GOtcha users guide

David Martin

July 28, 2006

1 Introduction

GOtcha is a tool for predicting Gene Ontology[1] (GO) term associations with gene products. GOtcha compares the query sequence to sequences in well-annotated genomes and transfers the annotation using a novel, tree-based algorithm.

The main features of GOtcha are:

1. Term-specific predictions. Each term has an associated accuracy estimate.
2. Calibrated, percentage-based scoring system that gives meaningful numbers for non-mathematicians.
3. Optional graphical output that allows a rapid visualisation of the relevant portions of the GO hierarchy

GOtcha encapsulates the external database cross references from GO and can output appropriate summary tables for further analysis.

1.1 Citation

If you publish work that makes use of GOtcha you should cite:

GOtcha: a new method for prediction of protein function assessed by the annotation of seven genomes David M.A.Martin , Matthew Berriman and Geoffrey J. Barton *BMC Bioinformatics* (2004) **5**,178

[1] http://www.geneontology.org

2 Installation

GOtcha is an application written in Perl. It is designed exclusively for command line use in batch mode so don't expect a fancy interactive GUI. A series of web scripts are available in PHP but these will require tailoring to your particular situation.

2.1 Dependencies

GOtcha requires the following Perl Modules. Many are part of the standard Perl distribution.

Module	Notes
BioPerl	Tested with 0.7 and 1.4. (Newer versions should work but 1.2 doesn't) www.bioperl.org
Archive::Tar	There are several versions of this module.
GD	Required for the graphics drawing. A newer version that supports PNG is required.

GOtcha also requires the following software to be available in the PATH If these applications cannot

Program	Package
blastall	from the NCBI BLAST2 suite (www.ncbi.nih.gov)
seqret	from the EMBOSS suite (www.emboss.org)
infoseq	from the EMBOSS suite (www.emboss.org)
dot	from the Graphviz suite (www.research.att.com/sw/tools/graphviz/)
convert	from the ImageMagick suite (www.imagemagick.org)

be found in the PATH, or if you wish to use a specific version, there are command line options to specify the location of the executables.

2.2 License

GOtcha is currently made availble under an academic collaboration license. You may use and modify GOtcha for academic purposes but may not distribute it without permission from the author.

2.3 Environment

It is much easier to run GOtcha if the environment variable GOTCHA_LIB is set to the root directory for the GOtcha installation. This will make command lines a lot shorter as many options no longer need to be set explicitly. A line such as

```
export GOTCHA_LIB=/path/to/GOtcha
```

should be included in the .bashrc (for local install) or /etc/profile (for global install).

or for csh users

```
setenv GOTCHA_LIB /path/to/GOtcha
```

in the .cshrc file (local) or /etc/cshrc (global)

2.4 Installation directory

The install directory /path/to/GOtcha is structured as:

/lib	Perl modules
/data	GOtcha format tables for GO and score calculation
/db	Blast databases
/bin	Perl scripts.
/docs	Documentation
/examples	Test data and examples

2.4.1 Installation steps

1. Create a directory in which to install GOtcha.
2. Set the environment variable GOTCHA_LIB to this directory.
3. Move to this directory and unpack the distribution there. This should create the directory structure above.
4. Check the list of Perl dependencies and ensure all the necessary modules are installed.
5. Determine where the local Perl interpreter is. You may need to edit the first line in each of the scripts in the /bin directory to reflect the location of Perl.
6. Determine where any site-specific installations of Perl modules are and amend the 'use lib' lines appropriately.

7. Add the $GOTCHA_LIB/bin directory to your path.
   `export PATH=$PATH:$GOTCHA_LIB/bin`
8. Ensure the GOtcha scripts are executable
   `chmod +x $GOTCHA_LIB/bin/*.pl`
9. Edit the rungotcha.pl script to include the line
   `use lib /path/to/GOtcha/lib`
   where /path/to/GOtcha is where you have installed GOtcha.
10. Try executing GOtcha as
    `rungotcha.pl --help`

If all the above went correctly then you should get the GOtcha help screen, a long list of command line options. If not then you will have to do some debugging.

