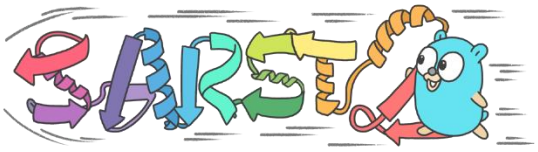
SARST2 사용자 매뉴얼

한국어 버전



목차

1. 이 소프트웨어에 대하여

SARST2 (Structural similarity search Aided by Ramachandran Sequential Transformation, version 2)는 고성능 단백질 구조 정렬 알고리즘입니다. 주어진 쿼리 단백질을 사용하여 데이터베이스에서 구조 유사성을 검색하는 것과 두 단백질 구조 간의 쌍별 구조 정렬을 모두 지원합니다.

이 소프트웨어는 다음 학술 논문과 함께 발표되었으며, 논문에 제공된 URL 을 통해 자주 업데이트될 예정입니다:

제목	SARST2, a high-throughput protein structure alignment algorithm for searching massive databases
저자	Wei-Cheng Lo*, Arieh Warshel, Chia-Hua Lo, Chia Yee Choke, Yan-Jie Li, Shih-Chung Yen, Jyun-Yi Yang and Shih-Wen Weng
연구소	Institute of Bioinformatics and Systems Biology, National Yang Ming Chiao Tung University, Hsinchu, Taiwan, Republic of China

*교신저자 (WadeLo@nycu.edu.tw)

다운로드 URLs:
https://github.com/NYCU-10lab/sarst
https://10lab.ceb.nycu.edu.tw/sarst2

2. 다운로드, 압축 해제 및 설치

SARST2 프로그램 및 사전 포맷된 대상 데이터베이스의 최신 버전은 위에 나열된 URL 에서 이용 가능합니다.

압축된 아카이브를 다운로드한 후, 아카이브 형식에 따라 tar, gzip, 또는 zip 유틸리티를 사용하여 파일을 압축 해제하십시오. 압축 해제 후, SARST2 프로그램 파일은 사전 컴파일된 실행 바이너리 형태로 제공되므로 별도의 설치가 필요하지 않습니다.

예를 들어, SARST2-v2.0.26-Linux.x86_64.tar.gz 아카이브를 다운로드했다면, Linux 에서 다음 명령어를 사용하여 압축을 해제할 수 있습니다:

```
tar xfp SARST2-v2.0.26-Linux.x86_64.tar.gz
```

압축 해제 후, 다음 명령어를 사용하여 sarst2 프로그램을 실행할 수 있습니다:

```
cd SARST2-v2.0.26-Linux.x86_64/bin
chmod +x sarst2
./sarst2 -h
```

3. 소프트웨어 폴더의 내용

bin	Linux, Windows, macOS 용 64 비트 실행 파일.
dbs	사전 포맷된 대상 데이터베이스: PDB-2022 및 SCOP-2.07.
doc	다국어 사용자 매뉴얼.

4. 빠른 시작 가이드

이 소프트웨어 패키지에는 아래 설명된 세 가지 주요 프로그램이 있습니다.

sarst2	SARST2 단백질 구조 정렬 알고리즘의 구현.
formatdb	사용자가 구조 검색 및 정렬을 위한 맞춤형 대상 데이터베이스를 생성할 수 있도록 하는 데이터베이스 포맷팅 도구.
readdb	사전 포맷된 대상 데이터베이스에 저장된 단백질 아미노산 및 선형으로 인코딩된 구조 서열을 추출하는 도구.

위 프로그램 중 어느 것이든 매개변수 없이 (또는 -h 와 함께) 실행하면 해당 프로그램에 대한 간략한 텍스트 기반 도움말 메시지가 표시됩니다.

