Table of Contents

Introduction

Purpose of this manual

Contained within this manual are details for configuring and running the application, Biofilter. If this is your first time to use the software, we highly recommend that you take a few minutes to download and work through one or more tutorials. Then, once familiar with the capabilities of the software, users can refer to this guide when making changes to the basic configuration settings.

Conventions Used

The following lists the conventions used throughout the document in order to distinguish commentary text from the actual commands run and the input users will provide.

Example Commands

Commands are listed as the user will input them, and are shown in `Courier New`. The commands are prefixed with a prompt, which should not be typed by the user. A prompt of `$` indicates a command to be run with normal user privileges, and a prompt of `#` indicates a command to be run with elevated privileges (such as installation). Examples are shown below:

```
$ sample-command
# sample-command-admin
```

Program Input/Output

Files that are used as either input or output will be listed in a gray box, as shown below.

```
Example Program Output.

Column1             Column2   Column3
really_long_data    short     more_data

                         (truncated)
```

Additionally, lines may be suppressed in some very long files, as shown above by the `(truncated)` line.

Common Parameters

There are a number of parameters which are used commonly across multiple configuration settings. In order to simplify the descriptions of the various properties of each command, we'll describe those properties here.

Integer

Parameters specified in this way just simply refer to a whole number. In general, these values should be equal to or greater than 0, except when specified otherwise.

Float

Values specified as float are decimal values.

Yes/No

These parameters accept a boolean, Yes/No type setting. Users can use ON/OFF or YES/NO to set them, and Biofilter recognizes the options regardless of case.

Filename

When a configuration refers to a file for input or output, the filename is generally used. This can be either a fully qualified path (such as /home/user/file.txt) or it can specified as a path relative to the directory where the application was run (such as ../data/goodfilename). It can also be just a plain filename as long as the file itself is available from the directory in which the application was run.

Label

A label refers to a parameter whose value can be any text string without whitespace. These labels are generally used for reporting but in many cases are used to determine filenames. As a result, users should avoid using unusual characters (such as “/”, “!”, or “#”) in the string that could possibly cause problems with filenames. Because spaces and tabs are used to separate each parameter on a given line, labels can not contain spaces.

Description

A description is a chunk of text that can contain spaces. It will always be at the very end of a line and is generally optional.

Installing Biofilter

Biofilter is packaged with the GNU autotools, so installation occurs in four steps: unpacking, configuration, compilation and installation. Each of those steps will be described below, but first the user must ensure that the prerequisites for running Biofilter are met, as well as the prerequisites for generating the supporting biological database, which we have called the Library of Knowledge Integration (LOKI).

Prerequisites

The following are prerequisites for building and running Biofilter. The packages that are needed only for building the LOKI database are indicated.

- A modern C++ compiler
- Boost Libraries for C++, version 1.46 or later
- SQLite, version 3.5.4 or later
- SOCI, with SQLite support compiled
- MySQL client libraries, version 5.1 or later (LOKI only)
- Python, version 2.6 or later (LOKI only)
    - suds for Python, version 0.4 or later
    - MySQLdb for Python
    - SQLite bindings for python

Unpacking

Biofilter is distributed as a zipped tarball, and the command for unpacking the distribution is:

```
$ tar -xvzf biofilter-1.1.0.tar.gz
```

This will unpack the source code into a directory called `biofilter-1.1.0`. For all of the following commands, we assume that you are in this directory.

```
$ cd ./biofilter-1.1.0
```

Configuration

In order to compile Biofilter, the user must first configure the software. This script will attempt to detect all of the prerequisites on the user's system, and this is the time for the user to specify system-specific options, such as the location of the installed program. The command is:

```
$ ./configure
```

The configure script can also take a number of helpful options, some of which are detailed below:

- **--help**
  This option will list all of the available options that can be passed to the configure script.

- **--prefix=[path]**
  This option tells Biofilter to install itself into the given path, which is useful if you do not have administrative access to the computer. By default, the program will be in [path]/bin, and the LOKI database will be in [path]/share. Note: when using this option, the path given must be an absolute path and cannot use any shell expansions, such as the "~" notation.

- **--disable-loki**
  This option disables the compilation of the LOKI biological database. Since the compilation of the database will take a few hours with a high speed internet connection, this option is helpful if you are installing a new version of Biofilter, but you want to leave the database unchanged.

- **--enable-debug**
  For the advanced users, this option will turn off all optimization and turn on debugging symbols, which can be helpful in diagnosing a problem with the Biofilter software.

Compilation

LOKI Setup

Due to the size of the LOKI database, it is not distributed along with the Biofilter code. We provide the means for a user to build the LOKI database by downloading the data directly from the sources. The LOKI database must be compiled before installation described further below.

In order to build the LOKI database, the user must have complete access to a MySQL database, with permissions including the ability to drop and create tables. The installer will not create a MySQL database during compilation; it will create LOKI tables within an existing MySQL database. Thus, the MySQL database must exist, and the user must have permissions to drop and create tables within the MySQL database.

The database settings are located in a file called "BioUpdater/dbsettings.py". The user must supply the database host, username, password and name. Alternatively, you can use the environment variables "DB_HOST", "DB_USER", "DB_PASS", and "DB_NAME" to override any or all of the values. The environment variables will take precedence over the values given in the configuration file. The default values of the parameters are given below:

Parameter	Value
hostname	localhost
username	root
password	
database	LOKI

Note that if the settings were incorrect and you received an error during the compilation of the LOKI database, you must follow the steps given in Rebuilding the Database below.

