biogizmo: Python Gadgets for Bioinformatics

1. Dependencies

- python 2
- biopython package
- pandas package
- biomart package
- sqlalchemy package
- Network access to UCSC Genome Browser, TogoWS API and BioMart.

2. Functionality

2.1 Fetch SNP Information via Biomart

Input any number of rsid, return chromosome, coordinate and strand information in hg19 build in BED format.

Usage:

```
usage: biogizmo.py snp [-h] -i RSID_LST [RSID_LST ...]

Fetch SNP information.

optional arguments:
  -h, --help            show this help message and exit
  -i RSID_LST [RSID_LST ...], --rsid RSID_LST [RSID_LST
...] rsid of interest; separate by space if there are
multiple
```

A quick test:

```
$ python2 biogizmo.py snp -i rs11 rs12
chr7    11364200        11364201        rs11    0       +
chr7    11337163        11337164        rs12    0       +
```

2.2 Fetch Human Genome Reference Sequences via Togows

Input chromsome and coordinate, return human genome reference sequence in hg19/hg38 in fasta format.

Usage:

```
usage: biogizmo.py hg-refseq [-h] --build {hg19,hg38} --
chrom CHROM
                             --chrom-start CHROM_START --
chrom-end CHROM_END

Fetch Human Genome Reference Sequence.

optional arguments:
  -h, --help            show this help message and exit
  --build {hg19,hg38}   human genome build
  --chrom CHROM         chrom of interest (Must start with
'chr'. E.g. 'chr1')
  --chrom-start CHROM_START
                        start position on chromosome (0-
based)
  --chrom-end CHROM_END
                        end position on chromosome (0-
based)
```

A quick test:

```
$ python2 biogizmo.py hg-refseq --build hg19 --chrom chr1
--chrom-start 830000 --chrom-end 830040
```

```
>hg19:chr1:830000-830040
CTCACAAGGCTTTTGCTGAGACCATCAAAGCCTTTAACCTC
```

2.3 Fetch ChIA-PET Clusters via UCSC Genome Browser and Togows

Query UCSC Genome Browser hg19 database to fetch cluster information, which would be parsed and submitted to Togows for cluster sequences. Output would be in fasta format.

UCSC Genome Browser keeps 15 tables of ChIA-PET information, based on the experiments on different cell lines and protein factors with replication. To make commands simpler, table-keys are used to represent the long table names.

table key	**table name**
Hct116Pol2Rep1	wgEncodeGisChiaPetHct116Pol2InteractionsRep1
Helas3Pol2Rep1	wgEncodeGisChiaPetHelas3Pol2InteractionsRep1
K562CtcfRep1	wgEncodeGisChiaPetK562CtcfInteractionsRep1
K562Pol2Rep1	wgEncodeGisChiaPetK562Pol2InteractionsRep1
K562Pol2Rep2	wgEncodeGisChiaPetK562Pol2InteractionsRep2
Mcf7CtcfRep1	wgEncodeGisChiaPetMcf7CtcfInteractionsRep1
Mcf7CtcfRep2	wgEncodeGisChiaPetMcf7CtcfInteractionsRep2
Mcf7EraaRep1	wgEncodeGisChiaPetMcf7EraaInteractionsRep1
Mcf7EraaRep2	wgEncodeGisChiaPetMcf7EraaInteractionsRep2
Mcf7EraaRep3	wgEncodeGisChiaPetMcf7EraaInteractionsRep3
Mcf7Pol2Rep1	wgEncodeGisChiaPetMcf7Pol2InteractionsRep1
Mcf7Pol2Rep2	wgEncodeGisChiaPetMcf7Pol2InteractionsRep2

Mcf7Pol2Rep3	wgEncodeGisChiaPetMcf7Pol2InteractionsRep3
Mcf7Pol2Rep4	wgEncodeGisChiaPetMcf7Pol2InteractionsRep4
Nb4Pol2Rep1	wgEncodeGisChiaPetNb4Pol2InteractionsRep1

A ChIA-PET cluster contains 2 blocks, so every cluster found by the command your enter would generate a fasta file of 2 sequences.