Linux, macOS

```
./sarst2
./formatdb
./readdb
./sarst2 -h
./formatdb -h
./readdb -h
```

Windows (Cmd 또는 PowerShell)

```
.\sarst2.exe
.\formatdb.exe
.\readdb.exe
.\sarst2.exe -h
.\formatdb.exe -h
.\readdb.exe -h
```

5. 매뉴얼: sarst2 (구조 비교 주 프로그램)

5.1 사용법

```
./sarst2    쿼리 구조                    대상 구조(들)              [옵션]
            ---------                    -----------
            > 하나의 PDB/CIF 파일        > 다음일 수 있습니다
                                         1. -db + 사전 포맷된 데이터베이스
                                         2. PDB/CIF 파일 목록
                                         3. PDB/CIF 파일 폴더
                                         4. 하나의 PDB/CIF 파일 (쌍별)
```

5.2 프로그램 옵션

-db	[str]	검색할 대상 구조의 타겟 데이터베이스. (기본값: 없음)
-brief	[int]	한 줄 요약을 표시할 대상 구조의 개수. (기본값: 500)
-detail	[int]	상세 정렬 데이터를 표시할 대상 구조의 개수. (기본값: 500)
-t	[int]	스레드(thread) 개수. 0 이상이어야 하며, 0 인 경우 모든 프로세서가 사용됩니다. (기본값: 0, 모든 프로세서)
-w	[int]	단어 크기. (기본값: 5)
-orderby	[int]	다음 요소 중 하나로 히트 목록을 정렬합니다. 1: Conf-score--, 2: TM-score--, 3: 서열 동일성--, 또는 4: RMSD++. 여기서 --/++는 내림차순/오름차순을 의미합니다. (기본값: 1, Conf-score--) (쌍별 정렬에서는 무시됨)
-mode	[int]	검색 모드, 1: 정확, 2: 균형, 3: 빠름, 그 외: 자동. (기본값: 자동)
-f	[int]	보조 필터 활성화, 0: 비활성화, 1: 활성화, 그 외: 자동. (기본값: 자동) (쌍별 정렬에서는 항상 0, 비활성화됨)

-C	[float]	Conf-score (신뢰 점수) 임계값. 0 과 1 사이여야 하며, 0 인 경우 임계값은 적용되지 않습니다. (기본값: 0.5) (쌍별 정렬에서는 항상 비활성화됨)
-pC	[float]	최종 pC-값 (즉, -log2(C))의 절단 값. 0 이상이어야 하며, 0 인 경우 절단 값은 적용되지 않습니다. (기본값: 1.0, -C = 0.5 와 동일) (쌍별 정렬에서는 항상 비활성화됨)
-e	[float]	각 필터 및 정제 단계에 적용되는 pC-값의 절단 값. 동일한 -e 및 -pC 가 주어졌을 때, -e 는 더 많은 관련 없는 히트를 버립니다. 0 이상이어야 하며, 0 인 경우 절단 값은 적용되지 않습니다. (기본값: 1.0) (쌍별 정렬에서는 항상 비활성화됨)
-tmcut	[float]	TM-score 절단 값. 0 이상이어야 하며, 0 인 경우 절단 값은 적용되지 않습니다. SARST2 에 의한 TM-score ≥ 0.7 은 패밀리 수준의 상동성을 의미할 수 있습니다. (기본값: 0.15) (쌍별 정렬에서는 항상 비활성화됨)
-mem	[T/F]	모든 대상 단백질 데이터를 메모리에 캐시합니다. (기본값: T) (쌍별 정렬에서는 항상 T, 활성화됨)
-q	[T/F]	빠른 출력 스타일. 결과를 단순화되고 파서 친화적인 형식으로 표시합니다. (기본값: F)
-sa	[T/F]	구조 기반 서열 정렬을 표시합니다. (기본값: T)
-mat	[T/F]	중첩을 위한 변환 행렬을 표시합니다. (기본값: F)
-nmsbj	[T/F]	대상 구조의 크기로 TM-score 를 정규화합니다. (기본값: F)
-nmavg	[T/F]	쿼리 구조와 각 대상 구조의 평균 크기로 TM-score 를 정규화합니다. (기본값: F)
-nmusr	[float]	TM-score 정규화를 위한 단백질 크기. 두 구조의 최소 크기 이상이어야 합니다. 그렇지 않으면 TM-score 가 1 보다 클 수 있습니다.

