Computing multiple string alignments

고급단백체정보학
수강생 이주호

발표 순서

- 도입부 (교재 이전 부분 및 14. 4 요약)
- Computing multiple string alignment (14. 5)
  - 이후 단원에서 나오는 three objective functions 소개
- Sum-of-pairs (SP) objective function (14. 6)

도입부

- 우리는 이제까지,
  ‘둘 이상의 sequence들을 비교하는 것’에 대한 다양한 요인들, 방안들을 살펴 보았습니다.

- 이러한 내용들이 취하는 당위성의 근간에는,
  교재 212p에 있는 ‘the first fact’가 있습니다.

도입부

- The first fact of biological seq. analysis
  - In biomolecular sequences (DNA, RNA, 아미노산 sequences), high sequence similarity usually implies…

    …significant functional or structural similarity.

도입부

- The first fact of biological seq. analysis
  - In biomolecular sequences (DNA, RNA, 아미노산 sequences), high sequence similarity usually implies…

    …significant functional or structural similarity.

  - 요약:
    - 결과값이 큰 단백체들은 보통…
      …하는 일, 생긴 것(꼬인 구조 등)도 비슷하다.

    - 이를 반대로 해석하면,
      '알고리즘은 생물학적 특성을 충분히 반영해야 한다' 가 됩니다.

도입부

- 발표자가 생각하는 '생물학적 특성' 반영 부분
  - Scoring matrix
    - Match, mismatch, indel에 대한 weight를 character별 차등 부여
  - Local alignment
    - 아직 모르는 긴 string과 잘 알고 있는 짧은 sequence를 비교
  - Gap
    - 돌연변이(mismatch)와는 달리,
      유전적 손실(indel)은 '어디에 몇 번' 일어났는지도 중요하므로,
      이에 대한 weight를 부여하기 위한 수단으로써 도입

도입부

- 발표자가 생각하는 '생물학적 특성' 반영 부분
  - Scoring matrix
  - Gap (weight function)
  - 특히 위의 두 장치들은, 생물학자들에게,
    자신의 의도에 맞는 결과값을 낼 수 있도록,
    그리고 그 작업의 결과를 아직 모르는 단백체에 대 볼 수 있도록
    연산에 대한 선택 가능성을 제공한다는 의의를 가집니다.

도입부

- 교재의 단원 나열 순서를 보면…
  - 11장 – 방금 봤던 각종 개념들과 기초적인 방법론 소개
  - 12장 – 이들을 refine하기 위한 기법들 소개
    - 더 빠르게 연산하는 방법(Four-Russians speedup 등)
    - 속도와 선택 가능성의 절충안(convex gap weight)
  - …와 같은 순서로 설명을 진행하고 있으며,
    주로 각 알고리즘의 '타당성'을 검토하는 데에
    대다수의 지면을 할애하고 있습니다.

도입부

- 그러나 여전히, 도출된 값의 '합당성'은 알고리즘의 input에 해당하는 weight function / matrix에 의존합니다.

- 그렇기 때문에, 교재의 13장에서는 input을 tune하며 원하는 결과를 얻기 위해 기존의 문제를 extend하는 방법들을 소개하고 있습니다.

도입부

- 13장에서 살펴 본 방법들:
  - 13. 1. Parametric sequence alignment
  - 13. 2. Computing suboptimal alignments
  - 13. 3. Chaining diverse local alignments

도입부

- 13장에서 살펴 본 방법들:
  - 13. 1. Parametric sequence alignment
    - 내가 원하는 alignment가 1등이 되는 input의 범위를 가늠
      (범위가 얼마나 큰 지 확인하거나, 범위 내의 다른 input을 생성)
  - 13. 2. Computing suboptimal alignments
    - Input에 대해 유사성 높은 alignment만 filtering
  - 13. 3. Chaining diverse local alignments
    - (아마도 filter를 거친) substring 단위 alignment들을 토대로
      (아마도 매우 긴) 전체 string의 alignment를 추론하는 것

도입부

- 13장의 방법들을 조합하면
  the first fact가 제시하는 목표에
  조금 더 가까이 다가갈 수 있게 됩니다.
  - 생물학적으로 유사한 작용을 하는 두 string에 대한
    더 높은 결과값을 내는 parameter 집합을 찾을 수 있습니다.

  - 이러한 연산을 여러 string들에 대해 적용하면서…
    - 유사한 sequence 집합(단백체群) 내에서의 교차검증을 통해
      진실에 더 가까운 parameter들을 선별할 수 있습니다.
    - 그 결과를 미지의 단백체에 대해 적용하여
      그 단백체의 생물학적 구조 / 작용을 예측할 수 있습니다.