Compiling Biofilter

During the compilation of Biofilter, the program is built and the LOKI database is generated (after the LOKI setup steps above), if desired. The command is:

```
$ make
```

Installation

Installation is the point at which the program and database are moved into their final locations, as defined during the configure step. Typically, the user will need administrative rights to complete the installation step. To install both Biofilter and LOKI, type:

```
# make install
```

If you only want to install Biofilter, you can type:

```
# make install-exec
```

And if you only want to install the LOKI database, you can type:

```
# make install-data
```

During the installation of LOKI, the database is copied to the destination directory, and it is named "`yyyy.mm.dd.knowledge.bio`", where "`yyyy.mm.dd`" is the date of creation of the knowledge database. However, for ease of use, the installer will also create a shortcut called simply "`knowledge.bio`" that will point to this installed file. This is designed so that a user may have multiple concurrent LOKI databases that each correspond to a different snapshot in time.

Rebuilding the Database

The LOKI database that Biofilter uses is static, and it will not capture updates made to the sources as the sources are updated. Thus, from time to time it becomes necessary for the user to rebuild the database with the most recent information. Assuming that the user configured Biofilter to build the LOKI database in the first step, the command to discard the current LOKI database from the build directory is:

```
$ make clean
```

From this point, the user can re-run the compilation and installation steps to regenerate and reinstall the LOKI database. Note that the old LOKI database will **NOT** be deleted from the installation directory, but the shortcut will be updated to the most recent database.

Using the Biofilter Application

The Biofilter application can be used for many purposes, and as a result, there are many options available to the user to customize the behavior of the program. In general, the execution of Biofilter is as follows:

```
$ biofilter [OPTIONS] config-file
```

Above, the `[OPTIONS]` are command-line options, which are described below, and the `config-file` is a configuration file which specifies the behavior of Biofilter. If options given in the configuration file and command line differ, the command line options will take precedence.

Configuration Parameters

The following parameters can be given in the configuration file. For each command, we give the calling syntax of the parameter as well as an example. In the syntax, any optional parameters will be listed in square brackets ([]). When a specific input file is needed, the format of the input file will be referenced in the "See also" section.

SETTINGS_DB

Syntax: **SETTINGS_DB filename**

```
SETTINGS_DB knowledge.bio
```

This option sets the location of the LOKI database to be used by Biofilter. The filename can either be given as an absolute or relative path. If the database is not found relative to the current working directory, Biofilter will search the data directory given during installation for the LOKI database.

MAX_GENE_COUNT

Syntax: **MAX_GENE_COUNT integer**

```
MAX_GENE_COUNT 30
```

This configuration option sets the maximum number of genes in a pathway to consider the pathway valid for generating gene/gene models. Some pathways are so encompassing that the considering the relationships between the genes would lead to an overwhelming amount of generated models. By increasing this value, more values will be created, and decreasing this value will create fewer models.

Note that this setting applies only to groups found in LOKI, and it will not restrict the size of groups loaded with the ADD_GROUP configuration option.

RS_SOURCE

Syntax: **RS_SOURCE filename**

```
RS_SOURCE Illumina-660Quad.txt
```

This option allows the user to limit the list of SNPs to those included in the file of interest. Typically, this will be a list of SNPs that are on a given platform, as the above example illustrates. The file must

be a list of only unique SNP RS identification numbers (the "RS" prefix removed), as described in the input files section.

See also: SNP List

MAP_SOURCE

Syntax: **MAP_SOURCE filename**

```
MAP_SOURCE variants.txt
```

This option is similar to the RS_SOURCE, except that it allows the user to include variants that may not have actual RS numbers. The format of this file must be PLINK 4-column format, which we summarize in the Variant List section. For more details on PLINK, see: http://pngu.mgh.harvard.edu/~purcell/plink/

See also: Variant List

ADD_GROUP

Syntax: **ADD_GROUP filename**

```
ADD_GROUP new_group.txt
```

This gives the user the option of adding in custom groups of genes that are known to the user but are not captured in the LOKI database. The input is a plain text file containing the type of group collection as well as the groups and their associated genes.

See also: Custom Groups

INCLUDE_GROUPS

Syntax: **INCLUDE_GROUPS integer [integer] [...]**

```
INCLUDE_GROUPS 1 1453
```

This configuration option allows the user to limit the search of information according to the given group IDs, which must match those found in the LOKI database. If given, only those groups (and their children) will be considered when generating models and annotations.

This option is especially helpful in limiting Biofilter to a specific list of databases. These IDs can be found in the Gene Definition report under the "Groups" column. Note that these IDs are internal to LOKI and are not available from public data sources. To use group names, please see the INCLUDE_GROUP_NAMES configuration option.

See also: Gene Definition

INCLUDE_GROUP_FILE

Syntax: **INCLUDE_GROUP_FILE filename**

```
INCLUDE_GROUP_FILE group_ids.txt
```

This option is provided as a convenient alternative to the INCLUDE_GROUPS command above. Instead of listing all of the group IDs in the configuration file, the user may use a separate file where the group IDs are listed individually, one per line.

INCLUDE_GROUP_NAMES

Syntax: **INCLUDE_GROUP_NAMES label [label] [...]**

```
INCLUDE_GROUP_NAMES  GO:0003674 hsa00010 Pfam
```

This option allows the user to limit the search to groups as in the INCLUDE_GROUPS command above, except that groups are identified by their name instead of their ID in the LOKI database. This is helpful if the user has interest in a specific pathway from a database included with Biofilter.

INCLUDE_GROUP_NAME_FILE

Syntax: **INCLUDE_GROUP_NAME_FILE filename**

```
INCLUDE_GROUP_NAME_FILE group_names.txt
```

Again, as above, this option is provided as an alternative to the INCLUDE_GROUP_NAMES command, providing a way for the user to supply a file containing a list of group names to include, one per line.

POPULATION

Syntax: **POPULATION label**

```
POPULATION NO-LD
```

This command sets the population on which to base the gene boundaries. The population "NO-LD" will always be included in the LOKI database, and this population is the default boundaries of a gene from either Entrez or Ensembl. Often the "NO-LD" boundaries of a gene will be referred to as the "true" boundaries of the gene.

GENE_BOUNDARY_EXTENSION

Syntax: **GENE_BOUNDARY_EXTENSION integer**

```
GENE_BOUNDARY_EXTENSION 1000
```

When using the "NO-LD" population, this option gives the user the ability to extend the gene boundaries by the given number of base pairs. Th extension occurs both upstream and downstream of the actual boundaries in the database.