-d	[float]	TM-score 스케일링을 위한 d0, 예: 5.0 옹스트롬 (Å).
-ml	[T/F]	머신러닝(기계 학습)을 적용합니다. (기본값: T) (쌍별 정렬에서는 항상 F, 비활성화됨)
-fdp	[str]	필터링 단계에 대한 동적 프로그래밍 알고리즘. 지원되는 옵션: NW (Needleman-Wunsch), SW (Smith-Waterman) (기본값: NW)
-rdp	[str]	정제 단계에 대한 동적 프로그래밍 알고리즘. 지원되는 옵션: NW (Needleman-Wunsch), SW (Smith-Waterman) (기본값: NW)
-swp	[str]	사용자가 지정한 스왑 파일 경로. 스왑 파일을 사용하면 메모리 비용을 줄일 수 있습니다. (기본값: 없음)
-Sout	[str]	구조 중첩 파일을 출력할 폴더. 폴더가 존재하지 않으면 생성됩니다. 중첩 파일의 수는 -detail 옵션에 의해 제한됩니다. (기본값: 없음)
-html	[str]	HTML 출력 폴더를 만듭니다. 중첩된 구조 파일도 HTML 폴더에 생성됩니다. (기본값: 없음)
-jsmol	[str]	HTML 출력에서 중첩된 구조를 표시하기 위한 JSmol JavaScript 패키지 경로를 설정합니다. 로컬 디스크 폴더 또는 HTTP(S) URL 일 수 있습니다. (기본값: 없음) (시험용 URL: "https://10lab.ceb.nycu.edu.tw/ext/jsmol")
-pssm_out	[str]	이 알고리즘에 적용된 구조 및 서열 코드의 PSSM 을 저장할 파일. (기본값: 없음)
-pssm_pC	[float]	PSSM 구성을 위한 pC-값 절단 값. (기본값: 0.05)
-h		도움말 메시지 (빠른 가이드)를 인쇄합니다.

5.3사용 예시: 데이터베이스 구조 유사성 검색

사전 포맷된 대상 데이터베이스에 대해 쿼리 구조를 검색합니다

```
./sarst2 Qry.pdb -db my_db/my_proteins.db -brief 10 -w 7 -e 0.1
./sarst2 Qry.pdb -db my_db/my_proteins.db -brief 10 -d 5.0 -sa F
```

위 예시에서 대상 데이터베이스는 "my_db" 폴더에 저장되어 있으며, "my_proteins.db"는 대상 데이터베이스 파일의 파일 스템입니다. 실제 데이터베이스 파일 목록은 **매뉴얼: formatdb** 를 참조하십시오.

5.4사용 예시: 다대일 정렬

나열된 대상 구조에 대해 쿼리 구조를 검색합니다

```
./sarst2 Qry.pdb Sbj1.pdb Sbj2.cif Sbj3.pdb -mat T
```

와일드카드 패턴으로 지정된 대상 구조 파일에 대해 쿼리 구조를 검색합니다

```
./sarst2 Qry.pdb "set1/*.pdb" "set2/1a???.cif" -nmavg T
```

이 예시에서 "set1/*.pdb"와 "set2/1a???.cif"는 따옴표로 묶여 있고 와일드카드 문자를 포함합니다. sarst2 프로그램은 이러한 와일드카드 패턴을 자동으로 확장하고 일치하는 파일 이름을 내부적으로 검색합니다. 패턴이 따옴표로 묶여 있지 않으면 운영 체제가 대신 와일드카드를 확장합니다. 일치하는 파일 수가 많을 경우, 결과 명령줄 인수 목록이 시스템 제한을 초과하여 명령이 실패할 수 있습니다. 따라서 sarst2 가 내부적으로 파일 목록을 처리하도록 와일드카드 패턴을 따옴표로 묶는 것을 권장하며, 운영 체제의 기본 동작에 의존하지 않도록 합니다.

대상 구조를 포함하는 여러 폴더에 대해 쿼리 구조를 검색합니다

```
./sarst2 Qry.pdb set1 set2 -nmavg T
```

이 예시에서 set1 과 set2 는 단백질 구조 파일을 포함할 수 있는 폴더입니다. sarst2 프로그램은 이 폴더의 모든 파일(set1/* 및 set2/*와 동일)을 자동으로 검색합니다. PDB 또는 CIF 형식으로 식별된 파일이 선택되어 쿼리 구조에 대해 정렬됩니다.

