

Supplementary material of “OptimJV3: an optimization tool for improving genomic data compression using genetic algorithms”

R. Ferrolho, A. J. Pinho, and D. Pratas

1 Datasets and materials

Table 1 presents all datasets (sequences) used in the experiments, excluding samples.

2 Random and Local Search Results - Supplementary Material

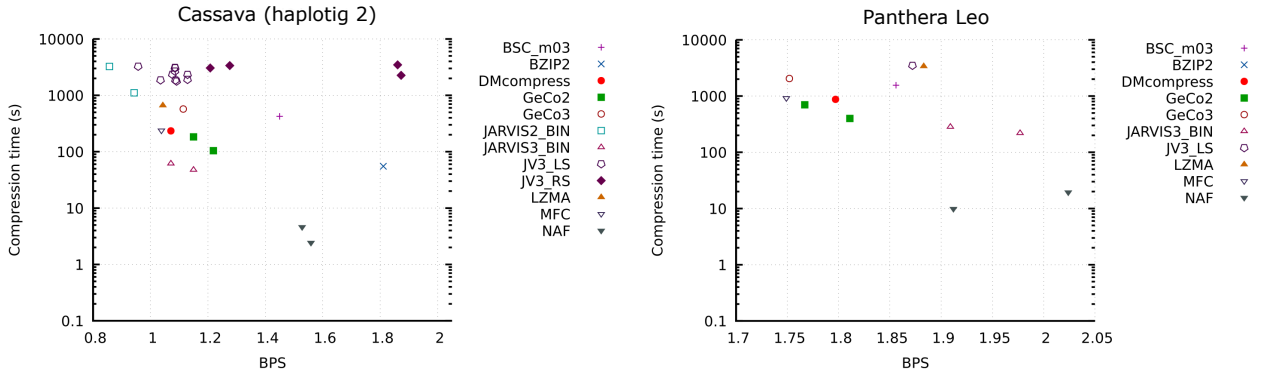


Figure S1: Results of Random Search (RS) for Cassava (haplotig2), and Local Search (LS) for Pathera Leo.

3 Reproducibility

3.1 Quick Demo

To clone the OptimJV3 project and change script permissions, execute the following instructions:

```
1 git clone https://github.com/cobilab/OptimJV3.git
2 chmod +x OptimJV3/scripts/*.sh
```

After cloning and activating the script permissions, the following demonstration, where a GA is applied to the compression the human chromosome Y (CY), can be executed in the scripts folder:

```
1 ./GetDSinfo.sh # map sequences into its DS, sorted by size; view sequences info
2 ./GA.sh -s cy -ga "demo" -lg 100 -t 10 # optimize compression of CY using canonical GA for 100
   generations and 10 threads
```

The results will then be stored in a folder with prefix "DS" within the "OptimJV3" project. To identify this folder name, the following command can be executed:

```
1 ./Main.sh -v
```

For instance, assuming that there are no other sequences installed, the output folder should be "DS1".

Table S1: Enumeration of all datasets considered; RS/LS breadth experiments span this full list, whereas MGA/MOGA deep analyses are reported for CY, HG, and Cassava H1/H2. Scripts and outputs for the remaining datasets are available in the repository. ID refers to the NCBI sequence ID or an abbreviation if from another source.

