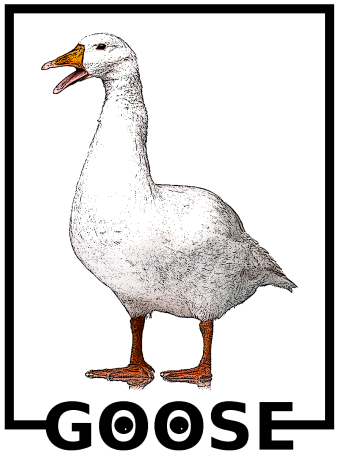

A toolkit for DNA sequence analysis and manipulation

D. Pratas (pratas@ua.pt)
A. J. Pinho (ap@ua.pt)

IEETA/DETI, University of Aveiro, Portugal

Version 1.7.17

Contents

1. Introduction

Recent advances in DNA sequencing have revolutionized the field of genomics, making it possible for research groups to generate large amounts of sequenced data, very rapidly and at substantially lower cost. Its storage have been made using specific file formats, such as FASTQ and FASTA. Therefore, its analysis and manipulation is crucial [1]. Several frameworks for analysis and manipulation emerged, namely `GALAXY` [2], `GATK` [3], `HTSeq` [4], `MEGA` [5], among others. In the majority, these frameworks require licenses and do not provide a low level access to the information, since they are commonly approached by scripting or interfaces.

We describe `GOOSE`, a (free) novel toolkit for analyzing and manipulating FASTA-FASTQ formats and sequences (DNA, amino acids, text), with many complementary tools. The toolkit is for Linux-based systems, built for fast processing. `GOOSE` supports pipes for easy integration. It includes tools for information display, randomizing, edition, conversion, extraction, searching, calculation and visualization. `GOOSE` is prepared to deal with very large datasets, typically in the scale Gigabytes or Terabytes.

The toolkit is a command line version, using the prefix "goose-" followed by the suffix with the respective name of the program. `GOOSE` is implemented in `C` language and it is available, under GPLv3, at:

```
https://pratas.github.io/goose
```

1.1 Installation

For `GOOSE` installation, run:

```
git clone https://github.com/pratas/goose.git
cd goose/src/
make
```

1.2 License

The license is **GPLv3**. In resume, everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed. For details on the license, consult: `http://www.gnu.org/licenses/gpl-3.0.html`.

2. FASTQ tools

Current available tools for FASTQ format analysis and manipulation include:

1. `goose-fastq2fasta`
2. `goose-fastq2mfasta`
3. `goose-fastqclustreads`
4. `goose-FastqExcludeN`
5. `goose-FastqExtractQualityScores`
6. `goose-FastqInfo`
7. `goose-FastqMaximumReadSize`
8. `goose-FastqMinimumLocalQualityScoreForward`
9. `goose-FastqMinimumLocalQualityScoreReverse`
10. `goose-FastqMinimumQualityScore`
11. `goose-FastqMinimumReadSize`
12. `goose-count`
13. `goose-extractreadbypattern`
14. `goose-fastqpack`
15. `goose-fastqsimulation`
16. `goose-FastqSplit`
17. `goose-FastqTrimm`
18. `goose-fastqunpack`
19. `goose-filter`
20. `goose-findnpos`

21. goose-genrandomdna
22. goose-getunique
23. goose-info
24. goose-mfmotifcoords
25. goose-mutatefastq
26. goose-newlineonnewx
27. goose-period
28. goose-permuteseqbyblocks
29. goose-randfastqextrachars
30. goose-real2binthreshold
31. goose-reducematrixbythreshold
32. goose-renamehumanheaders
33. goose-searchphash
34. goose-seq2fasta
35. goose-seq2fastq
36. goose-SequenceToGroupSequence
37. goose-splitreads
38. goose-wsearch

3. FASTA tools

1. `goose-fasta2seq`
2. `goose-fastaextract`
3. `goose-fastainfo`
4. `goose-mutatefasta`
5. `goose-randfastaextrachars`
6. `goose-geco`
7. `goose-gede`
8. `goose-mutatedna`
9. `goose-randseqextrachars`
10. `goose-reverse`: it reverses the order of a sequence.
11. `goose-reverselm`: it reverses the order of a large sequence. Low memory usage for large files.