5.5사용 예시: 쌍별 구조 정렬

하나의 대상 구조와 쿼리 구조를 정렬합니다

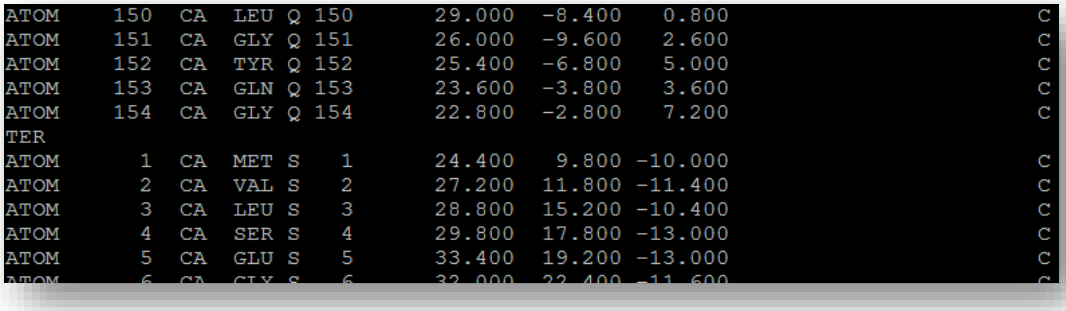
```
./sarst2 Qry.pdb Sbj.pdb -sa F
./sarst2 Qry.pdb Sbj.cif -mat T
```

5.6단백질 구조 중첩 파일 출력

쿼리-대상 구조 중첩 PDB 파일을 생성합니다

```
./sarst2 Qry.pdb -db prot/myDb -detail 100 -Sout output_folder
./sarst2 Qry.pdb "set1/*.cif" -detail 100 -Sout output_folder
```

-Sout output_folder 옵션을 사용하면, PDB 형식의 중첩된 단백질 구조가 사용자가 지정한 폴더로 출력됩니다. 생성되는 중첩 구조의 수는 -detail 옵션에 의해 정의됩니다. 각 출력 파일은 Qry-SbjSN.pdb 로 명명되며, 여기서 SN 은 히트 목록에 있는 대상 단백질의 일련 번호를 나타냅니다. 각 중첩 파일에서 쿼리 및 대상 단백질 구조의 체인 ID 는 각각 Q 와 S 입니다. 두 체인은 아래 그림과 같이 TER 레코드로 구분됩니다.



```
ATOM    150  CA  LEU Q 150      29.000  -8.400   0.800                       C
ATOM    151  CA  GLY Q 151      26.000  -9.600   2.600                       C
ATOM    152  CA  TYR Q 152      25.400  -6.800   5.000                       C
ATOM    153  CA  GLN Q 153      23.600  -3.800   3.600                       C
ATOM    154  CA  GLY Q 154      22.800  -2.800   7.200                       C
TER
ATOM      1  CA  MET S   1      24.400   9.800 -10.000                       C
ATOM      2  CA  VAL S   2      27.200  11.800 -11.400                       C
ATOM      3  CA  LEU S   3      28.800  15.200 -10.400                       C
ATOM      4  CA  SER S   4      29.800  17.800 -13.000                       C
ATOM      5  CA  GLU S   5      33.400  19.200 -13.000                       C
ATOM      6  CA  GLY S   6      32.000  22.400 -11.600                       C
```

그림에서 보듯이, 중첩 파일에는 알파 탄소(Cα) 원자만 나타납니다. 이는 SARST2 가 모든 계산을 Cα 좌표에만 기반하여 수행하기 때문입니다. 쿼리 구조의 방향은 모든 중첩 파일에서 고정된 상태로 유지되며, 각 대상 구조는 쿼리 구조와의 정렬에 따라 기하학적으로 변환(회전 및 이동)되어 중첩을 이룹니다.

중첩된 구조를 시각화하려면 RasMol 또는 RasWin (http://www.openrasmol.org/) 사용을 권장합니다. 중첩 파일에는 Cα 원자만 존재하므로, RasMol 의 표시 모드는 "backbone"으로 설정해야 합니다.

