)는 하기 알고리즘 2의 step 1 내지 step 5와 같이 수행될 수 있다.

Input:
$\mathcal{C}$:= *an RSD code of length* $2n$ *with generator matrix* $[I_n \mid M]$ *and minimum distance* d.

Output:
$\mathcal{D}$:= *a DNA code of length* $2n+2$ *with HD and RC constraint with* d *and DI constraint with* $e = 4$.

step 1. *[Obtain* $\rho(\mathcal{C})$ *and* $\mathcal{C}_0$*]*
Obtain $\rho(\mathcal{C})$ *generated by the matrix* $(I_n + MR_n \mid M + R_n)$ *and obtain* $\mathcal{C}_0$.
step 2. *[Decompose* $\mathcal{C}$ *into cosets]*
Decompose $\mathcal{C}$ *into cosets of* $\mathcal{C}_0$*, and let* $\mathscr{S}$ *be the set of cosets.*
step 3. *[Remove self-reverse-complementary codwords]*
If there exists $\mathbf{u}$ *in* $\mathcal{C}$ *such that* $\rho(\mathbf{u}) = \mathbf{1}$*, then the coset* $\mathbf{u} + \mathcal{C}_0$ *is the set of all self-reverse-complementary codewords in* $\mathcal{C}$*. Remove the coset from* $\mathscr{S}$.
step 4. *[Extract codewords from a coset]*
For a coset $\mathcal{C}_i \in \mathscr{S}$*, set* $\mathcal{C}_i' = \emptyset$*. Take a codeword* $\mathbf{x} \in \mathcal{C}_i$*. Set* $\mathcal{C}_i' := \mathcal{C}_i' \cup \{\mathbf{x}, \mathbf{x}^r\}$ *and* $\mathcal{C}_i := \mathcal{C}_i - \{\mathbf{x}, \mathbf{x}^r, \mathbf{x}+\mathbf{1}, \mathbf{x}^r+\mathbf{1}\}$*. Repeat this until* $\mathcal{C}_i$ *becomes empty.*
step 5. *[Iteration]*
Set $\mathcal{D}' = \mathcal{D}' \cup \mu(\mathcal{C}_i')$ *and repeat step 4 for all the cosets in* $\mathscr{S}$.
step 6. *[Extend* $\mathcal{D}'$ *for DI constraint]*
Extend all codeword in $\mathcal{D}'$ *to construct* $\mathcal{D}'_\psi = \{\mathbf{c}_\psi \mid \mathbf{c} \in \mathcal{D}'\}$.
step 7. *[Extract codewords of constant GC-weight]*
Append on $\mathcal{D}$ *the every codeword* $\mathbf{c}_\psi$ *in* $\mathcal{D}'_\psi$ *of GC-weight* $k = 2\lfloor n/2 \rfloor + 2$.
step 8. *[End]*
$\mathcal{D}$ *is a DNA code with HD and RC constraint with* d *and DI constraint with* $e = 4$ *and GC constraints with GC-weight* k.

[0156]

[0157] 알고리즘 2

[0159] 다음으로, 상기 제2 DNA 코드($\mathcal{D}'$)에 포함된 코드워드 각각의 n번째 기호와 n+1번째 기호 사이에 2개의 기호를 삽입하여 2n+2 길이를 가진 제3 DNA 코드를 생성한다(S230).

[0160] 본 단계(S230)에서, 2n 길이의 제2 DNA 코드($\mathcal{D}'$)에 2개의 기호를 삽입하여 2n+2 길이를 가진 제3 DNA 코드($\mathcal{D}'_\psi$)를 생성할 수 있다. 즉, 2n 길이를 가진 제2 DNA 코드($\mathcal{D}'$)의 모든 코드워드의 길이가 2n+2로 연장되도록 2개의 기호가 삽입될 수 있다. 본 단계(S230)는 상기 알고리즘 2의 step 6에 대응할 수 있다.

[0161] 여기서, 2n 길이를 가진 제2 DNA 코드($\mathcal{D}'$)는 1번째 기호에서 n번째 기호로 구성된 제1 그룹($\mathbf{u}_n$)과 n+1 번째 기호와 2n번째 기호로 구성된 제2 그룹($\mathbf{v}_n$)으로 구성될 수 있다. 즉, 2n 길이를 가진 제2 DNA 코드($\mathcal{D}'$)는 생성 매트릭스($I_n \mid M$)를 기초로 생성된 코드로 이에 포함된 코드워드들은 자기-쌍대 코드에 해당하며, 상보 정보 세트(complementary information set, CIS) 특성을 가질 수 있다. 상보 정보 세트는 전체 길이(2n) 중 절반(n)에

해당하는 기호 정보로 나머지 정보를 파악할 수 있는 코드워드를 의미한다. 즉, 2n 길이의 $(\mathbf{u}_n \mid \mathbf{v}_n)$로 구성된 제2 DNA 코드($\mathcal{D}'$)의 코드워드 중 $\mathbf{u}_n$는 $\mathbf{v}_n$을 기초로 결정될 수 있다. 구체적으로, $\mathbf{u}_n$는 $\mathbf{v}_n = \mathbf{u}_n M$와 같이 생성 매트릭스의 매트릭스를 통해 결정될 수 있으며, 반대의 경우도 적용될 수 있다.