4. Amino acid sequence tools

Current available amino acid sequence tools, for analysis and manipulation, are:

1. `goose-AminoAcidToGroup`
2. `goose-ProteinToPseudoDNA`

4.1 Program goose-AminoAcidToGroup

The `goose-AminoAcidToGroup` converts an amino acid sequence to a group sequence.
For help type:

```
./goose-AminoAcidToGroup -h
```

In the following subsections, we explain the input and output paramters.

Input parameters

The `goose-AminoAcidToGroup` program needs two streams for the computation, namely the input and output standard. The input stream is an amino acid sequence. The attribution is given according to:

```
Usage: ./goose-AminoAcidToGroup < in.prot > out.group
It converts a amino acid sequence to a group sequence.
Table:
Prot    Group
R   P
H   P  Amino acids with electric charged side chains: POSITIVE
K   P
-   -
D   N
E   N  Amino acids with electric charged side chains: NEGATIVE
-   -
S   U
T   U
N   U  Amino acids with electric UNCHARGED side chains
Q   U
-   -
C   S
U   S
G   S  Special cases
P   S
```

```
-   -
A   H
V   H
I   H
L   H
M   H  Amino acids with hydrophobic side chains
F   H
Y   H
W   H
-   -
*   *  Others
X   X  Unknown
```

It can be used to group amino acids by properties, such as electric charge (positive and negative), uncharged side chains, hydrophobic side chains and special cases. An example on such an input file is:

```
IPFLLKKQFALADKLVLSKLRQLLGGRIKMMPCGGAKLEPAIGLFFHAIGINIKLGYGMTETTATVSCWHDFQFNPNSIG
TLMPKAEVKIGENNEILVRGGMVMKGYYKKPEETAQAFTEDGFLKTGDAGEFDEQGNLFITDRIKELMKTSNGKYIAPQY
IESKIGKDKFIEQIAIIADAKKYVSALIVPCFDSLEEYAKQLNIKYHDRLELLKNSDILKMFE
```

Output

The output of the `goose-AminoAcidToGroup` program is a group sequence.
An example, for the input, is:

```
HSHHHPPUHHHHNPHHHUPHPUHHSSPHPHHSSSSHPHNSHHSHHHPHHSHUHPHSHSHUNUUHUHUSHPNHUHUSUUHS
UHHSPHNHPHSNUUNHHHPSSHHHPSHHPPSNNUHUHHUNNSHHPUSNHSNHNNUSUHHHUNPHPNHHPUUUSPHHHSUH
HNUPHSPNPHHNUHHHHHNHPPHHUHHHHSSHNUHNNHHPUHUHPHPNPHNHHPUUNHHPHHN
```

4.2 Program goose-ProteinToPseudoDNA

The `goose-ProteinToPseudoDNA` converts an amino acid (protein) sequence to a pseudo DNA sequence.
For help type:

```
./goose-ProteinToPseudoDNA -h
```

In the following subsections, we explain the input and output paramters.

Input parameters

The `goose-ProteinToPseudoDNA` program needs two streams for the computation, namely the input and output standard. The input stream is an amino acid sequence. The attribution is given according to:

```
Usage: ./goose-ProteinToPseudoDNA < in.prot > out.dna
It converts a protein sequence to a pseudo DNA sequence.
Table:
Prot  DNA
A     GCA
C     TGC
```

```
D     GAC
E     GAG
F     TTT
G     GGC
H     CAT
I     ATC
K     AAA
L     CTG
M     ATG
N     AAC
P     CCG
Q     CAG
R     CGT
S     TCT
T     ACG
V     GTA
W     TGG
Y     TAC
*     TAG
X     GGG
```

It can be used to generate pseudo-DNA with characteristics passed by amino acid (protein) sequences. An example on such an input file is:

```
IPFLLKKQFALADKLVLSKLRQLLGGRIKMMPCGGAKLEPAIGLFFHAIGINIKLGYGMTETTATVSCWHDFQFNPNSIG
TLMPKAEVKIGENNEILVRGGMVMKGYYKKPEETAQAFTEDGFLKTGDAGEFDEQGNLFITDRIKELMKTSNGKYIAPQY
IESKIGKDKFIEQIAIIADAKKYVSALIVPCFDSLEEYAKQLNIKYHDRLELLKNSDILKMFE
```