DS	Source	ID	Description	Size
DS1	NCBI	KT868810.1	Cutavirus strain BR-283 NS1 gene, partial cds; and putative VP1, hypothetical protein, VP2, and hypothetical protein genes	4.09668 KB
DS2	NCBI	CM029732.1	Pollicipes pollicipes isolate AB1234 mitochondrion, complete sequence	14.7363 KB
DS3	DNACorpus	BuEb	Bundibugyo ebolavirus genome	18.4961 KB
DS4	NCBI	OM812693.1	SARS-CoV-2 isolate HUN/Hun-1/2020, complete genome	29.0693 KB
DS5	AlcoR	alcor30KB	Synthetic sequence	29.2969 KB
DS6	DNACorpus	AgPh	Aggregatibacter phage S1249 genome	42.9395 KB
DS7	DNACorpus	YeMi	Yellowstone lake mimivirus genome	71.9619 KB
DS8	NCBI	NC_001664.4	Human betaherpesvirus 6A, variant A, isolate U1102, complete genome	155.643 KB
DS9	NCBI	NC_000898.1	Human herpesvirus 6B, complete genome	158.314 KB
DS10	NCBI	NC_000908.2	Mycoplasmoides genitalium G37, complete sequence	566.48 KB
DS11	DNACorpus	AeCa	Aeropyrum camini genome	1.51734 MB
DS12	DNACorpus	HePy	Helicobacter pylori genome	1.59056 MB
DS13	AlcoR	alcor2MB	Synthetic sequence with five low-complexity regions	2 MB
DS14	NCBI	NC_004461.1	Staphylococcus epidermidis ATCC 12228, complete sequence	2.3835 MB
DS15	DNACorpus	HaHi	Haloarcula hispanica genome	3.7098 MB
DS16	DNACorpus	EsCo	Escherichia coli genome	4.42662 MB
DS17	DNACorpus	PlFa	Plasmodium falciparum genome	8.5704 MB
DS18	DNACorpus	WaMe	Wallemia mellicola genome	8.72081 MB
DS19	DNACorpus	ScPo	Schizosaccharomyces pombe genome	10.1587 MB
DS20	NCBI	NC_000024.1	Human chromosome Y	21.6181 MB
DS21	DNACorpus	EnIn	Entamoeba invadens genome	25.1799 MB
DS22	DNACorpus	DrMe	Drosophila miranda chromosome 2	30.6906 MB
DS23	NCBI	BA000046.3	Pan troglodytes DNA, chromosome 22, complete sequence	31.1909 MB
DS24	NCBI	NC_000021.9	Human chromosome 21	38.2315 MB
DS25	DNACorpus	OrSa	Oriza sativa Japonica chromosome 1	41.2584 MB
DS26	NCBI	NC_073246.2	Gorilla gorilla gorilla isolate KB3781 chromosome 22	55.7969 MB
DS27	DNACorpus	DaRe	Danio rerio chromosome 3	59.6667 MB
DS28	NCBI	NC_072005.2	Pongo abelii isolate AG06213 chromosome 20	62.3704 MB
DS29	DNACorpus	AnCa	Anolis carolinensis genome	135.603 MB
DS30	NCBI	NC_000008.11	Human chromosome 8	138.062 MB
DS31	NCBI	NC_058373.1	Felis Catus chromosome B3	141.611 MB
DS32	DNACorpus	GaGa	Gallus gallus chromosome 2	141.651 MB
DS33	DNACorpus	HoSa	Human chromosome 4	180.962 MB
DS34	CNGB ¹	Cassava2	Cassava (haplotig 2)	673.607 MB
DS35	CNGB ²	Cassava	Cassava (haplotig 1)	727.074 MB
DS36	NCBI	GCA.008795835.1	Panthera Leo genome	2.22454 GB
DS37	³	HG	Human genome	2.9032 GB

¹ https://s3.ap-northeast-1.wasabisys.com/gigadb-datasets/live/pub/10.5524/102001_103000/102193/00_Assembly_Fasta/haplotigs/TME204.HiFi_HiC.haplotig2.fa

² https://s3.ap-northeast-1.wasabisys.com/gigadb-datasets/live/pub/10.5524/102001_103000/102193/00_Assembly_Fasta/haplotigs/TME204.HiFi_HiC.haplotig1.fa

³ https://s3-us-west-2.amazonaws.com/human-pangenomics/T2T/CHM13/assemblies/analysis_set/chm13v2.0.fa.gz

3.2 Setup

After cloning and changing script permissions, the following instruction should be executed to install all tools and sequences required to reproduce the experiments:

```
1 ./Setup.sh
```

Alternatively, in the scripts folder, the following instructions should be executed:

```
1 ./InstallTools.sh      # install listed compressors, GT0, and AlcoR
2 ./DownloadFASTA.sh    # downloads FASTA files
3 ./GetCassava.sh        # gunzip cassava files
4 ./GetAlcoRFASTA.sh    # simulates and stores 2 synthetic FASTA sequences
5 ./FASTA2seq.sh         # cleans FASTA files and stores raw sequence files
6 ./DownloadDNACorpus.sh # download raw sequences from a balanced sequence corpus
7 ./GetSample.sh -s cassava -sp 0.4 -mb 100 -so cassava100MB
8 ./GetSample.sh -s cassava -sp 0.4 -mb 50 -so cassava50MB
9 ./GetSample.sh -s cassava -sp 0.4 -mb 25 -so cassava25MB
10 ./GetSample.sh -s cassava -sp 0.4 -mb 12.5 -so cassava12d5MB
11 ./GetSample.sh -s human -sp 0.2 -mb 10 -so human10MB
12 ./GetDSinfo.sh        # map sequences into their ids, sorted by size; view sequences info
```

3.3 Reproducing the “Non-MGA results”

To reproduce this experiment, the following commands should be executed in the scripts folder:

```
1 bash -x ./GA.sh -s $sequence -ga "rs" -lr 0 -lg 1 1> out 2> err &
```

3.4 Reproducing the “Canonical MGA experiment”

To reproduce this experiment, the following commands should be executed in the scripts folder:

```
1 bash -x ./GA.sh -s "escherichia_coli" -ga "e0_gai_lr0_cmga" -lr 0 -lg 100 1> out 2> err &
2 bash -x ./GA.sh -s cy -ga "e0_gai_lr0_cmga" -lr 0 -lg 100 1> out 2> err &
3 bash -x ./GA.sh -s cassava -ga "e0_gai_lr0_cmga" -lr 0 -lg 20 1> out 2> err &
```

