User Manual

GENETIC ALGORITHM FOR OPERON PREDICTION IN PROKARYOTES

GAOPP

Operons by GENETIC ALGORITHM

Biomedical Informatics Division,

Rajendra Memorial Research Institute for Medical Sciences (I.C.M.R)

Patna, India.

Table of Content

---

1. INTRODUCTION TO THE TOOL

1.1 What is GAOPP:

GAOPP is standalone GUI tool for operon prediction. It uses unsupervised method Genetic Algorithm for identifying promoters in annotated prokaryotic species. It uses biological features like intergenic distance, Cluster of Gene Ontology and pathway involvement of each gene pair and clusters them in to operons. There are several computational methods are available for this purpose but none of them are GUI based. They need heavy data preparation, also. To meet these requirements GAOPP has been created.

It has three different evaluating functions to evaluate the fitness of each putative operon structure, can be found in literatures. These functions use biological properties like intergenic distance, involvement in metabolic pathway, and functionality from Clusters of Gene ontology (COG) gene functional families. This need needs the protein table file found at *National Centre for Biotechnology Information (NCBI)* FTP (ftp://ftp.ncbi.nih.gov/genomes/Bacteria/). For Pathway information *KyotoEncyclopedia of Genes and Genome (KEGG)* pathway database can be used. A track of experimental promoters in the target species can be used to predict promoters. Terminators can be predicted using *TranTerm* and the output file may be used to provide terminator coordinates in the genome. Windows version of the tool is currently available to download. Binaries for Linux platform will be released soon.

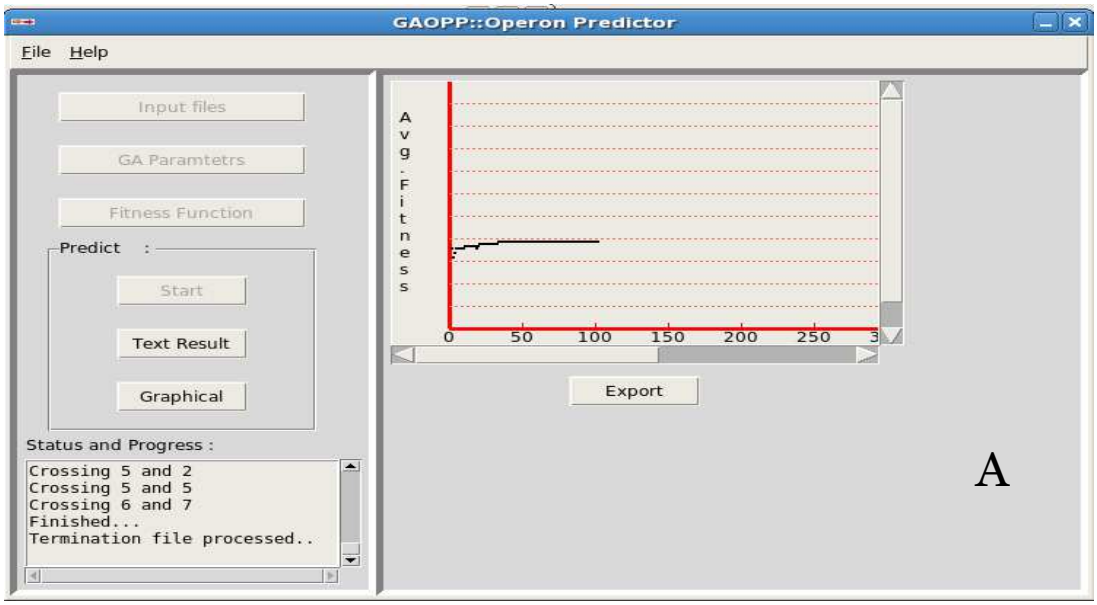
1.2 Installation on windows :

1. Download the zipped installation file and extract it.
2. To install the tool, simply double click on install.bat file.
3. It prompts you to enter installation directory. To accept default destination C:\GAOPP\ press ***y*** . Wait until the prompt closes. Double click on the shortcut icon at Desktop.

4. To run it from source code, it requires PERL5.8 above and Tkx module. Active perl can be used instead.
5. To uninstall the program, simply go to the folder you installed and delete GAOPP directory. Remove the Desktop shortcut.

1.3 System Requirement:

1. Operating system Windows 2000/XP/Vista/7 , Linux* (available soon)
2. To run from source code it requires perl5.8 or above and Tkx installed.
3. To run larger genome sequences it may require higher configuration.
4. Additional software like PDF reader and Post Script Viewer may be required.



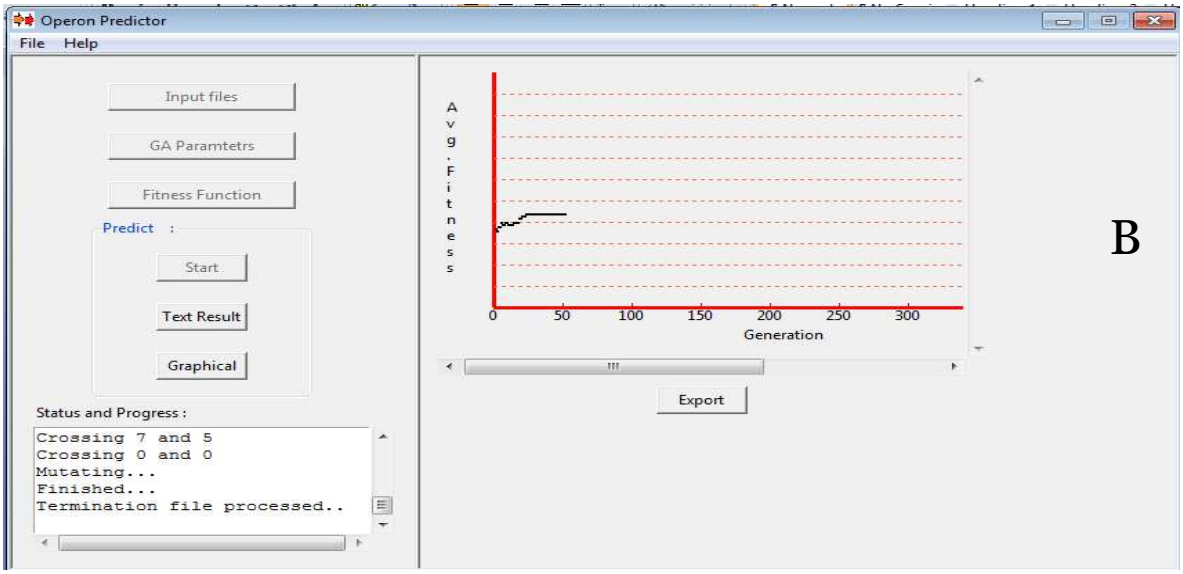
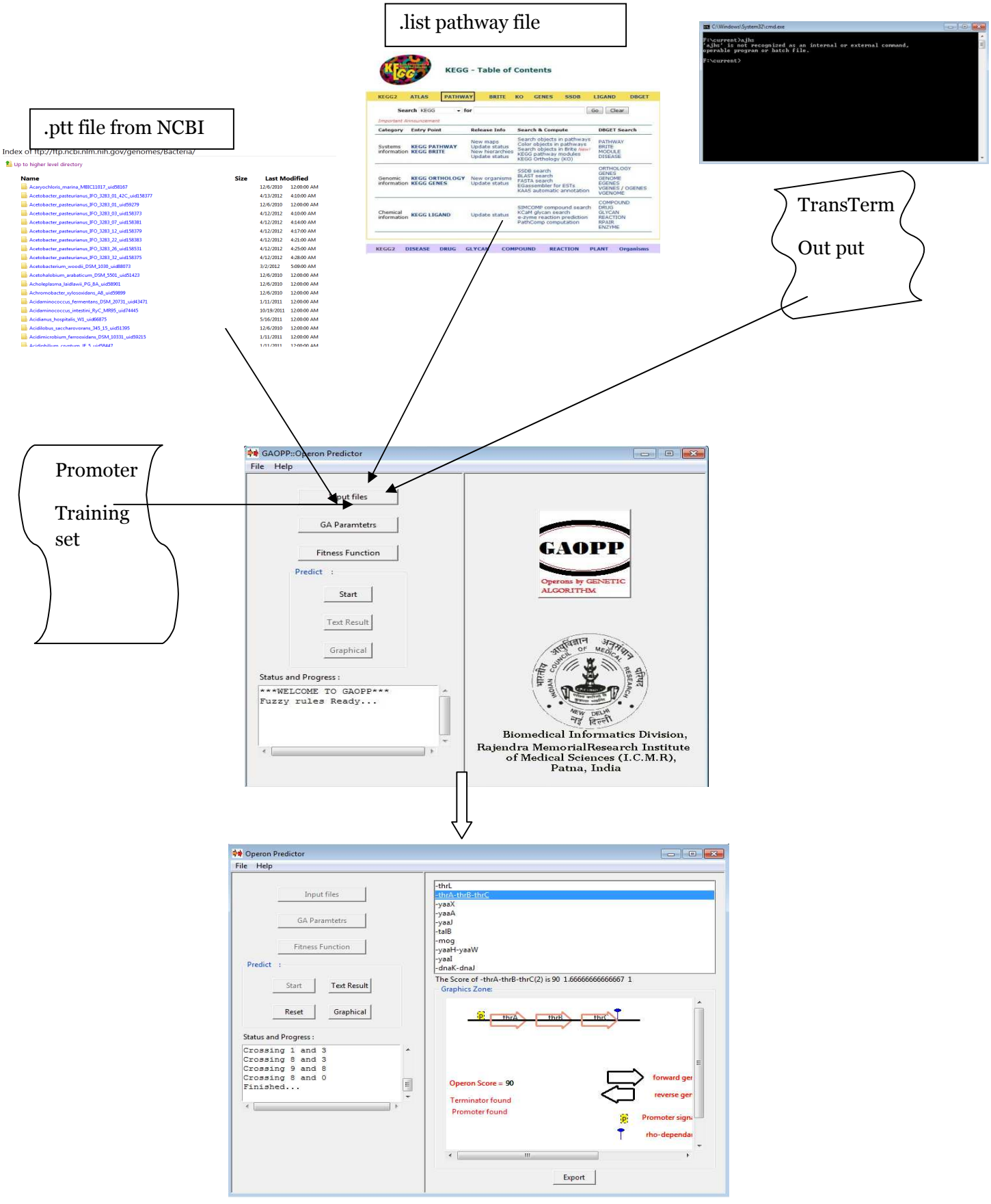


