

NAME

gpcommand – automate GenePool analyses

SYNOPSIS

```
gpcommand
  [-c filename] [--config_file=filename]
  [-l filename] [--log_file=filename]
  [-x] [--gpextract]
  [-z] [--gpanalyze]
  [-v level] [--verbose=level]
  [--help]
```

DESCRIPTION

gpcommand is the third of three programs in the GenePool system. It is a perl script that helps users run basic pooling analyses by reading a configuration file and automatically invoking the other two GenePool programs **gpextract**(1) and **gpanalyze**(1).

gpextract and **gpanalyze** are designed for direct use by researchers however each has a large number of command line options which can initially be difficult to understand and use correctly. **gpcommand** is a perl wrapper that will create and execute basic command lines that run **gpextract** and **gpanalyze**. All commands executed by **gpcommand** can optionally be echoed to a log file for later inspection.

A typical GenePool analysis will require a sequence of **gpcommand** calls. The section **EXAMPLES** below has more details.

Options

The following options are available:

-c filename | --config_file filename

The format of the configuration file is outlined in more detail in the **Configuration File** section below. The configuration file is the key to the operation of the **--gpextract** and **--gpanalyze** options since it specifies the CEL files that comprise the pooling experiment.

-l filename | --log_file filename

Use of a log file is optional and is only really of significant interest when used with the **--gpextract** and **--gpanalyze** options in which case it contains a record of the exact command lines that were constructed to execute the **gpextract** and **gpanalyze** programs.

-x | --gpextract

This is the first of three "action" options. It uses the information read from the configuration file (option **-c**) to construct and execute a series of calls to the **gpextract** program. It is highly recommended that you pair this option with the **-l logfile** option so that (a) you can see how the **gpextract** commandline works; and (b) so that you have a record of exactly how your CEL files were extracted.

-z | --gpanalyze

This is the second of three "action" options. It uses the information read from the configuration file (option **-c**) to construct and execute a series of calls to the **gpanalyze** program. You cannot run **--gpanalyze** unless you have already run **--gpextract** since the commands executed by **--gpanalyze** rely on reading the extracted data files created during the **--gpextract** run. As with **--gpextract** it is highly recommended that you pair this option with the **-l logfile** option so you can see the commands that were actually run. Once users are familiar with the GenePool system they will probably use **gpanalyze** directly rather than using this **gpcommand** option.

Configuration File

Much of the functioning of **gpcommand** is driven by a configuration file which should be created for each pooling experiment. The configuration file can have any name although the default is **genepool.ini** and if a file with the default name exists in the current directory then it will be used if no configuration file is explicitly specified. The configuration file is based on the familiar Windows INI file format and an example is shown here:

```
[ EXPERIMENT ]
```

```

Name = MEL01
Description = Melanoma test data for GenePool
[PLATFORM1]
Vendor = Affymetrix
ChipType = 500K
Enzyme = Nsp
CDFFilename = Mapping250K_Nsp.cdf
Case1 = Case_Nsp_1.CEL
Case2 = Case_Nsp_2.CEL
Case3 = Case_Nsp_3.CEL
Control1 = Control_Nsp_1.CEL
Control2 = Control_Nsp_2.CEL
Control3 = Control_Nsp_3.CEL
[PLATFORM2]
Vendor = Affymetrix
ChipType = 500K
Enzyme = Sty
Case1 = Case_Sty_1.CEL
Case2 = Case_Sty_2.CEL
Case3 = Case_Sty_3.CEL
Control1 = Control_Sty_1.CEL
Control2 = Control_Sty_2.CEL
Control3 = Control_Sty_3.CEL

```

You must have exactly one EXPERIMENT section and you must have at least one PLATFORM section but you can have as many PLATFORM blocks as your experiment requires - 1 for Affymetrix 10K, 2 for Affymetrix 100K, 2 for Affymetrix 500K etc.

The section titles (inside [square braces]) should be all capital letters. The attribute lines (variable=value format) can have zero or more spaces around the = sign so you can choose to line up values vertically as shown in the PLATFORM2 section of the example above. The variable name (to the left of the equals sign) is case insensitive but the value (to the right of the equals sign) *is* case sensitive since it includes file names etc. Comment lines can start with a pound sign or a semicolon but there is no facility for inline comments so you can't put a comment on the end of an attribute line or a section title line.