3 Running GOtcha

GOtcha needs to know the following:

- Which databases you wish to use for annotation?
- Which evidence codes you wish to include/exclude (deafult: all included)?

It also needs a sequence with which to work (STDIN by default) and will create a directory for the results.

If GOTCHA_LIB is set then GOtcha can find the databases and data files in the GOtcha installation. All these locations can be overwritten on the command line.

GOtcha will automatically attempt to read the sequence format and type. Sequence type can be overwritten (Nucleic acid, Protein or Auto (default)) on the command line.

As GOtcha is overrun by command line options, the best way to see how to run it is to look at some examples and then read the reference section to get the precise fine tuning you want.

3.1 Examples

3.1.1 Basic GOtcha runs

In this example we read in a sequence, use a standard set of databases to annotate, and write the results to a directory. This uses all the default settings, uses all evidence codes and performs no cutoff filtering.

```
rungotcha.pl --infile myseq.fasta --outfile results --searchdb go_ath,go_fb,go_human,go_mal,go_s
```

This will create a directory `results` with a number of files in, listed in the table below.

Table 1: File list for an example GOtcha run

dot.C.imap	dot-generated image map for Cellular Component
dot.F.imap	dot-generated image map for Molecular Function
dot.P.imap	dot-generated image map for Biological Process
go_ath.blast	BLAST search results for database go_ath
go_ath.gotcha	GOtcha results for just go_ath [2]
go_fb.blast	BLAST search results for database go_fb
go_fb.gotcha	GOtcha results for just go_fb
go_gvc.blast	BLAST search results for database go_gvc
go_human.blast	BLAST search results for database go_human
go_human.gotcha	GOtcha results for just go_human
go_mal.blast	BLAST search results for database go_mal
go_sgd.blast	BLAST search results for database go_sgd
go_sgd.gotcha	GOtcha results for just go_sgd
go_wb.blast	BLAST search results for database go_wb
go_wb.gotcha	GOtcha results for just go_wb
gotcha.C.dot	Dot input file for Cellular Component
gotcha.F.dot	Dot input file for Molecular Function
gotcha.P.dot	Dot input file for Biological Process
gotcha.imap	HTML fragment containing the GOtcha results
gotcha.js	Javascript file needed for web page functionality
gotcha.res	plain text GOtcha results.
gotcha.sql	SQL formatted GOtcha results for database entry
gotcha_C.png	Dot generated image for Cellular Component
gotcha_F.png	Dot generated image for Molecular Function
gotcha_P.png	Dot generated image for Biological Process
gotcha_web_C.png	rescaled image for web
gotcha_web_F.png	rescaled image for web
gotcha_web_P.png	rescaled image for web
gotcha_zoom_C.png	rescaled image for pan/zoom box
gotcha_zoom_F.png	rescaled image for pan/zoom box
gotcha_zoom_P.png	rescaled image for pan/zoom box
job.status	job status file for tracking errors
query.seq	input search sequence in FASTA format

This is a heady collection of files and it will grow as more complex options are included.

3.1.2 Controlling annotation parameters

The annotation is derived from the annotation databases specified with `--searchdb`. The P-scores have been calibrated against the complete database set so should not be regarded as reliable if you change the annotation databases.

It is often desirable to remove automated annotations from the source databases. GOtcha allows the specification of evidence codes in two ways:

- `--includecode`
  Only codes to be included are listed
- `--excludecode`
  All codes listed are excluded

Typically we will want to exclude automated annotations from the analysis. If we do not get any useful hints from GOtcha as to the function we are looking for then we can easily include them again.

```
rungotcha.pl --infile myseq.fasta --outfile results_noiea --searchdb
go_ath,go_fb,go_human,go_mal,go_sgd,go_wb,go_gvc --excludecode IEA
```

This will create a new directory, `results_noiea`, and place the results in there. Attempting to use the same directory will result in GOtcha exiting with a warning[3].