REPORT_PREFIX

Syntax: **REPORT_BOUNDARY label**

```
REPORT_PREFIX myReport
```

This option allows the user to set a prefix for all of the output files that are generated by Biofilter. This is useful when running Biofilter repeatedly, as it will prevent older results from being overwritten.

IMPLICATION_IDX_DUPLICATE_WEIGHT

Syntax: **IMPLICATION_IDX_DUPLICATE_WIEGHT float**

```
IMPLICATION_IDX_DUPLICATE_WEIGHT 0.25
```

When calculating the implication index of a particular model, often a pair of genes will appear in more than one group. When this happens in a disease independent source, the implication score is incremented by 1 for each source a pairing if found in, and then by the IMPLICATION_IDX_DUPLICATE_WEIGHT for each duplicate pairing found in a source. Thus, if two genes are found in three pathways in a disease independent source with this configuration value set to 0.25, the implication index will be increased by 1.5 for this source.

BINARY_MODEL_ARCHIVE

Syntax: **BINARY_MODEL_ARCHIVE Yes/No**

```
BINARY_MODEL_ARCHIVE No
```

Enabling this option allows for the production of binary model files in order to save disk space. If enabled, the gene/gene and SNP/SNP models will be printed in a proprietary binary format. This format is highly implementation dependent, and may not be portable among computers.

See also: Binary Files

DISEASE_DEPENDENT_LEVEL

Syntax: **DISEASE_DEPENDENT_LEVEL ALL_MODELS/GROUP_LEVEL/DD_ONLY**

```
DISEASE_DEPENDENT_LEVEL ALL_MODELS
```

With this configuration option, users can choose to filter the gene/gene model results by how closely they are related to a disease dependent source. This configuration option may take one of the three following values:

- **ALL_MODELS**
  When this value is used (the default), Biofilter will generate all gene/gene models, regardless of their relation to a disease dependent source.
- **GROUP_LEVEL**
  When this value is used, Biofilter will generate all gene/gene models for groups that contain at least one gene in a disease dependent group. Note that a model may be generated from two genes not in any disease dependent group if they are in the same group as a third gene which is also in a disease dependent group.
- **DD_ONLY**
  When this value is used, Biofilter will only generate gene/gene models in which one of the genes in the model is contained in a disease dependent group.

GENOMIC_BUILD

Syntax: **GENOMIC_BUILD string**

```
GENOMIC_BUILD 37
```

This setting tells Biofilter the build of the genome that the input data is based on. This is especially important for input that is based on position, as in the MAP_SOURCE configuration option. If the input data is not the same build as used internally by Biofilter, the software will perform a "lift over" of the input data into the genomic build used by Biofilter.

DETAILED_REPORTS

Syntax: **DETAILED_REGeneGeneModelPORTS Yes/No**

```
DETAILED_REPORTS On
```

When this option is set to "On", Biofilter will add in more details in the output reports. See the Output Files section for more information on what information is included only in the detailed reports.

MARKER_INFO_REPORT

Syntax: **MARKER_INFO_REPORT Yes/No**

```
MARKER_INFO_REPORT On
```

When this option is set to "On", Biofilter will produce a marker report which lists the SNPs considered by Biofilter along with their chromosome and base pair locations.

See also: Marker Info Report

SNP_REPORT

Syntax: **SNP_REPORT Yes/No**

```
SNP_REPORT On
```

When this option is "On", Biofilter will produce a report which lists all of the SNPs used by Biofilter along with any genes that contain the given SNP.

See also: SNP Report

SNP_GENE_REPORT

Syntax: **SNP_GENE_REPORT Yes/No**

```
SNP_GENE_REPORT On
```

When this option is "On", Biofilter will produce a report that details the relationship between all of the SNPs considered to be in genes and how they are related.

See also: SNP/Gene Relationship Report

GENE_COVERAGE

Syntax: **GENE_COVERAGE filename|ALL**

```
GENE_COVERAGE gene_list.txt
```

This option is a filename containing a list of gene aliases to restrict a gene coverage report. If "ALL" is given instead of a file, all genes in the database containing at least one marker will be listed in the gene coverage report. Note: this option does not affect the generation of models; it is only used in the Error: Reference source not found command line option, which precludes the generation of gene/gene or SNP/SNP models.

See also: Gene Aliases, Gene Coverage Report

GENE_REPORT

Syntax: **GENE_REPORT Yes/No**

```
GENE_REPORT On
```

When set to "On", Biofilter will generate a report that lists all of the genes that were used to generate models, along with a great deal of detail about the genes and their contained SNPs.

See also: Gene Definition

COVERAGE_RS

Syntax: **COVERAGE_RS filename**

```
COVERAGE_RS rs_ids.txt
```

This option allows the user to specify a list of RSIDs in generating a gene coverage report. The list of RSIDs must be the integer portion of the RSID, listed one per line. To include multiple platforms, this option can be given more than once in a configuration file.

See also: SNP List, Gene Coverage Report

COVERAGE_MAP

Syntax: **COVERAGE_MAP filename**

```
COVERAGE_MAP map_ids.txt
```

This option allows a user to specify a list of markers on a platform in generating a gene coverage report. The format of the file must conform to the Variant List specification. To include multiple map-based platforms, the user may specify this option more than once in a configuration file.

See also: Variant List, Gene Coverage Report

MINIMUM_IMPLICATION_INDEX

Syntax: **MINIMUM_IMPLICATION_INDEX float**

```
MINIMUM_IMPLICATION_INDEX 2.0
```

When generating models, this option gives the minimum implication score required to consider a model. Output models are ordered by implication score, and this can be a means to reduce the number of testable models that are generated so that the multiple testing burden is reduced. Note that all gene/gene models are generated, but only those with an implication score surpassing this threshold are

actually stored in memory. Also, only SNP/SNP models with implication scores above the threshold are ever generated, so this parameter can serve to reduce a computational burden in model generation.