Fig -Different feel and look on Linux platform (A) and Windows Platform (B)



GAOPP: Genetic Algorithm for Operon Prediction in Prokaryotes

2. Working with GAOPP:

2.1 Input files:

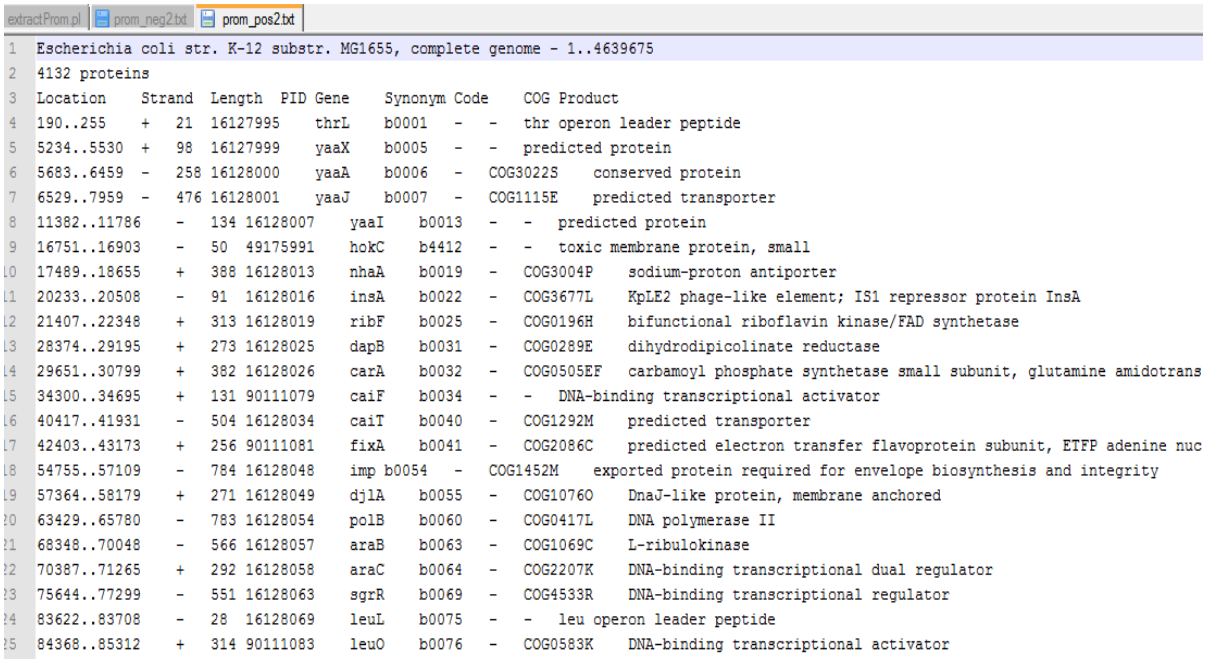
Download the required files like .ptt file and pathway file. Note down the KEGG organism code if you are planning to use pathway data, organism code has to be specified. Check that the .ppt file and pathway file are in the following format:

```
Escherichia coli str. K-12 substr. MG1655, complete genome - 1..4639675
4132 proteins
Location     Strand      Length      PID    Gene  Synonym     Code  COG    Product
190..255     +      21    16127995    thrL  b0001 -      -     thr operon leader peptide
337..2799    +      820   16127996    thrA  b0002 -      COG0460E,COG0527E fused aspartokinase I and homoserine
2801..3733   +      310   16127997    thrB  b0003 -      COG0083E    homoserine kinase
3734..5020   +      428   16127998    thrC  b0004 -      COG0498E    threonine synthase
5234..5530   +      98    16127999    yaaX  b0005 -      -     predicted protein
5683..6459   -      258   16128000    yaaA  b0006 -      COG3022S    conserved protein
6529..7959   -      476   16128001    yaaJ  b0007 -      COG1115E    predicted transporter
8238..9191   +      317   16128002    talB  b0008 -      COG0176G    transaldolase B
9306..9893   +      195   16128003    mog   b0009 -      COG0521H    predicted molybdochelatase
9928..10494  -      188   16128004    yaaH  b0010 -      COG1584S    conserved inner membrane protein associated
10643..11356       -      237   16128005    yaaW  b0011 -      COG4735S    conserved protein
```

```
path:eco00010   eco:b0114   eco:aceE ko:K00163 ec:1.2.4.1
path:eco00010   eco:b0115   eco:aceF ko:K00627 ec:2.3.1.12
path:eco00010   eco:b0116   eco:lpd ko:K00382 ec:1.8.1.4
path:eco00010   eco:b0356   eco:frmA ko:K00121 ec:1.1.1.1 ec:1.1.1.284
path:eco00010   eco:b0688   eco:pgm ko:K01835 ec:5.4.2.2
path:eco00010   eco:b0755   eco:gpmA ko:K01834 ec:5.4.2.1
path:eco00010   eco:b0756   eco:galM ko:K01785 ec:5.1.3.3
path:eco00010   eco:b1002   eco:agp ko:K01085 ec:3.1.3.10
```

For promoter prediction, a promoter training set need to be specified. A Perl script provided with the program may be used to extract the promoter and non promoter training sets. Simply run script specifying your input files and sequence file. The input files have same .ptt file format. To generate the positive input file, edit the .ptt file keeping only those genes which contains upstream promoter signals, and delete others. Similarly, for negative input file only those genes not having upstream promoter sequence. Run extractProm.pl :

