eggNOG-mapper·····

···FASTA····eggNOG-mapper·····ID·····

eggNOG-mapper·····2.1.13···DIAMOND·····
·····FASTA ID·····FASTA·····
·····

·· | ·· | ···

·· | ····part01/02 | ID·····

·· | eggNOG-mapper 2.1.13 | ·····

·· | DIAMOND·CPU 4 | ·····

·· | ·····query ID·· | ·····

·····

·····FASTA·····output_dir·····
·····

```
#!/usr/bin/env bash
set -euo pipefail
db=/db/eggnog-compatible-2.1.13
test -d "$db"
for part in part01 part02; do test -s "/data/$part.faa"; done
mkdir eggnog-parts-001
root="$PWD/eggnog-parts-001"
emapper.py --version > "$root/version.txt" 2>&1
for part in part01 part02; do
  mkdir "$root/$part"
  /usr/bin/time -v -o "$root/$part/resources.txt" \
    emapper.py -i "/data/$part.faa" --itype proteins -m diamond \
    --data_dir "$db" --cpu 4 -o "$part" --output_dir "$root/$part" \
    > "$root/$part/run.log" 2>&1
  test -s "$root/$part/$part.emapper.annotations"
done
```

·····

·····FASTA·····ID·····query·····
·····ID····SQLite·····

```
from pathlib import Path
import os
root=Path("eggnog-parts-001")
fasta_dir=Path(os.environ.get("FASTA_DIR","/data"))
parts=["part01","part02"]
header=None; seen=set(); missing=[]; rows=[]
for part in parts:
    fasta=fasta_dir/(part+".faa")
    input_ids=[]
    with fasta.open() as handle:
        for line in handle:
            if line.startswith(">"):
                fields=line[1:].split()
                if not fields: raise ValueError("empty protein ID")
                input_ids.append(fields[0])
    if len(input_ids)!=len(set(input_ids)): raise ValueError("duplicate input IDs")
    local_header=None; local_seen=set()
    path=root/part/(part+".emapper.annotations")
    with path.open() as handle:
        for line in handle:
            if line.startswith("#query\t"):
                local_header=line.rstrip("\n")
                if header is None: header=local_header
                if local_header!=header: raise ValueError("annotation headers differ")
            elif line.strip() and not line.startswith("#"):
                if local_header is None: raise ValueError("missing header")
                fields=line.rstrip("\n").split("\t")
                if len(fields)!=len(header.split("\t")): raise ValueError("column count differs")
                query=fields[0]
                if query not in set(input_ids): raise ValueError("annotation ID absent from part FASTA: "+query)
                if query in local_seen or query in seen: raise ValueError("duplicate annotation ID: "+query)
                local_seen.add(query);seen.add(query);rows.append(line.rstrip("\n"))
    if local_header is None: raise ValueError("empty annotation output")
    missing.extend(sorted(set(input_ids)-local_seen))
out=root/"merged.annotations.tsv"
with out.open("x") as handle: handle.write(header+"\n"+"\n".join(rows)+"\n")
with (root/"queries_without_annotation.txt").open("x") as handle:
    handle.write("\n".join(missing)+("\n" if missing else ""))
print("annotation rows",len(rows),"queries without rows",len(missing))
```

·····

·····
·····query·····

……
……ID……ID……
……

……

……
……

……CPU……
……I/O……

……query…seed ortholog……eggNOG-mapper‥
……

….

eggNOG-mapper 2.1.13…· https://github.com/eggnogdb/eggnog-mapper/blob/2.1.13/emapper.py

eggNOG-mapper…… https://github.com/eggnogdb/eggnog-mapper/blob/2.1.13/USAGE.md

eggNOG-mapper 2.1.13 release https://github.com/eggnogdb/eggnog-mapper/releases/tag/2.1.13