도입부

- 그러나, 이러한 접근 방법만으로는
  현실 속의 단백체, 특히 유전과 관련한 현상을
  합당하게 설명하지 못 하는 듯 보입니다.
  - 단순히 string pair들에 대한 비교 결과들을 모으는 것 만으로는
    합당한 결과값을 내는 scoring matrix, gap weight function을
    찾기 어려웠습니다.

  - 그래서, 교재의 14장에서부터는
    단백체群에 대한 더 입체적인 분석에 필요한
    multiple string alignment 개념을 소개하고 있으며,
    그와 동시에 또 다른 명제, the second fact를 제시하고 있습니다.

도입부

- The second fact of biological seq. comparison
  - Evolutionarily and functionally related molecular strings
    can differ significantly throughout much of the string
    and yet preserve the same three-dimensional structure(s),
    or the same two-dimensional substructure(s) (motifs, domains),
    or the same active sites,
    or the same or related residues (DNA or amino acid).

도입부

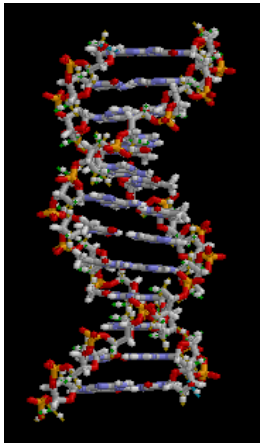
- The second fact of biological seq. comparison
  - Evolutionarily and functionally related molecular strings
    can differ significantly throughout much of the string
    and yet preserve the same three-dimensional structure(s),
    or the same two-dimensional substructure(s) (motifs, domains),
    or the same active sites,
    or the same or related residues (DNA or amino acid).

  - …이 문장의 의미를 그림으로 나타내면 다음과 같습니다:

도입부

GATACCATCGGGAATTAACCCCA…

A String



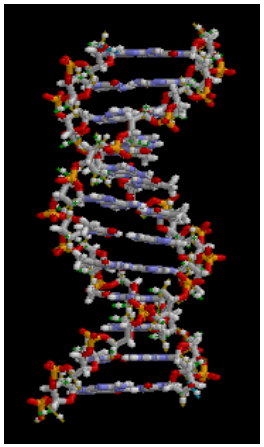
A DNA

A Squirrel

도입부

GATACCATCGGGAATTAACCCCA…

A String



A DNA

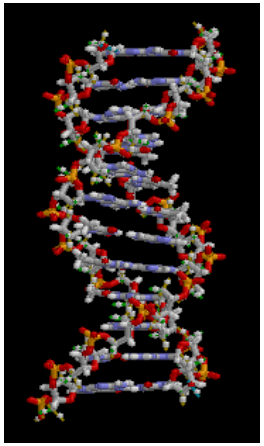
어떤 단백체 하나에 대한 unique한 string을 만들 수 있습니다.

A Squirrel

도입부

GATACCATCGGGAATTAACCCCA…

A String



A DNA

A Squirrel

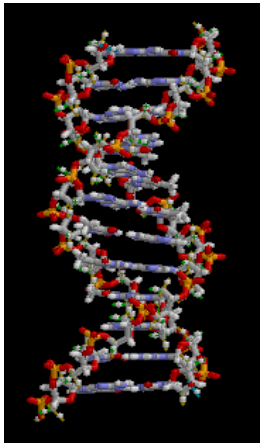
단백체의 물리적 구조는 1차원 string에 드러나 있지는 않습니다.

그러나, '단백체가 어떤 물리적 구조를 가질지'에 대한 정보 또한
string 내에 함축되어 있기 때문에
특정 string 표현에 대해 unique한 단백체를 specify할 수 있습니다.

도입부

DNA에는 유기체의 형질에 대한 정보가 있습니다.

유사한 유기체들의 DNA는 서로 유사하며,
오늘날 우리는 그 반대 또한 중요하게 고려하고 있습니다.
(유사한 DNA를 가진 유기체들을 '유사하다'고 여깁니다)



A DNA

A Squirrel

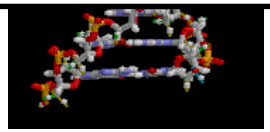
도입부

따라서, the first fact를 다시 떠올려 보면,
유사한 유기체들의 string들은 반드시 서로 유사해야 합니다.

정확히는,
우리는 이러한 string들에 대해 높은 결과값을 내야 합니다.