Usage:

```
usage: biogizmo.py chia-pet [-h] --table-key

{Nb4Pol2Rep1,K562CtcfRep1,Hct116Pol2Rep1,Mcf7Pol2Rep4,Mcf7
Pol2Rep3,Mcf7Pol2Rep2,Mcf7Pol2Rep1,Mcf7EraaRep2,K562Pol2Re
p1,Helas3Pol2Rep1,Mcf7EraaRep3,Mcf7CtcfRep1,Mcf7CtcfRep2,M
cf7EraaRep1,K562Pol2Rep2}
                            --scope-type {within,overlap}
--chrom CHROM
                            --chrom-start CHROM_START --
chrom-end CHROM_END
                            --out-dir OUT_DIR

Fetch ChIA-PET cluster sequences.

optional arguments:
  -h, --help            show this help message and exit
  --table-key
{Nb4Pol2Rep1,K562CtcfRep1,Hct116Pol2Rep1,Mcf7Pol2Rep4,Mcf7
Pol2Rep3,Mcf7Pol2Rep2,Mcf7Pol2Rep1,Mcf7EraaRep2,K562Pol2Re
p1,Helas3Pol2Rep1,Mcf7EraaRep3,Mcf7CtcfRep1,Mcf7CtcfRep2,M
cf7EraaRep1,K562Pol2Rep2}
                        short name of the UCSC Genome
Browser ChIA-PET tables
  --scope-type {within,overlap}
```

```
                        when 'within', search ChIA-PET
clusters within range
                        ['start', 'end']; when 'overlap',
search ChIA-PET
                        clusters overlapping range
['start', 'end']
  --chrom CHROM         chrom of interest (Must start with
'chr'. E.g. 'chr1')
  --chrom-start CHROM_START
                        start position to search on
chromosome (0-based)
  --chrom-end CHROM_END
                        end position to search on
chromosome (0-based)
  --out-dir OUT_DIR     directory to for the output fasta
files
```

A quick test:

```
$ python2 biogizmo.py chia-pet --table-key K562CtcfRep1 --
scope-type overlap --chrom chr1 --chrom-start 830000 --
chrom-end 860000  --out-dir .
./cluster0_K562CtcfRep1_chr1_839813_999431.fasta
./cluster1_K562CtcfRep1_chr1_839717_874070.fasta
./cluster2_K562CtcfRep1_chr1_839841_856780.fasta
./cluster3_K562CtcfRep1_chr1_839974_848569.fasta
4 cluster fasta files written to .
```

2.4 Pairwise Alignment via Biopython

Perform global (Needleman-Wunsch) or local (Smith-Waterman) alignment to 2 DNA sequences. Match score and mismatch/gap-opening/gap-extension penalties can be specified. By default, alignments and scores would be reported; if --score-only is present, only

the best alignment would be given.

Usage:

```
usage: biogizmo.py pw-aln [-h] --alg {global,local} --
input-fasta INPUT_FASTA
                          --match MATCH --mismatch
MISMATCH --gap-open
                          GAP_OPEN --gap-extend GAP_EXTEND
[--score-only]

Perform pairwise alignment.

optional arguments:
  -h, --help            show this help message and exit
  --alg {global,local}  whether to perform global
alignment (Needleman-Wunsch)
                        or local alignment (Smith-
Waterman)
  --input-fasta INPUT_FASTA
                        the fasta file of input sequences
(must contain
                        exactly 2 sequences)
  --match MATCH         match score (must be positive)
  --mismatch MISMATCH   mismatch penalty (cannot be
positive)
  --gap-open GAP_OPEN   penalty when opening a gap (cannot
be positive)
  --gap-extend GAP_EXTEND
                        penalty when extending an existing
gap (cannot be
                        positive)
  --score-only          If specified, only the best
alignment score will be
```

```
                                reported.
```

A quick test:

```
$ python2 biogizmo.py pw-aln --alg global --match 1 --
mismatch -1 --gap-open -1 --gap-extend -1 --input-fasta
cluster0_K562CtcfRep1_chr1_839813_999431.fasta --score-
only
35.0
```

3. Future Work

I am working on SNP classificaion problems and ChIA-PET might be a good source of features. biogizmo helps me perform further inverstigation on ChIA-PET sequences.

Future work may include:

- Search for repeats on ChIA-PET sequences
- Motif analysis based on ChiA-PET cluster alignment
- Feature engineering for SNPs within ChiA-PET clusters