3.5 Reproducing the “CY experiments”

To reproduce these experiments, the following command should be executed in the scripts folder:

```
1 bash -x ./Main.sh -s cy -lg 100 -t 10 1> out 2> err &
```

3.6 Reproducing the “Cassava sampling experiment”

To reproduce this experiment, run:

```
1 ./SamplingDemoCassava.sh
```

3.7 Reproducing the “Human genome sampling experiments”

To reproduce this experiment for human genome samples greater than 10 MB, run:

```
1 ./SamplingDemo.sh
```

As for the 10 MB experiment, the following command should be executed to run a GA algorithm for 500 generations:

```
1 bash -x ./GA.sh -s human10MB -lg 500 1> out 2> err &
```

4 Parameters and options of OptimJV3

All implemented features are listed in the following scripts:

```
1 ./InstallTools.sh -h
2 ./GetDSinfo.sh -h
3 ./Main.sh -h
4 ./GA.sh -h
5 ./Initialization.sh -h
6 ./Run.sh -h
7 ./Evaluation.sh -h
```

```

8 ./Selection.sh -h
9 ./CrossMut.sh -h
10 ./GetSamples.sh -h
11 ./PlotGA.sh -h
12 ./PlotGAcmp.sh -h

```

4.1 Install tools menu

To access the menu of the script that installs all required tools, the following command can be executed

```
1 ./InstallTools.sh -h
```

This command will output the following content

```

1 -----
2
3 OptimJV3 - optimize JARVIS3 CM and RM parameters
4
5 Program options -----
6
7 -h|--help.....Show this
8 -iwc|--install-with-conda.....Install some tools with
9                                     conda
10 -iwb|--install-with-both.....Install all tools with
11                                     conda and globally
12
13 -----

```

with the options/parameters available.

4.2 Download FASTA sequences menu

To access the menu of the script that downloads a set of FASTA sequences, or a specific FASTA sequence given its NCBI ID, the following command can be executed

```
1 ./DownloadFASTA.sh -h
```

This command will output the following content

```

1 -----
2
3 CompressSequences - JARIVS3 Optimization Benchmark
4 Download FASTA files script
5
6 Program options -----
7
8 --help|-h.....Show this
9 -id.....Download sequence by NCBI id
10
11 -----

```

with the options/parameters available.

4.3 Map sequences to dataset ID menu

The GetDSinfo.sh maps and displays information about sequences, their dataset id, and their size. The command to access its features is:

```
1 ./GetDSinfo.sh -h
```

This command will output the following content

```

1 -----
2
3 OptimJV3 - optimize JARVIS3 CM and RM parameters
4
5 Program options -----
6
7 -h|--help.....Show this
8 -v|--view-ds|--view-datasets.....View datasets info
9
10 -----

```

with the options/parameters available.

4.4 Main menu

The main script runs a set of pre-configured GAs and applied to a set of sequences.

The command to access the main menu with its features is

```
1 ./Main.sh -h
```

This command will output the following content

```
1 -----
2
3 OptimJV3 - optimize JARVIS3 CM and RM parameters
4
5 Program options -----
6
7 -h|--help.....Show this
8 -v|--view-ds|--view-datasets...View sequence names, size
9       of each in bytes, MB, and BG, and their group
10 -s|--seq|--sequence.....Select sequence by its name
11 -sg|--sequence-grp|--seq-group..Select group of sequences
12                                   by their size
13 -ds|--dataset.....Select sequence by its dataset number
14 -dr|--drange|--dsrange|--dataset-range.....Select
15       sequences by range of dataset numbers
16 -fg|--first-gen|--first-generation.....Define first
17                                   generation number
18 -lg|--last-gen|--last-generation.....Define last
19                                   generation number
20 -rg|--range-gen|--range-generation....Define generation
21                                   range
22 -t|--nthreads....Define number of threads to run JARVIS3
23                                   in parallel
24 -sd|--seed.....Define pseudo-random seed
25 -si|--seed-increment.....Define seed increment
26
27 example 1: ./Main.sh -s human
28 example 2: ./Main.sh -s cassava -s human
29
30 -----
```

with the options/parameters available.

4.5 GA menu

The GA script applies a GA to a set of sequences.