[0162] 이러한, 제1 그룹($\mathbf{u}_n$)과 제2 그룹($\mathbf{v}_n$) 사이에 2개의 기호가 삽입될 수 있다. 삽입된 2개의 기호는 오리지널 코드워드와 변형된 코드워드와의 차이점을 추적할 수 있는 기준이 될 수 있다.

[0163] 여기서, 2개의 기호는 동일한 종류의 기호가 반복되어 삽입될 수 있다. 즉, AA, GG, CC 또는 TT가 삽입될 수 있다. 구체적으로, 2개의 기호는 n번째 기호와 상이한 종류의 기호로 결정될 수 있다. 2n 길이의 제2 DNA 코드($\mathcal{D}'$)에 포함된 코드워드들에 이러한 조건으로 2개의 기호가 삽입될 수 있으며, 2n+2 길이의 제3 DNA 코드($\mathcal{D}'_\psi$)가 생성될 수 있다.

[0164] 또한, 2개의 기호는 제2 DNA 코드($\mathcal{D}'$)의 각 코드워드의 GC 개수를 고려하여 결정될 수 있다. 예를 들어, 제2 DNA 코드($\mathcal{D}'$)에 포함된 코드워드 $\mathbf{c} = (c_1, c_2, \cdots c_{2n})$에 삽입되는 기호 $\psi(\mathbf{c})$는 하기 수학식 3과 같이 n번째 기호와의 관계 및 코드워드 $\mathbf{c} = (c_1, c_2, \cdots c_{2n})$의 GC 개수를 고려하여 삽입될 수 있으며, $\mathbf{c}_\psi = (c_1, \cdots, c_n, \psi(\mathbf{c}), \psi(\mathbf{c}), c_{n+1}, \cdots, c_{2n})$이 생성될 수 있다.

[0166] [수학식 3]

$$\psi : \mathcal{D} \to \{A, C, G, T\} \ by$$

[0167]
$$\psi(\mathbf{c}) = \begin{cases} C & if\ c_n \neq C \ and\ n_C(\mathbf{c}) + n_G(\mathbf{c}) = 2\lfloor n/2 \rfloor, \\ G & if\ c_n = C \ and\ n_C(\mathbf{c}) + n_G(\mathbf{c}) = 2\lfloor n/2 \rfloor, \\ A & if\ c_n \neq A \ and\ n_C(\mathbf{c}) + n_G(\mathbf{c}) \neq 2\lfloor n/2 \rfloor, \\ T & if\ c_n = A \ and\ n_C(\mathbf{c}) + n_G(\mathbf{c}) \neq 2\lfloor n/2 \rfloor, \end{cases}$$

[0169] 본 단계(S230)에서, GC 개수를 고려하여 2개의 기호가 삽입됨에 따라, 종래 GC 개수가 충족되지 않았던 제2 DNA 코드($\mathcal{D}'$)의 일부 코드워드들의 GC 개수가 충족될 수 있다. 즉, 2n+2 길이의 제3 DNA 코드($\mathcal{D}'_\psi$)가 생성됨에 따라 GC 개수를 충족하는 코드워드들의 개수가 제2 DNA 코드($\mathcal{D}'$)일 때보다 더 많아지게 된다. 즉, GC 개수를 충족하는 코드워드의 개수가 최대화(maximized)될 수 있다. 도 7은 해밍 거리 제약(HD), 가역-상보 제약(RC) 및 레벤슈타인 거리 제약(DI)를 충족하는 제3 DNA 코드의 일 예시로, 도 4에 예시된 제2 DNA 코드에 2개의 기호가 삽입되어 8길이의 DNA 코드가 생성된 것을 알 수 있다.