GATACCATCGGGAATT

A String



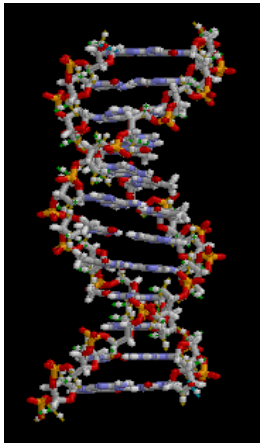
A DNA

A Squirrel

도입부

GATACCATCGGGAATTAACCCCA…

A String



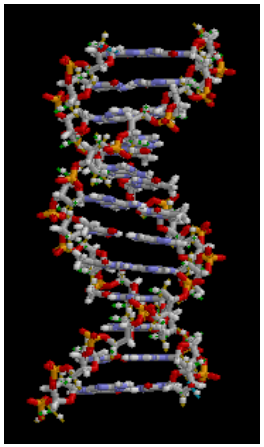
A DNA

이러한 관점 아래에서 the second fact는,
(string에서 잘 드러나지 않는) DNA의 물리적 구조와 같은 요소들이
유기체의 형질에 영향을 줄 수 있음을 나타내고 있습니다.

도입부

GATACCATCGGGAATTAACCCCA…

A String



A DNA

완벽히 일치하지는 않겠지만,
DNA의 물리적 구조를 프로그래밍 언어의 'scope'에 비유한다면
The second fact의 중요성을 다음 코드 예제를 통해 확인할 수 있을 것입니다.

도입부

```c
#include <stdio.h>

#define print(x) do { printf("%d₩n", (x)); } while ( 0 )


void f() { int x = 3; { int x = 5; } print(x);   return; }

void g() { int x = 3; { int x = 5;   print(x); } return; }


int main()
{
    f();
    g();
    return 0;
}
```

도입부

```c
#include <stdio.h>

#define print(x) do { printf("%d₩n", (x)); } while ( 0 )

void f() { int x = 3; { int x = 5; } print(x);    return;

void g() { int x = 3; { int x = 5;    print(x); } retur

int main()
{
    f();
    g();
    return 0;
}
```

모든 단백체에 대해 적용되는 자연적 매커니즘은
C 코드에서 library나 전처리 지시자로 나타낼 수 있을 것입니다.

도입부

```
#include <stdio.h>

#define print(x) do { printf("%d₩n", (x)); } while ( 0 )

void f() { int x = 3; { int x = 5; } print(x);    return; }

void g() { int x = 3; { int x = 5;    print(x); } return; }

int main()
{
    f();
    g();
    return 0;
}
```

두 함수 본문을 단순 string으로 본다면 매우 유사합니다.
그러나 우리는 { }가 실행에 큰 영향을 준다는 것
(semantics를 고려해야 함)을 잘 알고 있으며...

도입부

```
#include <stdio.h>

#define print(x) do { printf("%d₩n", (x)); } while ( 0 )

void f() { int x = 3; { int x = 5;                } print(x);    return; }

void g() { int x = 3; { int x = 5;    print(x); }              return; }

int main()
{
    f();
    g();
    return 0;
}
```

따라서, 이렇게 align하도록 weight function을 구성해야
합당한 결과값(그리 유사하지 않음)을 얻을 수 있음을 알고 있습니다.

도입부

- The second fact of biological seq. comparison
  - Evolutionarily and functionally related molecular strings
    can differ significantly throughout much of the string
    and yet preserve the same three-dimensional structure(s),
    or the same two-dimensional substructure(s) (motifs, domains),
    or the same active sites,
    or the same or related residues (DNA or amino acid).

  - 네, 실제 단백체의 염기 배열은 C 코드보다 더 복잡할 것이므로, string을 다룰 때 물리적 구조를 고려하는 것은 타당합니다.

Computing Multiple Str. Alignments

- 이제 알고리즘적 관점으로 돌아와서, multiple string alignment를 구하기 위한 방법을 살펴 보도록 하겠습니다. (교재 342p)

Computing Multiple Str. Alignments

- 우선, 두 string을 다룰 때와 마찬가지로, global multiple string alignment와 local multiple string alignment의 두 가지를 생각해 볼 수 있습니다.
  - 이 중 global은 두 string에 대한 정의를 확장한 것과 같습니다.
  - Local의 경우 이전의 관념과 조금 다른 정의를 가지고 있으며, 이를 다음 슬라이드와 같이 나타낼 수 있습니다:

Computing Multiple Str. Alignments

- (교재 342p)
  - Given a set of $k > 2$ strings $\boldsymbol{S} = \{S_1, S_2, \ldots, S_k\}$,
    a local multiple alignment of $\boldsymbol{S}$ is obtained by selecting one substring S_i' from each string $S_i \in \boldsymbol{S}$ and then globally aligning those substrings.

Computing Multiple Str. Alignments

- (교재 342p)
  - Given a set of $k > 2$ strings $\boldsymbol{S} = \{S_1, S_2, \ldots, S_k\}$,
    a local multiple alignment of $\boldsymbol{S}$ is obtained by selecting one substring S_i' from each string $S_i \in \boldsymbol{S}$ and then globally aligning those substrings.