The command to access the GA menu with the options/parameters of OptimJV3 is

```
1 ./GA.sh -h
```

This command will output the following content

```
1 -----
2
3 OptimJV3 - optimize JARVIS3 CM and RM parameters
4
5 Program options -----
6
7 -h|--help.....Show this
8 -v|--view-ds|--view-datasets...View sequence names, size
9       of each in bytes, MB, and BG, and their group
10 -fg|--first-generation...Specify first generation number
11 -lg|--last-generation.....Select last generation number
12                                   by their size
13 -a|-ga|--genetic-algorithm...Define (folder) name of the
14                                   genetic algorithm
```

```

15 -s|--seq|--sequence.....Select sequence name
16 -sg|--seq-grp|--sequence-group....Select sequence group
17 -ds|--dataset.....Select sequence by its dataset number
18 -dr|--drange|--dsrange|--dataset-range.....Select
19     sequences by range of dataset numbers
20 -ps|--psize|--population|--population-size.....Define
21     population size
22 -sd|--seed.....Define pseudo-random seed
23 -si|--seed-increment.....Define seed increment
24
25 Program options (initialization) -----
26
27 -hei|--heuristic-initialization.....Activate heuristic
28     initialization/local initialization
29 -hyi|--hybrid-initialization.....Activate hybrid
30     initialization
31 -hhp|--hybrid-heuristic-percentage....Define percentage
32     of tests generated by heuristic/local search
33 -hhp|--hybrid-heuristic-percentage.....Define number
34     of tests generated by heuristic/local search
35 -mCM|--m-cm|--min-cm....Define minimum number of context
36     models (CMs)
37 -MCM|--M-cm|--max-cm.....Define maximum number of
38     context models (CMs)
39 -mRM|--m-rm|--min-rm.....Define minimum number of
40     copy/repeat models (RMs)
41 -MRM|--M-rm|--max-rm.....Define maximum number of
42     copy/repeat models (RMs)
43 -lr|--learning-rate.....Define learning rate
44 -hs|--hidden-size.....Define hidden size
45 -sing|--seeding.....Activate seeding feature
46 to populate with few hardcoded solutions. Only works for
47     human genome
48
49 Program options (run) -----
50
51 -t|--nthreads....Define number of threads to run JARVIS3
52     in parallel
53
54 Program options (evaluation) -----
55
56 --moga|--moga-wm|--moga-weighted-metric...Activate Multi
57     Objective Genetic Algorithm (MOGA) using weighted
58     metric function
59 --moga-ws|--moga-weighted-sum...Activate Multi-Objective
60     Genetic Algorithm (MOGA) using weighted sum function
61 -pe|--p-exp|--p-expoent.....Define expoent for MOGA
62     weighted metric function
63 -w1|--wBPS|--w-bps|--weight-bps....Bits Per Symbol (BPS)
64     weight for MOGA function
65 -w2|--wCTIME|--w-ctime|--weight-ctime...Compression time
66     weight for MOGA function
67
68 Program options (selection) -----
69
70 -sl|--sel|--selection.....Choose selection
71     operator: 'elitist' (default), 'roulette', 'rank', or
72     'tournament'
73 -ns|--num-sel-cmds...Define number of commands to select
74 -sr|--selection-rate...Define rate of commands to select
75

```

```

76 Program options (crossover) -----
77
78 -cr|-ccr|--command-crossover-rate..Define crossover rate
79                                     of a selected pair of commands
80 -mrc|--model-crossover-rate.....Define crossover rate
81                                     of a selected pair of CMs or RMs
82 -cc|--command-crossover.....Choose command crossover
83 operator: 'mrc' (metameric random crossover) (default),
84                                     (metameric canonical crossover) 'mcc'
85 -c|--crossover.....Choose model crossover operator:
86   'xpoint' (default), 'uniform', 'average', 'discrete',
87                                     'flat', 'heuristic'
88
89 Program options (mutation) -----
90
91 -hm|--heuristic-mutation.....Activate heuristic mutation
92                                     for narrower range mutations
93 -mr|-cmr|--command-mutation-rate....Define mutation rate
94                                     of a command
95 -pmr|--parameter-mutation-rate.....Define mutation rate
96                                     of a parameter
97
98 Program options (stop criteria) -----
99
100 -sc|--stop-criteria.....Define stop criteria: '1' to
101     halt program when no offspring is produced,
102     else (default) stop at last generation

```

with the options/parameters available.

4.6 Initialization menu

The initialization script starts a GA by initializing a population of commands that compress a chosen sequence.

To access the initialization menu, the following menu can be executed

```

1 ./Initialization.sh -h

```

This command will output the following content

```

1 -----
2
3 OptimJV3 - optimize JARVIS3 CM and RM parameters
4
5 Program options -----
6
7 -h|--help.....Show this
8 -v|--view-ds|--view-datasets..View sequence names, size
9                                     of each in bytes, MB, and BG, and their group
10 -s|--seq|--sequence.....Select sequence by its name
11 -sg|--sequence-grp|--seq-group.Select group of sequences
12                                     by their size
13 -a|-ga|--genetic-algorithm...Define (folder) name of the
14                                     genetic algorithm
15 -s|--seq|--sequence.....Select sequence name
16 -sg|--seq-grp|--sequence-group....Select sequence group
17 -ds|--dataset.....Select sequence by its dataset number
18 -dr|--drange|--dsrange|--dataset-range.....Select
19                                     sequences by range of dataset numbers
20 -ps|--psize|--population|--population-size.....Define
21                                     population size
22 -sd|--seed.....Define pseudo-random seed
23 -si|--seed-increment.....Define seed increment
24