3.1.3 Multiple Analyses

The most time consuming part of the analysis is running the initial database searches. These results can be reprocessed with minimal time penalty by using the `--reprocess <name>` option. This will look for a set of results in the directory named by outfile and only run BLAST if necessary. The results files will have the `<name>` appended to the filename[4], allowing individual parameter sets to be readily identified.

When `--reprocess` is used, GOtcha will overwrite existing results if the filenames match without warning.

3.1.4 HTML output

GOtcha can produce HTML output that can be incorporated into a web page. The HTML output is produced when the flag `--nopng` is not used and is written to the file `gotcha.imap`*[.ext]* where

[3] Unless the `--reprocess` option is used

[4] Except in certain cases where the `<name>` is included before the file extension

ext is the argument given to `--reprocess`. The HTML in this file is a fragment that sources a javascript file (`gotcha`*[.ext]*`.js`) that is essential for the dynamic part of the HTML as well as a number of image files for the tree representations. It is designed to be included in a properly formed HTML document in the parent directory to the results.

The document path for links (`href`) and images (`src`) is by default set to the same as the directory in which the results are written; ie. if the results are written to `gotcha_output` then all the `href` and `src` links will take the form `href="gotcha_output/filename"`. This link prefix can be modified with the `--linkprefix` *[path]* which will then insert that prefix for all links in the HTML.

If you wish to inclure a web server address (ie a fully qualified URL) then specify this with the `--webpath [http://my.server.com:port/]` option.

By default GOtcha does not produce a full HTML page. It can be persuaded to do so by specifying `--html 'page title'` and will then produce an `index`*prefix*`.html` file in the output directory where *prefix* is any argument given to `--reprocess` and `'page title'` is a user specified text string to use as the title for the web page..

Some ontologies, particularly Biological Process, can produce a large graph that is mostly nodes with a very low P-score.These can be eliminated using the `--cutoff [integer]` option with a value between 0 and 100. All nodes with a score below that specified will not be shown in the graphical output but will still be listed. This has no effect on the text output.

3.1.5 High throughput optimisation

If you are analysing a large number of sequences and want to improve the performance of GOtcha, here are some guidelines.

- use `--nopng` to disable graphical output. This will dramatically speed up the process.
- Minimise the number and size of files you generate by using the `--tar` and `--outcomp` options.
- You can minimise the file read/write to disk by using a RAM disk as the temporary area, and as an area with which to build the tar archives (if you use the `--tar` option). The path to this can be specified with `--fastdir [path]`.
- If you want results with and without IEA then use the `--reprocess [tag]` option.

GOtcha runs very well in parallel environments. It has been successfully run using Grid Engine, CONDOR and LSF.

3.1.6 SQL output

GOtcha can write an SQL instruction set for import of results directly into a relational database. It expects to find a table `gotcharesult` with the following structure:

Table 2: SQL table structure for SQL output

contigid	integer	Integer identifier for the sequence. This should be a reference to a table of sequence identifiers. contigid is set by the option `--contigid`
seqdb	string	Name for the sequence database from which a sequence is obtained. This name is set by the option `--seqdb`
reprocess	string	Name of the run derived from the `--reprocess` option
goterm	integer	The numerical part of the GO identifier
iscore	float	Value of the I-score for the assignment
var	float	Standard deviation for the I-score
cscore	float	C-score for the assignment
pscore	integer	Percent score for the assignment.
ontology	string	Ontology to which the term belongs.

3.2 Configuration files

Long command lines are unwieldy and prone to errors being transferred. Almost all command line options can be placed in a file to be read by GOtcha with the option `--config` *filename*. GOtcha will read options in the order

1. Built in defaults
2. Configuration file
3. Command line.

The last option read is that which is taken, so default settings can be overridden by the command line.