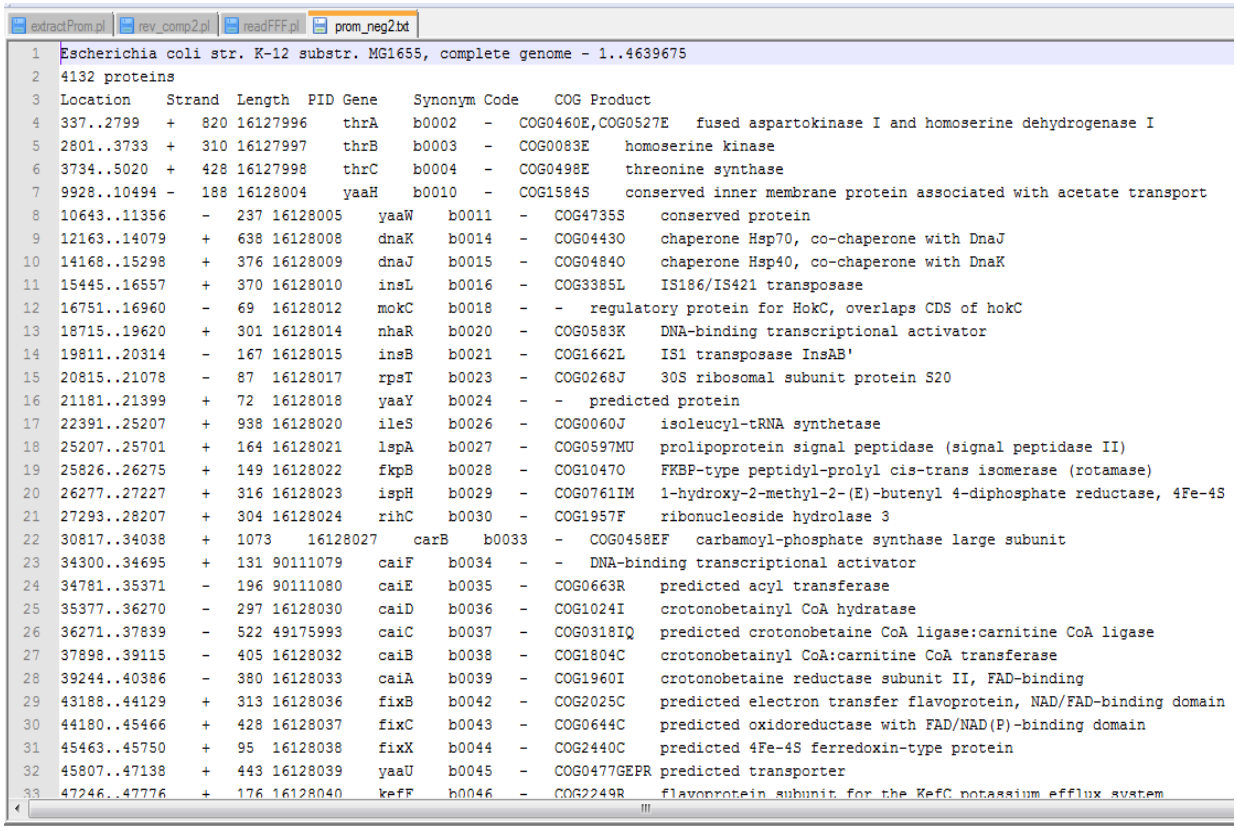
Perl extraxtProm.pl -pos <positive.ptt> -neg <negetive.ptt> -seq <nucl.fna> -out <output.txt>



extractProm.pl | prom_neg2.txt | prom_pos2.txt

```
Escherichia coli str. K-12 substr. MG1655, complete genome - 1..4639675
4132 proteins
Location    Strand  Length  PID Gene    Synonym Code    COG Product
190..255    +   21  16127995    thrL    b0001   -   -   thr operon leader peptide
5234..5530  +   98  16127999    yaaX    b0005   -   -   predicted protein
5683..6459  -   258 16128000    yaaA    b0006   -   COG3022S    conserved protein
6529..7959  -   476 16128001    yaaJ    b0007   -   COG1115E    predicted transporter
11382..11786    -   134 16128007    yaaI    b0013   -   -   predicted protein
16751..16903    -   50  49175991    hokC    b4412   -   -   toxic membrane protein, small
17489..18655    +   388 16128013    nhaA    b0019   -   COG3004P    sodium-proton antiporter
20233..20508    -   91  16128016    insA    b0022   -   COG3677L    KpLE2 phage-like element; IS1 repressor protein InsA
21407..22348    +   313 16128019    ribF    b0025   -   COG0196H    bifunctional riboflavin kinase/FAD synthetase
28374..29195    +   273 16128025    dapB    b0031   -   COG0289E    dihydrodipicolinate reductase
29651..30799    +   382 16128026    carA    b0032   -   COG0505EF   carbamoyl phosphate synthetase small subunit, glutamine amidotrans
34300..34695    +   131 90111079    caiF    b0034   -   -   DNA-binding transcriptional activator
40417..41931    -   504 16128034    caiT    b0040   -   COG1292M    predicted transporter
42403..43173    +   256 90111081    fixA    b0041   -   COG2086C    predicted electron transfer flavoprotein subunit, ETFP adenine nuc
54755..57109    -   784 16128048    imp b0054   -   COG1452M    exported protein required for envelope biosynthesis and integrity
57364..58179    +   271 16128049    djlA    b0055   -   COG10760    DnaJ-like protein, membrane anchored
63429..65780    -   783 16128054    polB    b0060   -   COG0417L    DNA polymerase II
68348..70048    -   566 16128057    araB    b0063   -   COG1069C    L-ribulokinase
70387..71265    +   292 16128058    araC    b0064   -   COG2207K    DNA-binding transcriptional dual regulator
75644..77299    -   551 16128063    sgrR    b0069   -   COG4533R    DNA-binding transcriptional regulator
83622..83708    -   28  16128069    leuL    b0075   -   -   leu operon leader peptide
84368..85312    +   314 90111083    leuO    b0076   -   COG0583K    DNA-binding transcriptional activator
```