```

```

25 Program options (initialization) -----
26
27 -hei|--heuristic-initialization.....Activate heuristic
28         initialization/local initialization
29 -hyi|--hybrid-initialization.....Activate hybrid
30         initialization
31 -hhp|--hybrid-heuristic-percentage....Define percentage
32         of tests generated by heuristic/local search
33 -hhp|--hybrid-heuristic-percentage.....Define number
34         of tests generated by heuristic/local search
35 -mCM|--m-cm|--min-cm...Define minimum number of context
36         models (CMs)
37 -MCM|--M-cm|--max-cm.....Define maximum number of
38         context models (CMs)
39 -mRM|--m-rm|--min-rm.....Define minimum number of
40         copy/repeat models (RMs)
41 -MRM|--M-rm|--max-rm.....Define maximum number of
42         copy/repeat models (RMs)
43 -lr|--learning-rate.....Define learning rate
44 -hs|--hidden-size.....Define hidden size
45 -sing|--seeding.....Activate seeding feature
46 to populate with few hardcoded solutions. Only works for
47         human genome

```

with the options/parameters available.

4.7 Run menu

The run script is used to run the initial population or offspring from a certain generation.

To access the menu of the run script, the following command can be executed

```
1 ./Run.sh -h
```

This command will output the following content

```

1 -----
2
3 OptimJV3 - optimize JARVIS3 CM and RM parameters
4
5 Program options -----
6
7 -h|--help.....Show this
8 -v|--view-ds|--view-datasets...View sequence names, size
9         of each in bytes, MB, and BG, and their group
10 -s|--seq|--sequence.....Select sequence by its name
11 -sg|--sequence-grp|--seq-group.Select group of sequences
12         by their size
13 -a|--ga|--genetic-algorithm...Define (folder) name of the
14         genetic algorithm
15 -ds|--dataset.....Select sequence by its dataset number
16 -dr|--drange|--dsrange|--dataset-range.....Select
17         sequences by range of dataset numbers
18 -ps|--psize|--population|--population-size.....Define
19         population size
20 -sd|--seed.....Define pseudo-random seed
21 -si|--seed-increment.....Define seed increment
22
23 Program options (run) -----
24
25 -t|--nthreads....Define number of threads to run JARVIS3
26         in parallel
27 -to|--timeout.....Define timeout (default: 1 hour)

```

with the options/parameters available.

4.8 Evaluation menu

This script is used to evaluate and sort the current generation based on their compression, expressed in BPS, and compression time.

To access the menu of evaluation script, the following command can be executed

```
1 ./Evaluation.sh -h
```

This command will output the following content

```
1 -----
2
3 OptimJV3 - optimize JARVIS3 CM and RM parameters
4
5 Program options -----
6
7 -h|--help.....Show this
8 -v|--view-ds|--view-datasets...View sequence names, size
9       of each in bytes, MB, and BG, and their group
10 -s|--seq|--sequence.....Select sequence by its name
11 -sg|--sequence-grp|--seq-group.Select group of sequences
12                               by their size
13 -a|--ga|--genetic-algorithm...Define (folder) name of the
14                               genetic algorithm
15 -s|--seq|--sequence.....Select sequence name
16 -sg|--seq-grp|--sequence-group....Select sequence group
17 -ds|--dataset.....Select sequence by its dataset number
18 -dr|--drange|--dsrange|--dataset-range.....Select
19       sequences by range of dataset numbers
20 -ps|--psize|--population|--population-size.....Define
21       population size
22 -sd|--seed.....Define pseudo-random seed
23 -si|--seed-increment.....Define seed increment
24
25 Program options (evaluation) -----
26
27 --moga|--moga-wm|--moga-weighted-metric...Activate Multi
28       Objective Genetic Algorithm (MOGA) using weighted
29       metric function
30 --moga-ws|--moga-weighted-sum...Activate Multi-Objective
31       Genetic Algorithm (MOGA) using weighted sum function
32 -pe|--p-exp|--p-expoent.....Define expoent for MOGA
33       weighted metric function
34 -w1|--wBPS|--w-bps|--weight-bps....Bits Per Symbol (BPS)
35       weight for MOGA function
36 -w2|--wCTIME|--w-ctime|--weight-ctime...Compression time
37       weight for MOGA function
```

with the options/parameters available.

4.9 Selection menu

This script selects commands from the population as potential candidates for producing offspring.