[0171] 여기서, 각 코드워드마다 2개의 기호가 삽입되어 2n+2 길이를 가진 제3 DNA 코드($\mathcal{D}'_\psi$)는 $dd(\mathbf{u}, \mathbf{v}) \geq$ 를 충족할 수 있게 된다. 길이가 2n+2인 제3 DNA 코드($\mathcal{D}'_\psi$)의 서로 다른 두 코드워드(codeword, 부호어) $x_\psi = (x_1, x_2, \cdots, x_n, \psi(x), \psi(x), x_{(n+1)}, \cdots, x_{2n})$와 $y_\psi = (y_1, y_2, \cdots, y_n, \psi(y), \psi(y), y_{n+1}, \cdots, y_{2n})$를 통해 제3 DNA 코드($\mathcal{D}'_\psi$)가 $dd(\mathbf{u}, \mathbf{v}) \geq 4$를 충족하는 지 여부를 설명하도록 한다.

[0172] 제3 DNA 코드($\mathcal{D}'_\psi$)의 서로 다른 두 코드워드 x_ψ와 y_ψ는 상기 수학식 3의 규칙에 따라 2개의 기호($\psi(x)\ \psi(y)$)가 각각 삽입되어 길이가 2n+2로 확장된 상태이며, x_k와 y_k는 각각 코드워드 x_ψ와 y_ψ의 k번째 기호를 의미한다. 삽입된 기호($\psi(x)\ \psi(y)$)의 조건에 따라 다음과 같은 두가지 경우로 구분하여 설명을 하도록

한다.

[0174] 1. $\psi(x) \neq \psi(y)$ 인 경우,

[0175] 코드워드 x_ψ 와 y_ψ 는 서로 삽입된 두 기호($\psi(x),\psi(x)$, $\psi(y),\psi(y)$)가 서로 상이하므로 최소한 두개의 기호($\psi(x),\psi(x)$)를 지운 자리에 두개의 기호($\psi(y),\psi(y)$)를 삽입해야 서로 같은 코드워드로 만들 수 있다. 따라서 거리 dd의 정의에 따라 $\mathrm{dd}(x_\psi, y_\psi)$ 는 4이상을 충족한다.

[0177] 2. $\psi(x) = \psi(y)$ 인 경우,

[0178] 코드워드 x_ψ , y_ψ 가 서로 다르므로 $\mathrm{dd}(x_\psi, y_\psi) \neq 0$ 가 성립될 수 있다. 또한, 코드워드 x_ψ 와 y_ψ 의 길이가 2n+2로 동일하기 때문에 거리 dd의 정의에 따라 $\mathrm{dd}(x_\psi, y_\psi)$ 는 짝수이다. 즉, $\mathrm{dd}(x_\psi, y_\psi)$ 가 2가 아닌 경우, $\mathrm{dd}(x_\psi, y_\psi)$ 는 4이상을 충족하게 된다.

[0179] 여기서, $\mathrm{dd}(x_\psi, y_\psi)$ =2라고 가정하자. 즉, 코드워드 x_ψ 에 포함된 하나의 기호를 제거하고 적당한 자리에 다시 하나의 기호를 삽입하면 코드워드 y_ψ 와 똑같은 코드워드를 얻을 수 있다고 가정하자. 코드워드 x_ψ 와 y_ψ 는 수학식 3에 따라 생성되었으므로 $x_n \neq \psi(x) = \psi(y) \neq y_n$ 이다. 즉, $x_n \neq \psi(y)$ 이고 $\psi(x) \neq y_n$ 이므로 코드워드 x_ψ 의 하나의 기호를 제거하고 다시 적당한 자리에 하나의 기호를 삽입하여 코드워드 y_ψ 와 똑같이 만들려면, 제거된 기호와 삽입된 기호는 모두 코드워드 x_ψ 의 앞 n개 기호($x_1, x_2, \cdots, x_n$ 제1 그룹) 사이에서만, 또는 뒤의 n개 기호($x_{(n+1)}, \cdots, x_{2n}$, 제2 그룹) 사이에서만 존재할 수 있다. 예를 들어 제1 그룹(앞의 n개)중 하나를 지우고, 제2 그룹(뒤의 n개) 중 하나에 기호를 끼워넣었다면, 기호 $\psi(x)$ 이 앞으로 한 칸 이동하게 되고, $\psi(x) \neq y_n$ 이므로 모순이다. 앞의 n개 기호 사이에서만 하나를 제거했다 다시 삽입해서 같은 코드워드가 된다면 코드워드 x_ψ 와 y_ψ 의 뒤 n개의 기호는 원래부터 동일한 것이여야 하는데, 이는 RSD 코드의 특성상 불가능하다. 즉, RSD 코드의 코드워드는 상보 정보 세트(complementary information set, CIS) 특성을 가지므로 서로 다른 코드워드들의 앞의 n개 기호들이 동일하다면 뒤의 n개의 기호들도 동일하고, 뒤의 n개의 기호들이 동일하다면 앞의 n개의 기호들도 동일해야 하기 때문이다. 마찬가지 이유로 뒤의 n개 기호 사이에서만 하나를 제거했다 다시 삽입해서 같은 코드워드를 만드는 것도 불가능하다. 따라서, $\mathrm{dd}(x_\psi, y_\psi) \neq 2$ 이고, 결과적으로 $\mathrm{dd}(x_\psi, y_\psi)$ 는 4이상인 것을 알 수 있다.