A configuration file should look like:

```
searchdb database1
searchdb database2
blastmat /local/blast/data
```

GOtcha will in any case attempt to read the file `$GOTCHA_LIB/data/gotcha.conf`. An example file, `gotcha.conf_example` is included in the data directory of the distribution.

3.3 Reference

This is a comprehensive listing of the GOtcha command line options. They have been grouped by type. Options described as multi can be specified with multiple values, eg. `--searchdb foo --searchdb bar` or `--searchdb foo,bar` or a combination of the two.

Table 3: Basic runtime options

`--infile [filename]`	optional	Sequence file defaults to STDIN
`--outfile [directory]`	optional	Output directory for results. Defaults to `gotcha.[jobid]`
`--searchdb [dbname]`	required multi	Blast databases to use.
`--seqtype [T|F|A]`	optional	Protein (T) or DNA (F) sequence. Defaults to Autodetect (A)
`--excludetaxa [taxon]`	optional multi	taxa to exclude (using NCBI taxonomic id)
`--includecode [code]`	optional multi	Only use links with these evidence codes. Cannot be used with `--excludecode`
`--excludecode [code]`	optional multi	Only use links without these evidence codes. Cannot be used with `--includecode`
`--config [filename]`	optional	Read configuration options from the file specified. Any command line options will override the settings in this file.

Table 4: Configuration Options

`--blastdb [directory]`	optional	Directory containing BLAST databases Defaults to `$BLASTDB` if set
`--blastmat [directory]`	optional	Directory containing BLAST matrices Defaults to `$BLASTMAT` if set
`--goidx [filename]`	optional[5]	Filename for the GO database index file. Defaults to `$GOTCHA_LIB/data/terms.idx`
`--linkidx [filename]`	optional[6]	Filename for the GO database links index file. Defaults to `$GOTCHA_LIB/data/links.idx`
`--scoreidx [filename]`	optional	Filename for the GO database scores index file. Defaults to `$GOTCHA_LIB/data/scores.idx`
`--tmpdir [directorypath]`	optional	Path to store temporary files. default `/tmp`
`--dotfontpath [directorypath]`	optional	path to truetype fonts required by `dot`. Default `/usr/share/fonts/truetype`
`--dotfontname [name]`	optional	Name of truetype font used by `dot`. Default `Arial`
`--embosspath [directory]`	optional	Set the path to the EMBOSS executables
`--blastpath [directory]`	optional	Location of the `blastall` program.
`--fastdir [dirname]`	optional	Location to build results

Table 5: Advanced Run Options

`--cutoff [integer]`	optional	Restrict HTML output to terms with a probability higher than nDefault 0.
`--reprocess [name]`	optional	Do not rerun searches, just reprocess the results.
`--mindatapoints [integer]`	optional	minimum number of datapoints to use when assigning probabilities (default 40)
`--debug [integer]`	optional	Print debugging output to STDERR 0=no debug info. 1=default. 2-5 increasing verbosity.

Table 6: Graphics and Output Options

`--nopng`	optional	Do not produce graphics. (saves space for high throughput analysis)
`--linkprefix [path]`	optional	prefix to relative WWW links. Default gotcha.[jobid]
`--html [`*title*`]`	optional	Produce an `index.html` file with title [*title*].
`--webpath [server]`	optional	use `server` in fully qualified URI for links
`--tar [filename]`	optional	Produce results as a tar archive, not as single files.
`--incomp`	optional	existing tar file of results is compressed.
`--outcomp`	optional	Compress tar file upon creation
`--xref [dbname]`	optional	provide a list of database cross references for dbname in gotcha.dbname
`--topblast`	optional	Provide a list of GO terms derived from the top annotated hit of each database as top-blast.hits[.reprocess]
`--contigid [integer]`	optional	Numerical id for the sequence for insertion in SQL output. Default 0.
`--seqdb [name]`	optional	Database name for SQL output