MAX_SNP_MODEL_COUNT

Syntax: **MAX_SNP_MODEL_COUNT integer**

```
MAX_SNP_MODEL_COUNT 1500
```

This option provides a maximum number of SNP/SNP models to create. The SNP/SNP models are ordered by their implication index, with higher implication scores receiving priority. Note that this number is a hard cutoff, so there may be additional models at the same implication score as the last model provided that were not output. This parameter can be used to reduce the computational burden by reducing the number of SNP/SNP models generated to a known, fixed constant.

EXPORT_SNP_MODELS

Syntax: **EXPORT_SNP_MODELS Yes/No**

```
EXPORT_SNP_MODELS On
```

When set to "On", Biofilter will output a list of SNP/SNP models based on the Gene/Gene models that are generated. The models will be ranked in order of their implication index, with higher implication scores appearing first. Note that when this option is enabled, MARKER_INFO_REPORT will also be enabled.

See also: SNP/SNP Models

EXPORT_GENE_MODELS

Syntax: **EXPORT_GENE_MODELS Yes/No**

```
EXPORT_GENE_MODELS On
```

When set to "On", Biofilter will output a list of Gene/Gene models that are identified in the LOKI database as biologically relevant. The models will be ranked in order of their implication index, just as the SNP/SNP models are. Note that when this option is enabled, GENE_REPORT will also be enabled.

See also: Gene/Gene Models

Optional Commands

In default operation, Biofilter generates gene/gene and SNP/SNP models for testing purposes. However, certain features of Biofilter can be activated by using certain command-line flags. When any of the flags below are used, the Biofilter will perform only the given task, and models will not be performed. For the most part, these tasks are used to gather data from the LOKI database or to get help about the correct usage of Biofilter.

Typically, command line options are called with two dashes (--) preceding the name of the option. However, come options are available as a short version, which are preceded by a single dash (-). When available, the optional short version is in parentheses. Any arguments that the options take are given in angle brackets (<>), and are explained in the text.

--help (-h)

Prints a summary of the available command-line options and exits.

--sample-config (-S)

This parameter takes no arguments and causes the output of the Biofilter to generate a basic configuration based on the default settings (and any that have been overridden by other parameters). No other execution is performed.

--list-populations (-P)

Lists the populations available in the database in use. Populations are used to adjust the gene boundaries to include additional SNPs that are observed to be within an LD threshold. By default, there is only one population, given by "NO-LD". Note that if this option is given, no analysis will be done; the only output will be the list of populations available which is written to the screen.

--report-gene-coverage

See also: GENE_COVERAGE, COVERAGE_RS, COVERAGE_MAP, Gene Coverage Report

When this option is given on the command line, Biofilter will generate a gene coverage report based on the list of genes given by GENE_COVERAGE and the list of markers defined by COVERAGE_RS and COVERAGE_MAP. Note that when this option is provided to Biofilter, no models will be generated, and the only output will the the Gene Coverage Report.

--groups (-G) <label|ALL>

See also: Group List

When this option is selected on the command line, Biofilter will print a list of the groups or pathways matching the search criteria to the screen. The given label is a comma-separated list of criteria to search for matching groups. If "ALL" is given, Biofilter will list all available groups contained in the LOKI database. When searching for groups, Biofilter searches both the group name as well as the description, and any group matching one or more criteria is returned.

The following example will list all groups related to either "NOZZLE" or "GNVR":

```
biofilter config -G NOZZLE,GNVR
```

--genes <label|ALL> <label|ALL>

See also: Gene List

When this option is selected, Biofilter will print a list of genes and their associated aliases from the LOKI database. The first label is a comma-separated list of aliases to search for. The aliases may be all or part of a gene identifier. For example, searching for an alias of "AZ" will produce all genes that can be identified by an alias containing the letters "AZ". The second label gives a comma-separated list of alias types to restrict the search to.

The following example searches for a gene with the alias “BRCA1” or “ABCF”, restricted to aliases found in the Entrez database:

```
biofilter config --genes BRCA1,ABCF Entrez
```

Command-Line Options

Below is a list of all of the available command line options and the manner in which they are called. These options will not affect the manner in which Biofilter runs, and these options typically override options given in the configuration file, which can help the user when making rapid small adjustments to the configuration.

--DB <filename>

See also: SETTINGS_DB

This option overrides the SETTINGS_DB value in the configuration file, and uses the given LOKI database. This is helpful for using a LOKI database that was downloaded earlier than the most recent one.

--marker-info

See also: MARKER_INFO_REPORT

Sets the value of the MARKER_INFO_REPORT configuration option to “On”, overriding the value given in the configuration file.

--binary (-b) <Yes/No>

See also: BINARY_MODEL_ARCHIVE

When this option is given on the command line, it overrides the BINARY_MODEL_ARCHIVE setting from the configuration file and uses the value given.

--detailed (-D)

See also: DETAILED_REPORTS

When selected on the command line, this overrides the configuration setting for DETAILED_REPORTS from the configuration file and produces detailed reports for all appropriate outputs.

--cov-rs <filename>

See also: COVERAGE_RS

When selected, uses the given filename in the COVERAGE_RS option in addition to any value given in the configuration file.

--cov-map <filename>

See also: COVERAGE_MAP

When selected, uses the given filename in the COVERAGE_MAP option in addition to any value given in the configuration file.

--add-group (-d) <filename>

See also: ADD_GROUP

When given on the command line, this allows the user to define a custom group, as defined in the Custom Groups section of the input file formats. This argument adds custom groups in addition to the groups given in the configuration file.

--gene-file (-g) <filename|ALL>

See also: GENE_COVERAGE

When selected, uses the given filename in the GENE_COVERAGE option, overriding any value given in the configuration file.

--list-genes

See also: GENE_REPORT

Sets the value of the GENE_REPORT to "ON", overriding this setting from the configuration file.