Fig: Positive promoter input file

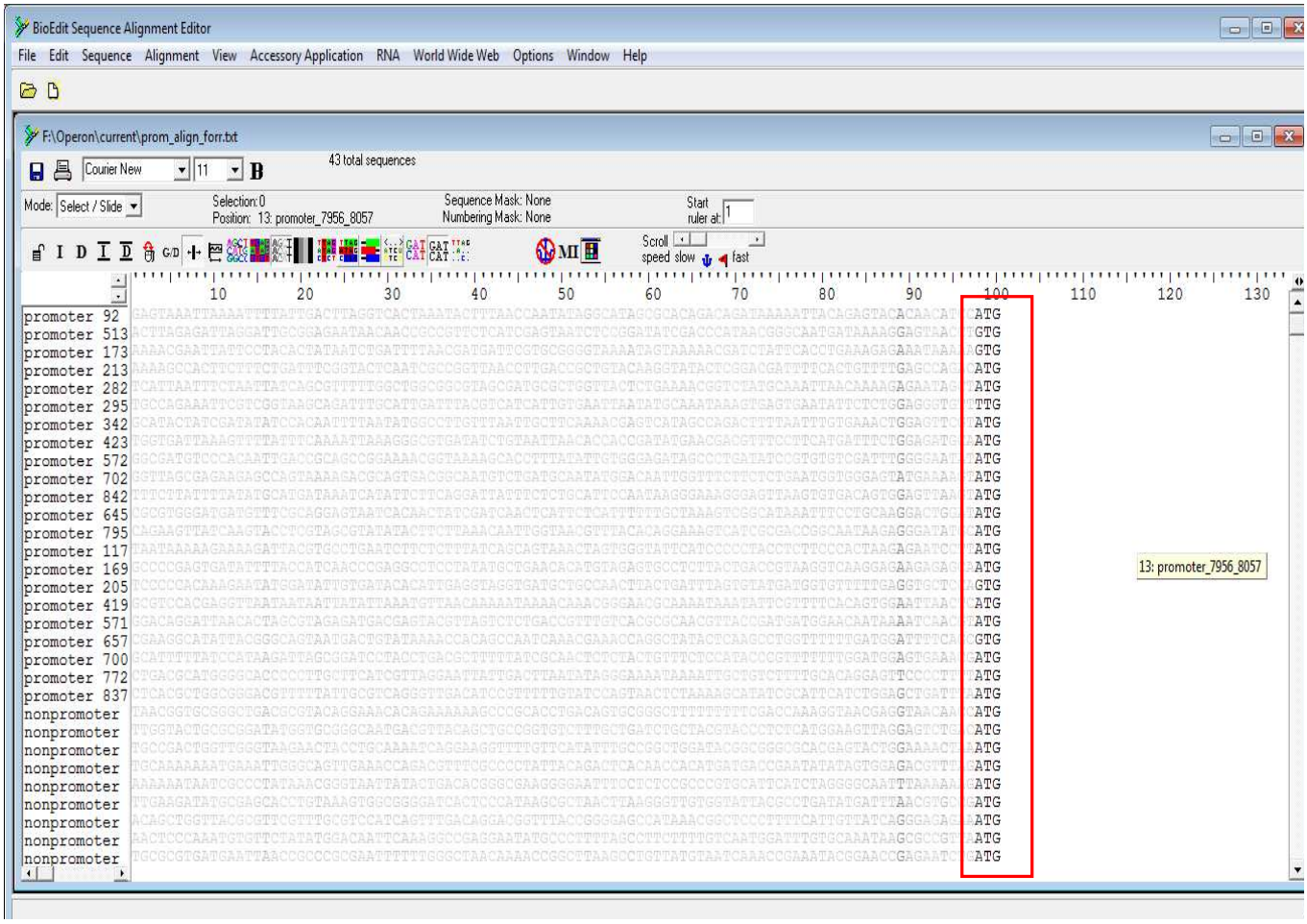


extractProm.pl | rev_comp2.pl | readFFF.pl | prom_neg2.txt

```
Escherichia coli str. K-12 substr. MG1655, complete genome - 1..4639675
4132 proteins
Location    Strand  Length  PID Gene    Synonym Code    COG Product
337..2799   +   820 16127996    thrA    b0002   -   COG0460E,COG0527E   fused aspartokinase I and homoserine dehydrogenase I
2801..3733  +   310 16127997    thrB    b0003   -   COG0083E    homoserine kinase
3734..5020  +   428 16127998    thrC    b0004   -   COG0498E    threonine synthase
9928..10494 -   188 16128004    yaaH    b0010   -   COG1584S    conserved inner membrane protein associated with acetate transport
10643..11356    -   237 16128005    yaaW    b0011   -   COG4735S    conserved protein
12163..14079    +   638 16128008    dnaK    b0014   -   COG0443O    chaperone Hsp70, co-chaperone with DnaJ
14168..15298    +   376 16128009    dnaJ    b0015   -   COG0484O    chaperone Hsp40, co-chaperone with DnaK
15445..16557    +   370 16128010    insL    b0016   -   COG3385L    IS186/IS421 transposase
16751..16960    -   69  16128012    mokC    b0018   -   -   regulatory protein for HokC, overlaps CDS of hokC
18715..19620    +   301 16128014    nhaR    b0020   -   COG0583K    DNA-binding transcriptional activator
19811..20314    -   167 16128015    insB    b0021   -   COG1662L    IS1 transposase InsAB'
20815..21078    -   87  16128017    rpsT    b0023   -   COG0268J    30S ribosomal subunit protein S20
21181..21399    +   72  16128018    yaaY    b0024   -   -   predicted protein
22391..25207    +   938 16128020    ileS    b0026   -   COG0060J    isoleucyl-tRNA synthetase
25207..25701    +   164 16128021    lspA    b0027   -   COG0597MU   prolipoprotein signal peptidase (signal peptidase II)
25826..26275    +   149 16128022    fkpB    b0028   -   COG1047O    FKBP-type peptidyl-prolyl cis-trans isomerase (rotamase)
26277..27227    +   316 16128023    ispH    b0029   -   COG0761IM   1-hydroxy-2-methyl-2-(E)-butenyl 4-diphosphate reductase, 4Fe-4S
27293..28207    +   304 16128024    rihC    b0030   -   COG1957F    ribonucleoside hydrolase 3
30817..34038    +   1073    16128027    carB    b0033   -   COG0458EF   carbamoyl-phosphate synthase large subunit
34300..34695    +   131 90111079    caiF    b0034   -   -   DNA-binding transcriptional activator
34781..35371    -   196 90111080    caiE    b0035   -   COG0663R    predicted acyl transferase
35377..36270    -   297 16128030    caiD    b0036   -   COG1024I    crotonobetainyl CoA hydratase
36271..37839    -   522 49175993    caiC    b0037   -   COG0318IQ   predicted crotonobetaine CoA ligase:carnitine CoA ligase
37898..39115    -   405 16128032    caiB    b0038   -   COG1804C    crotonobetainyl CoA:carnitine CoA transferase
39244..40386    -   380 16128033    caiA    b0039   -   COG1960I    crotonobetaine reductase subunit II, FAD-binding
43188..44129    +   313 16128036    fixB    b0042   -   COG2025C    predicted electron transfer flavoprotein, NAD/FAD-binding domain
44180..45466    +   428 16128037    fixC    b0043   -   COG0644C    predicted oxidoreductase with FAD/NAD(P)-binding domain
45463..45750    +   95  16128038    fixX    b0044   -   COG2440C    predicted 4Fe-4S ferredoxin-type protein
45807..47138    +   443 16128039    yaaU    b0045   -   COG0477GEPR predicted transporter
47246..47776    +   176 16128040    kefF    b0046   -   COG2249R    flavoprotein subunit for the KefC potassium efflux system
```

Fig: Negative promoter inputfile

Your file is ready if all the sequences must have A(G)TG at the right side.

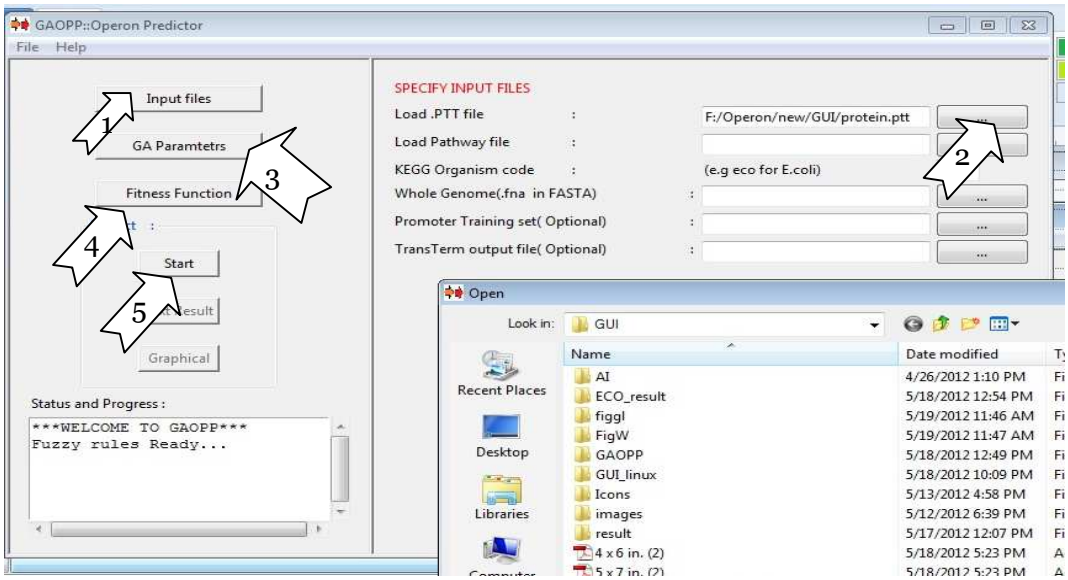


In order to generate the terminator coordinates, we have provided a compiled transterm binary executable and *expterm.dat* file. This will run **only on Linux** platform (see transTerm usage file) Run the following command on Linux Terminal:

transterm -p expterm.dat seq.fasta annotation.ptt > output.tt

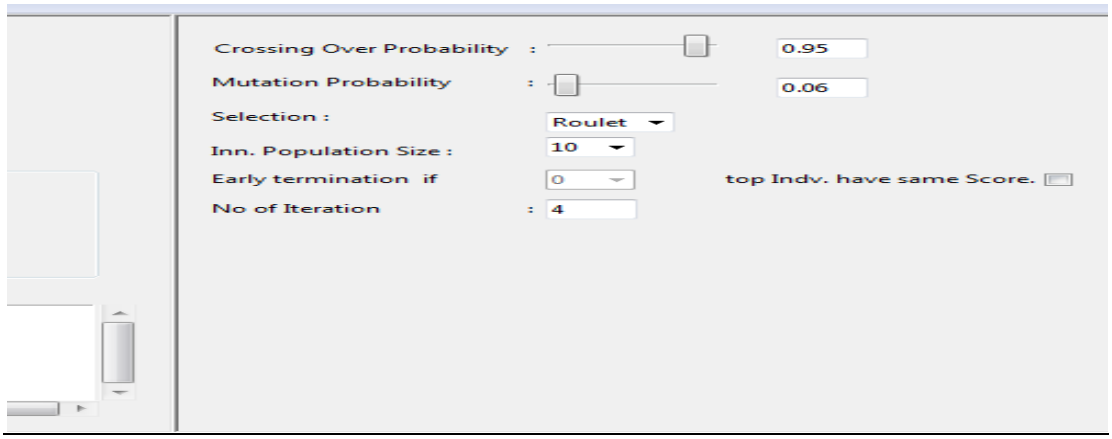
Remember to keep name of .ptt file and FASTA identifier in sequence file, exactly the same. And provide the sequence file earlier than .ptt file as the command line argument. The output file is written after '>'.

To load the input files click on the respective buttons and click on browse to load the files. Providing incorrect files causes anonymous error or result may be ambiguous.



2.2 Genetic Algorithm Parameters:

Clicking on GA parameters button opens the parameter panel:



2.2.a Operator Probability:

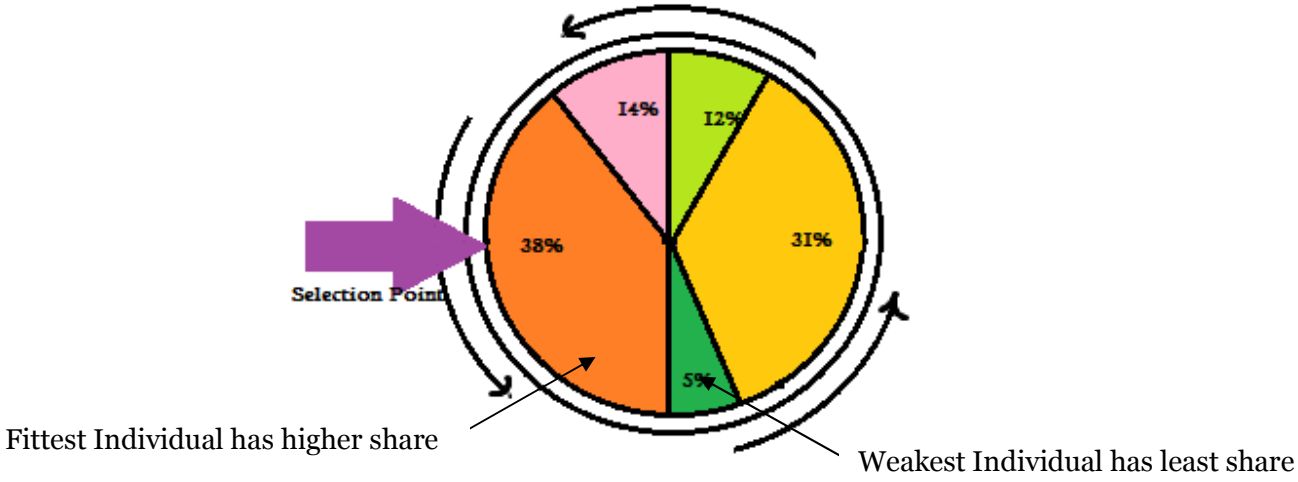
To implement genetic algorithm operators like Mutation and crossing over user need to set the probability. The probability indicates how often the operon has to be implemented. Generally a high cross over probability and low mutation probability combination gives optimized result. Use the sliders to adjust the probability.

2.2.b Selection:

A selection procedure selects an individual solution to be act as a parent for crossing over and generate offspring for next generation. There are two options for selecting the parents i. Roulette Wheel Selection ii. Best Individual selection.

i. Roulette Wheel Selection:

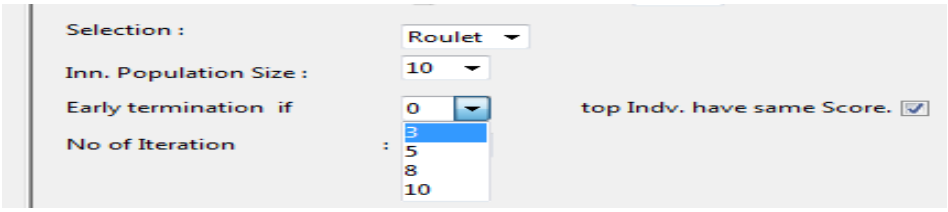
It selects an individual stochastically form the current generation by simulating rotation of a wheel with an objective to select the fittest individual. During the process individuals having higher fitness score has higher probability to get selected in comparison to less fit individuals.



ii. Best Individual: This method selects only the best individual from the generation. When user opts for this option, a higher mutation probability is advisable.

2.2.c Early Termination:

On attaining the best plausible solution, all the individuals will look much alike and mutation and crossing over does not make any change to the population. Hence continuing the process is worthless. Click on this the check box if user wants to terminate the evolution process when specified number of individuals in the current generation has same score.



Initial Population Number must be higher than the number of individuals checked for early termination.

2.2. No. of Iterations:

This option explicitly specifies how many generations are to be evolved to find the best possible solution. Set this option as per your convenience. Until and unless early termination is not defined the program will run until the specified generation.

2.3 Fitness Function:

Click on Fitness Function Button to change the fitness function. Selecting a fitness function gives the literature reference used for calculating the score.

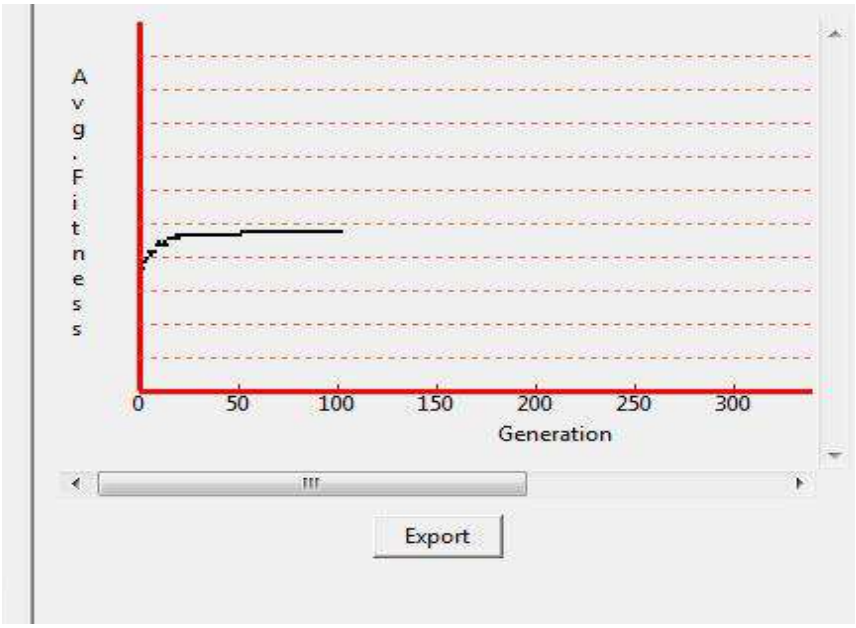
Fuzzy Fitness Finder (Jacob et.al) function takes a long run about 10-12 hrs for whole genome. Remember to set early termination option when FFF is used.