  - 요약:
    - 각 string들에 대한 substring들을 선택하고
    - 해당 substring들을 global 방식으로 align합니다.

    - 두 string을 다룰 때에는 '앞 뒤 indel을 무시'하는 것으로 여겼으며, 이는 큰 맥락에서 본다면 여기서의 정의와 크게 다르지 않습니다.

Computing Multiple Str. Alignments

- Local alignment가 그러했듯,
  local multiple alignment 또한
  '크게 유사한 구간이 존재하는지'를 중시하며,
  이는 the second fact와도 일맥상통합니다.
  - 다만 아직까지는 global multiple alignment가
    의미 있는 결론들을 더 자주 도출하고 있으며,
    따라서 교재에서도 이 쪽을 중점적으로 다루고 있습니다.

Computing Multiple Str. Alignments

- Objective function
  - Multiple string alignment를 구하기 위해서는, 이전에 배운 edit distance나 similarity에서처럼 alignment의 score를 매기기 위한 척도 및 계산 방법(objective function)이 필요할 것입니다.
  - 그러나 모두가 납득할 수 있는 적절한 척도는 아직 등장하지 않았으며, 학계에서 주로 사용되는 alignment 방법들은 자체적으로 optimal alignment를 결정하는 대신 생물학자가 직접 여러 후보들 중 하나를 선택하는 방식을 취하고 있습니다.
  - 그럼에도 불구하고 (우리는 생물학자가 아니므로),
    교재에서는 세 가지 objective function을 예로 들어 소개하면서 multiple str. alignment에 대한 독자의 이해를 꾀하고 있습니다.

Computing Multiple Str. Alignments

- 교재에서 소개하는 objective function들
  - Sum of pairs (SP) function (14. 6)
  - Consensus function (14. 7)
  - Tree function (14. 8)
  - 오늘 발표에서는 SP function에 대해 살펴보도록 하겠습니다.

Sum-of-pairs objective function

- 이후 내용 요약:
  - Sum-of-pairs score 정의
  - Alignment 문제 정의
    - 이 문제는 NP-complete임이 증명되어 있습니다.
  - Optimal alignment를 정직하게 계산하는 방법 간략히 소개
    - 당연히 P 시간 안에 끝나지 않습니다.
  - Bounded-error approximation 방법 매우 간략히 소개
    - Score가 optimal의 두 배 미만인 alignment들을 P 시간 안에 찾을 수 있습니다.

Sum-of-pairs objective function

- Sum-of-pairs score
  - 어떤 multiple alignment $\mathcal{M}$에 대해,
    모든 string pair에 대한 pairwise global alignment를 구하고,
    이들 각각의 score를 모두 더한 값이 $\mathcal{M}$의 SP score가 됩니다.

Sum-of-pairs objective function

- Sum-of-pairs score
  - 어떤 multiple alignment $\mathcal{M}$에 대해,
    모든 string pair에 대한 pairwise global alignment를 구하고,
    이들 각각의 score를 모두 더한 값이 $\mathcal{M}$의 SP score가 됩니다.

A	A	G	A	A	-	A
A	T	-	A	A	T	G
C	T	G	-	G	-	G

이와 같이 $\mathcal{M}$이 주어졌고,
pairwise scoring scheme이 $ms + id$로 주어졌을 때…

Sum-of-pairs objective function

- Sum-of-pairs score
  - 어떤 multiple alignment $\mathcal{M}$에 대해,
    모든 string pair에 대한 pairwise global alignment를 구하고,
    이들 각각의 score를 모두 더한 값이 $\mathcal{M}$의 SP score가 됩니다.

A	A	G	A	A	-	A
A	T	-	A	A	T	G
C	T	G	-	G	-	G

4

각 string pair에 대해 pairwise global alignment를 구합니다.

Sum-of-pairs objective function

- Sum-of-pairs score
  - 어떤 multiple alignment $\mathcal{M}$에 대해,
    모든 string pair에 대한 pairwise global alignment를 구하고,
    이들 각각의 score를 모두 더한 값이 $\mathcal{M}$의 SP score가 됩니다.

A	A	G	A	A	-	A
A	T	-	A	A	T	G
C	T	G	-	G	-	G

5

각 string pair에 대해 pairwise global alignment를 구합니다.

Sum-of-pairs objective function

- Sum-of-pairs score
  - 어떤 multiple alignment $\mathcal{M}$에 대해,
    모든 string pair에 대한 pairwise global alignment를 구하고,
    이들 각각의 score를 모두 더한 값이 $\mathcal{M}$의 SP score가 됩니다.