[0181] 즉, 각 코드워드마다 2개의 기호가 삽입되어 2n+2 길이를 가진 제3 DNA 코드($\mathcal{D}'_\psi$)는 $dd(\mathbf{u},\mathbf{v}) \geq 4$ 를 충족하는 것을 알 수 있다.

[0182] 2n+2 길이의 제3 DNA 코드($\mathcal{D}'_\psi$)는 삽입된 2개의 기호를 기준으로 코드의 삽입/삭제 변형을 감지하고, 오리지널 코드워드로의 복원이 가능할 수 있다. 여기서, $dd(\mathbf{u},\mathbf{v}) \geq 4$ 를 충족하는 제3 DNA 코드($\mathcal{D}'_\psi$)는 적어도 단일 오류를 검출하고, 오리지널 코드워드로의 복원이 가능하다.

[0183] 하기 알고리즘 3과 같은 과정을 통해 2n+2 길이의 제3 DNA 코드($\mathcal{D}'_{\psi}$)에 변형이 발생된 경우, 변형된 DNA 코드($\mathcal{D}_{\phi}$)에서 오리지널 제3 DNA 코드($\mathcal{D}'_{\psi}$)에 포함된 코드로 복원이 가능할 수 있다.

Input:
$\mathbf{x}$:= *a vector which is received through a single deletion/insertion channel from* $\mathcal{D}_{\phi}$.

Output:
$\mathbf{c}$:= *a codeword in a code* $\mathcal{D}'_{\psi}$

Step 1. [Check the length of $\mathbf{x}$*]*
Check the length m *of* $\mathbf{x}$.
- *If* $m = 2n+1$, *then a deletion error occurs. Go to step 2.*
- *If* $m = 2n+3$, *then an insertion error occurs. Go to step 3.*
- *If* $m = 2n+2$, *then no deletion/insertion error occurs. Go to step 4.*

Step 2. [Find the position where a deletion occurs]
Decompose $\mathbf{x} = \mathbf{u}_n \oplus (x_{n+1}) \oplus \mathbf{v}_n$, *where* $\mathbf{u}_n$ *and* $\mathbf{v}_n$ *are vectors of length* n.
- *If* $u_n = x_{n+1}$, *then the deletion error occurs outside of* $\mathbf{v}_n$. *Take* $\mathbf{y} = \mathbf{v}_n$ *with the information set* $\{n+1, \ldots, 2n\}$. *Go to step 5.*
- *If* $u_n \neq x_{n+1}$, *then the deletion error occurs outside of* $\mathbf{u}_n$. *Take* $\mathbf{y} = \mathbf{u}_n$ *with the information set* $\{1, \ldots, n\}$. *Go to step 5.*

Step 3. [Find the position where an insertion occurs]
Decompose $\mathbf{x} = \mathbf{u}_n \oplus (x_{n+1}, x_{n+2}, x_{n+3}) \oplus \mathbf{v}_n$, *where* $\mathbf{u}_n$ *and* $\mathbf{v}_n$ *are vectors of length* n.
- *If* $x_{n+1} = x_{n+2}$, *then the insertion error occurs outside of* $\mathbf{u}_n$. *Take* $\mathbf{y} = \mathbf{u}_n$ *with the information set* $\{1, \ldots, n\}$. *Go to step 5.*
- *If* $x_{n+1} \neq x_{n+2}$, *then the insertion error occurs outside of* $\mathbf{v}_n$. *Take* $\mathbf{y} = \mathbf{v}_n$ *with the information set* $\{n+1, \ldots, 2n\}$. *Go to step 5.*

Step 4. [No error occurs]
Take $\mathbf{c} = \mathbf{x}^{\{n+1, n+2\}}$. *Go to step 6.*

Step 5. [Obtain $\mathbf{c}$*]*
Obtain $\mathbf{c}$ *generated by* $\mathbf{y}$ *with the information set from the generator matrix of* $\mathcal{D}'_{\psi}$.
Go to step 6.

Step 6. [End]
$\mathbf{c}$ *is the codeword in* $\mathcal{D}'_{\psi}$.