Output

The output of the `goose-ProteinToPseudoDNA` program is a DNA sequence.
An example, for the input, is:

```
ATCCCGTTTCTGCTGAAAAAACAGTTTGCACTGGCAGACAAACTGGTACTGTCTAAACTGCGTCAGCTGCTGGGCGGCCG
TATCAAAATGATGCCGTGCGGCGGCGCAAAACTGGAGCCGGCAATCGGCCTGTTTTTTCATGCAATCGGCATCAACATCA
AACTGGGCTACGGCATGACGGAGACGACGGCAACGGTATCTTGCTGGCATGACTTTCAGTTTAACCCGAACTCTATCGGC
ACGCTGATGCCGAAAGCAGAGGTAAAAATCGGCGAGAACAACGAGATCCTGGTACGTGGCGGCATGGTAATGAAAGGCTA
CTACAAAAAACCGGAGGAGACGGCACAGGCATTTACGGAGGACGGCTTTCTGAAAACGGGCGACGCAGGCGAGTTTGACG
AGCAGGGCAACCTGTTTATCACGGACCGTATCAAAGAGCTGATGAAAACGTCTAACGGCAAATACATCGCACCGCAGTAC
ATCGAGTCTAAAATCGGCAAAGACAAATTTATCGAGCAGATCGCAATCATCGCAGACGCAAAAAAATACGTATCTGCACT
GATCGTACCGTGCTTTGACTCTCTGGAGGAGTACGCAAAACAGCTGAACATCAAATACCATGACCGTCTGGAGCTGCTGA
AAAACTCTGACATCCTGAAAATGTTTGAG
```

5. General purpose tools

1. `goose-comparativemap`: visualisation of comparative maps. It builds a image given specific patterns between two sequences.
2. `goose-BruteForceString`: it generates, line by line, multiple combinations of strings up to a certain size.
3. `goose-char2line`: it transforms each char into a char in each line.
4. `goose-sum`: it adds the second column value to the first column value.
5. `goose-min`: it finds the minimum value between two column values.
6. `goose-minus`: it substracts the second column value to the first column value.
7. `goose-max`: it finds the mmaximum value between two column values.
8. `goose-extract`: it extracts a subsequence of a sequence by coordinates.
9. `goose-segment`: it segments a sequence given a certain threshold.

Bibliography

[1] H. Buermans and J. Den Dunnen, "Next generation sequencing technology: advances and applications," *Biochimica et Biophysica Acta (BBA)-Molecular Basis of Disease*, vol. 1842, no. 10, pp. 1932–1941, 2014.

[2] B. Giardine, C. Riemer, R. C. Hardison, R. Burhans, L. Elnitski, P. Shah, Y. Zhang, D. Blankenberg, I. Albert, J. Taylor *et al.*, "Galaxy: a platform for interactive large-scale genome analysis," *Genome research*, vol. 15, no. 10, pp. 1451–1455, 2005.

[3] M. A. DePristo, E. Banks, R. Poplin, K. V. Garimella, J. R. Maguire, C. Hartl, A. A. Philippakis, G. Del Angel, M. A. Rivas, M. Hanna *et al.*, "A framework for variation discovery and genotyping using next-generation dna sequencing data," *Nature genetics*, vol. 43, no. 5, pp. 491–498, 2011.

[4] S. Anders, P. T. Pyl, and W. Huber, "Htseq–a python framework to work with high-throughput sequencing data," *Bioinformatics*, p. btu638, 2014.

[5] S. Kumar, G. Stecher, and K. Tamura, "Mega7: Molecular evolutionary genetics analysis version 7.0 for bigger datasets," *Molecular Biology and Evolution*, p. msw054, 2016.

[6] D. Pratas, A. J. Pinho, and P. J. S. G. Ferreira, "Efficient compression of genomic sequences," in *Proc. of the Data Compression Conf., DCC-2016*, Snowbird, Utah, Mar. 2016, pp. 231–240.

[7] D. Pratas, "Compression and analysis of genomic data," Ph.D. dissertation, University of Aveiro, 2016.