To access the selection menu, the following command can be executed

```
1 ./Selection.sh -h
```

This command will output the following content

```
1 -----
2
3 OptimJV3 - optimize JARVIS3 CM and RM parameters
4
5 Program options -----
6
```

```

7 -h|--help.....Show this
8 -v|--view-ds|--view-datasets...View sequence names, size
9   of each in bytes, MB, and BG, and their group
10 -s|--seq|--sequence.....Select sequence by its name
11 -sg|--sequence-grp|--seq-group.Select group of sequences
12   by their size
13 -a|-ga|--genetic-algorithm...Define (folder) name of the
14   genetic algorithm
15 -s|--seq|--sequence.....Select sequence name
16 -sg|--seq-grp|--sequence-group....Select sequence group
17 -ds|--dataset.....Select sequence by its dataset number
18 -dr|--drange|--dsrange|--dataset-range.....Select
19   sequences by range of dataset numbers
20 -ps|--psize|--population|--population-size.....Define
21   population size
22 -sd|--seed.....Define pseudo-random seed
23 -si|--seed-increment.....Define seed increment
24
25 Program options (selection) -----
26
27 -sl|--sel|--selection.....Choose selection
28   operator: 'elitist' (default), 'roulette', 'rank', or
29   'tournament'
30 -ns|--num-sel-cmds...Define number of commands to select
31 -sr|--selection-rate...Define rate of commands to select

```

with the options/parameters available.

4.10 Crossover and Mutation menu

The crossover and mutation script is used to produce offspring.

To access the menu of this script, the following command can be executed

```
1 ./CrossMut.sh -h
```

This command will output the following content

```

1 -----
2
3 OptimJV3 - optimize JARVIS3 CM and RM parameters
4
5 Program options -----
6
7 -h|--help.....Show this
8 -v|--view-ds|--view-datasets...View sequence names, size
9   of each in bytes, MB, and BG, and their group
10 -s|--seq|--sequence.....Select sequence by its name
11 -sg|--sequence-grp|--seq-group.Select group of sequences
12   by their size
13 -a|-ga|--genetic-algorithm...Define (folder) name of the
14   genetic algorithm
15 -s|--seq|--sequence.....Select sequence name
16 -sg|--seq-grp|--sequence-group....Select sequence group
17 -ds|--dataset.....Select sequence by its dataset number
18 -dr|--drange|--dsrange|--dataset-range.....Select
19   sequences by range of dataset numbers
20 -ps|--psize|--population|--population-size.....Define
21   population size
22 -sd|--seed.....Define pseudo-random seed
23 -si|--seed-increment.....Define seed increment
24
25 Program options (crossover) -----
26

```

```

27 -cr|-ccr|--command-crossover-rate..Define crossover rate
28         of a selected pair of commands
29 -mrc|--model-crossover-rate.....Define crossover rate
30         of a selected pair of CMs or RMs
31 -cc|--command-crossover.....Choose command crossover
32 operator: 'mrc' (metameric random crossover) (default),
33         (metameric canonical crossover) 'mcc'
34 -c|--crossover.....Choose model crossover operator:
35     'xpoint' (default), 'uniform', 'average', 'discrete',
36         'flat', 'heuristic'
37
38 Program options (mutation) -----
39
40 -hm|--heuristic-mutation....Activate heuristic mutation
41         for narrower range mutations
42 -mr|-cmr|--command-mutation-rate....Define mutation rate
43         of a command
44 -pmr|--parameter-mutation-rate.....Define mutation rate
45         of a parameter

```

with the options/parameters available.

4.11 Get Sample menu

The features of the script that outputs a sample file can be found by running the following command

```
1 ./GetSample.sh -h
```

This command will output the following content

```

1 -----
2
3 OptimJV3 - optimize JARVIS3 CM and RM parameters
4
5 Program options -----
6
7 -h|--help.....Show this
8 -s|--sequence.....Choose sequence name/file
9 -sp|--start-percentage....Define Percentage of the full
10         sequence where sample begins
11 -sz|-mb|--size-mb.....Define sample size
12 -so|--sample-output.....Define sample output filename
13
14 -----
15

```

with the options/parameters available.

4.12 Get Samples menu

The features of the script that outputs four sample files from a sequence - with sizes of 100MB, 50MB, 25MB and 12.5MB - can be found by running the following command

```
1 ./GetSamples.sh -h
```

This command will output the following content

```

1 -----
2
3 OptimJV3 - optimize JARVIS3 CM and RM parameters
4
5 Program options -----
6
7 -h|--help.....Show this
8 -s|--seq|--sequence.....Choose sequence name/file

```

```

9 -----
10

```

with the options/parameters available.