[0184]

[0185] 알고리즘 3

[0187] 알고리즘 3은 RSD 코드의 코드워드가 앞쪽 n개 기호들(제1 그룹)이 결정되면 뒤쪽 n개의 기호들(제2 그룹)이 생성 매트릭스를 이용하여 결정될 수 있고, 뒤쪽 n개 기호들(제2 그룹)이 결정되면 앞쪽 n개의 기호들(제1 그룹)이 생성 매트릭스를 이용하여 결정될 수 있는 성질을 이용하여, 코드워드의 복원을 수행하는 것이다. 즉 코드워드의 삭제/삽입 오류가 없는 부분이 앞쪽 n개 부분인지, 뒤쪽 n개 부분인지를 결정할 수 있다면, 삽입/삭제 오류는 제거,복원될 수 있다.

[0188] 먼저 수신된 코드워드의 길이가 2n이라면 삽입/삭제 오류는 없는 것으로 가정할 수 있다. 수신된 코드워드의 길이가 2n+1이라면 삽입오류가, 2n-1이라면 삭제 오류가 발생했음을 가정할 수 있다. 여기서, 제3 DNA 코드($\mathcal{D}'_{\psi}$)의 모든 코드워드는 n번째 기호와, 삽입된 두개의 기호는 서로 상이하며, 삽입된 두개의 기호는 서로 동일한 것으로 구성되어 있다. 이러한 점을 이용하여 앞쪽 n개의 기호, 혹은 뒤쪽 n개의 기호 부분 중 어느 부분에 삽입/삭제 오류가 발생하였는지 결정할 수 있다(알고리즘 3의 step2, step 3 부분).

[0189] 삭제 오류가 발생한 경우, 수신된 코드워드의 n번째 기호와 n+1번째 기호가 상이하다면 삭제 오류는 앞의 n개의 기호 중에서는 발생하지 않은 것을 알 수 있다. 즉, 앞의 n개의 기호를 기초로 생성 매트릭스를 이용하여 전체 코드워드의 복원이 가능하다. 수신된 코드워드의 n번째 기호와 n+1번째 기호가 동일하다면, 삭제 오류가 앞의 n개의 기호에서 발생한 것을 알 수 있고, 뒤의 n개의 기호를 이용하여 전체 코드워드를 복원할 수 있다.

[0190] 삽입 오류가 발생한 경우, 수신된 코드의 n+1번째 기호와 n+2번째 기호가 동일하다면 삽입 오류는 앞의 n개의 기호 중에서는 발생하지 않은 것을 알 수 있다. 즉, 앞의 n개의 기호를 이용하여 전체 코드를 복원하면 된다. 수신된 코드의 n+1번째 기호와 n+2번째 기호가 상이하다면, 뒤의 n개의 기호를 이용하여 전체 코드워드를 복원할 수 있다.

[0192] 다음으로, 제3 DNA 코드에서 고정된 GC 개수 제약을 충족하는 코드워드를 추출하여 제4 DNA 코드를 구성한다(S240).

[0193] 이전 단계에서 고정된 GC 개수 제약을 충족시키도록 2개의 기호가 코드워드에 삽입되어 2n+2 길이를 가진 제3 DNA 코드($\mathcal{D}'_{\psi}$)를 생성하였어도, 해당 제3 DNA 코드($\mathcal{D}'_{\psi}$)에는 GC 개수가 충족되지 않은 일부 코드들이 포함되어 있는 상태이다. 본 단계(S240)를 통해 GC 개수가 정해진 개수(k)를 충족시키는 코드워드만이 추출되어 제4 DNA 코드를 구성할 수 있다. 즉, 상술한 과정을 통해 생성된 제4 DNA 코드는 해밍 거리 제약(HD), 가역-상보 제약(RC), 고정된 GC 개수 제약뿐만 아니라, DI 제약 또한 충족할 수 있다.

[0194] GC 개수를 4개로 제약하는 경우, 도 7의 예시에서 GC 개수 제약을 충족하지 못하는 코드워드 GGGAAGGG, AAATTAAA는 본 단계(S240)에서 추출되지 못하며, 나머지 코드워드 30개가 추출되어 도 8과 같은 제4 DNA 코드를 구성할 수 있다. 도 8의 제4 DNA 코드는, 길이 8, 해밍 제약 거리(d)가 3인 조건, GC 개수 4, 레벤슈타인 거리가 4이상으로, 단일 변형의 복원이 가능한 코드에 해당하며, 30개의 코드워드로 구성되는 것을 알 수 있다. 또한, 본 실시예에서 두 개의 기호가 GC 비중을 고려하여 삽입되었기에, 도 5와 도 8을 비교하면, GC 개수를 충족하는 코드워드의 개수(15개 -> 30개)가 더 많아지게 된 것을 알 수 있다. 본 발명의 다른 실시예에 따른 DNA 코드 생성 방법은 단일 변형에 대한 복원 기능뿐만 아니라, GC 개수 제약을 충족하는 DNA 코드워드의 개수가 최대화될 수 있어, 더욱 많은 개수의 코드워드를 생성이 가능해진다.