--snp-report

See also: SNP_REPORT

Sets the value of the to "ON", overriding this setting from the configuration file.

--map-snps-to-gene

See also: SNP_GENE_REPORT

Sets the value of the SNP_REPORT to "ON", overriding this setting from the configuration file.

--build (-B) <label>

See also: GENOMIC_BUILD

When selected, uses the given build in the GENOMIC_BUILD option, overriding any value given in the configuration file.

--PREFIX <label>

See also: REPORT_PREFIX

When selected, overrides the REPORT_PREFIX option from the configuration file with the given prefix.

--snps (-s) <filename>

See also: RS_SOURCE

When selected, uses the given filename in the RS_SOURCE option, overriding any value given in the configuration file.

--set-population (-p) <label>

See also: POPULATION

When selected, uses the given label for the POPULATION configuration option, overriding the setting found in the configuration file.

--gene-boundary <integer>

See also: GENE_BOUNDARY_EXTENSION

When selected, uses the given integer for the GENE_BOUNDARY_EXTENSION configuration option, overriding the value in the configuration file.

--variants (-v) <filename>

See also: MAP_SOURCE

When selected, overrides the MAP_SOURCE configuration option with the given filename.

--write-models (-W) <float> <integer>

See also: EXPORT_GENE_MODELS, MINIMUM_IMPLICATION_INDEX, MAX_SNP_MODEL_COUNT

When selected, sets EXPORT_GENE_MODELS configuration option to “ON”. The float and integer arguments are optional (but must be given in order), and if supplied, will override the MINIMUM_IMPLICATION_INDEX and MAX_SNP_MODEL_COUNT, respectively.

--export-snp-models (-X) <float> <integer>

See also: EXPORT_SNP_MODELS, MINIMUM_IMPLICATION_INDEX, MAX_SNP_MODEL_COUNT

When selected, sets EXPORT_SNP_MODELS configuration option to “ON”. The float and integer arguments are optional (but must be given in order), and if supplied, will override the MINIMUM_IMPLICATION_INDEX and MAX_SNP_MODEL_COUNT, respectively.

Input File Formats

This section lists all possible input files that can be given to Biofilter. With very few exceptions, input files are space delimited ASCII files.

Gene Aliases

The gene aliases are a means for a user to restrict searches on the LOKI database to a set of predefined genes. Internally, LOKI uses Ensembl gene IDs as the canonical name of the gene, but the genes listed in this file can be alternative names, such as those found in Entrez or Uniprot. When using aliases for the gene, only non-ambiguous aliases are considered valid in this file.

Example file:

```
NMT1
FURIN
RD1
S100B
ATP2A2
```

SNP List

The SNP Source file contains all SNPs to be used in the analysis. Generally, this will match the SNPs from the platform to be used in the analysis. However, it is also possible to use a highly restricted set for other types of analysis (such as identifying which genes a set of interesting SNPs might be found in.)

The format is very simple. List all RS IDs in their integer format (removing the "RS" before each number). Each ID should be separated by whitespace. An example is shown below:

```
10000169
10000185
10000201
1000022
10000226
1000025
10000255
10000266
```

Variant List

The variant list is a means for a user to specify SNPs or other variants which do not necessarily have an RSID associated with them. This file and the SNP List above are mutually exclusive, with the variant List taking precendence. The format of this file is a PLINK 4-column map file with the columns being Chromosome, ID, Genetic distance (not used), and base pair location. An example is shown below:

```
4  rs10000169  0  77575270
4  rs10000185  0  7560658
4  rs10000201  0  11829395
13 rs1000022   0  99259220
```

```
4  rs10000226  0  87957991
6  rs1000025   0  134127012
4  rs10000255  0  162644668
4  rs10000266  0  40399629
```

Custom Groups

In addition to the groups that have been defined in the LOKI database, Biofilter allows a user to define groups of genes and submit this custom group in an input file. This can be useful in defining disease-dependent groups of genes. The custom group must be formatted according to the Group File Format defined below.

Group Definition

A user can create many custom groups within a single file, and each file defines a single high-level set of groups that are all related in some way. A single group may contain a collection of genes as well as other children groups, thereby allowing for a hierarchical structure to be defined in the custom group file. When calculating a model's implication index, each custom group file is considered to be a single source, comparable to the database sources loaded from LOKI.

Group File Format

Custom groups are defined using a plain text file with a specific format. Each file defines a set of related groups and forms a single meta-group. The file must follow the following format, which will be explained below:

```
[Source Name] [Source Type] [Source Description]
GROUP [Name] [Description]
([alias] [alias]*)|(CHILDREN [group] [group]+)|(GROUP ...)
```

[Source Name] [Source Type] [Source Description]

The first line of the file must contain the name of the collection of groups, along with the type of collection.

The **"Source Name"** must be a string with no spaces, and it must be unique from any other source already defined in the LOKI database. The number of sources is very limited, and if the name of this group is not the same as any database of biological knowledge, there should be no namespace conflicts.

The **"Source Type"** defines the type of information contained within this custom collection of groups. This value must be a string taking one of the following values:

- **DISEASE_INDEPENDENT**
  This option is for a collection of groups that is not related to any particular disease.
- **DISEASE_DEPENDENT**
  This option is for a collection of groups that is related to a particular disease of interest.
- **GENE_COLLECTION**
- **SNP_COLLECTION**

The **"Source Description"** is an optional string designed to help the user keep track of the actual meaning behind the group. The description may contain any character except a newline ("\n").

GROUP [Name] [Description]

This line defines the beginning of a new group. This line must be given on the second line, and it may occur on subsequent lines within the file.

The **"Name"** must be a string with no spaces, and it must be unique from any other group name defined within the current custom group file. This name can be used to identify the current group as a child of another group.

The **"Description"** is an optional string used to describe the group.

CHILDREN [group] [group]+

This line defines associations between groups within the custom group collection. The first group given is considered the parent, and all subsequent groups are the children. Note that there must be at least two groups given in this line.