A	A	G	A	A	-	A
A	T	-	A	A	T	G
C	T	G	-	G	-	G

5

이 때, 양 쪽 모두 space인 cell은
(pairwise scoring scheme과 무관하게) 연산에서 제외합니다.

Sum-of-pairs objective function

- Sum-of-pairs score
  - 어떤 multiple alignment $\mathcal{M}$에 대해,
    모든 string pair에 대한 pairwise global alignment를 구하고,
    이들 각각의 score를 모두 더한 값이 $\mathcal{M}$의 SP score가 됩니다.

A	A	G	A	A	-	A
A	T	-	A	A	T	G
C	T	G	-	G	-	G

5

각 string pair에 대해 pairwise global alignment를 구합니다.

Sum-of-pairs objective function

- Sum-of-pairs score
  - 어떤 multiple alignment $\mathcal{M}$에 대해,
    모든 string pair에 대한 pairwise global alignment를 구하고,
    이들 각각의 score를 모두 더한 값이 $\mathcal{M}$의 SP score가 됩니다.

A	A	G	A	A	-	A
A	T	-	A	A	T	G
C	T	G	-	G	-	

4 + 5 + 5 = 14

이렇게 구한 score들을 모두 더하면 $\mathcal{M}$의 SP score가 됩니다.

Sum-of-pairs objective function

- Sum-of-pairs problem
  - 어떤 string 집합에 대해,
    SP score가 최소가 되는 multiple alignment $\mathcal{M}$을 찾는 것.

Sum-of-pairs objective function

- Sum-of-pairs problem
  - 어떤 string 집합에 대해,
    SP score가 최소가 되는 multiple alignment $\mathcal{M}$을 찾는 것.

  - SP score 자체는,
    이전에 배운 두 string간 alignment 방법을 그대로 적용하여
    (matrix를 채우는 방식의 dynamic programming을 사용하여)
    손쉽게 구할 수 있습니다.

Sum-of-pairs objective function

- Sum-of-pairs problem
  - 사실, 이전 방식을 그대로 적용하면, 다차원 matrix를 사용하여 $\mathcal{M}$ 자체를 바로 구할 수도 있습니다.
    - 이 경우 길이가 n인 string k개에 대해 $\Theta(n^k)$ 시간이 걸립니다.

Sum-of-pairs objective function

- Sum-of-pairs problem
  - 사실, 이전 방식을 그대로 적용하면, 다차원 matrix를 사용하여 $\mathcal{M}$ 자체를 바로 구할 수도 있습니다.
    - 이 경우 길이가 n인 string k개에 대해 $\Theta(n^k)$ 시간이 걸립니다.
  - 그러나 이 경우 n이 1000 미만일 때 k가 5 이상만 되어도 impractical한 시간이 걸리게 됩니다.
  - 교재가 소개하는 논문에서는 제한적 조건(특정 단백체들의 일반적 길이에 기반한) 아래 적용 가능한 speedup 방법을 제시하고 있으나, 일반적인 해결 방안으로 보기는 어렵습니다.

Sum-of-pairs objective function

- A bounded-error approximation method
  - 이러한 시간적 난점을 감안하여,
    일반적으로 사용되는 SP alignment method들은
    정답을 직접 찾는 대신…
    - '정답이 존재할 수 있는 score 범위'를 찾고,
    - 해당 범위 내에 존재하는 alignment들(정답 '후보'들)을 함께 찾으며,
    - 이러한 연산을 P 시간 안에 수행해 줍니다.

Sum-of-pairs objective function

- A bounded-error approximation method
  - 이러한 시간적 난점을 감안하여,
    일반적으로 사용되는 SP alignment method들은
    정답을 직접 찾는 대신…
    - '정답이 존재할 수 있는 score 범위'를 찾고,
    - 해당 범위 내에 존재하는 alignment들(정답 '후보'들)을 함께 찾으며,
    - 이러한 연산을 P 시간 안에 수행해 줍니다.

  - 이러한 approximation method는
    교재 13장에서의 관점과 유사하게,
    생물학자가 실제 단백체의 작용에 기반한 선택을 할 수 있도록
    의미 있는 후보 집합을 제시할 수 있다는 의미 또한 가집니다.