4.13 Plot GA menu

The features of the script that plots the evolutionary plots and histograms of a single GA can be found by running the following command

```
1 ./PlotGA.sh -h
```

This command will output the following content

```

1 -----
2
3 OptimJV3 - optimize JARVIS3 CM and RM parameters
4
5 Program options -----
6
7 -h|--help.....Show this
8 -a|--ga|--genetic-algorithm...Define (folder) name of the
9                               genetic algorithm
10 -s|--seq|--sequence.....Choose sequence name/file
11 -ds|--dataset.....Select sequence by its dataset number
12 -pb|--percentage-best.....Define percentage of best
13                               individuals to plot
14 -b|--best.....Define number of best individuals to plot
15 -fg|--first-generation...Specify first generation number
16 -lg|--last-generation.....Select last generation number
17 -br|--b-range.....Define x-axis (BPS range)
18 -trs|--trange-s...Define y-axis (time range, in seconds)
19 -trm|--trange-m...Define y-axis (time range, in minutes)
20 -trh|--trange-h...Define y-axis (time range, in hours)
21 -hi|--hist-interval.....Define bin size for histogram
22
23 -----

```

with the options/parameters available.

4.14 Plot GA comparison menu

The features of the script that plots the comparison between GAs of a specific experiment can be found by running the following command

```
1 ./PlotGAcmp.sh -h
```

This command will output the following content

```

1 -----
2
3 OptimJV3 - optimize JARVIS3 CM and RM parameters
4
5 Program options -----
6
7 -h|--help.....Show this
8 -e|--experiment-number.....Define experiment number:
9     1. Compare GAs for different initialization methods
10    2. Compare GAs for different population sizes
11    3. Compare MOGAs
12    4. Compare GAs for different selection methods
13    5. Compare GAs for different crossover methods
14 -ds|--dataset.....Select sequence by its dataset number
15 -fg|--first-generation...Specify first generation number
16 -lg|--last-generation.....Select last generation number

```

```

17 -br|--b-range.....Define x-axis (BPS range)
18 -trs|--trange-s...Define y-axis (time range, in seconds)
19 -trm|--trange-m...Define y-axis (time range, in minutes)
20 -trh|--trange-h....Define y-axis (time range, in hours)
21
22 -----

```

with the options/parameters available.

5 Parameters and options of CompressSequences

```

1 ./InstallTools.sh -h
2 ./DownloadFASTA.sh -h
3 ./GetDSInfo.sh -h
4 ./RunTestsExample.sh
5 ./ProcessBenchRes.sh -h
6 ./Plot.sh -h

```

5.1 Install tools menu

See “Install tools menu” subsection from “Parameters and options of OptimJV3” section.

5.2 Download FASTA sequences menu

See “Download FASTA sequences menu” subsection from “Parameters and options of OptimJV3” section.

5.3 Map sequences to dataset ID menu

See “Map sequences to dataset ID menu” subsection from “Parameters and options of OptimJV3” section.

5.4 Plot menu

The features of the script that plots the results of benchmark data can be found by running

```

1 ./Plot.sh -h

```

This command will output the following content

```

1 -----
2
3 CompressSequences - benchmark
4 Plot Script
5
6 Program options -----
7
8 -s|--sequence.....Select sequence
9                               of each
10 --sequence|--seq|-s.....Select sequence by its name
11 --sequence-group|--seq-grp|-sg.Select group of sequences
12                               by their size
13     1. Sequences with size lower than 1MB
14     2. Sequences with size between 1MB and 100MB
15     3. Sequences with size between 100MB and 1GB
16     4. Sequences with size between 1GB and 3GB
17     5. Sequences with size greater than 3GB
18 -br|--b-range.....Define x-axis (BPS range)
19 -trs|--trange-s...Define y-axis (time range, in seconds)
20 -trm|--trange-m...Define y-axis (time range, in minutes)
21 -trh|--trange-h....Define y-axis (time range, in hours)
22 -m|--mode....Select data to plot ('bench', 'NGA', 'all')
23
24 -----

```

with the options/parameters available.

5.5 Run menu

The features of the script that runs the generated JARVIS3 commands can be found by running

```
1 ./Run.sh -h
```

Alternatively, to access the menu of a similar script which runs fewer tests the following instruction can be executed:

```
1 ./RunLess.sh -h
```

Both instructions will output the following content

```
1 -----
2
3 CompressSequences - benchmark
4 Run Script
5
6 Program options -----
7
8 -h|--help.....Show this
9 -v|--view-ds|--view-datasets...View sequence names, size
10      of each in bytes, MB, and GB, and their group
11 -s|--seq|--sequence.....Select sequence by its name
12 -sg|--sequence-grp|--seq-group.Select group of sequences
13      by their size
14 -a|--ga|--genetic-algorithm...Define (folder) name of the
15      genetic algorithm
16 -ds|--dataset.....Select sequence by its dataset number
17 -dr|--drange|--dsrange|--dataset-range.....Select
18      sequences by range of dataset numbers
19 -g|--gen-num.....Define generation number
20 -to|--timeout.....Define timeout
21 -t|--nthreads....Define number of threads to run JARVIS3
22      in parallel
```

with the options/parameters available.