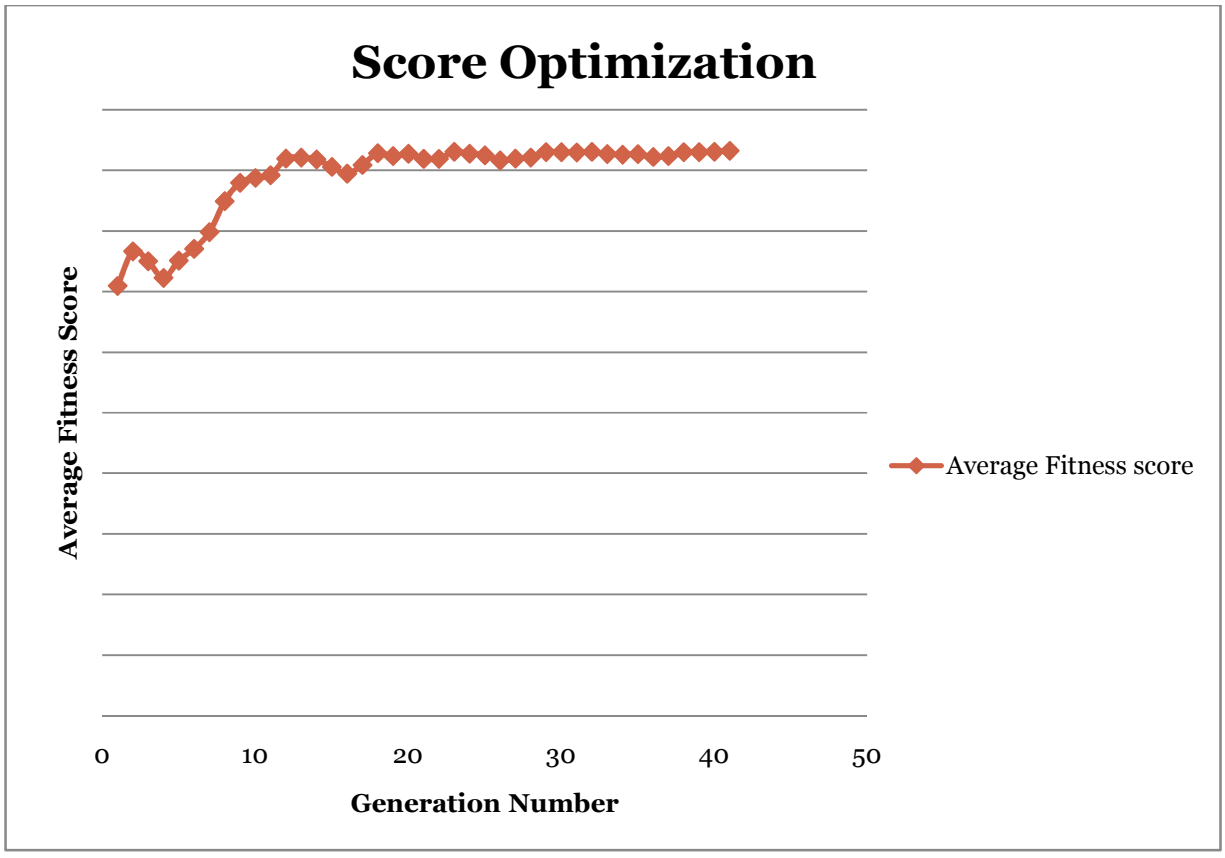
Rule based Fitness function is a heuristic one and can be used for quicker evaluation and doesn't guarantee better prediction.

2.4 Result Visualization:

Optimization process starts when start button is clicked. Like most standard GA software average fitness score in each generation plotted. This shows a uprising curve for successful optimization process. If the cure is not reliable (not uprising) user need to adjust the probabilities and run the program again.

Click on export button to save the plot in postscript format (.ps) to view it later in any post script viewer like **ghostviwer**. Otherwise the *progress.xls* file can be open after the run and select the two columns and plot using XY sctter.


Avg. Fitness
0
50
100
150
200
250
300
Generation
Export


Score Optimization
Average Fitness Score
0
10
20
30
40
50
Generation Number
Average Fitness score

Operon clusters along with their corresponding scores are displayed in the result panel when Result in Text Button is clicked. Result exported to hard disc.

A Graphical viewer has been designed to represent individual operon clusters along with the promoter and terminator signals. The list of operons is displayed on the top. Selecting an cluster displays its total score at the bottom of list. Double clicking on a particular entry loads the entire operon map with terminator and promoter signals. Map in postscript format can be exported.

