CgiHunter

CATCGTGGACGAT

# User Manual

This documentation introduces the basic functions of the CgiHunter software tool for CpG island annotation of DNA sequences.

**Table of content**

## 1. Installation & Setup

CgiHunter has been developed in the programming language python version 2.4. To use the software python 2.4 has to be installed on your computer. A python interpreter can be obtained at http://www.python.org/download/ for Unix and Windows systems.

To install the CgiHunter software either use the windows installer or the setup.py script via '*python setup.py install*'.
You can start the CgiHunter Graphical User interface with the command '*python CgiHunterGUI.py*' or on windows machines by clicking on the CgiHunterGUI icon.

Optionally, on 32-bit systems the python package 'psyco' can be installed for a performance improvement of the computation. It is available from http://psyco.sourceforge.net/download.html .

## 2. Functionality overview

CgiHunter can compute bias-free CpG island annotations for arbitrary DNA sequences. The sequences should be provided in the FASTA format.
There are three different ways to use CgiHunter:

a) A Graphical User Interface (CgiHunterGUI.py)
b) By Command line call (CgiHunter.py)
c) By using XML files (CgiHunter.py)

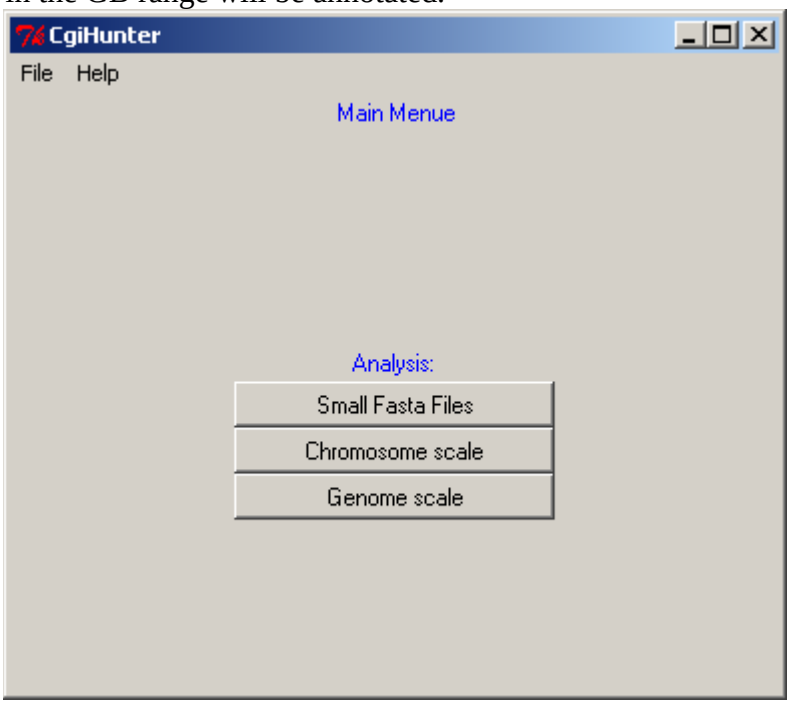
The GUI is mainly designed for casual users and for the preparations of XML files, while option b) and c) can be easily included into computational pipelines and other frameworks.

## 3. Graphical User Interface

The GUI can be started either by the command '*python CgiHunterGUI.py*' or by double clicking on the CgiHunterGUI icon on windows systems.