[0195] 본 발명의 실시예에 따른 DNA 코드 생성 방법은 종래 계산될 수 없었던 길이 및 조건의 DNA 코드를 해밍 거리 제약(HD), 가역-상보 제약(RC), 고정된 GC 개수 제약뿐만 아니라, DI 제약을 더 충족하는 DNA 코드를 생성할 수 있다. 즉, 하나의 변형(삭제, 삽입)을 가지 DNA 코드를 원형대로 복원하는 것이 지원될 수 있다. 이에 따라, DNA 컴퓨팅 및 DNA 데이터 저장 등의 기술이 연구 개발되는 것이 더욱 지원될 수 있다.

[0196] 상술한 구체적인 실시예들에서, 발명에 포함되는 구성 요소는 제시된 구체적인 실시예에 따라 단수 또는 복수로 표현되었다. 그러나, 단수 또는 복수의 표현은 설명의 편의를 위해 제시한 상황에 적합하게 선택된 것으로서, 상술한 실시 예들이 단수 또는 복수의 구성 요소에 제한되는 것은 아니며, 복수로 표현된 구성 요소라 하더라도 단수로 구성되거나, 단수로 표현된 구성 요소라 하더라도 복수로 구성될 수 있다.

[0197] 이상에서 설명한 실시예들에 따른 DNA 코드 생성 방법에 의한 동작은, 적어도 부분적으로 컴퓨터 프로그램으로 구현되고 컴퓨터로 읽을 수 있는 기록매체에 기록될 수 있다. 실시예들에 따른 DNA 코드 생성 방법에 의한 동작을 구현하기 위한 프로그램이 기록되고 컴퓨터가 읽을 수 있는 기록매체는 컴퓨터에 의하여 읽혀질 수 있는 데이터가 저장되는 모든 종류의 기록장치를 포함한다. 컴퓨터가 읽을 수 있는 기록매체의 예로는 플래시 메모리 타입(flash memory type), 하드디스크 타입(hard disk type), 멀티미디어 카드 마이크로 타입(multimedia card micro type), 카드 타입의 메모리(예를 들어 SD 또는 XD 메모리 등), 램, 롬 등이 있다. 또한 컴퓨터가 읽을 수 있는 기록매체는 네트워크로 연결된 컴퓨터 시스템에 분산되어, 분산 방식으로 컴퓨터가 읽을 수 있는 코드가 저장되고 실행될 수도 있다. 또한, 본 실시예를 구현하기 위한 기능적인 프로그램, 코드 및 코드 세그먼트(segment)들은 본 실시예가 속하는 기술 분야의 통상의 기술자에 의해 용이하게 이해될 수 있을 것이다. 상기 컴퓨터 프로그램은 상술한 컴퓨터에서 실행될 수 있는 명령어 세트로 구성되며, 상기 컴퓨터의 형태에 따라 그 명칭이 상이할 수 있다. 예를 들어, 상기 컴퓨터의 형태가 스마트 폰인 경우, 상기 컴퓨터 프로그램은 앱으로 지칭될 수 있다.

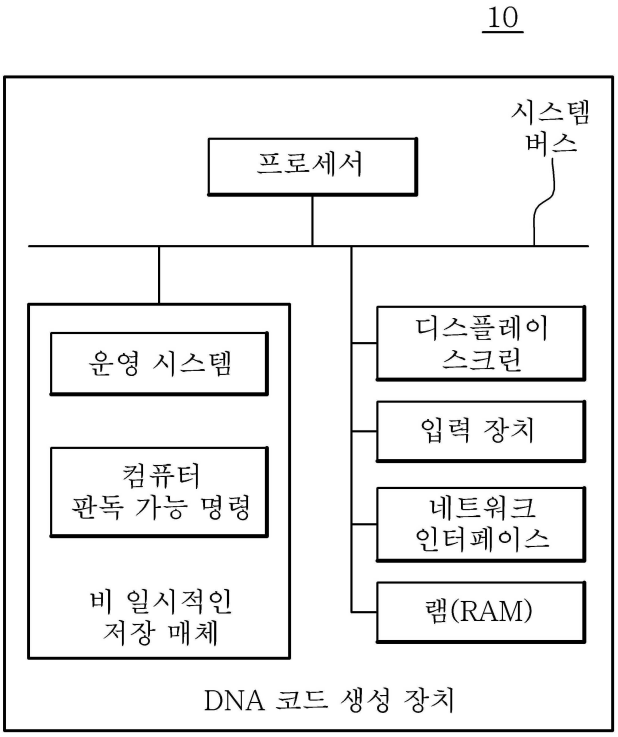
[0198] 한편 발명의 설명에서는 구체적인 실시예에 관해 설명하였으나, 다양한 실시예들이 내포하는 기술적 사상의 범위에서 벗어나지 않는 한도 내에서 여러 가지 변형이 가능함은 물론이다. 그러므로 본 발명의 범위는 설명된 실시예에 국한되어 정해져서는 아니되며 후술하는 청구범위뿐만 아니라 이 청구범위와 균등한 것들에 의해 정해져야 한다.

