MTGIpick

MTGIpick Enables Robust Identification of Genomic Islands from a Single Genome

---

Qi Dai*, Cong Wang, Chaohui Bao, Sheng Ma, Yabing Hai, Yunfei Wang, Xiaoqing Liu, Yuhua Yao, Wenwen Huo, Zhenyu Xuan, Min Chen* and Michael Q Zhang*. MTGIpick enables robust identification of genomic islands from a single genome.

To be submitted to Nucleic Acids Research.
Version 05 March 2015

Qi Dai
College of Life Sciences,
Zhejiang Sci-Tech University, Hangzhou 310018,
China

Contents

1. Overview

MTGIpick is a software that implements multiscale statistical algorithm to predict genomic islands from a single genome. It uses small-scale test with large-scale features to score small region deviating from the host and large-scale statistical test with small-scale features to identify multi-window segments for identification of genomic islands. MTGIpick can identify genomic islands from a single genome, without annotated information of genomes or prior knowledge from other datasets. In simulations with alien fragments from artificial and real genomes, MTGIpick reported robust results across different experiments. From real biological data, MTGIpick demonstrated better performance compared with existing methods, and identified genomic islands with more accurate size.

MTGIpick was written in Matlab and Java, compiled in Windows，Linux and Mac, and run on those platforms. We have supplied a version of MTGIpick. The output of MTGIpick consists of the genomic signatures, score of each region and predicted genomic islands.

Contact: daiailiu04@yahoo.com

2. Obtain and install

2.1. Requirements

MTGIpick has been compiled and tested under Sun Java interpreter and Matlab. MTGIpick can be used in Windows-, Linux- and Mac-based platforms. Java Virtual Machine and MATLAB Compiler Runtime (MCR) are required for MTGIpick setup on your platform. However, we strongly advise the use of openjdk (JDK) instead of the Oracle version of java virtual machine when working in linux-based or Mac-based machines as the Oracle version may result in some exceptions during the analyses.

Software	On window	On Linux(x86_64)	On Mac
Java Virtual Machine	jdk 1.7	jdk 1.7	jdk 1.7
MATLAB Compiler Runtime	mcr 8.4	mcr 8.4	mcr 8.4

2.2. Download

There are two ways to download the MTGIpick:

1) You can download the MTGIpick package with JDK 1.7 and MCR 8.4 from our web (http://bioinfo.zstu.edu.cn/MTGI)

A) Windows (MTGIpick.zip)

- MCRInstaller.exe	# MCR 8.4 for Windows
- MTGI_setup.exe	# main program
- Example.fasta	# A sequence in FASTA format
- README.txt	# Documentation.

B) Linux (MTGIpick.zip)

- jdk-7u79-linux-x64.tar.gz	# JDK 1.7 for Linux
- MCRInstaller.zip	# MCR 8.4 for Linux
- MTGI_linux.jar	# Main program
- install_jdk-mcr_linux.sh	# Install jdk 1.7 and mcr 8.4
- run_MTGI_linux.sh	# Run MTGIpick software
- Example.fasta	# A sequence in FASTA format
- README.txt	# Documentation.

C) Mac (MTGIpick.zip)

- jdk-7u25-macosx-x64.dmg	# JDK 1.7 for Mac
- MCRInstaller.zip	# MCR 8.4 for Mac
- MTGI_mac.jar	# Main program
- install_mcr_mac.sh	# Install mcr 8.4
- run_MTGI_mac.sh	# Run MTGIpick software
- Example.fasta	# A sequence in FASTA format
- README.txt	# Documentation.

2) If you download the MTGIpick package without JDK 1.7 or MCR 8.4 from our web (http://bioinfo.zstu.edu.cn/MTGI), download the JDK 1.7 and MCR 8.4 for your platform from the following Web:

JDK: http://www.oracle.com/technetwork/java/javase/downloads/java-archive-downloads-javase7-521261.html#jdk-7u25-oth-JPR

MCR: http://www.mathworks.com/products/compiler/mcr/?refresh=true

Make sure that the JDK 1.7 and MCR 8.4 are saved into the folder of the MTGIpick.