### 3.1 Main menu

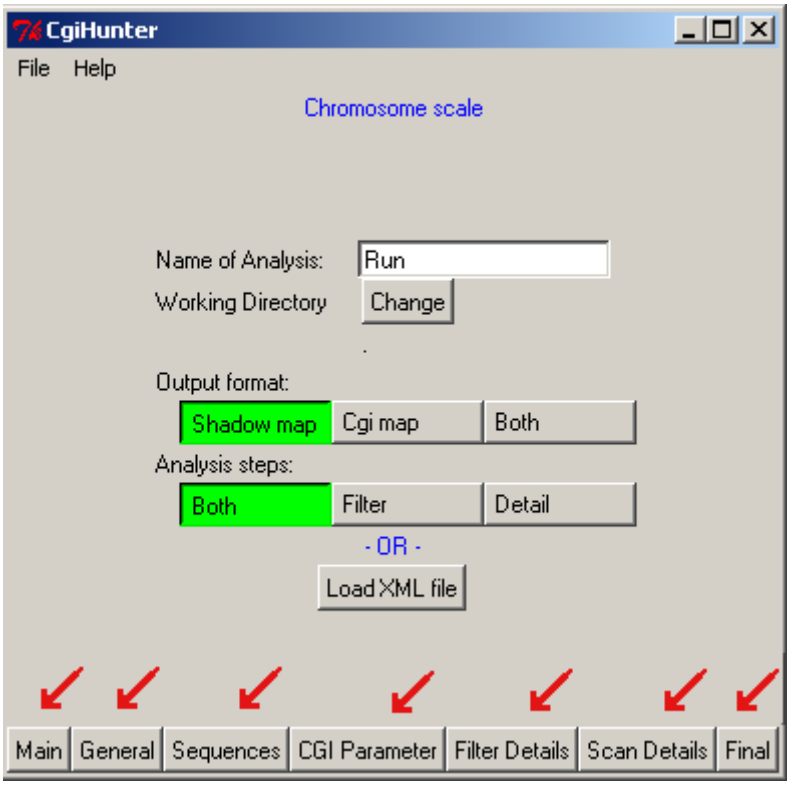
The main menu has three options: 'Small Fasta Files' is appropriate if a number of small sequences should be analyzed (10 KB), 'Chromosome scale' may be used if sequences in the MB range should be scanned and 'Genome scale' should be chosen whenever sequence data in the GB range will be annotated.




### 3.2 'Small Fasta files' and 'Chromosome scale'

Both analysis modes are divided into different steps and very similar. The only difference between these modes is that the 'Chromosome scale' contains two additional steps called 'Filter Details' and 'Scan Details'.

You can navigate between these steps by using the buttons in the navigation panel at the bottom of the window (indicated in the picture by the red arrows). You can always jump back and forth between the different steps and even into the main menu without loosing your settings. At any point you can save the recent configuration by clicking on the "File" pull-down menu in the upper left corner and choosing the "Save conf" option.




All steps will be explained in detail in the following subsections.

### 3.2.1 General

Here you can load an xml file specifying the options of your analysis and afterwards updating the configurations individually for all steps. If no xml file exists, you may define the name of the analysis and the directory in which all temporary data and the results should be stored. This folder should already exist and several subfolders will be generated automatically.
Further, you can choose the output format of the analysis. CgiHunter can either annotate all base pairs that are contained in any CpG island (Shadow map), or, if different possible CpG island overlap with each other, select the optimal (by default the largest) island (Cgi map).
Note that the Shadow map is computed faster, but the annotated regions will be a fusion of smaller CpG islands and therefore may not directly fulfill the criterions you have selected (see publication for details).
Finally, it can be chosen if either the filter step or the annotation step or both steps should be performed.

### 3.2.2 Sequences

Here the sequences that should be scanned can be specified. Either add each sequence individually or add a whole folder of sequence data at once. Each sequence will be identified by its genome assembly and its chromosome name / filename.

### 3.2.3 CGI Parameter

In this window you can select the minimal criteria a CpG island has to fulfill.
All parameters are expected as integers, with the first two values in percent (50-100) and the minimal length in nucleotides per window.

### 3.2.4 Filter Details

For large sequences, the configuration of the filter step can have a major influence on the runtime of CgiHunter. In this window the number of filter steps and their resolution can be selected. For more details see the publication.