*[alias] [alias]**

This line is a whitespace-separated list of gene names that can be found in the LOKI database. Currently, only non-ambiguous gene aliases are considered valid inputs.

Examples

Because this is possibly the most complex input file available to Biofilter, we have provided a couple examples below. The simple group definition should be sufficient for anyone attempting to use a list of genes that are associated with a given disease. The more complex example illustrates an interrelated pathology.

Simple Group Definition

This file is a single group containing a simple collection of genes that are associated with Alzheimer's

```
ALZHEIMERS DISEASE_DEPENDENT Alzheimer's Collection
GROUP alz-assoc Genes associated with Alzheimer's
AGT
APH1A
APOA1BP
APOA2
CAMK1G
CFH
CHRNB2
```

Complex Group Collection

The following example shows a slightly more complex collection of interrelated groups, still using the Alzheimer's data above, but split into two groups, one with genes starting with the letter “A” and one with genes starting with the letter “C”. Additionally, there is a parent super-group that contains both subgroups. Also, this file demonstrates the inclusion of more than one gene on a single line, as can be see in the “alz-assoc-A” group.

```
ALZ-COMPLEX DISEASE_DEPENDENT Alzheimer's Complicated
GROUP alz-assoc-A Genes assoc. w/ Alzheimer's (beg. w/ A)
AGT       APH1A
APOA1BP   APOA2

GROUP alz-assoc-C Genes assoc. w/ Alzheimer's (beg. w/ C)
CAMK1G
CFH
CHRNB2

GROUP alz-master Master group for Alzheimer's
CHILDREN alz-master alz-assoc-A alz-assoc-C
```

Model Production

Overview

Biofilter uses biological information about gene-gene relationships and gene-disease relationships to construct multi-SNP models for conducting statistical analysis. Rather than annotating the independent effect of each SNP in a GWAS dataset, Biofilter allows the explicit detection and modeling of interactions between a set of SNPs. In this manner, Biofilter process provides a tool to discover significant multi-SNP models with non-significant main effects that have established biological plausibility. This approach has the added benefit of reducing both the computational and statistical burden of exhaustively evaluating all possible multi-SNP models.

Model production is gene-centric, and thus requires that any SNPs to be considered be mapped to genes. The gene mapping takes place internally using local copies of current data sources such as Ensembl, HapMap and dbSNP. A structured mapping is made based on relationships from one of the knowledge sources, and this information is used to identify candidates for SNP-SNP models.

The biological knowledge used by the Biofilter is derived from various sources which are identified as "Meta Groups" as well as optional user defined groupings. Currently, the data-sources represented include: Gene Ontology, KEGG, Net Path, pfam, Reactome and PharmGKB, drawn from the supporting biological database, which we have called the Library of Knowledge Integrattion (LOKI).

There are two basic types of data sources. Disease-dependent sources are user defined and relate a gene to the disease phenotype being studied (i.e. previously associated SNP). Using the appropriate group file format, disease-dependent genes can be included as a list or combined into multiple groups and relationships. Characterizing disease-dependent genes impacts the implication score; the score is calculated so that more weight is given to models pertaining to the specified input. Disease independent sources link more than one gene together. The goal is to identify pairs of genes with some prior evidence of putative epistasis. The databases collectively referred to as LOKI are disease independent sources, because they provide key relationships between genes in important biological processes (Bush 2009). However, the user can also define additional custom disease-independent groups.

Users can provide a set of SNPs that reflect the platform on which their analysis will be run. This can be a GWAS platform such as Illumina Human 1M-DUo BeadChip or one designed for the user's specific study (see Input File Formats: SNP List and Variant List). If a SNP in the input file does not exist according to Biofilter, it is ignored in the analysis and added to the Missing SNPs file. As a result, only those SNPs available in the Biofilter's local copy will be considered. All variants in the variant list input would be included because there is no internal representation of all possible base pair positions that contain SNPs.

Calculation of Implication Index

To rank the strength of the potential interactions, Biofilter uses an implication score, which is a measure of how many times the two genes or SNPs in the model are associated with each other. Since SNP/SNP models are generated from the gene/gene models, the implication index of a SNP/SNP model is defined by the generating gene/gene model's implication index.

For a gene/gene model, the implication index is calculated by counting the number of unique sources that associate the two genes. When a pair of genes are contained in the same group twice or more in the same source, the implication index is increased by 1 for each group for disease-dependent sources and by the value given by IMPLICATION_IDX_DUPLICATE_WEIGHT for each group in a disease-independent source. As an example, if IMPLICATION_IDX_DUPLICATE_WEIGHT was 0.25 and two genes were in three groups from a disease independent source and two groups from a disease dependent source, the implication index would be 3.5 ([1 + 0.5*(3-1)] + [1 + 1*(2-1)]).

Output Files

The output files that Biofilter produces are typically delimited ASCII text files. The files are differentiated by their suffix, which is given in each section. The naming convention for the files is "<prefix>-<suffix>", where the prefix is either given in the configuration file or is the name of the configuration file with any extensions removed.

Occasionally, reports will have more information available using the DETAILED_REPORTS configuration option. This additional information will typically be included as extra columns in a report, and they will be indicated as such, but the given examples will not be shown.