5.7대화형 HTML 결과 페이지 생성

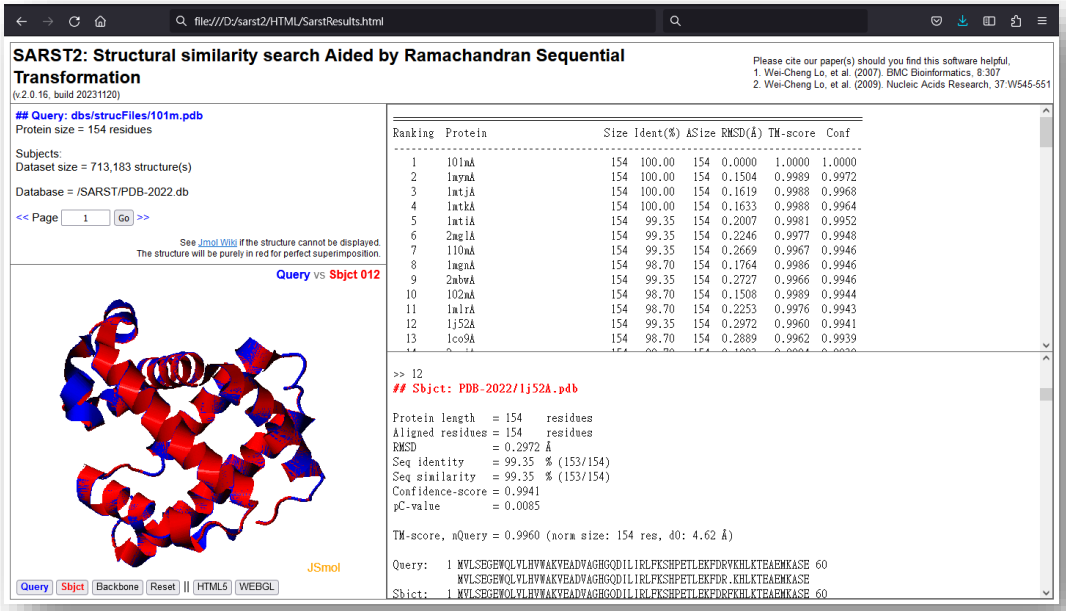
온라인 JSmol 스크립트가 포함된 HTML 문서를 생성합니다

```
./sarst2 Qry.pdb -db my_db/my_proteins.db -html output_folder
-jsmol "https://10lab.ceb.nycu.edu.tw/ext/jsmol"
```

로컬 JSmol 스크립트 폴더(Windows)가 포함된 HTML 문서를 생성합니다

```
./sarst2 Qry.pdb "set1/*.cif" -detail 100 -html output_folder
-jsmol file:///D:/bioinfo/software/jsmol
```

"-html output_folder" 옵션을 사용하면, HTML 결과 문서와 해당 중첩 단백질 구조 파일이 지정된 폴더에 생성됩니다. "-html" 옵션은 JSmol 패키지(버전 2013)의 URL 또는 로컬 경로를 지정하는 "-jsmol"과 함께 사용되어야 합니다. JSmol 은 웹 브라우저에서 실행되는 대화형 3D 분자 구조 뷰어로, 대부분의 주요 최신 브라우저를 지원합니다.



HTML 출력 폴더의 메인 파일은 SarstResults.html 이며, 웹 브라우저에서 열어야 합니다. 다른 HTML 파일은 내부 프레임을 사용하여 메인 페이지에 임베드됩니다. "sup"이라는 하위 폴더도 생성되며, 여기에는 쿼리와 히트 목록의 각 대상 단백질 간의 구조 중첩 파일이 저장됩니다.

운영 체제, 브라우저 또는 안티바이러스 소프트웨어에 따라, 브라우저가 JavaScript 를 실행하고 "sup" 폴더의 중첩 구조 파일에 접근할 수 있도록 보안 설정을 조정해야 JSmol 3D 뷰어가 제대로 작동할 수 있습니다.

6. 매뉴얼: formatdb (데이터베이스 제작 프로그램)

6.1 사용법

```
./formatdb      대상 구조(들)          -db 데이터베이스      [옵션]
                -----------
                > 다음일 수 있습니다
                1. PDB/CIF 파일 목록
                2. PDB/CIF 파일을 포함하는 폴더
                3. PDB/CIF 파일 경로를 나열하는 일반 텍스트 파일
```