The use of generic PLATFORM sections should allow us to mix data from multiple Affymetrix Mapping arrays of different density, and platforms from other vendors including Illumina. For example, an experiment could contain 3 platforms - the Illumina 320K chip and the Nsp/Sty pair of chips that constitute the Affymetrix 500K platform. A corollary of this is that the attributes that are valid within a PLATFORM section are determined by the VENDOR= and CHIPTYPE= attributes.

The list of currently valid values for the VENDOR= attribute line within PLATFORM sections is **Affymetrix**, and **Illumina**.

The list of currently valid values for the CHIPTYPE= attribute line within PLATFORM sections is **10K**, **100K**, and **500K** if the Vendor is Affymetrix and **300K**, and **550K** if the Vendor is Illumina.

EXAMPLES

There are two steps to a GenePool analysis (corresponding to the 2 executables **gpextract** and **gpanalyze**) and the example below shows the basic sequence:

```

gpcommand -c mypool.ini -l mypool.log --gpextract
gpcommand -c mypool.ini -l mypool.log --gpanalyze

```

Before making any **gpcommand** calls, the user must define the mypool.ini configuration file as outlined in the **Configuration File** section above. Once the user has created a configuration file and copied it into the directory containing all of the CEL files, the user should **cd** into that directory and run the first command in the sequence above.

The first command uses the configuration file to work out all of the CEL files that are part of this analysis

and it runs **gpextract** over each one and logs all of the **gpextract** commands to the log file **mypool.log** for later inspection. For reference, this command took approximately 20 minutes on a 2.6GHz Pentium 4 for a pooling experiment that used 36 Affy 500K chips - 9 cases and 9 controls, each with Nsp and Sty chips. So as a rule of thumb, on a reasonably modern PC you should get close to two CEL files processed per minute.

The user should then run the second command which uses the configuration file to predict the names of the extracted files that **gpextract** would have created from each of the CEL files. It then runs 5 basic analyses using **gpanalyze**. Each of the analyses is written to an output file that starts with the name of the pooling experiment as specified in the EXPERIMENT section of the configuration file, and ends with the string *Output.txt*. For example, if the pooling experiment were named MEL01, then the five analysis output files created would be called:

```
MEL01_Consistency_UnweightedOutput.txt
MEL01_Consistency_WeightedOutput.txt
MEL01_Silhouette_EuclideanOutput.txt
MEL01_Silhouette_ManhattanOutput.txt
MEL01_Silhouette_ModManhattanOutput.txt
```

For reference, this command took approximately 2 minutes on a 2.6GHz Pentium 4 to run the 4 "canned" analyses on a pooling experiment that used 36 Affy 500K chips - 9 cases and 9 controls, each with Nsp and Sty chips.

Here are a few other **gpcommand** commandline examples:

No configuration file is specified so **gpcommand** expects to find a file called **genepool.ini** in the current directory. There is also no log file specified so there will be no logging:

```
gpcommand --gpextract
```

KNOWN ISSUES

Please see the **genepool(1)** manpage.

AUTHORS

John Pearson <jpearson@tgen.org>
Waibhav Tembe <wtembe@tgen.org>
David Craig <dcraig@tgen.org>
Matt Huentelman <mhuentelman@tgen.org>

SEE ALSO

genepool(1), **gpextract(1)**, **gpanalyze(1)**.

COPYRIGHT

GenePool is copyright 2006 by The Translational Genomics Research Institute. All rights reserved. This License is limited to, and you may use the Software solely for, your own internal and non-commercial use for academic and research purposes. Without limiting the foregoing, you may not use the Software as part of, or in any way in connection with the production, marketing, sale or support of any commercial product or service. For commercial use, please contact licensing@tgen.org. By installing this Software you are agreeing to the terms of the LICENSE file distributed with this software.

In any work or product derived from the use of this Software, proper attribution of the authors as the source of the software or data must be made. The following URL should be cited:

<http://bioinformatics.tgen.org/software/genepool/>