Sum-of-pairs objective function

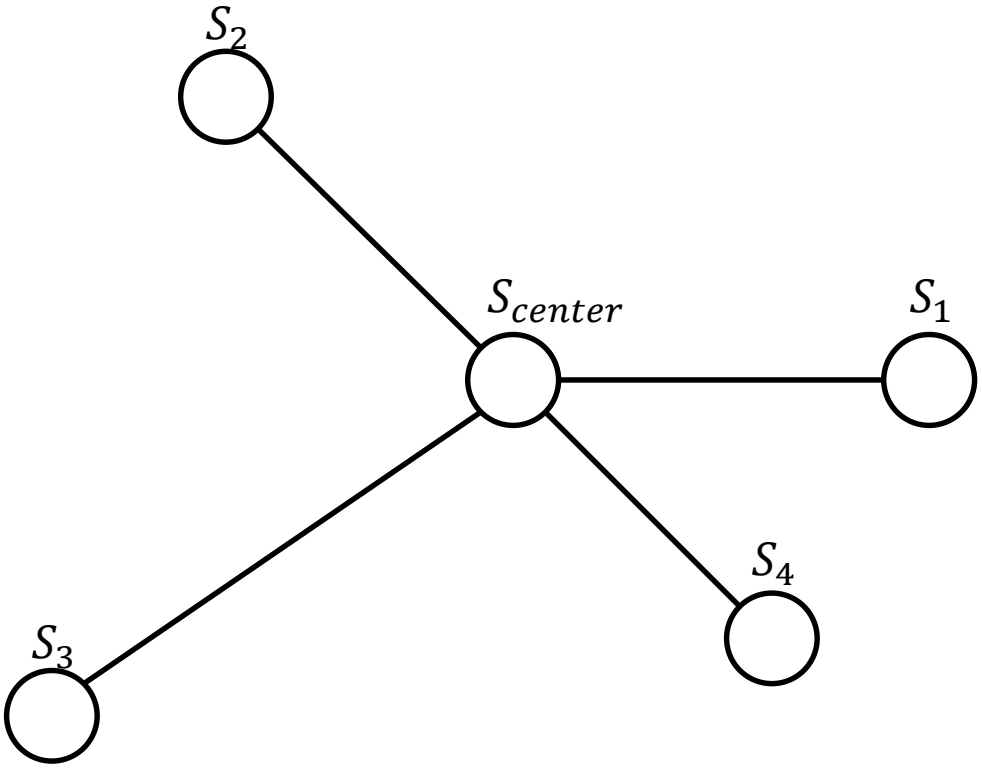
- Approximation의 main idea
  - 기본적으로 approximation은 '부등식'을 통해 수행됩니다.
  - SP score는 pairwise alignment의 합입니다.
  - 따라서, string들 사이의 관계를 graph로 미리 형성한 다음, '인접한' string들에 대해서는 직접 alignment를 구하고, 그렇지 않은 string과의 alignment를 간접적으로 추측함으로써 approximation을 수행하게 됩니다.

Sum-of-pairs objective function

- Approximation의 main idea
  - 기본적으로 approximation은 '부등식'을 통해 수행됩니다.
  - SP score는 pairwise alignment의 합입니다.
  - 따라서, string들 사이의 관계를 graph로 미리 형성한 다음, '인접한' string들에 대해서는 직접 alignment를 구하고, 그렇지 않은 string과의 alignment를 간접적으로 추측함으로써 approximation을 수행하게 됩니다.