부호의 설명

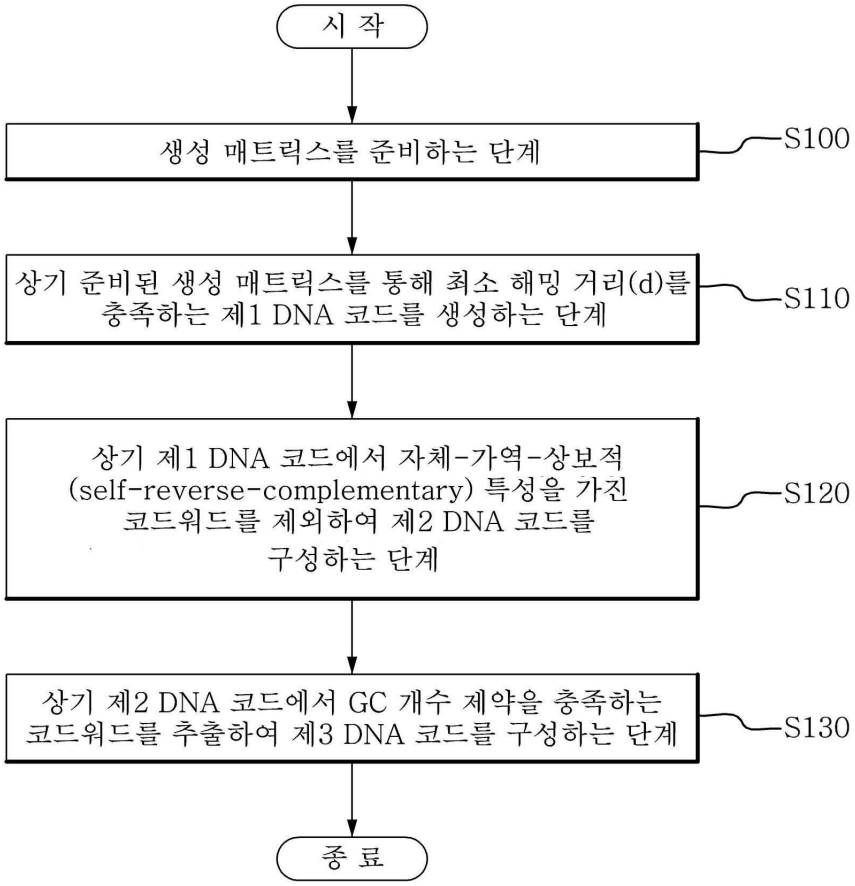
[0199] 10: DNA 코드 생성 장치

도면

도면1


10
시스템
버스
프로세서
운영 시스템
컴퓨터
판독 가능 명령
비 일시적인
저장 매체
디스플레이
스크린
입력 장치
네트워크
인터페이스
램(RAM)
DNA 코드 생성 장치

도면2



도면3

생성 매트릭스

$$\begin{pmatrix} 1 & 0 & 0 & 1 & \bar{\omega} & \bar{\omega} \\ 0 & 1 & 0 & \omega & 0 & \bar{\omega} \\ 0 & 0 & 1 & \omega & \omega & 1 \end{pmatrix}$$