2.3. Install and Run

1) Windows (Tested on win7)

Before installing MTGIpick, make sure that the MCR 8.4 for windows is saved into the same folder of MTGIpick software. Install MCR 8.4 first, and run MTGIpick setup directly.

2) Linux (Tested on CentOS-7.0-1406-x86_64)

Before installing MTGIpick, make sure that the JDK 1.7 (jdk-7u79-linux-x64.tar.gz) and MCR 8.4 for Linux are saved into the same folder of MTGIpick software. Please follow the following steps for installing and running MTGIpick:

Step 1

To install jdk 1.7 (jdk-7u79-linux-x64.tar.gz) and mcr 8.4, it requires a simple command line as follow:

```
> bash install_jdk-mcr_linux.sh
```

Step 2

To run the MTGIpick software, just type a simple command line as follow (Once the first step has run, execute the second step to run MTGIpick):

```
>bash run_MTGI_linux.sh
```

3) Mac (Tested on OS X 10.10 Yosemite)

Before installing MTGIpick, make sure that the JDK 1.7 (jdk-7u25-macosx-x64.dmg) and MCR 8.4 for Mac are saved into the same folder of MTGIpick software. Please follow the following steps for running MTGIpick:

Step 1

Install jdk (jdk-7u25-macosx-x64.dmg) by simply clicking

Step 2

To install mcr 8.4, it requires a simple command line as follow:

```
> bash install_mcr_mac.sh
```

Step 3

To run MTGIpick software, just type a simple command line as follow (Once the first two steps have run, execute the third step to run MTGIpick):

```
> bash run_MTGI_mac.sh
```

3. Tutorial

3.1. Functions and features

1. IST-LFS: an iteration of small-scale t-test with large-scale feature selection to quantifying the compositional difference between a region and the ‘native’ genome regions.

2. ILST-DSFS：an iteration of large-scale statistical testing using dynamic signals from small-scale feature selection to identify some multi-window segments.

3. CG-MJSD: A boundary detection method based on CG-based segmentation and Markovian Jensen–Shannon divergence.

4. The methods and parameters could be chosen according to your purpose.

3.2. Input

Input file to *MTGIpick* should be DNA sequences, and it have to be in FASTA format. For example, the file name is sequence1.fasta, its content looks like this:

```
>cya_GI1_93960_99000_GI2_208020_223020_GI3_408480_410520
CCCCATTCCCCCCATTCCCTCCTTTTCCACCATACCCTCTTTTCCCCTCGTTGCCCCCAA
ATTTTTACGCATTTCCCCATTAATGCGATGATCCCAGCGCGAAAGCATCTGTGATTAAGA
CGTCTATCAATTTATACTCGTTAGGGTTTTTTCTTCGGTGGTACCATCTGGGCGCCTACG
CATCTGTCATATCTTCCTCTCGCCTACCAGAGCCATTAGGGAAATAGCTTCAGAGCTATG
CTTGTTTTGGTTTTGACAGAACCTTTCCGATGTCGAGATCGCCGAAATTGCCGGATTGAT
```

```
ATTGGTTTCGGGAAATCATCCGTCCCGCTGAACTGTTTCTTTATCTTCCGTCGCATTTAC
TTAATCCCGGCTGTATTGATCATATCTGACTTTCTATCTTGCGCCGTGTACCTGCCGTTA
GCACCCCCCGTAGCCTCTCTATCCCCGTTTCCGGTATACTCTTGCAGCAGAGCAGCTCTG
GCATACCACTAAGCGCGTATGTAGAGGCTTCGTTCATGCAACATTTGATCGGGGAGATTT
ATTTTCTTGGGAGACTCCAAGCCGTTCTTGCAGACCTTTTGCCTGCGGTAGCGTGAAGCG
>……
```

Note:

The input file should be DNA sequences with no tabs, spaces, and wraps in the sequence.

3.3. Methods

3.3.1 IST-LFS method

1) Framework

IST-LFS is a proposed small-scale t-test with large-scale feature selection that was used to quantify the compositional differences of a region from the host in the MTGIpick. It is efficient at detecting horizontal gene transfers or genomic islands with small sizes. The steps are described below:

a) Split a genome into n non-overlapping windows of size 1kb.

b) Calculate the frequencies of the tetranucleotides in each window as genomic signatures.

c) Extract the signatures of the host with the help of the confidence intervals on the windows' variances.

d) Calculate the kurtosis of each tetranucleotide across n windows and select the windows with a larger kurtosis as informative signatures.

e) Measure the divergence of the ith window from the host using the two-sample t-test.

f) Select windows whose scores are large enough to be considered to be statistically significant.

g) Delete the selected windows and update all of windows of the genome; then, repeat steps d-f until there is no window to be found.

h) Refine the boundaries of the predicted genomic islands using the CG-MJSD method.

2) Parameters of IST-LFS Software

Word size: the length of k-mer.

Windowed transform: the total number of the windows used in genomic transformation.

Iteration time: the periods of time that are repeated to select windows whose scores are large enough to be considered statistically significant.

Core feature size: the size of selected features by the proposed kurtosis.

Eye window size: the size of eye windows used in the proposed divergence measure based on two-sample t-test.

Time standard error: the standard deviation of the mean of the window scores to select windows associated with putative GIs.

Upstream/downstream of ‘raw’ genomic islands: the length of sequences around ‘raw’ genomic islands to refine the boundaries of predicted genomic islands.

3.3.2 MTGIpick method

1) Framework

MTGIpick is a novel method for the robust identification of GIs using the multiscale statistical testing. The steps are described below:

a) Split a genome into n non-overlapping windows of size 1kb.

b) Calculate the frequencies of the tetranucleotides in each window as genomic signatures.

c) At a smaller scale, we propose an iteration of small-scale t-tests with large-scale feature selection (IST-LFS) to quantify the compositional differences of a region from the host.

d) At a large scale, we investigate the variability of higher moments of each tetranucleotide and design an iteration of large-scale statistical testing using dynamic signals from small-scale feature selection (ILST-DSFS), to identify large, multi-window segments.

e) For each multi-window region detected by the ILST-DSFS, we split it into several distinct segments according the GC-content bias, from which we detect genomic islands with respect to their IST-LFS scores.

f) Refine the boundaries of the predicted genomic islands using the CG-MJSD method.

2) Parameters of MTGIpick Software

Word size: the length of k-mer.

Windowed transform: the total number of the windows used in genomic transformation.

Iteration time: the periods of time that are repeated to select windows whose scores are large enough to be considered statistically significant.

Core feature size: the size of selected features by the proposed kurtosis.

Eye window size: the size of eye windows used in the proposed divergence measure based on two-sample t-test.

Time standard error: the standard deviation of the mean of the window scores to select windows associated with putative GIs.

Neighbourhood size: the total number of the windows surrounding the ith window used in the calculation of the higher moments.

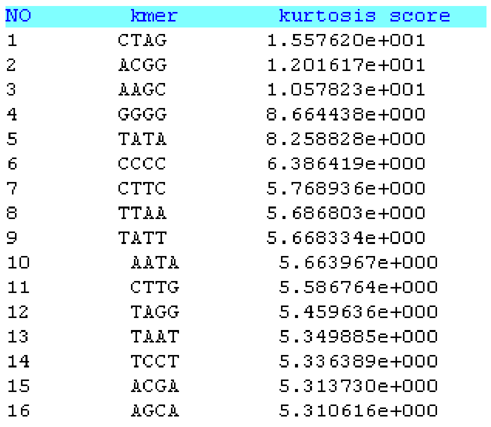
Long window size: chooses the number of the subsequent continued windows of the ith window.

Dynamic feature size: the size of selected features by the proposed kurtosis within the ith long sliding window.

Upstream/downstream of 'raw' genomic islands: the length of sequences around 'raw' genomic islands to refine the boundaries of predicted genomic islands.

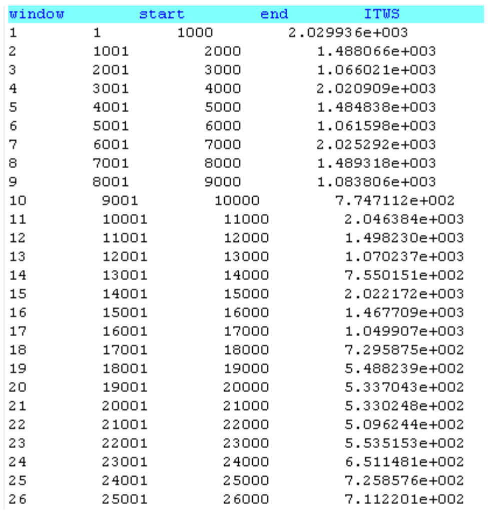
3.4. Outputs

The output of MTGIpick consists of genomic signatures, score of each region and predicted genomic islands. They are stored in the same directory where the input file is stored. The genomic signatures are displayed like this:



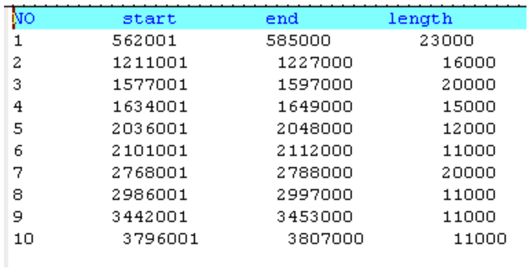
```
NO         kmer          kurtosis score
1         CTAG          1.557620e+001
2         ACGG          1.201617e+001
3         AAGC          1.057823e+001
4         GGGG          8.664438e+000
5         TATA          8.258828e+000
6         CCCC          6.386419e+000
7         CTTC          5.768936e+000
8         TTAA          5.686803e+000
9         TATT          5.668334e+000
10         AATA          5.663967e+000
11         CTTG          5.586764e+000
12         TAGG          5.459636e+000
13         TAAT          5.349885e+000
14         TCCT          5.336389e+000
15         ACGA          5.313730e+000
16         AGCA          5.310616e+000
```

The score of each region is displayed like this:



window	start	end	ITWS
1	1	1000	2.029936e+003
2	1001	2000	1.488066e+003
3	2001	3000	1.066021e+003
4	3001	4000	2.020909e+003
5	4001	5000	1.484838e+003
6	5001	6000	1.061598e+003
7	6001	7000	2.025292e+003
8	7001	8000	1.489318e+003
9	8001	9000	1.083806e+003
10	9001	10000	7.747112e+002
11	10001	11000	2.046384e+003
12	11001	12000	1.498230e+003
13	12001	13000	1.070237e+003
14	13001	14000	7.550151e+002
15	14001	15000	2.022172e+003
16	15001	16000	1.467709e+003
17	16001	17000	1.049907e+003
18	17001	18000	7.295875e+002
19	18001	19000	5.488239e+002
20	19001	20000	5.337043e+002
21	20001	21000	5.330248e+002
22	21001	22000	5.096244e+002
23	22001	23000	5.535153e+002
24	23001	24000	6.511481e+002
25	24001	25000	7.258576e+002
26	25001	26000	7.112201e+002

And the predicted genomic islands are



NO	start	end	length
1	562001	585000	23000
2	1211001	1227000	16000
3	1577001	1597000	20000
4	1634001	1649000	15000
5	2036001	2048000	12000
6	2101001	2112000	11000
7	2768001	2788000	20000
8	2986001	2997000	11000
9	3442001	3453000	11000
10	3796001	3807000	11000

4. References

1. Dhillon BK, Chiu TA, Laird MR, Langille MG, Brinkman FS: IslandViewer update: Improved genomic island discovery and visualization. Nucleic Acids Res 2013, 41: W129-132.
2. Aaron JA, Rajeev K, Azad AR, Jeffrey GL: Detection of genomic islands via segmental genome heterogeneity. Nucleic Acids Res 2009, 37:5255-5266.
3. Jaron KS, Moravec JC, Martinkova N: SigHunt: horizontal gene transfer finder optimized for eukaryotic genomes. Bioinformatics 2014, 30:1081-1086.
4. Langille MG, Hsiao WW, Brinkman FS: Detecting genomic islands using bioinformatics approaches. Nature Rev Microbiol 2010, 8:373-382.