### 3.2.5 Scan Detail

Here the optimality criterions for the ‘Cgi map’ output format can be selected. The score for each candidate island is computed by multiplying the values of its characterizing parameters (length, GC content, ratio of observed over expected CpG frequency and repeat content) with the displayed weights and adding up the results.

$$score = \sum_{i=1}^{4} weight_i \cdot value_i$$

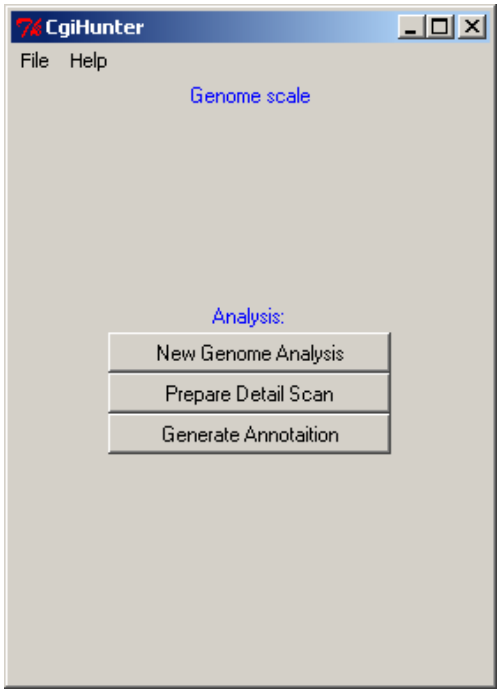
By default only the length of CpG islands is considered. By adjusting the weights in a range between -1 and 1 more emphasis can be given to other aspects of the candidate islands. For example, if only the weight for GC content is set to one, while the others are switched to zero, always the candidate island with the highest GC content will be selected in a detail scan if multiple candidate islands overlap.

### 3.2.6 Final

After specifying all parameters you can save your analysis as xml file by clicking on ‘Save as XML file’. You can now decide if you want to execute your analysis directly from the GUI by clicking on ‘Start’ or from the command line by using the option –x to send your xml file directly to CgiHunter.py (e.g. *python CgiHunter.py –x <name of xml file>*).
If you use the GUI variant, a status window will open. By clicking on the ‘Refresh’ button you can keep track of the analysis (during the filter step the GUI reacts with great delays, if you scan large files). Do not close the main window, as this will terminate the analysis. It will be announced in green letters, when the refresh button is clicked after the analysis is finished.

## 3.3 Genome Scale

This type of analysis has three working steps: the definition of a ‘New Genome Analysis’, the ‘Prepare Detail Scan’ step, and finally the ‘Generate Annotation’ step.




This workflow enables the CgiHunter software to separate large analysis tasks in well separated subtasks. These can then be processed independently from each other i.e. in parallel on separate computers or on computer clusters. The results are then reassembled again in the last step.

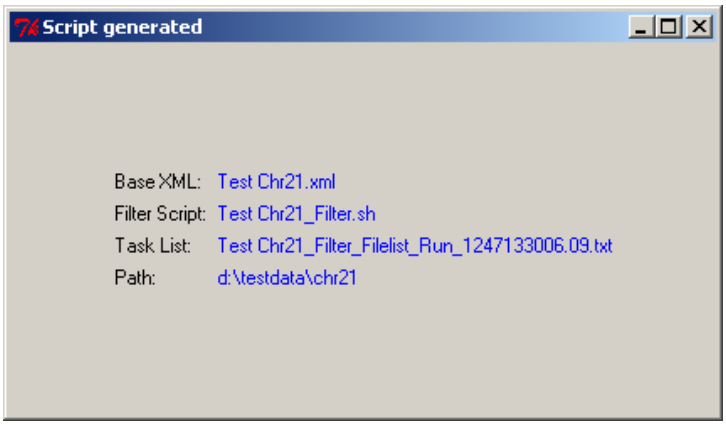
This has the advantage that on the one hand the runtime can be accelerated by using more CPUs and on the other hand a crash of the program (e.g. caused by hardware problems) does not affect tasks that are already finished.