Marker Info Report

Suffix: **marker-info.map**

This file gives the list of all SNPs used in the generation of models in Biofilter. When generating SNP/SNP models, this file will always be produced, regardless of the MARKER_INFO_REPORT configuration setting. The file is a tab-delimited text file with the following columns:

- **ID**
  This column lists the unique identifier for the given marker. Typically, this is an RSID, but if using MAP_SOURCE, this is the ID given in the input file.
- **Chrom**
  The chromosome of the marker.
- **Pos**
  The base pair location of the marker.
- **Index**
  This column lists the internal index of the marker. This is used in the SNP/SNP model report, as seen in the SNP/SNP Models section.
- ***Role*** (Detailed Reports only)
  The final column will list the role of the SNP, as found in the LOKI database.

```
ID    Chrom    Pos  Index
RS1   1        7    0
RS2   1        14   1
RS3   1        21   2
RS5   1        35   3
RS7   1        49   4
RS11  1        77   5
RS23  2        21   6
```

Region Details Report

Suffix: **gene-report.csv**

This file is a comma-separated text file that lists all of the pertinent information about the genes used in Biofilter. The output is very similar to the Gene Definition report, except that SNPs are only printed

when detailed reports are used, and this report may be produced independently of generating gene/gene models. The columns in this report are:

- **Gene Name**
  The first column holds the canonical name of the gene.
- **Chrom**
  The second column lists the chromosome that the gene can be found on.
- **Eff. Start, Eff. Stop**
  Columns 3 and 4 list the population-specific base pair boundaries of the gene being looked at.
- **True Start, True Stop**
  Columns 5 and 6 list the canonical (NO-LD population) base pair boundaries of the gene.
- **Alias List**
  Column 7 lists all available aliases of the gene, separated by a colon (:). Note that ambiguous aliases are listed in this list.
- ***SNPs*** (Detailed Reports only)
  The final column lists the IDs of the SNPs that are associated with this gene, separated by a colon (:). Note that this column gives the IDs of the SNP as opposed to the internal SNP index.

```
Gene Name, Chrom, Eff. Start, Eff. Stop, True Start, True Stop, Alias List
G1,        1 ,    5,          15,        5,          15,        G1:R1
G2,        1 ,    25,         35,        25,         35,        G2:G23:R2
G3,        1 ,    45,         55,        45,         55,        G23:G3:R3
G5,        1 ,    30,         50,        30,         50,        G5:R5
```

SNP Report

Suffix: **snp-report.csv**

This file is a comma-separated text file that lists the relationships between markers and genes that Biofilter finds in the LOKI database. The file has the following columns:

- **Chrom**
  The chromosome of the marker.
- **RSID**
  The unique identifier of the marker. Usually, this will be an RSID, but when MAP_SOURCE is used, this is the identification string given to Biofilter.
- **Gene Name**
  A colon-separated list of all of the canonical names of the genes that contain the given marker.

```
Chrom,    RSID,     Gene Name
1 ,       RS1,
1 ,       RS2,      G1
1 ,       RS3,
1 ,       RS5,      G2:G5
1 ,       RS7,      G5
1 ,       RS11,     G4
2 ,       RS23,     G6
```

SNP/Gene Relationship Report

Suffix: **snp-gene-map.csv**

This file is a comma-delimited text file that details the location of the SNPs in relation to the associated genes. While a SNP is typically considered to be "inside" a gene, when using multiple populations, it is helpful to differentiate the term to describe if a SNP was included due to inclusion in the canonical gene or rather due to the gene expansion. The columns of the file are:

- **Chrom**
  The cromosome of the marker.
- **RSID**
  The unique identifier of the marker. Usually, this will be an RSID, but when MAP_SOURCE is used, this is the identification string given to Biofilter.
- **Gene Name**
  The name of the gene containing the marker. Note that if a given marker is in multiple genes, each SNP/gene combination will be on a separate line, as the location within the gene may be different for each SNP/gene pair.
- **Location w/in Gene**
  This describes the location of the marker within the associated gene. This value can take one of the following values:
  - **Interior**
    The SNP is located both within the canonical (NO-LD population) boundaries of the gene as well as the boundaries of the gene defined by the population. Note: when using the NO-LD population, this is the only valid value for this column.
  - **Flanking**
    The SNP is located outside the canonical boundaries of the gene, but inside the population specific boundaries.
  - **Exterior**
    The SNP is located outside the population specific boundaries of the gene. This should only be seen if the SNP was explicitly linked to the gene in question (which is currently not available as of Biofilter v1.1).

```
Chrom,    RSID,     Gene Name,     Location w/in Gene
1,        RS2,      G1,            Interior
1,        RS5,      G2,            Interior
1,        RS5,      G5,            Interior
1,        RS7,      G5,            Interior
1,        RS11,     G4,            Flanking
2,        RS23,     G6,            Flanking
```

Gene/Gene Models

Suffix: **model-archive.gene-gene**

This report is a tab-delimited text file containing the list of the gene-gene models that were produced by Biofilter based on the configuration options. The gene IDs that are given in this file are the internal

IDs used by Biofilter, and they can be translated into actual gene names through the use of the Gene Definition report, which will always be given when this output is produced.

The columns of the file are:

- **Gene 1, Gene 2**
  The first two columns are the internal indices of the genes for the given model. The indices can be converted to actual genes through the use of the Gene Definition report.
- **Implication Index**
  The third column lists the implication index of the model. See the Model Production section for details on how the implication index is calculated.

Example file:

```
Gene 1    Gene 2    Implication Index
0         1         2
1         2         1
```

Gene Definition

Suffix: **model-archive.genes**

This report is a tab-delimited text file containing the list of genes used in model production by Biofilter. This list will contain all genes that contain at least one SNP from the input data source, and the columns are as follows:

- **Gene Idx**
  Column 1 lists the gene index used in the Gene/Gene models. This is the internal index used by Biofilter, and this file gives the conversion from gene index to actual gene.
- **Name**
  The canonical name of the gene
- **True Begin, True End**
  Columns 3 and 4 list the canonical (NO-LD) boundaries of the gene, in base pair locations.
- **Eff. Begin, Eff. End**
  Columns 5 and 6 list the population specific boundaries of the gene. When using the NO-LD population, these columns should be identical to columns 3 and 4.
- **Groups**
  This column lists all of the groups that this gene has been found in. The groups are separated by source by a pipe (|). Within each source, the group ids are separated by either an exclamation point (!) for disease independent sources, or a tilde (~) for disease dependent sources. The first group in a source is always prefixed with the appropriate separator so the user can determine the correct category of source even if there is only one group from the source.
- **Aliases**
  Column 8 lists all of the aliases available for the gene, separated by a pipe (|). This list even includes potentially ambiguous aliases for this gene.