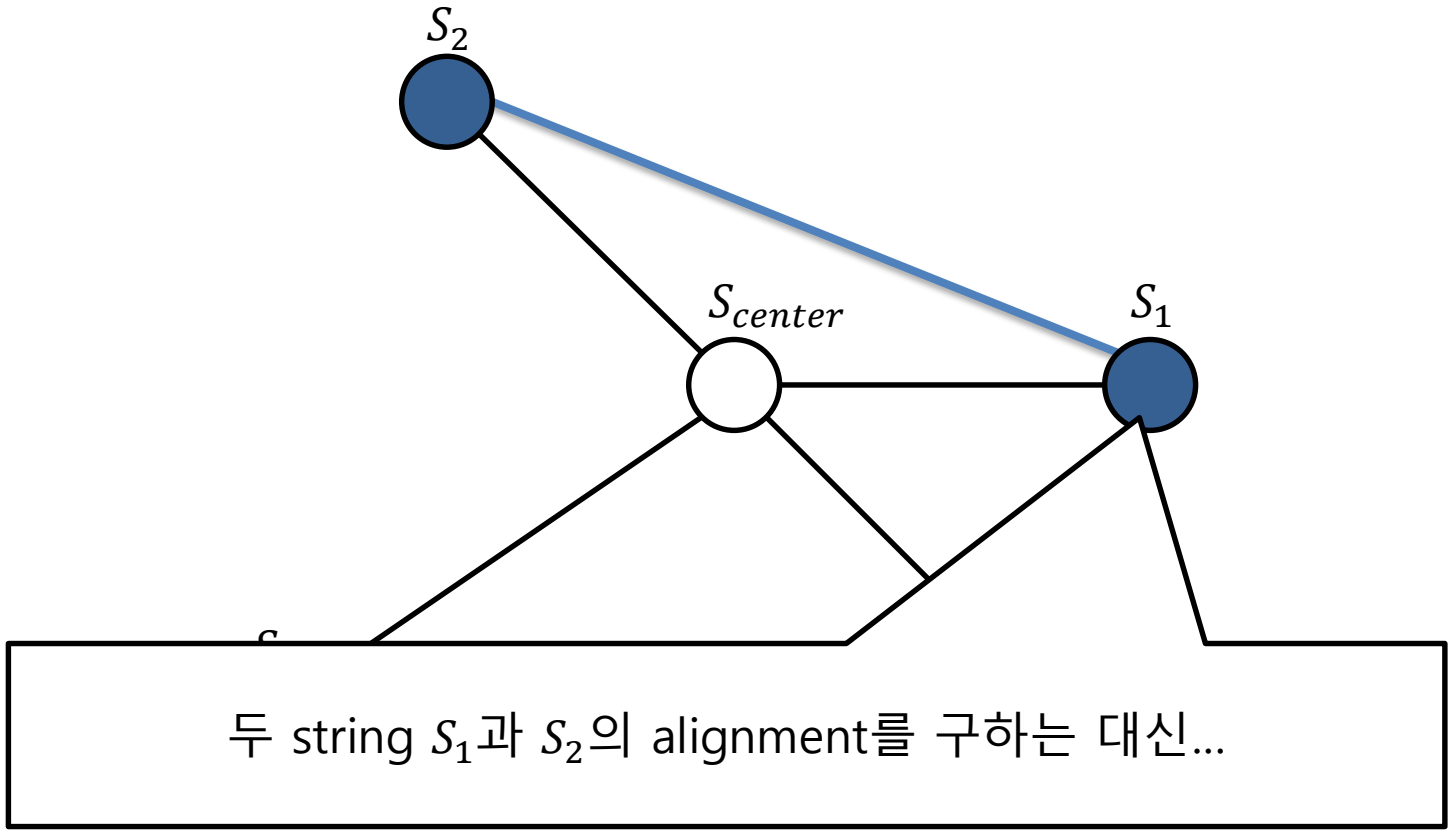
Sum-of-pairs objective function

- The center star method



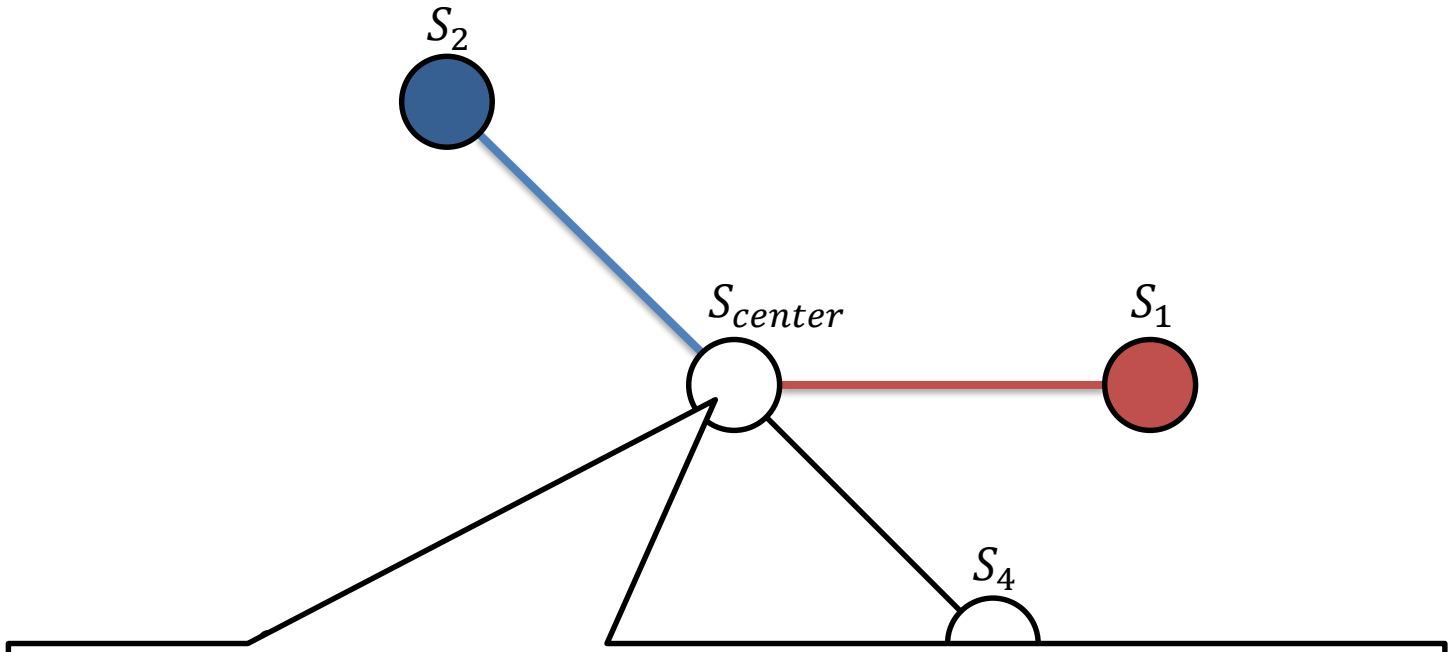
Sum-of-pairs objective function

- The center star method