5.6 Process Benchmark results menu

The features of the script that processes the benchmark data can be found by running

```
1 ./ProcessBenchRes.sh -h
```

This command will output the following content

```
1 -----
2
3 OptimJV3 - optimize JARVIS3 CM and RM parameters
4
5 Program options -----
6
7 -h|--help.....Show this
8 -v|--view-ds|--view-datasets....View sequences and size
9      of each
10
11 -----
```

with the options/parameters available.

5.7 JARVIS3 Parameters

JARVIS3 [1] Context Models (-cm) and Repeat Models (-rm) are the following:

```
1
2 -cm [NB_C]:[NB_D]:[NB_I]:[NB_G]/[NB_S]:[NB_E]:[NB_R]:[NB_A]
3 Template of a context model.
4 Parameters:
5 [NB_C]: (integer [1;14]) order size of the regular context
6      model. Higher values use more RAM but, usually, are
7      related to a better compression score.
8 [NB_D]: (integer [1;5000]) denominator to build alpha, which
9      is a parameter estimator. Alpha is given by 1/[NB_D].
10      Higher values are usually used with higher [NB_C],
11      and related to confident bets. When [NB_D] is one,
12      the probabilities assume a Laplacian distribution.
13 [NB_I]: (integer {0,1,2}) number to define if a sub-program
14      which addresses the specific properties of DNA
```

```

sequences (Inverted repeats) is used or not. The
number 1 turns ON the sub-program using at the same
time the regular context model. The number 2 does
only contemplate the inversions only (NO regular). The
number 0 does not contemplate its use (Inverted repeats
OFF). The use of this sub-program increases the
necessary time to compress but it does not affect the
RAM.
[NB_G]: (real [0;1)) real number to define gamma. This value
represents the decayment forgetting factor of the
regular context model in definition.
[NB_S]: (integer [0;20]) maximum number of editions allowed
to use a substitutional tolerant model with the same
memory model of the regular context model with
order size equal to [NB_C]. The value 0 stands for
turning the tolerant context model off. When the
model is on, it pauses when the number of editions
is higher than [NB_C], while it is turned on when
a complete match of size [NB_C] is seen again. This
is probabilistic-algorithmic model very useful to
handle the high substitutional nature of genomic
sequences. When [NB_S] > 0, the compressor used more
processing time, but uses the same RAM and, usually,
achieves a substantial higher compression ratio. The
impact of this model is usually only noticed for
higher [NB_C].
[NB_R]: (integer {0,1}) number to define if a sub-program
which addresses the specific properties of DNA
sequences (Inverted repeats) is used or not. It is
similar to the [NR_I] but for tolerant models.
[NB_E]: (integer [1;5000]) denominator to build alpha for
substitutional tolerant context model. It is
analogous to [NB_D], however to be only used in the
probabilistic model for computing the statistics of
the substitutional tolerant context model.
[NB_A]: (real [0;1)) real number to define gamma. This value
represents the decayment forgetting factor of the
substitutional tolerant context model in definition.
Its definition and use is analogous to [NB_G].

... (you may use several context models)

-rm [NB_R]:[NB_C]:[NB_B]:[NB_L]:[NB_G]:[NB_I]:[NB_W]:[NB_Y]
Template of a repeat model.
Parameters:
[NB_R]: (integer [1;10000] maximum number of repeat models
for the class. On very repetitive sequences the RAM
increases along with this value, however it also
improves the compression capability.
[NB_C]: (integer [1;14]) order size of the repeat context
model. Higher values use more RAM but, usually, are
related to a better compression score.
[NB_B]: (real (0;1]) beta is a real value, which is a
parameter for discarding or maintaining a certain
repeat model.
[NB_L]: (integer (1;20]) a limit threshold to play with
[NB_B]. It accepts or not a certain repeat model.
[NB_G]: (real [0;1)) real number to define gamma. This value
represents the decayment forgetting factor of the
regular context model in definition.
[NB_I]: (integer {0,1,2}) number to define if a sub-program
which addresses the specific properties of DNA
sequences (Inverted repeats) is used or not. The
number 1 turns ON the sub-program using at the same
time the regular context model. The number 0 does
not contemplate its use (Inverted repeats OFF). The
number 2 uses exclusively Inverted repeats. The
use of this sub-program increases the necessary time
to compress but it does not affect the RAM.
[NB_W]: (real (0;1)) initial weight for the repeat class.
[NB_Y]: (integer {0}, [1;50]) maximum cache size. This will
use a table cache with the specified size. The size
must be in balance with the k-mer size [NB_C].

```

References

- [1] M. J. P. Sousa et al. JARVIS3: an efficient encoder for genomic data. *Bioinformatics*, 7(7), 2025.