도면4

제2 DNA 코드

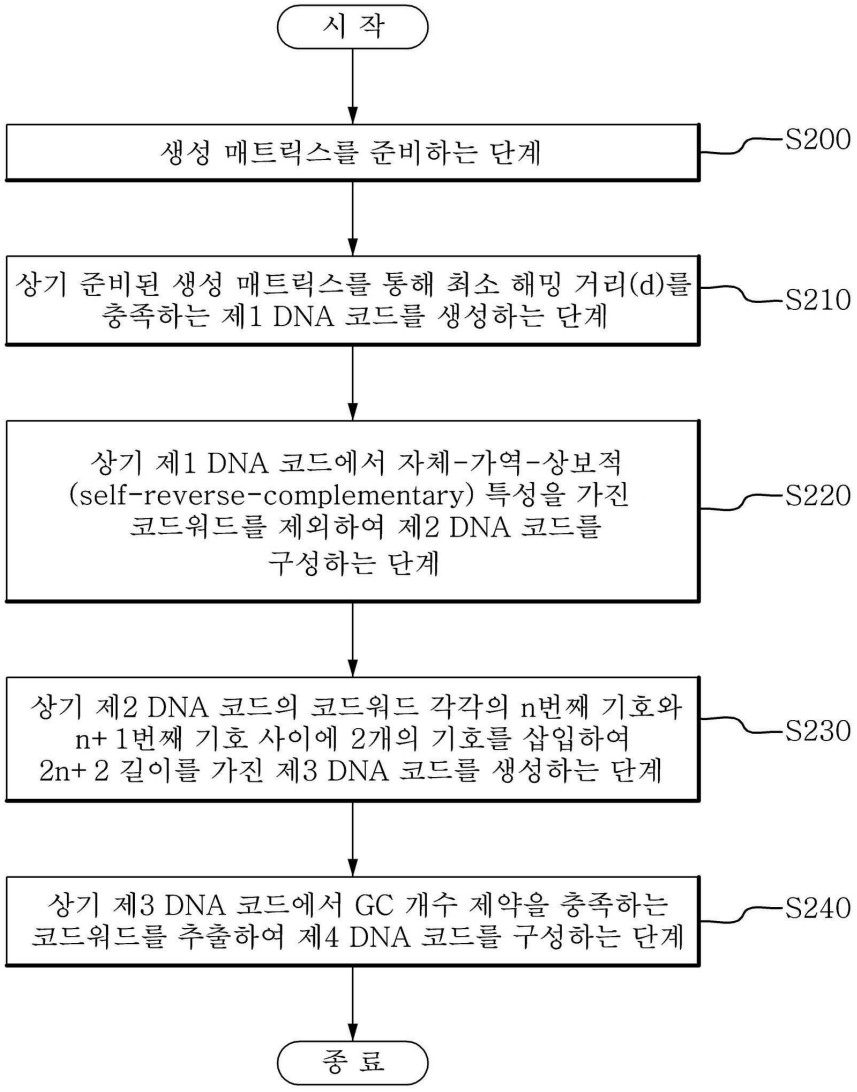
GGGGGG	TCGGCT	ATGGTA	CAGGAC
CGTTGC	ACTTCA	TTTTTT	GATTAG
CGGCAA	AACGGC	ACGCTC	CTCGCA
GGTAAT	TAATGG	TCTATG	GTATCT
TGGTCC	CCTGGT	GCGTGA	AGTGCG
CTGTAT	TATGTC	AAGTTG	GTTGAA
AGGATT	TTAGGA	CCGAAG	GAAGCC
GTGAGC	CGAGTG	TAGACA	ACAGAT

도면5

제3 DNA 코드

TCGGCT	CAGGAC	CGTTGC	CGGCAA
AACGGC	ACGCTC	CTCGCA	TGGTCC
CCTGGT	GCGTGA	AGTGCG	CCGAAG
GAAGCC	GTGAGC	CGAGTG	

도면6



도면7

제3 DNA 코드

GGGAAGGG	ACGTTGCA	TAGCCGAT	CTGTTGTC
CGATTAGC	TCACCACT	AAATTAAA	GTACCATG
CGGTTCTT	TTCTTGGC	TCGTTCAC	CACTTGCT
GGACCTTA	ATTCCAGG	ACACCTAG	GATCCACA
AGGTTACC	CCATTGGA	GCGTTAGT	TGATTGCG
CAGCCATA	ATACCGAC	TTGCCAAG	GAACCGTT
TGGCCTAA	AATCCGGT	CCGTTTTG	GTTAAGCC
GAGTTTGC	CGTAAGAG	ATGCCTCT	TCTCCGTA

도면8

제4 DNA 코드

ACGTTGCA	TAGCCGAT	CTGTTGTC	CGATTAGC
TCACCACT	GTACCATG	CGGTTCTT	TTCTTGGC
TCGTTCAC	CACTTGCT	GGACCTTA	ATTCCAGG
ACACCTAG	GATCCACA	AGGTTACC	CCATTGGA
GCGTTAGT	TGATTGCG	CAGCCATA	ATACCGAC
TTGCCAAG	GAACCGTT	TGGCCTAA	AATCCGGT
CCGTTTTG	GTTAAGCC	GAGTTTGC	CGTAAGAG
ATGCCTCT	TCTCCGTA		