Sum-of-pairs objective function

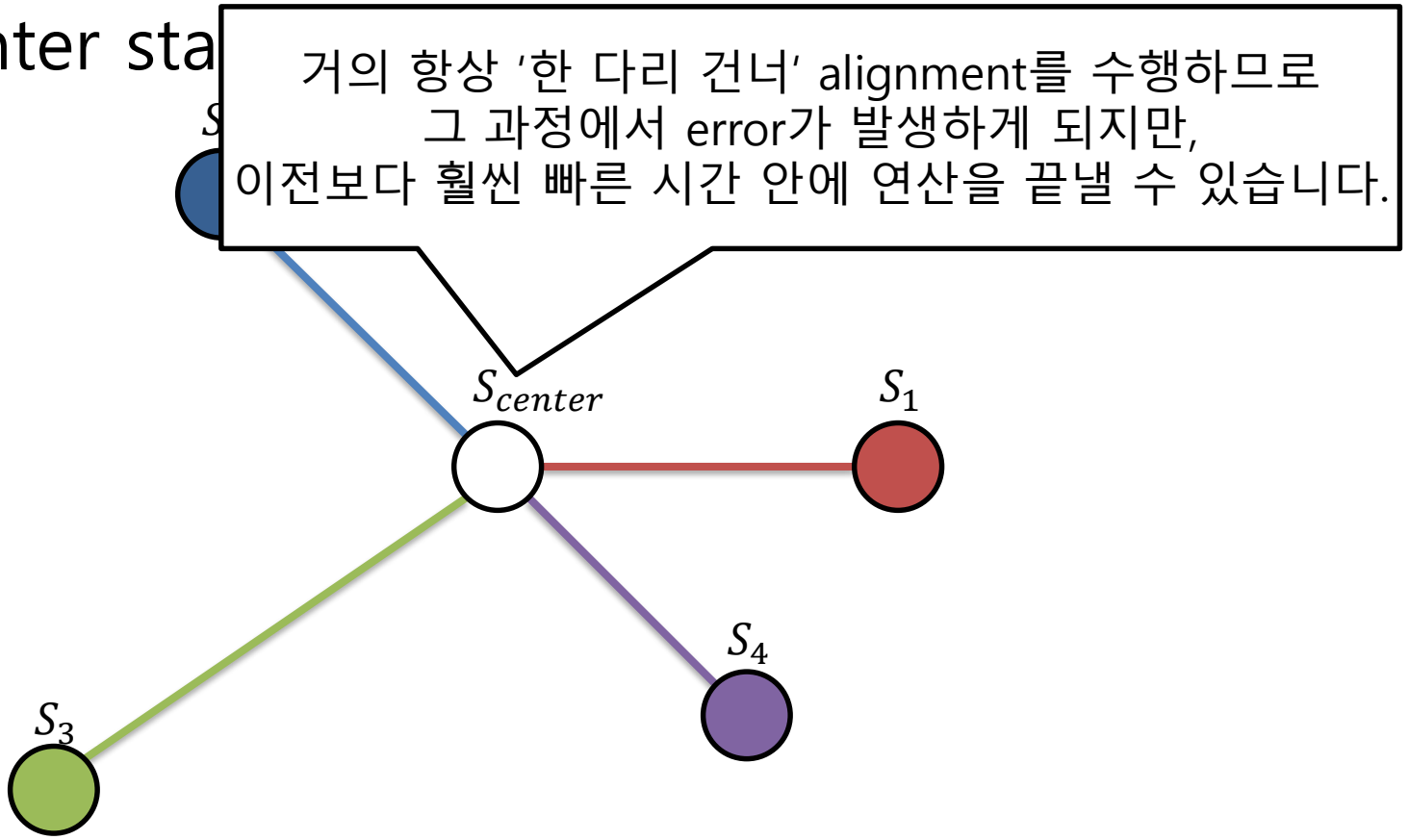
- The center star method



한 가운데 있는 string S_{center}와의 alignment를 각각 구하고
그 둘의 합을 진짜 alignment 대신 사용합니다.

Sum-of-pairs objective function

- The center sta



Q&A