CD-HIT·····

··CDS·····cd-hit-est·····ID··.clstr·····

·····CDS·····cd-hit-est·····
·····ID·····ID·····

·· | ···· | ····

-c 0.95 | ····· | ·····

-G 1 | ····· | ·····

-aS/-aL 0.9 | ····· | ·····

-M/-T | ··MB···· | ·····

·····

·····ID·····cd-hit-est·····
·····

```
#!/usr/bin/env bash
set -euo pipefail
genes=/data/all_samples.cds.fna
test -s "$genes"
binary="$(command -v cd-hit-est)"
mkdir cdhit-catalog-001
root="$PWD/cdhit-catalog-001"
sha256sum "$binary" "$genes" > "$root/inputs-and-tool.sha256"
/usr/bin/time -v -o "$root/resources.txt" \
  cd-hit-est -i "$genes" -o "$root/catalog.fna" \
  -c 0.95 -n 10 -G 1 -aS 0.9 -aL 0.9 -r 1 -d 0 \
  -M 16000 -T 4 > "$root/run.log" 2>&1
test -s "$root/catalog.fna"
test -s "$root/catalog.fna.clstr"
python3 - "$genes" "$root/catalog.fna" "$root/catalog.fna.clstr" <<'PY'
import sys
def ids(path):
    with open(path) as f:return [l[1:].split()[0] for l in f if l.startswith('>')]
original,representatives=ids(sys.argv[1]),ids(sys.argv[2])
assert original and len(original)==len(set(original))
assert representatives and len(representatives)==len(set(representatives))
assert set(representatives)<=set(original)
clusters=members=stars=0
with open(sys.argv[3]) as f:
    for line in f:
        if line.startswith('>Cluster '):clusters+=1
        elif line.strip():
            members+=1
            if line.rstrip().endswith('*'):stars+=1
assert members==len(original) and clusters==stars==len(representatives)
print('input, representatives, clusters',len(original),len(representatives),clusters)
PY
```

·····

CD-HIT····FASTA··.clstr·····ID·····
·····

```
import sys
def read_ids(path):
    ids=[]
    with open(path) as handle:
        for line in handle:
            if line.startswith(">"):
                fields=line[1:].split()
                if not fields: raise ValueError("empty FASTA identifier")
                ids.append(fields[0])
    if not ids or len(ids)!=len(set(ids)):
        raise ValueError("FASTA IDs must be nonempty and unique")
    return ids
original=read_ids(sys.argv[1]); representatives=read_ids(sys.argv[2])
clusters={}; current=None
with open(sys.argv[3]) as handle:
    for line in handle:
        if line.startswith(">Cluster "):
            current=int(line.split()[1]); clusters[current]=[]
        elif line.strip():
            if current is None or ">" not in line: raise ValueError("malformed .clstr")
            member=line.split(">",1)[1].split("...",1)[0].split()[0]
            clusters[current].append((member,line.rstrip().endswith("*")))
members=[member for rows in clusters.values() for member,_ in rows]
stars=[member for rows in clusters.values() for member,is_star in rows if is_star]
if len(members)!=len(original) or set(members)!=set(original):
    raise ValueError("cluster membership differs from input")
if len(members)!=len(set(members)):
    raise ValueError("an input ID occurs more than once")
if set(stars)!=set(representatives) or len(stars)!=len(clusters):
    raise ValueError("representatives do not match cluster stars")
print("IDs and cluster membership OK",len(original),len(representatives),len(clusters))
```

……

……
……CDS……
….

……SHA256……..clstr…FASTA……-M……
……I/O……ID……
……

……

……CDS……
……

cd-hit-est……ID…
……FASTA……
..

……ID……
……

….

CD-HIT…… https://github.com/weizhongli/cdhit/blob/master/doc/cdhit-user-guide.tex

CD-HIT…… https://github.com/weizhongli/cdhit