- **SNPs**
  Column 9 lists the indexes of the SNPs included in this gene, separated by a pipe (|). See Marker Info Report for details on converting a SNP index to an RSID.

Gene Idx	Name	True Begin	True End	Eff. Begin	Eff. End	Groups	Aliases	SNPs
0	G1	5	15	5	15	!1!2|~5	G1|R1	0|1
1	G2	25	35	25	35	!1|!3	G2|G23|R2	3
2	G3	45	55	45	55	!2	G23|G3|R3	4
3	G5	30	50	30	50		G5|R5	3|4

SNP/SNP Models

Suffix: **model-archive.snp-snp**

This file is a tab-delimited listing of all of the SNP/SNP models generated by Biofilter. The models are organized by implication index, and the file has the following columns:

- **SNP 1, SNP 2**
  The first two columns list the SNP indices of the SNPs in the given model. See Marker Info Report for information on converting the SNP index to an RSID.
- **Implication Index**
  This is the implication index of the model, which is a proxy for the strength of the model. See Model Production for details on the calculation of the implication index.

SNP 1	SNP 2	Implication Index
0	3	2
1	3	2
3	4	1

Gene Coverage Report

Suffix: **gene-coverage.csv**

This file is a comma-delimited report produced by the Error: Reference source not found command line option. The report contains the number of markers in each gene from platforms found using the COVERAGE_RS and COVERAGE_MAP configuration options. The user may limit the genes listed by passing a file of gene aliases to the GENE_REPORT configuration option. The columns of the file are as follows:

- **Gene**
  The first column of the file gives the canonical name of the gene.
- **Total**
  The second column gives the total number of markers contained in all files that were given. Each gene should have at least one marker associated with it in this file.
- ***All SNPs*** (Detailed Report only)
  This column gives a list of the SNP IDs that are associated with the gene, separated by a colon (:). Note that this is combined across all input files; the contribution from individual files is given later in the report.
- **<filename>**
  This column lists the total number of markers from the file associated with a gene. This column is repeated for each input platform given.

- ***<filename> SNPs*** (Detailed Report only)
  This column gives the list of the SNP IDs from the file that are associated with the given gene. The list of IDs is separated by a colon (:).

```
Gene,     Total,    test_map.txt,  test_snps.txt
G1,       3,        3,             2
G2,       1,        1,             1
G3,       1,        1,             1
G5,       2,        2,             2
```

Missing SNPs

Suffix: **missing-snps.txt**

Unlike the other output files, this is not a delimited file, and it is intended to be read by the operator. It lists all of the SNPs that were unable to be converted into a base pair location for use in generating models. This file is only produced when using the RS_SOURCE configuration option and when at least one of the input RS numbers could not be located in the LOKI databse.

```
The following SNPs were unable to be found in the
variations file:
     RS10046131
     RS10046212
     RS10046325
     RS10047718
     RS10047744
     RS1006093
```

Group List

Suffix: (output to screen – use redirection)

This report, produced with the "--groups" command line option, lists all of the available groups in the LOKI database that meet the search criteria. The output is tab-delimited text with the following columns:

- **ID**
  The first column is the internal ID of the group. This ID can be found in the Gene Definition report when generating gene/gene models.
- **Name**
  The second column lists the name of the group, as given by the originating data source.
- **Description**
  The final column is the description of the pathway, as given by the original data source.

An example of the output is given below. To save space, the description of the groups was truncated.

```
ID     Name     Description
89555  PF08744  NOZZLE is a transcription factor that (...)
94454  PF13807  This domain is found between two (...)
```

Gene List

Suffix: (output to screen – use redirection)

This report, produced with the "--genes" command line option, lists all of the available genes in the LOKI database that meet the search criteria. The output is tab-delimited text with the following columns:

- **Name**
  The first column lists the canonical name of the gene.
- **Alias**
  The second column lists an alias for this gene. Note that each line will have only a single gene/alias pair, so it is possible for a gene to appear on multiple lines.
- **Chrom**
  Column 3 lists the chromosome on which the gene is located.
- **Start, End**
  Columns 4 and 5 list the start and ending base pair location of the gene, respectively. Note that this start and stop position are for the given population, if available.
- **Description**
  Column 6 gives a description of the gene, as found in the LOKI database.
- **Alias Type**
  The final column lists the type of alias for the gene, which corresponds to the source of the information. As of Biofilter 1.1, the following alias types are available:
  - *Ensembl*
  - *Protein Accession ID*
  - *mRNA Accession ID*
  - *Entrez ID*
  - *Entrez Gene*
  - *Uniprot*
  - *Uniprot/SWISSPROT*

An example of the output is given below. For space considerations, the description of the gene is often truncated.

```
Name    Alias   Chrom Start     End       Description          Alias Type
ABCF1   ABCF1   6     30539169  30559308  ATP-binding (...)    Entrez Gene
BRCA1   BRCA1   17    41196311  41277499  breast cancer (...)  Entrez Gene
ABCF2   ABCF2   7     150904922 150924316 ATP-binding (...)    Entrez Gene
BRCA1P1 BRCA1P1 17    41320086  41321970  BRCA1 pseudogene 1   Entrez Gene
ABCF3   ABCF3   3     183903862 183911794 ATP-binding (...)    Entrez Gene
```

Binary Files

Some of the files listed above have the option of being printed in a strictly binary format to save space in the directory. This is especially helpful if the user is generating a large number of SNP/SNP or gene/gene models. To enable output in binary format, simply enable the BINARY_MODEL_ARCHIVE configuration option. When this option is enabled, the output is written without any delimiters or headers, and an integer is printed that indicates the number of lines

contained within the file. The reports that are available in a binary format are Gene/Gene Models and SNP/SNP Models.

WARNING: This feature has not been well tested, and may cause compatibility issues among different machines. If the only issue is long-term storage space, the user may be able to compress the text output sufficiently to avert the problem.