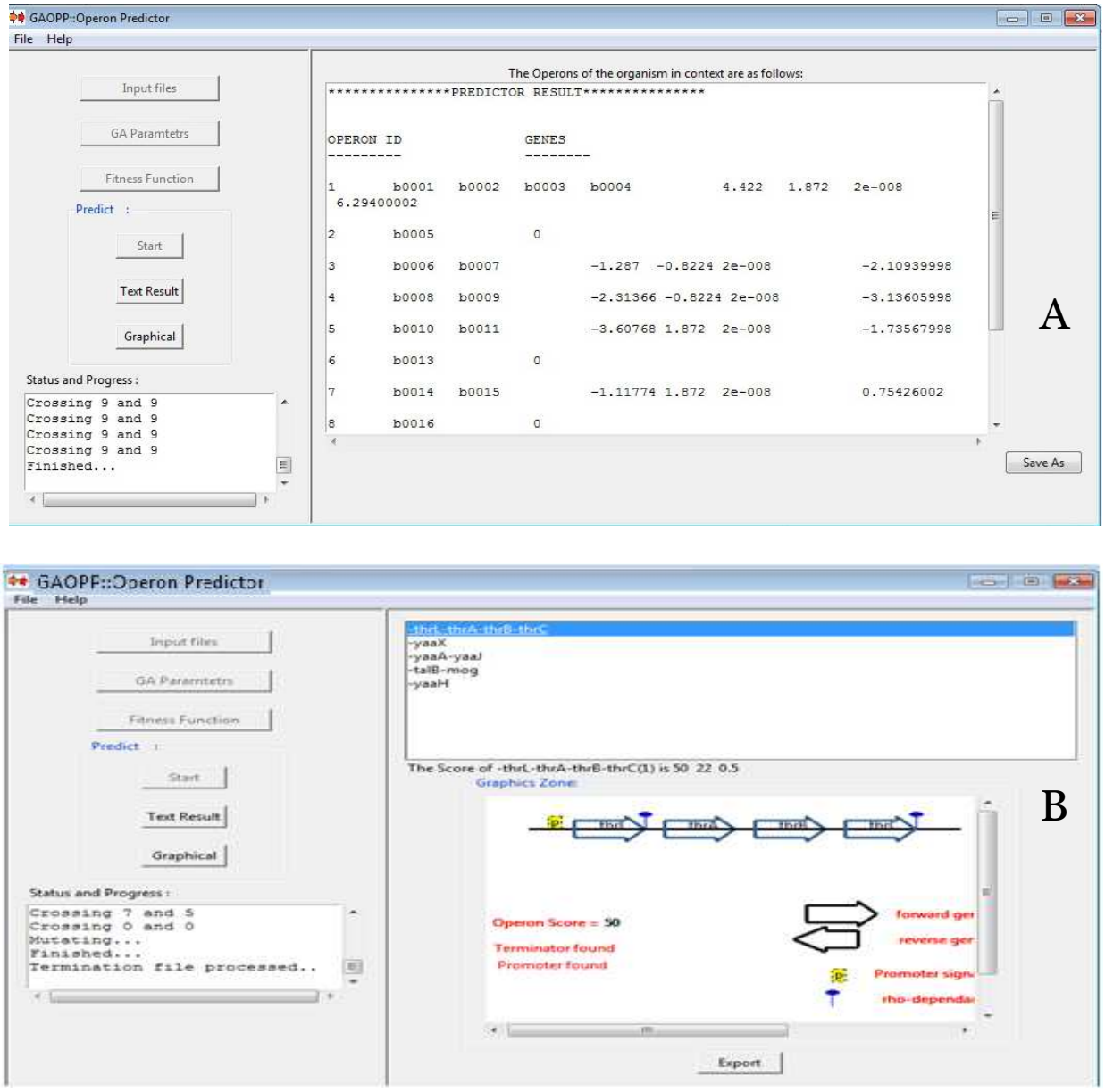
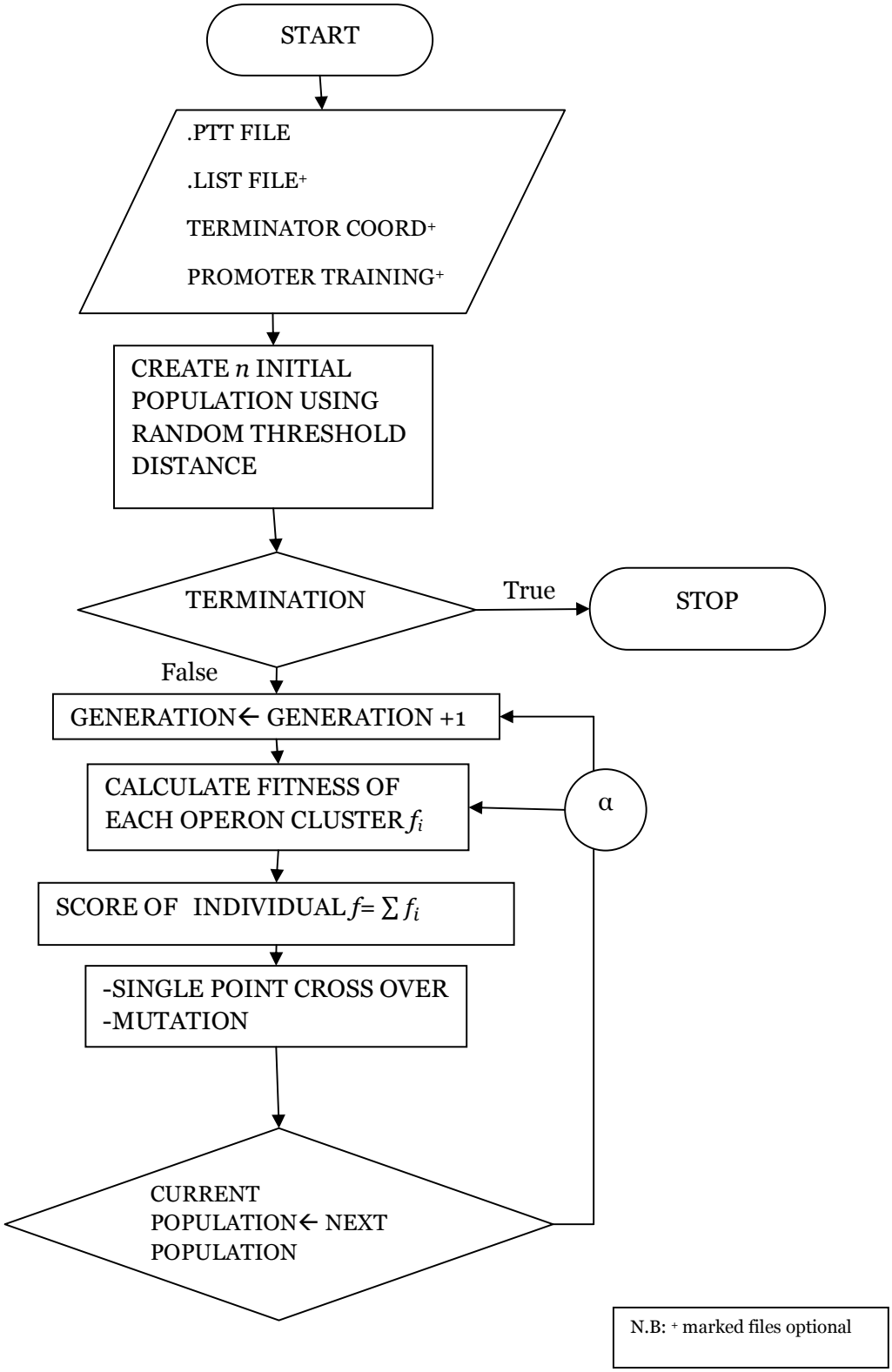
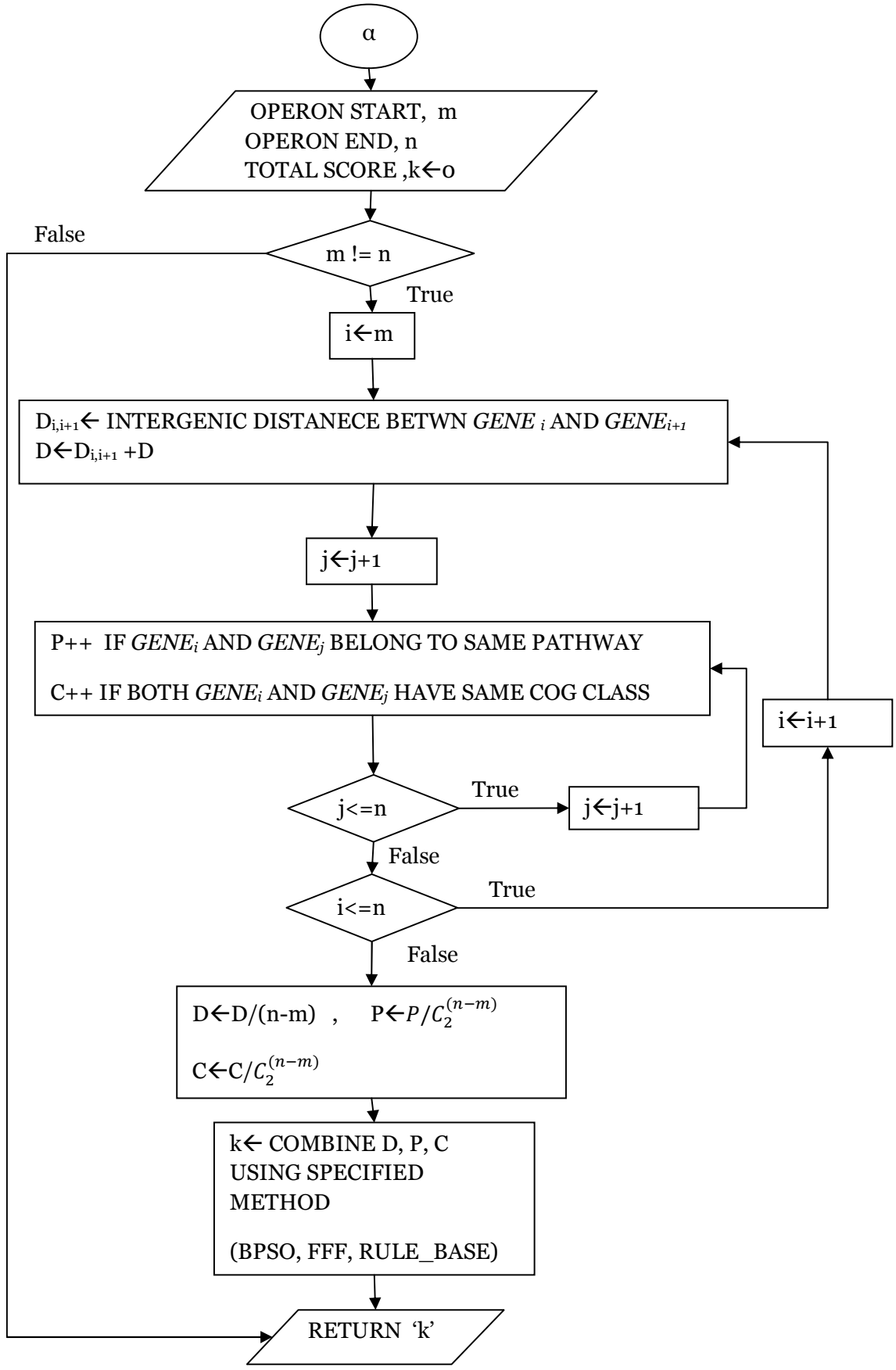


Fig: Output panel: Result in Text form (A) and Result in Graphical (B). Graphical Result Shows visualizes regulatory signals.

