


TOGA

Dostupne verzije i moduli

Verzija	Modul
1.1.2	bioinfo/TOGA/1.1.2

Korištenje

 TOGA je program koji ovisno o opsegu posla i veličini gena, može biti dosta zahtjevan.

Primjer je napravljen po uzoru na : <https://github.com/hillerlab/TOGA>

Podaci skinuti za ovaj primjer: - Human 2bit: <https://hgdownload.cse.ucsc.edu/goldenpath/hg38/bigZips/hg38.2bit>

- Mouse 2bit: <https://hgdownload.cse.ucsc.edu/goldenpath/mm10/bigZips/mm10.2bit>

Detaljnije objašnjenje same komande i drugih argumenata : <https://github.com/hillerlab/TOGA>

TOGA.sge

```
#!/bin/sh
#$ -N toga_example
#$ -pe p28-mpisingle 28
#$ -R y
#$ -cwd

module load bioinfo/TOGA/1.1.2

toga.py test_input/hg38.mm10.chr11.chain test_input/hg38.genCode27.chr11.bed hg38.2bit mm10.2bit --pn test -i
supply/hg38.wgEncodeGencodeCompV34.isoforms.txt --chn 100 --cjn 10 --u12 supply/hg38.U12sites.tsv --ms --nc
nextflow_config
```

Objašnjenje primjera

Preporučljivo je izvoditi TOGA na cijelom čvoru. Ovisno o broju poslova koje se da sa flagovima **--chn** i **--cjn**. U primjeru je sa **--chn** ili **--chain_jobs_num** dato 100 poslova, koji se relativno brzo izvode. Ovisno o tome koliko smo resursa dali u datoteci **extract_chain_features_config.nf** u direktoriju **nextflow_config** on će toliko koristiti. Isto se tako odnosi na **--cjn** ili **--cesar_jobs_num** koji koriste veliku količinu RAM memorije, ovisno o veličini genoma, tako da se treba paziti koliko se daje RAM-a po poslu kako ne bi prešlo dostupnu količinu memorije (inače poslovi se zruše i dobivaju se JAVA greške). Niže su objašnjeni detalji nextflow konfiguracije koju TOGA koristi kako bi se izvršavali poslovi.

Sadržaj nextflow_config direktorija

```
[mhrzenja@teran nextflow_config]$ ls -l
total 2
-rw-r--r-- 1 mhrzenja grid 366 Svi 3 16:09 call_cesar_config_template.nf
-rw-r--r-- 1 mhrzenja grid 198 Svi 3 16:10 cesar_bigmemo_config.nf
-rw-r--r-- 1 mhrzenja grid 71 Svi 3 16:09 extract_chain_features_config.nf
```

call_cesar_config_template.nf

```
process.executor = "local"
process.penv = "p28"
process.time = "24h"
processor.cpus = 2
process.memory = "4GB"
```

extract_chain_features_config.nf

```
process.executor = "local"
process.penv = "p28"
process.cpus = 2 #dajemo 2 dretve po poslu, brzo se izvode stoga je u redu dati po 2 threada na 100 poslova
```

cesar_bigmem_config.nf

```
#Dajemo 2 dretve po poslu i 4GB memorije
process.executor = "local"
process.penv = "p28"
process.memory = "4GB"
executor.cpus = 2
```

Vrijeme trajanja ovog primjera posla:

Primjer sa nextflow

```
start_time    Wed May  3 16:10:31 2023
end_time      Wed May  3 18:48:48 2023
```

Primjer posla bez --nc(bez nextflowa):



Ovaj primjer se odnosi bez konfiguracijskih datoteka, tj bez korištenja nextflow flag-a u komandi. Bez korištenja parametara posao se svojevoljno širi po čvoru i zauzima koliko može resursa, stoga kada dođu na red veliki memorijski procesi oko 20-30% poslova faila jer ostanu bez dostupne memorije. Stoga je potrebno ograničiti broj poslova **-chn** i **-cjn**.

Pošto u primjeru zahtjevamo manje resursa tj. ne cijeli čvor brže se dođe na red za izvođenje. Treba pripaziti sa ovom opcijom da ne zadamo previše poslova jer će program ometati druge korisnike. U ovom primjeru 2 big mem posla su zauzela po 8-10 GB radne memorije stoga dajemo $2 \times 15 = 30$ GB da se zauzme.

TOGA_no_nextflow.sge

```
#!/bin/sh
#$ -N tg
#$ -pe p28-mpisingle 2
#$ -cwd
#$ -R y
#$ -l memory=15

module load bioinfo/TOGA/1.1.2

toga.py test_input/hg38.mm10.chr11.chain test_input/hg38.genCode27.chr11.bed hg38.2bit mm10.2bit --pn test -i
supply/hg38.wgEncodeGencodeCompV34.isoforms.txt --chn $NSLOTS --cjn $NSLOTS --u12 supply/hg38.U12sites.tsv --ms
```

Vrijeme trajanja ovog primjera posla:

Primjer bez nextflow-a

start_time	Thu May 4 09:08:06 2023
end_time	Thu May 4 20:10:18 2023

Instalacija:

```
#Stvaranje conda env za TOGA deps
conda create -p /apps/virtenv/TOGA/cenv
conda activate /apps/virtenv/TOGA/cenv
conda install -c bioconda nextflow

#Instaliravanje TOGA programa
git clone https://github.com/hillerlab/TOGA.git
mv TOGA/ 1.1.2/
cd 1.1.2
pip install -r requirements.txt
module load gcc/9
./configure.sh
```