6.2 프로그램 옵션

-db	[str]	생성할 대상 구조의 타겟 데이터베이스. (기본값: 없음)
-flist	[str]	PDB/CIF 파일 경로를 나열하는 일반 텍스트 파일. 이 인수는 일반적인 대상 파일 인수와 함께 사용될 수 있습니다. (기본값: 없음)
-t	[int]	스레드(thread) 개수. 0 이상이어야 하며, 0 인 경우 모든 프로세서가 사용됩니다. (기본값: 0, 모든 프로세서)
-split	[int]	데이터베이스를 하위 집합으로 분할하며, 각 하위 집합에는 이 옵션으로 지정된 수의 대상 구조가 포함됩니다. 데이터베이스 분할은 데이터베이스 파일이 디스크의 파일 크기 제한을 초과하는 것을 방지하는 데 도움이 됩니다. (기본값: 없음)
-save_disk	[T/F]	디스크 공간을 절약하기 위해 원자 좌표를 소수점 셋째 자리에서 첫째 자리로 반올림합니다. (기본값: F)
-keep_order	[T/F]	데이터베이스에 저장된 대상 구조의 순서를 입력 순서와 동일하게 유지합니다. T 로 설정하면 데이터베이스 생성이 느려집니다. (기본값: F)
-h		도움말 메시지 (빠른 가이드)를 인쇄합니다.

6.3 사용 예시

나열된 대상 구조 파일에 대한 타겟 데이터베이스를 생성합니다

```
./formatdb Sbj1.pdb Sbj2.cif Sbj3.pdb -db myDb -keep_order T
```

myDb 로 시작하는 파일 이름을 가진 여러 데이터베이스 파일이 생성됩니다. -keep_order 를 활성화하면 명령줄 인수에 나열된 순서에 따라 대상 데이터베이스의 대상 구조 순서가 보존됩니다.

와일드카드가 있는 나열된 대상 구조 파일에 대한 타겟 데이터베이스를 생성합니다

```
./formatdb "set1/*.pdb" "set2/*.cif" Sbj1.pdb Sbj2.cif -db myDb
```

대상 파일을 나열할 때 와일드카드가 있는 인수와 없는 인수를 혼합해도 괜찮습니다. 프로그램이 파일 확장을 올바르게 처리할 수 있도록 와일드카드 인수를 따옴표로 묶는 것이 좋습니다.

대상 구조 파일 목록을 기반으로 타겟 데이터베이스를 생성합니다

```
./formatdb -flist protlist.txt Sbj1.pdb Sbj2.cif -db myDb
```

protlist.txt 파일에는 한 줄에 하나의 파일 경로가 포함된 파일 경로 목록이 있어야 합니다.

대상 구조 파일을 포함하는 폴더에서 타겟 데이터베이스를 생성합니다

```
./formatdb folder1 folder2 -db myDb -save_disk T -split 50000
```

-save_disk 를 활성화하면 Cα 좌표가 소수점 첫째 자리로 반올림되어 저장 공간을 절약합니다. "-split 50000" 옵션은 여러 하위 집합 데이터베이스를 생성하며, 각 하위 집합에는 최대 50,000 개의 구조가 포함됩니다. 이 분할 옵션은 포맷된 데이터베이스 파일의 크기가 일부 운영 체제 또는 디스크 형식에서 지원하는 최대 파일 크기를 초과할 수 있는 경우 특히 유용합니다.

7. 매뉴얼: readdb (데이터베이스 서열 읽기 프로그램)

7.1 사용법

```
./readdb    대상 데이터베이스          출력 파일          [-seq 서열 유형]
            ---------------          --------          --------------
            > SARST2 사전             > FASTA           > 다음일 수 있습니다
              포맷되어야 합니다          형식이어야합니다    1. AA
                                                       2. AAT
                                                       3. SARST
                                                       4. SSE
```

7.2 프로그램 옵션

-seq	[str]	출력 서열 유형. AA 아미노산 서열 AAT 5-심볼 아미노산 유형 서열 SARST SARST 라마찬드란 코드 서열 SSE 4-심볼 2 차 구조 요소 서열 (기본값: AA)
-h		도움말 메시지 (빠른 가이드)를 인쇄합니다.

7.3 사용 예시

SARST2 대상 데이터베이스에서 대상 서열을 추출합니다

```
./readdb my_db/my_proteins.db seqs.fasta
./readdb my_db/my_proteins.db seqs.fasta -seq SARST
./readdb my_db/my_proteins.db seqs.fasta -seq AAT
```

-seq 가 지정되지 않으면 기본 출력 서열 유형은 아미노산 서열입니다. 출력 서열 파일 (seqs.fasta) 은 FASTA 형식으로 저장됩니다. readdb 를 실행하기 전에 출력 파일이 이미 존재하면 덮어쓰여집니다.