### 3.3.1 New Genome Analysis

The definition of the whole analysis is similar to the ‘Chromosome scale’ tasks described in 3.3. The only difference is that in the final step the analysis cannot be started directly. Instead, a script is generated, that guides the parallel execution of the analysis. Depending on your infrastructure, you can choose one of three script types:

1. Array script
2. Batch script
3. Separated scripts

If your computer cluster supports array scripts, it is the best choice to solve this task. Simply generate the script and submit it to the scheduling program of your cluster. Alternatively, the batch script can be used. It simply contains a list of direct CgiHunter calls for each subtask. If you manage you resources manually, you should choose the third option. By specifying the number of available computer nodes, you receive one script for each node that you then can start separately on each machine.
In every case a dialog opens that asks for the path of the CgiHunter implementation that should be used for the analysis. This file must be in a path that is visible for the computers that will afterwards perform the computation.
After specifying this path, a number of xml files are generated that guide the Filter Scan of the given genome and placed in the subfolder ‘FilterTasks’. Furthermore, a window opens that communicates the names and the location of the three coordinating files for the parallel processing.

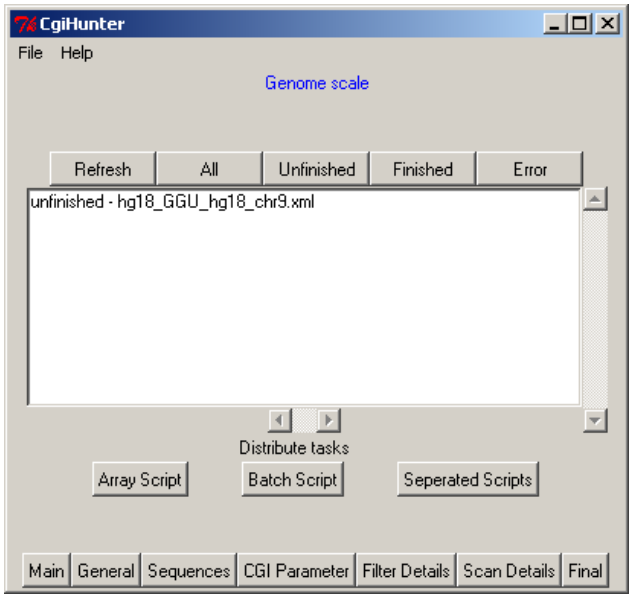



The 'Base XML' is a description of the overall genome scan. The 'Filter script' is the proposal for the script governing the computation process and may have to be adapted to your computational environment. The 'Task List', finally, contains the file-paths of all subtasks.

If none of the offered script variants suite the needs of your computer system, you have to design your own workflow to perform all these subtasks. In the simples case this is a script that reads line by line each file-path from the Task List and uses them as argument for CgiHunter.py. You can use the –X option of *CgiHunter.py* for this purpose. Furthermore, to distribute the task list on different machines, you can use in addition the -e and –o options. For example *python CgiHunter.py –X <tasklist> -e 3 –o 10* will execute every third out of ten files from <tasklist> on the given computer. Thus, it is fairly simple to distribute the remaining tasks on nine additional machines by changing the –e parameter (this approach is used in the 'Separated scripts').

For more advice on how to design the script you may read "CgiHunter_WholeGenomeAnnotation.pdf". After all subtasks are finished, you can proceed with the 'Prepare Detail Scan' step.

### 3.3.2 Prepare Detail Scan



You should only proceed with this step, if you have finished the computation of all subtasks in the previous step. By pressing the button ‘Prepare Detail Scan’ a dialog will ask for the ‘Base XML’ generated in the previous step. You then have the opportunity to specify the ‘Scan Details’. In the ‘Final’ screen the status of all subtasks is displayed. By pressing the Buttons on the top panel, you can display only those subtasks with a special status. Eventually, you have to eliminate error sources, like insufficient disk space, and restart some of the tasks.
When all tasks are finished (you can use the refresh button to update the task states), the tasks for the Detail Scan can be distributed.
You can then define the degree of distribution for the parallel processing. The smaller the ‘number of clusters per task’ is the more independent subtasks will be generated. Subsequently, you will be again asked for the location of the CgiHunter.py file and a window similar to that in step 3.3.1 is opened.

### 3.3.3 Generate Annotations

In this step all results from the ‘Detail Scan’ can be combined into a joint annotation. At first you will be asked to specify the ‘Base XML’ of the ‘Prepare Detail Scan’. You can recognize it by the ‘_Detail’ infix added to the original filename.
All subtasks will be then displayed in a list. You can check which files are already finished and which are still running by changing between the views ‘All’, ‘Unfinished’, ‘Finished’ and ‘Error’. Use the refresh button to update the task states. Upon completion of all subtasks you can press the ‘Generate annotations’ button. This will rebuild the annotation for each FASTA file that has been included in the analysis. The resulting BED files will be placed in the ‘Result’ subdirectory of your working directory. Additionally all annotations will be joined into a common file.

### 3.3.4 Optimization

The Detail Scan of CgiHunter generates a considerable amount of temporary data, depending on the CpG density and length of the analyzed DNA sequence. Therefore, make sure to provide enough hard disk space in case of heavy parallelization. A single Cgi map detail scan can in rare cases produce in the peak temporary data in the GByte range.
If possible, you can distribute the computation among separated file systems / hard drives to reduce possible interference when more than one process is in such a peak.

## 4. Command line

Direct calls to CgiHunter.py are possible, but it is strongly encouraged to only use them to execute predefined XML files (e.g. *python CgiHunter.py –x <name of xml file>*). If direct configuration of CgiHunter via command line is demanded, use the –h or –help option to find an overview of all command line options.

| Option | Type | Description |
|---|---|---|
| -x or --xml | string | Relative or absolute path the xml files that specifies the annotation**. Most other options will be ignored**. |
| -X or --tasklist | string | Relative or absolute path of a task list. That is a file that contains in each line a filepath that is ready to be processed by –x. All xml files in the task list will by executed subsequently. |
| -e or --every | int | Works only in combination with -X. If set e.g. to 5 while –o is set to 7 only every fifth task out of seven will be executed. |
| -o or --out_of | int | Works only in combination with -X. If set e.g. to 7 while –e is set to 1 only ever first task out of seven will be executed |
| - a or --assembly | string | The genome assembly/species the DNA sequence belongs to |
| -c or --chrom | string | The chromosome the DNA sequence belongs to |
| -d | int | Debug level – the higher the number the more status messages will be shown |
| -f or --comb | string | Relative or absolute path to a clustermap. The clustermap is the output of the algorithm's filter step. |
| -F or --force | flag | The execution of an xml file will be forced, even if it is marked as finished |
| -g or --getxml | string | If configuration is performed by command line this option can be used to retrieve the xml file that contains all options. |
| --gc | float | Minimal content of cytosine and guanine a region have to posses to qualify as a CpG island\n Only values between 50 and 100 are valid. Example: --gc 50 |
| --merge | flag | The file given with –x will be interpreted as 'Base XML' of a parallel computation. All available results from the subtasks will be merged. **Make sure that all computations have terminated correctly, before you merge results.** |
| -p or --path | string | Working of the annotation. The temp data, logs and results will be stored in subfolders in this directory |
| -s or --source | string | Absolute or relative path of the fasta file that contains the DNA sequence |
| -r or –ratio | float | Minimal ratio of observed CpGs over expected CpGs a region have to posses to qualify as a CpG island. Only values between 50 and 100 are valid. Example: -r 60 |
| -w or --min_win | int | Minimal length threshold of CpG island |
| -W or --max_win | int | Optional upper bound to CpG island length. Per default set to a million base-pairs |

## 5. XML files

Each CgiHunter analyses can be completely specified as XML document. These documents can be either generated by using the CgiHunterGUI, by modifying the template document included in the CgiHunter distribution or by your own custom techniques.
Each XML file has a status option that is initially set to ‘unfished’. Upon completion of a CgiHunter call this mark will be set to ‘finished’. CgiHunter will automatically ignore tasks that are marked as ‘finished’ to ensure that already processed files are not reanalyzed. To reset the status you can either load the XML file in the CgiHunterGUI and save it again under the same name or directly modify the status in the file.
If the computation of a file is interrupted, the status will contain the respective error message.
XML files can be executed directly from the command line (e.g. *python CgiHunter.py –x <name of xml file>*) or loaded in the GUI and then processed by clicking on ‘Start’ in the ‘Final’ screen.

## 6. CgiHunter End User License Agreement

This is a legal agreement ("Agreement") between you ("Licensee", either an individual or a single entity) and

Max Planck Institut fuer Informatik
Stuhlsatzenhausweg 85
66123 Saarbruecken
Germany,

("MPII" for short) about using the CgiHunter program ("Software") that provides a tool for computational analysis of genomes.

Please read the following Agreement carefully.

By using the software or downloading the source code you indicate that you have read and accepted the provisions of the Agreement and that you agree to be bound by all terms and conditions set forth herein. If you do not agree to any of the terms of this Agreement, do not use the Software and destroy any parts or results from the Software in your possession immediately.

1) Copyright

CgiHunter, including but not limited to the program code, sample programs, any associated files and documentations (the "Software"), is owned by MPII and is protected by copyright laws.

2) License

A license for using the CgiHunter web service or source code is provided free of charge to researchers working at academic, non-profit organizations on non-commercial projects.
Any commercial use of the software is strictly forbidden (please contact MPII for a free, time-limited, commercial test license).

3) Limited Warranty and Liability

Access to the software is provided on an 'as is' basis, and there are no warranties or conditions with respect to its fitness for purpose, its operational state, character, quality, or freedom from defects, or the non-infringement of rights of third parties.

The Licensee acknowledges that Software furnished hereunder is under test. The Licensee is solely responsible for determining the suitability of the Software and accepts full responsibility and risks associated with the use of the Software. In no event will MPII be liable for any damages, including but not limited to any loss of revenue, profit, or data, however caused, directly or indirectly, by the Software or by this Agreement.

4) Maintenance and Support

MPII is not obliged to provide maintenance or support to you. Nor do we guarantee availability of the webserver.

5) Distribution

No distribution is to be made of the Software by you.

6) Privacy

MPII is committed to respecting the privacy and data security of the users of the software. In general, any uploaded data or conducted analyses are exclusively available under the same account data from where they were generated. MPI staff will access and / or view your data only when this is necessary for debugging purposes.
However, due to the software being in test state it may happen that private data becomes erroneously accessible to third persons. MPII cannot take responsibility for any damages caused by such an event.

7) Reproduction of Information

All information is generated for personal and academic, non-commercial use only and, in this context, may be reproduced, in part or in whole and by any means, without charge or further permission from MPII. However, we stipulate that the software has to be cited appropriately.

8) Termination

If the Licensee fails to comply with any term of this Agreement, this Agreement is terminated and the Licensee has no further right to use the Software. On termination, the Licensee shall have no claim on or arising from the Software. The Software and any results generated by it shall be destroyed.

9) Applicable Law and Court of Jurisdiction

This Agreement is made and shall be construed in accordance with the laws of Germany. Court of Jurisdiction is Saarbruecken, Germany.

10) Construction Clause

If for any reason a court of competent jurisdiction finds any provision of this Agreement, or portion thereof, to be unenforceable, that provision of the Agreement will be enforced to the maximum extent permissible so as to affect the intent of the parties, and the remainder of this Agreement will continue in full force and effect.

11) Entire Agreement

This Agreement constitutes the entire agreement between the Licensee and MPII.
It replaces all other representations.
All modifications or extensions of this Agreement need to be put down in writing.