3. Algorithm:

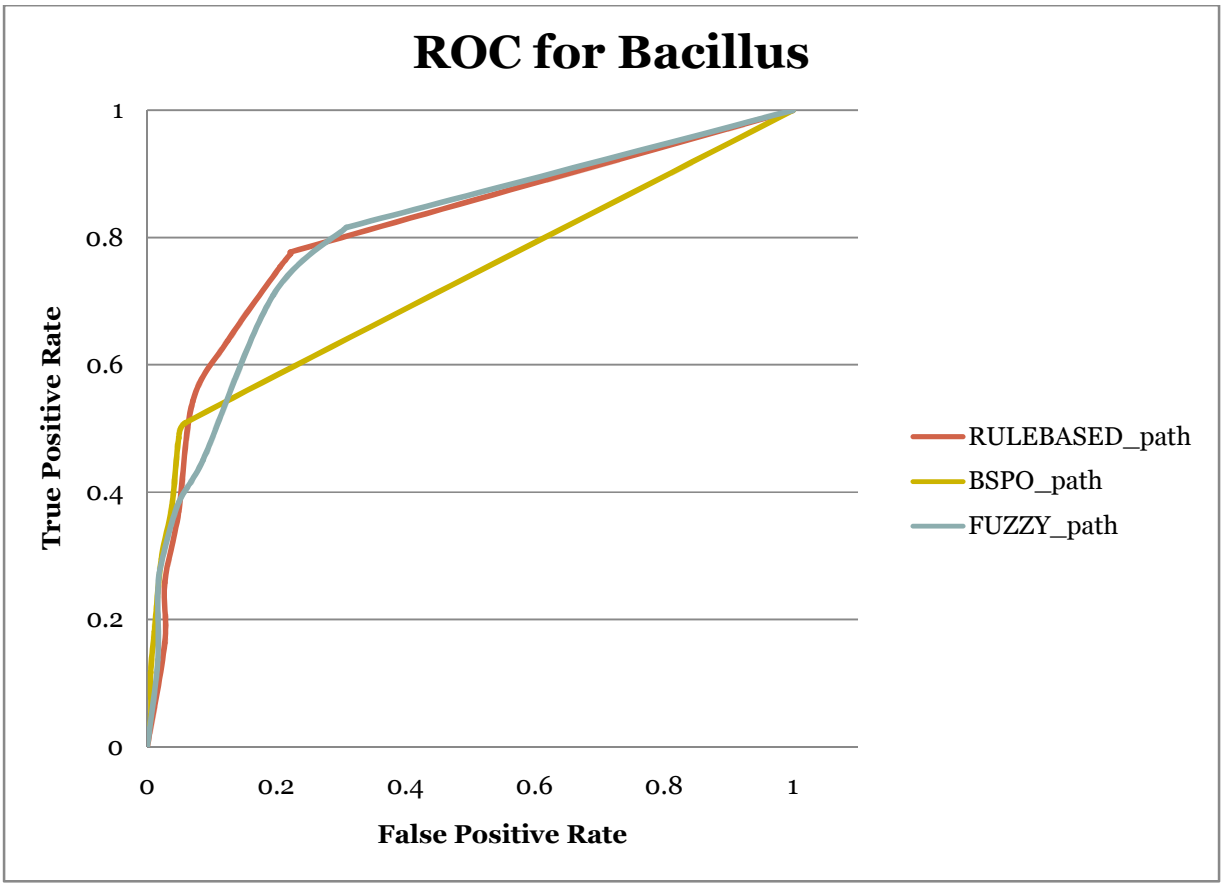

START
.PTT FILE
.LIST FILE+
TERMINATOR COORD+
PROMOTER TRAINING+
CREATE n INITIAL POPULATION USING RANDOM THRESHOLD DISTANCE
TERMINATION
True
STOP
False
GENERATION← GENERATION +1
CALCULATE FITNESS OF EACH OPERON CLUSTER f_i
α
SCORE OF INDIVIDUAL $f = \sum f_i$
-SINGLE POINT CROSS OVER
-MUTATION
CURRENT POPULATION← NEXT POPULATION
N.B: + marked files optional

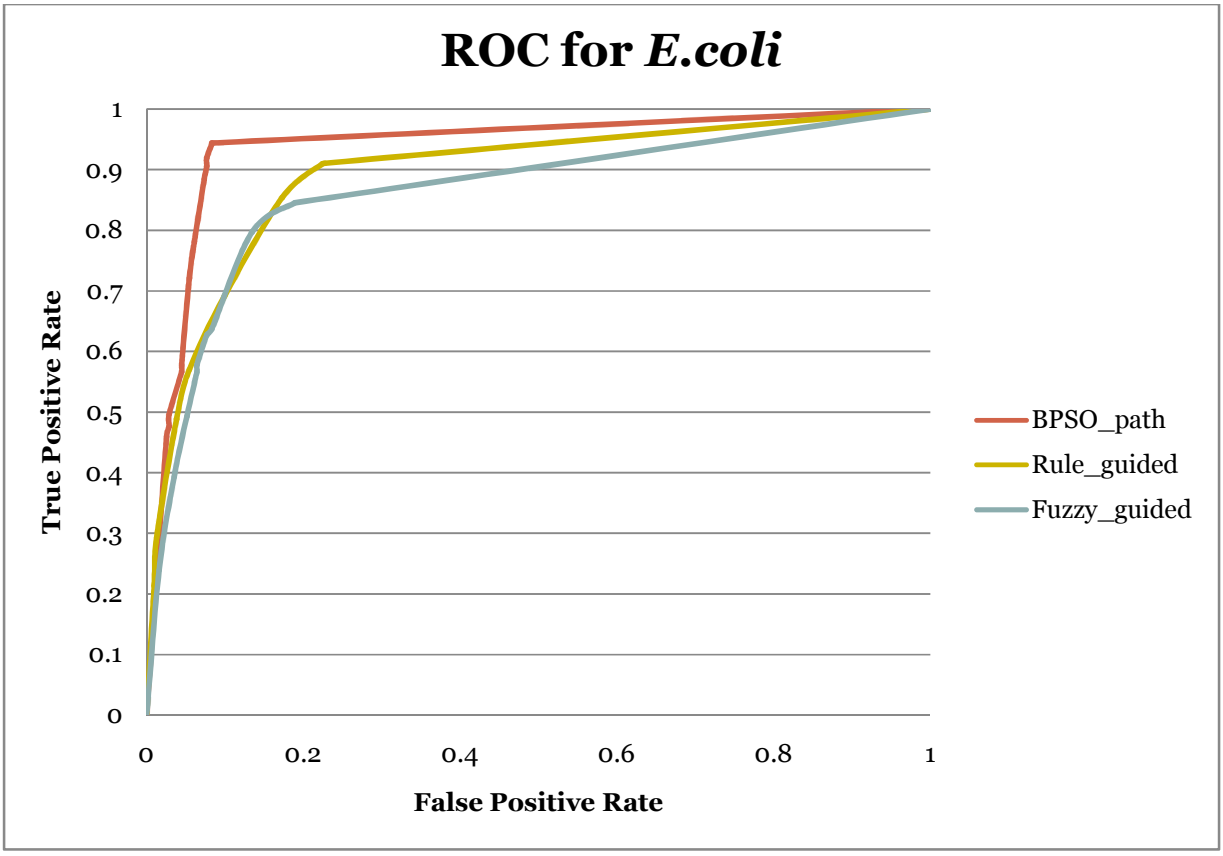


GAOPP: Genetic Algorithm for Operon Prediction in Prokaryotes

Evaluation:

We used GAOPP for available test sets like *Escherichia coli* K-12 substr-MG1655and *Bacillus subtilis*. We created positive and negative gene pairs from available experimental data. The predicted operons were compared with these available test set. From these observations we constructed Receiver operating curve.




ROC for E.coli
True Positive Rate
False Positive Rate
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
0.2
0.4
0.6
0.8
1
BPSO_path
Rule_guided
Fuzzy_guided

Reference: