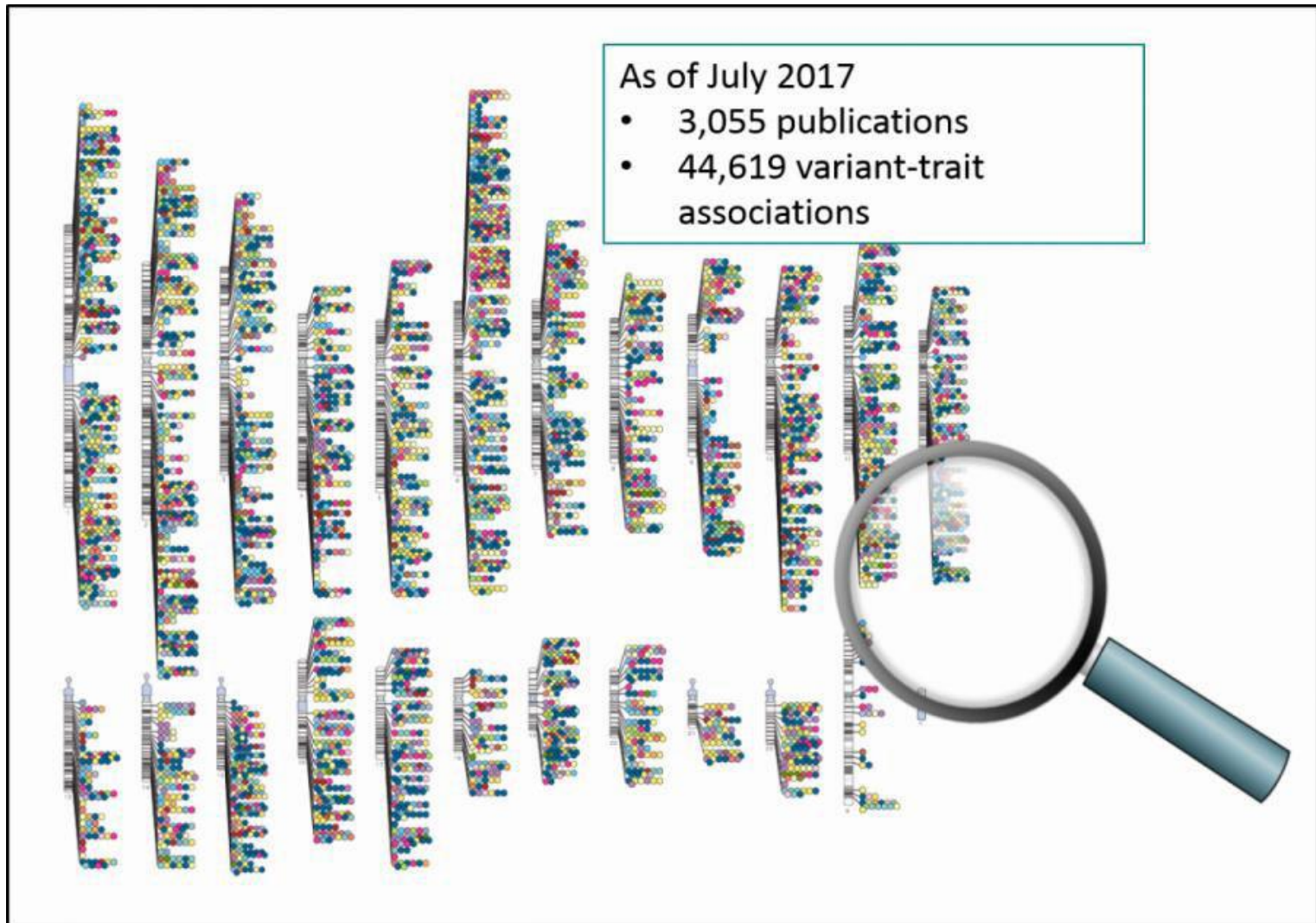


# **Gene-Gene /SNP-SNP Interaction: BIOFILTER**

GBIO00009

Archana Bhardwaj  
University of Liege



**The combinatorial problem of jointly analyzing the millions of genetic variations accessible by high-throughput genotyping**



# NIH Public Access

## Author Manuscript

*Pac Symp Biocomput.* Author manuscript; available in PMC 2010 April 26.

Published in final edited form as:

*Pac Symp Biocomput.* 2009 ; : 368–379.

## **Biofilter: A Knowledge-Integration System for the Multi-Locus Analysis of Genome-Wide Association Studies\***

**William S. Bush, Scott M. Dudek, and Marylyn D. Ritchie**

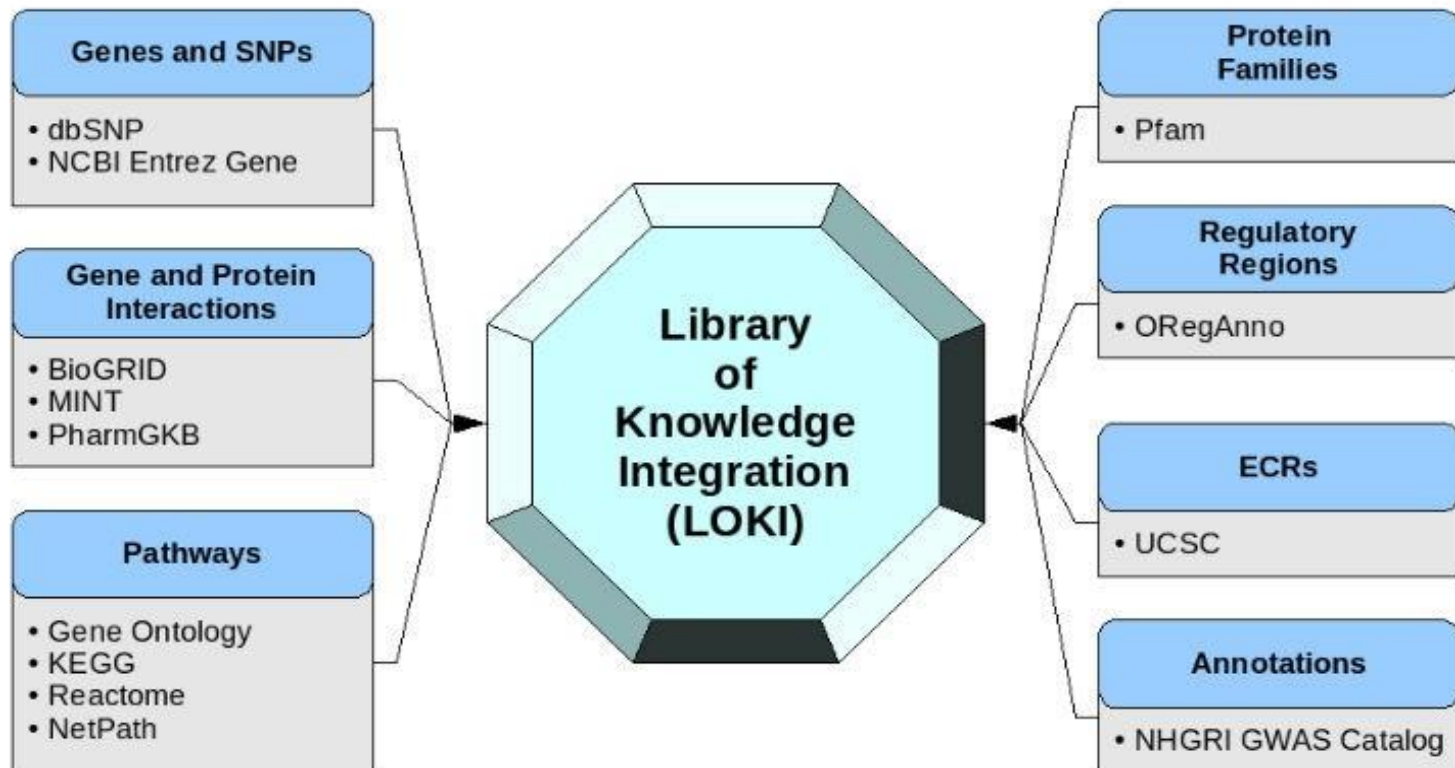
Center for Human Genetics Research, Vanderbilt University, Nashville, TN 37232, USA

### **Abstract**

Genome-wide association studies provide an unprecedented opportunity to identify combinations of genetic variants that contribute to disease susceptibility. The combinatorial problem of jointly analyzing the millions of genetic variations accessible by high-throughput genotyping technologies is a difficult challenge. One approach to reducing the search space of this variable selection problem is to assess specific combinations of genetic variations based on prior statistical and biological knowledge. In this work, we provide a systematic approach to integrate multiple public databases of gene groupings and sets of disease-related genes to produce multi-SNP models that have an established biological foundation. This approach yields a collection of models which can be tested statistically in genome-wide data, along with an ordinal quantity describing the number of data sources that support any given model. Using this knowledge-driven approach reduces the computational and statistical burden of large-scale interaction analysis while simultaneously providing a biological foundation for the relevance of any significant statistical result that is found.

- Biofilter uses publicly available databases to establish relationships between gene-products

## LOKI: Library of Knowledge Integration





# LOKI DB : dbSNP

NCBI

**dbSNP**  
Short Genetic Variations

dbVar ClinVar GaP PubMed Nucleotide Protein

Search small variations in dbSNP or large structural variations in dbVar

Search Entrez dbSNP for  Go

Have a question about dbSNP? Try searching the SNP FAQ Archive!  Go

**ANNOUNCEMENT**

**dbSNP and dbVar no longer accept submissions for non-human organism data. Please read more [here](#).**

**GENERAL**

RSS Feed

Contact Us

Organism Data

dbSNP Homepage

NCBI Variation Resources

Announcements

dbSNP Summary

FTP Download

SNP SUBMISSION

DOCUMENTATION

SEARCH

RELATED SITES

**Search by IDs on All Assemblies**

Note: rs# and ss# must be prefixed with "rs" or "ss", respectively (i.e. rs25, ss25)

ID:  Reference cluster ID(rs#)

Search Reset

**Submission Information**

- [By Submitter](#)
- [New Submitted Batches](#)
- [Method](#)
- [Population](#)
- [Publication](#)

**Batch**

- Enter List
  - [NCBI Assay ID\(ss\)](#)
  - [Reference SNP ID\(rs\)](#)

SNP

[Create alert](#) [Advanced](#)

Display Settings:  20 per page, Sorted by SNP\_ID

[Send to](#)

## Search results

Items: 1 to 20 of 336845724

[First](#) [Prev](#) Page  of 16842287 [Next](#) [Last](#)

☐ rs248 [*Homo sapiens*]

1.

ATTTTCTTTTCTTCCAAAGGAGGA [A/G] TTAACTACCCTCTGGACAATGTCC

Chromosome: 8:19953315

Gene: LPL (GeneView)

Functional Consequence: synonymous codon

Clinical significance: Likely benign

Validated: by 1000G,by cluster,by frequency,by hapmap,by submitter

Global MAF: A=0.0387/194

HGVS: NC\_000008.10:g.19810826G>A, NC\_000008.11:g.19953315G>A,  
NG\_008855.1:g.19245G>A, NM\_000237.2:c.435G>A, NP\_000228.1:p.Glu145

[PubMed](#) [View](#)

☐ rs268 [*Homo sapiens*]

2.

TGCAACAATCTGGGCTATGAGATCA [A/G] TAAAGTCAGAGCCAAAAGAAGCAGC

Chromosome: 8:19956018

Gene: LPL (GeneView)

Functional Consequence: missense

Allele Origin: A(germline)/G(germline)

Clinical significance: Pathogenic

Validated: by 1000G,by cluster,by frequency,by hapmap

Global MAF: G=0.0052/26

HGVS: NC\_000008.10:g.19813529A>G, NC\_000008.11:g.19956018A>G,  
NG\_008855.1:g.21948A>G, NM\_000237.2:c.953A>G, NP\_000228.1:p.Asn318Ser

14/11/2018

# LOKI DB : KEGG database

<http://www.genome.jp/kegg/pathway.html>



## KEGG PATHWAY Database

Wiring diagrams of molecular interactions, reactions and relations

hsa for “human”

Menu **PATHWAY** BRITE MODULE KO GENES LIGAND NETWORK DISEASE DRUG DBGET

Select prefix  
map

Enter keywords  
hsa  [Help](#)

[ [New pathway maps](#) | [Update history](#) ]

### Pathway Maps

**KEGG PATHWAY** is a collection of manually drawn [pathway maps](#) representing our knowledge on the molecular interaction, reaction and relation networks for:

- 1. Metabolism**  
[Global/overview](#) [Carbohydrate](#) [Energy](#) [Lipid](#) [Nucleotide](#) [Amino acid](#) [Other amino](#) [Glycan](#)  
[Cofactor/vitamin](#) [Terpenoid/PK](#) [Other secondary metabolite](#) [Xenobiotics](#) [Chemical structure](#)
- 2. Genetic Information Processing**
- 3. Environmental Information Processing**
- 4. Cellular Processes**
- 5. Organismal Systems**
- 6. Human Diseases**
- 7. Drug Development**

KEGG PATHWAY is a reference database for **Pathway Mapping**.

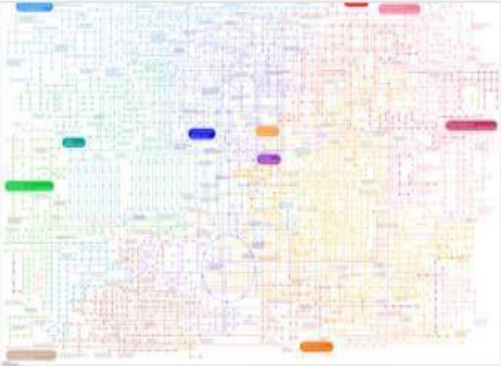
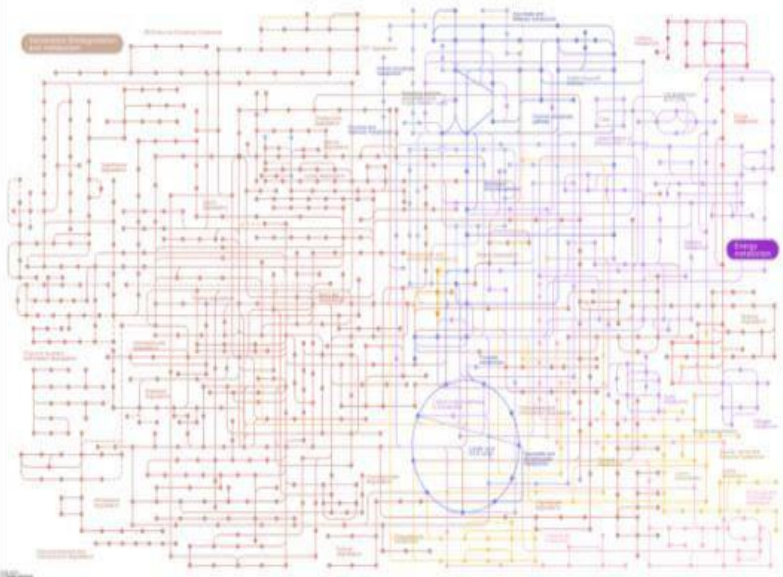
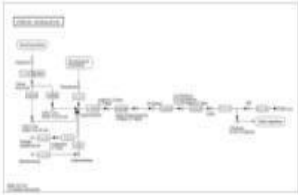
### Pathway Identifiers

Each pathway map is identified by the combination of 2-4 letter prefix code and 5 digit number (see [KEGG Identifier](#)). The prefix has the following meaning:

map	manually drawn reference pathway
ko	reference pathway highlighting KOs
ec	reference metabolic pathway highlighting EC numbers
rn	reference metabolic pathway highlighting reactions
<org>	organism-specific pathway generated by converting KOs to gene identifiers

and the numbers starting with the following:

01	global map (lines linked to KOs)
----	----------------------------------

map01100		Metabolic pathways	...15983 (kshB), 1.14.13.142, R09860 R09885 K16047 ( <i>hsaA</i> ), K16048 ( <i>hsaB</i> ), 1.14.14.12, R09819 K16049 (hs...	Neomycin, kanamycin and gentamicin biosynthesis Glycosaminoglycan biosynthesis - chondroitin sulfa...
map01120		Microbial metabolism in diverse environments	...99.5, R00295 3.12.1.1, 3.12.1.1, R01930 K08352 ( <i>phsA</i> ), K08353 ( <i>phsB</i> ), K08354 ( <i>phsC</i> ), 1.8.5.5, R10149...	Vitamine B6 metabolism Xylene degradation Glyoxylate and dicarboxylate metabolism Aminobenzoate ...
map00984		Steroid degradation	...125A), 1.14.13.141, R11357 R09885 R09885 K16047 ( <i>hsaA</i> ), K16048 ( <i>hsaB</i> ), 1.14.14.12, R09819 K16049 (hs...	STEROID DEGRADATION Cholest-4-en-3-one 1.1.3.6 1.14.13.141 (25S)-3-Oxo-cholest-4-en-26-oate 9alpha-...



# LOKI DB : BioGRID Database

**BioGRID** 3.4

home help wiki tools contribute stats downloads partners about us

## Welcome to the Biological General Repository for Interaction Datasets

BioGRID is an interaction repository with data compiled through comprehensive curation efforts. Our current index is version **3.4.155** and searches **63,959** publications for **1,507,991** protein and genetic interactions, **27,785** chemical associations and **38,559** post translational modifications from major model organism species. All data are **freely** provided via our search index and available for download in standardized formats.

INTERACTION STATISTICS

LATEST DOWNLOADS

### Search the BioGRID

Search by identifiers, keywords, and gene names...

p53

Homo sapiens

SUBMIT GENE SEARCH Q

Advanced Search

Search Tips

Featured Datasets

By Gene

By Publication

## AREAS OF INTEREST TO HELP YOU GET STARTED



### Build and Download Interaction Datasets

Create custom interaction datasets by protein or by publication. You can also download our entire dataset in a wide variety of standard formats.



### Online Tools and Resources

We've developed tools that make use of BioGRID data. Check out the list of tools to see if we can help you work with our data.



### Link To Us or Submit Interactions

Send us your datasets or link to the BioGRID directly from your own website or database. Full details on how to contribute are available here.



### View Our Interaction Statistics

Find out how many organisms, proteins, publications, and interactions are available in the current release of the BioGRID.

## BIOGRID FUNDING AND PARTNERS



more partners



# Result Summary

Gene / Identifier Search

p53

Homo sapiens

GO

## TP53

*Homo sapiens*

BCC7, LFS1, P53, TRP53

tumor protein p53

UBI NEDD FAT10 SUMO

GO Process (61)

GO Function (25)

GO Component (14)

### EXTERNAL DATABASE LINKOUTS

HGNC | OMIM | VEGA | Entrez Gene | RefSeq | UniprotKB | Ensembl | HPRD

Download 3000 Published Interactions For This Protein

### Stats & Options

### Current Statistics

Publications: 1101

High Throughput

Low Throughput

514 (18%)

2877 Physical Interactions

2363 (82%)

104 (81%)

128 Genetic Interactions

24 (19%)

### Search Filters

Customize how your results are displayed...

No Filter: Show All Associations



Switch View: Interactors (1034) Interactions (3005) Network Chemicals (2) PTM Sites (4)

Displaying 1 - 300 of 1034 total unique interactors

< Previous | 1 2 3 4 | Next >

Sort By: [Evidence] [Alphabetical]

**MDM2** | ACTFS, HDMX, hdm2

MDM2 proto-oncogene, E3 ubiquitin protein ligase

UBI NEDD FAT10 SUMO

413 1

[details]

**EP300** | RP1-85F18.1, KAT3B, RSTS2, p300

E1A binding protein p300

UBI SUMO

85 1

[details]

# Download Biofilter

- Go to following link

**<https://ritchielab.org/software/biofilter-download-1>**

- Download Biofilter 2.4.1

# Use of Biofilter software (1)

- We can annotate genomic location or region based data, such as results from association studies, or CNV analyses, with relevant biological knowledge for deeper interpretation.
- We can filter genomic location or region based data on biological criteria, such as filtering a series SNPs to retain only SNPs present in specific genes within specific pathways of interest.

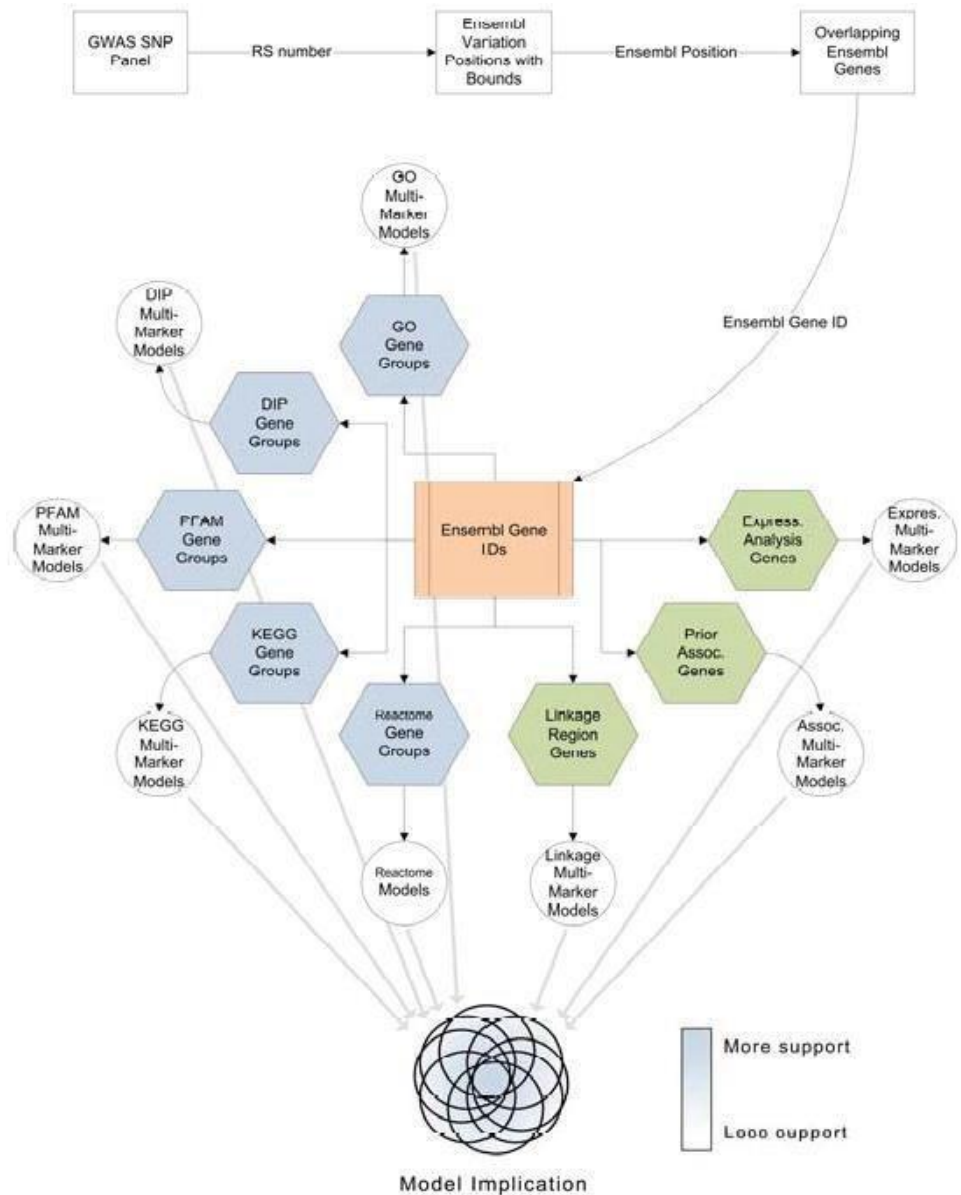


# Use of Biofilter software (2)

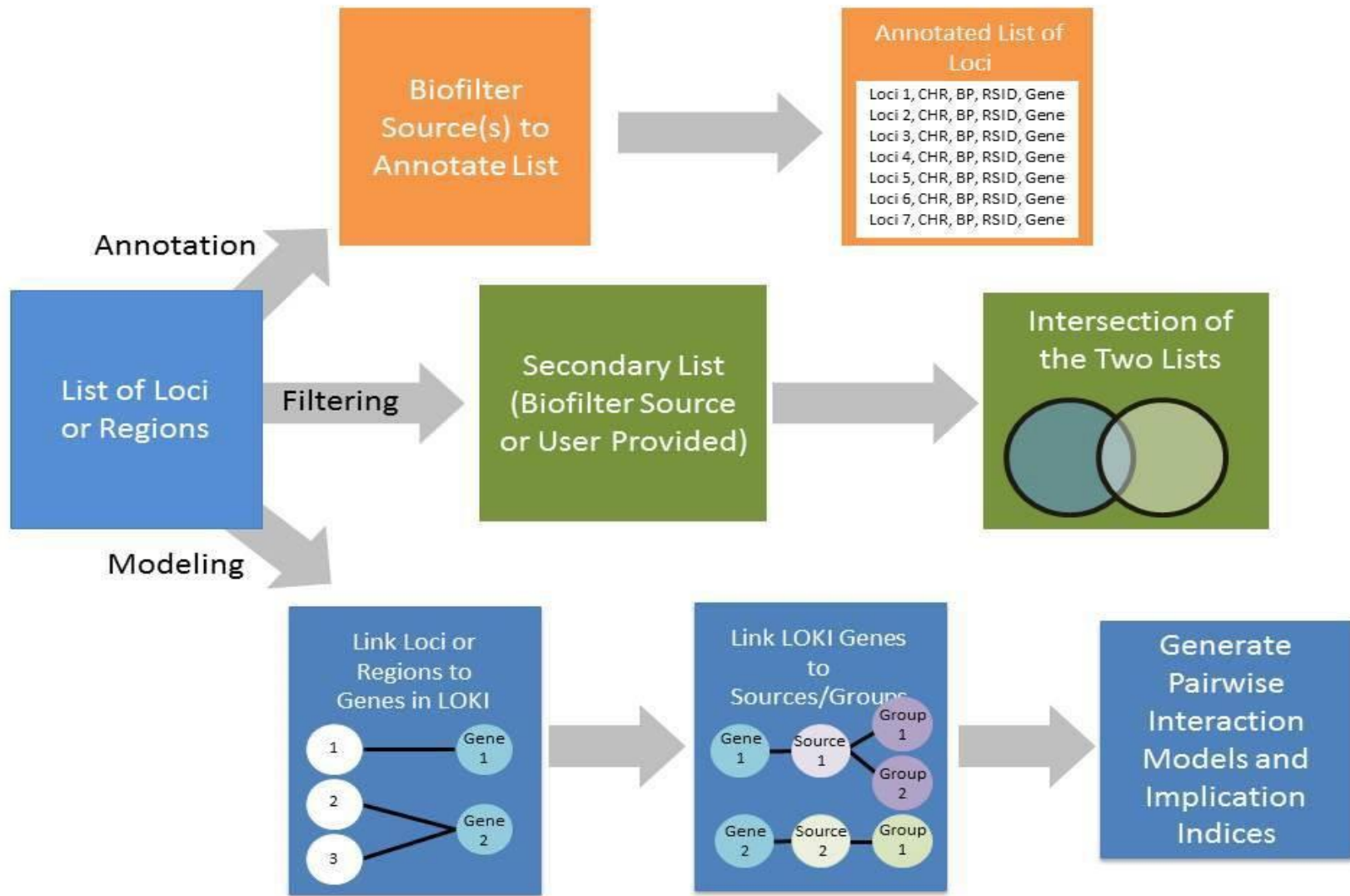
- Biofilter allows researchers to annotate and/or filter data as well generate gene-gene interaction models based on existing biological knowledge.
- We can generate Predictive Models for gene-gene, SNP-SNP, or CNV-CNV interactions based on biological information, with priority for models to be tested based on biological relevance, thus narrowing the search space and reducing multiple hypothesis-testing.

# Biofilter : Overview

- **GWAS platform SNPs** are mapped to Ensembl gene Ids.
- **Multi-marker models** are generated from SNPs within **knowledge-related genes**.
- **Derived models** are overlaid to assess overall model implication.









# Biofilter : Three Analysis mode



# Biofilter Data types

## *Data Types*

Biofilter can work with and understand the relationships between six basic types of data:

<b>SNP</b>		Specified by an RS number, i.e. "rs1234". Used to refer to a known and documented SNP whose position can be retrieved from the knowledge database.
<b>Position</b>		Specified by a chromosome and basepair location, i.e. "chr1:234". Used to refer to any single genomic location, such as a single nucleotide polymorphism (SNP), single nucleotide variation (SNV), rare variant, or any other position of interest.
<b>Region</b>		Specified by a chromosome and basepair range, i.e. "chr1:234-567". Used to refer to any genomic region, such as a copy number variation (CNV), insertion/deletion (indel), gene coding region, evolutionarily conserved region (ECR), functional region, regulatory region, or any other region of interest.
<b>Gene</b>		Specified by a name or other identifier, i.e. "A1BG" or "ENSG00000121410". Used to refer to a known and documented gene, whose genomic region and associations with any pathways, interactions or other groups can be retrieved from the knowledge database.
<b>Group</b>		Specified by a name or other identifier, i.e. "lipid metabolic process" or "GO:0006629". Used to refer to a known and documented pathway, ontological group, protein interaction, protein family, or any other grouping of genes, proteins or genomic regions that was provided by one of the external data sources.
<b>Source</b>		Specified by name, i.e. "GO". Used to refer to a specific external data source.



# Biofilter : Filtering mode

- Given any combination of input data, Biofilter can cross-reference the input data using the relationships stored in the knowledge database to generate a filtered dataset of any supported type (or types).
- For example, a user can provide a list of SNPs (such as those covered by a genotyping platform) and a list of genes (such as those thought to be related to a particular phenotype) and request a filtered set of SNPs. Biofilter will use LOKI's knowledge of SNP positions and gene regions to filter the provided
- SNP list, removing all those that are not located within any of the provided genes.

# Biofilter : Annotation mode

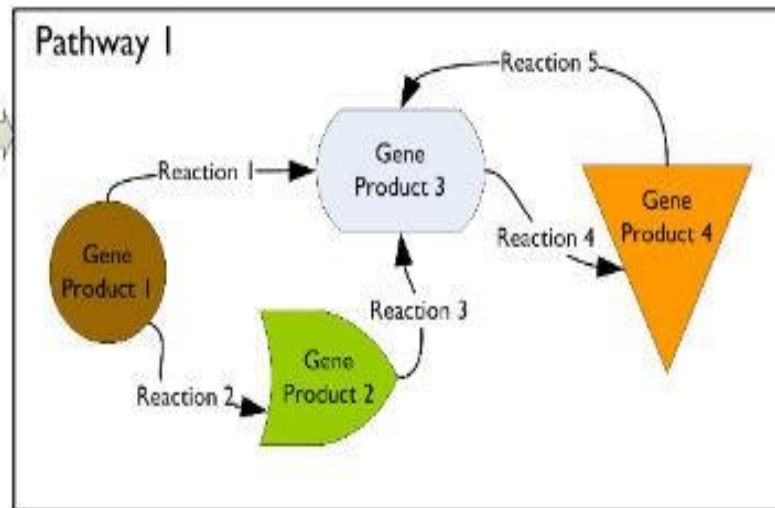
- The annotations are based on the relationships stored in the knowledge database; unlike filtering, any data which cannot be annotated as requested (such as a SNP which is not located within any gene) will still be included in the output, with the annotation columns of the output simply left blank.
- For example, a list of SNPs can be annotated with positions to generate a new list of all the same SNPs, but with extra columns containing the chromosome and genomic position for each SNP (if any). Any SNP with multiple known positions will be repeated, and any SNP with no known position will have blank in the added columns

# Biofilter : Annotation mode

## Single Locus Statistical Results

SNP 1, Rs101841,  $p = 0.000163$   
SNP 2, Rs182645,  $p = 0.000268$   
SNP 3, Rs23876,  $p = 0.00324$   
SNP 4, Rs378645,  $p = 0.004354$   
SNP 5, Rs37564,  $p = 0.02341$   
SNP 6, Rs8751,  $p = 0.03412$   
SNP 7, Rs86745,  $p = 0.03685$   
SNP 8, Rs41254,  $p = 0.04675$

## Biofilter Analysis



## Annotated Statistical Results

### Results in the Same Gene

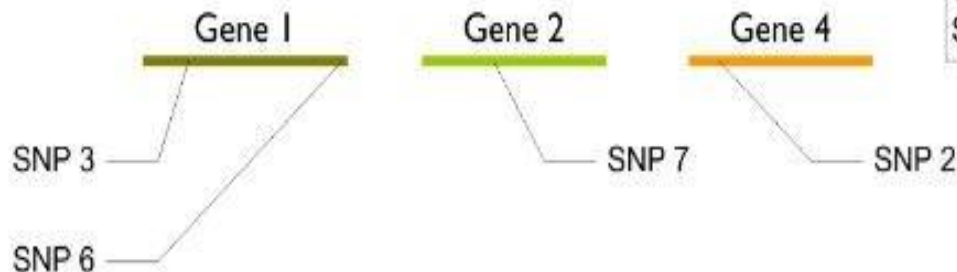
SNP 3, Rs23876,  $p = 0.00324$   
SNP 6, Rs8751,  $p = 0.03412$

### Results in the Same Pathway

SNP 2, Rs182645,  $p = 0.000268$   
SNP 3, Rs23876,  $p = 0.00324$   
SNP 6, Rs8751,  $p = 0.03412$   
SNP 7, Rs86745,  $p = 0.03685$

### Results with Biological Interaction

SNP 3, Rs23876,  $p = 0.00324$   
SNP 6, Rs8751,  $p = 0.03412$   
SNP 7, Rs86745,  $p = 0.03685$



# Biofilter : Model analysis mode(1)

- The last of Biofilter's primary analysis modes is a little different from filtering and annotation.
- In addition to simply cross-referencing any given data with the other available prior knowledge, Biofilter can also search for repeated patterns within the prior knowledge which might indicate the potential for important interactions between SNPs or genes.

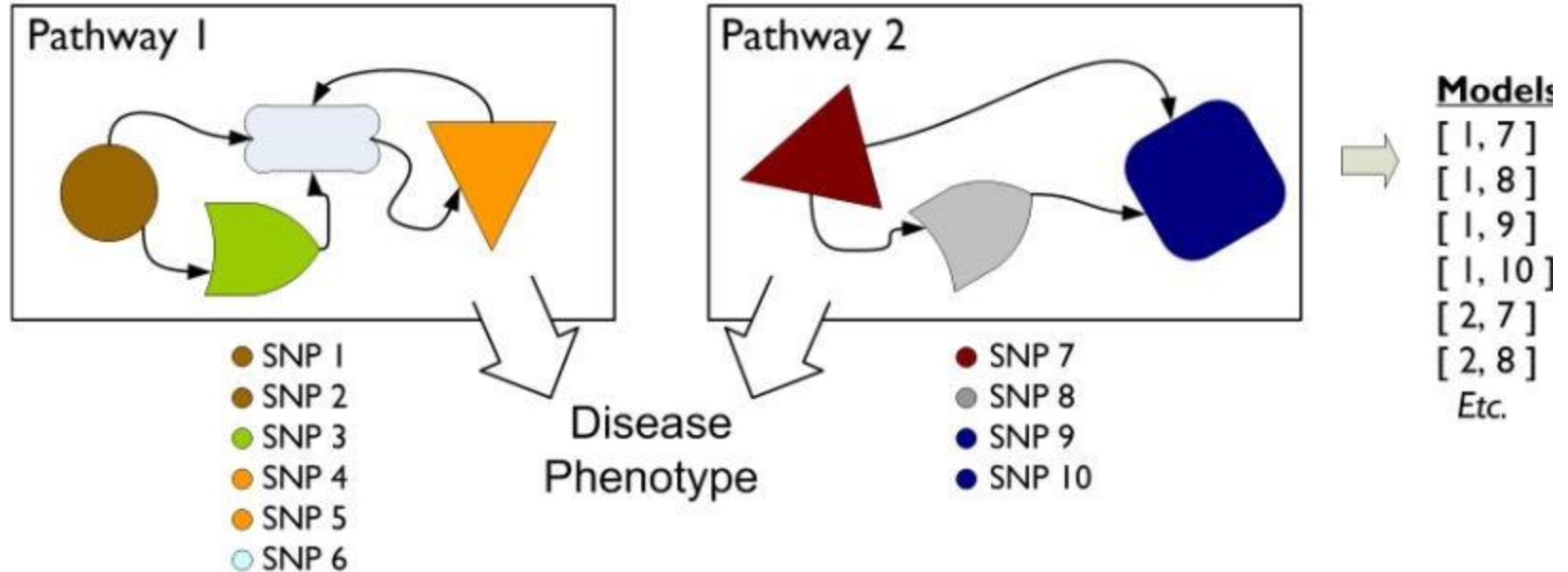


# **Biofilter : Model analysis mode(2)**

**The key idea behind this analysis is that If the same two genes appear together in more than one grouping, they're likely to have an important biological relationship; if they appear in multiple groups from several independent sources, then they're even more likely to be biologically related in some way.**

# Biofilter : Model analysis mode (3)

• Biofilter has access to thousands of such groupings and can analyze all of them to identify the pairs of genes or SNPs appearing together in the greatest number of groupings and the widest array of original data sources. These pairs can then be tested for significance within a research dataset, avoiding the prohibitive computational and multiple testing Burden of an exhaustive pairwise analysis.



# Compiling Prior Knowledge: Loki.db

- The LOKI prior knowledge database must be generated before Biofilter can be used. This is done with the “loki-build.py” script which was installed along with Biofilter. There are several options for this utility which are detailed below, but to get started, you just need “--knowledge” and “--update”:

*loki-build.py --verbose --knowledge loki.db --update*

- This will download and process the bulk data files from all supported knowledge sources, storing the result in the file “loki.db” (which we recommend naming after the current date, such as “loki20140521.db”).

# Updating Prior Knowledge: Loki.db (1)

▪ ***--update***     ***Arguments: [source] [...]***     ***Default: all***

Instructs the build script to process the bulk data from the specified sources and update their representation in the knowledge database. If no sources are specified, all supported sources will be updated.

▪ ***--update-except***     ***Arguments: [source] [...]***     ***Default: none***

Similar to “--update” but with the opposite meaning for the specified sources: all supported sources will be updated **except for the ones specified.**     If no sources are specified, none are excluded, and all supported sources are updated.

▪ ***--option***     ***Arguments: <source> <options>***     ***Default: none***

Passes additional options to the specified source loader module. The options string must be of the form “option1=value,option2=value” for any number of options and values. Supported options and values for each source can be shown with “--list-sources”.

# Updating Prior Knowledge : Loki.db (2)

▪ *--force-update* *Argument: none*

The build script will normally only update from a sources if it detects that an update is necessary, either because new data files have been downloaded from the source or because the source's loader module code has been updated. With this option, the build script will update all specified sources, even if it believes no update is necessary.



# LD Profiles : GWAS information

- Biofilter and LOKI allow for gene regions to be adjusted by the linkage disequilibrium (LD) patterns in a given population.
- When comparing a known gene region to any other region or position (such as CNVs or SNPs), areas in high LD with a gene can be considered part of the gene, even if the region lies outside of the gene's canonical boundaries.

**This step require use of additional tool**

# Biofilter : Command lines vs Configuration

- Biofilter can be run from a command-line terminal by executing

*biofilter.py or python biofilter.py*

- All options can either be provided directly on the command line

**biofilter.py --option-name**

- configuration files could be given as input such as

*biofilter.py analysis.config*

# Biofilter : Configuration file

Input files:

input1	input2
#snp	#snp
rs9	rs14
rs11	rs15
rs12	rs16
rs13	rs17
rs14	rs18
rs15	rs19
rs16	

Configuration:

```
KNOWLEDGE test.db
SNP_FILE input1
SNP_FILE input2
FILTER snp
```

▪ ***biofilter.py test.config***

# Biofilter : Command lines vs Configuration

- Options on the command line are lower-case, start with two dashes and may contain single dashes to separate words (such as “-- snp-file”),
- while in a configuration file the same option would be in upper-case, contain no dashes and instead use underscores to separate words (i.e. “SNP\_FILE”).
- Many command line options also have alternative shorthand versions of one or a few letters, such as “-s” for “--snp-file” and “--aag” for “--allow-ambiguous-genes”.

# Configuration Options

- ***--help / HELP***

Displays the program usage and immediately exits.

- ***--version / VERSION***

Displays the software versions and immediately exits. Note that Biofilter is built upon LOKI and SQLite, each of which will also report their own software versions.

- ***--report-configuration / REPORT\_CONFIGURATION***

Argument: [yes/no]     Default: no

Generates a Biofilter configuration file which specifies the current effective value of all program options, including any default options which were not overridden.



# Prior Knowledge Options

- ***--knowledge / KNOWLEDGE***

Argument: <file>      Default: *none*

- ***--report-genome-build / REPORT\_GENOME\_BUILD***

Argument: [yes/no]      Default: *yes*

- ***--report-gene-name-stats / REPORT\_GENE\_NAME\_STATS***

Argument: [yes/no]      Default: *no*

- ***--report-group-name-stats / REPORT\_GROUP\_NAME\_STATS***

Argument: [yes/no]      Default: *no*

- ***--allow-unvalidated-snp-positions / ALLOW\_UNVALIDATED\_SNP\_POSITIONS***

Argument: [yes/no]      Default: *yes*

- ***--allow-ambiguous-snps / ALLOW\_AMBIGUOUS\_SNPS***

# Primary Input Data Options

- ***--snp / SNP***

Arguments: <snp> [snp] [...] Default: *none*

- ***--snp-file / SNP\_FILE***

Arguments: <file> [file] [...] Default: *none*

- ***--position / POSITION***

Arguments: <position> [position] [...] Default: *none*

- ***--position-file / POSITION\_FILE***

Arguments: <file> [file] [...] Default: *none*

- ***-region / REGION***

Arguments: <region> [region] [...] Default: *none*

- ***--region-file / REGION\_FILE***

Arguments: <file> [file] [...] Default: *none*

# Output Options : Mode of analysis

## ***--filter / FILTER***

Argument: <type> [type] [...] Default: *none*

Perform a filtering analysis which outputs the specified type

## ***--annotate / ANNOTATE***

Argument: <type> [type] [...] [:] <type> [type] [...] Default: *none*

## ***--model / MODEL***

Argument: <type> [type] [...] [:] [type] [...] Default: *none*

# Filter mode : search SNPs that correspond to a list of genes

• input1

input2

#snp

#gene

rs11

ACE

rs12

rs13

rs14

rs15

rs16

• Test.config

KNOWLEDGE test.db

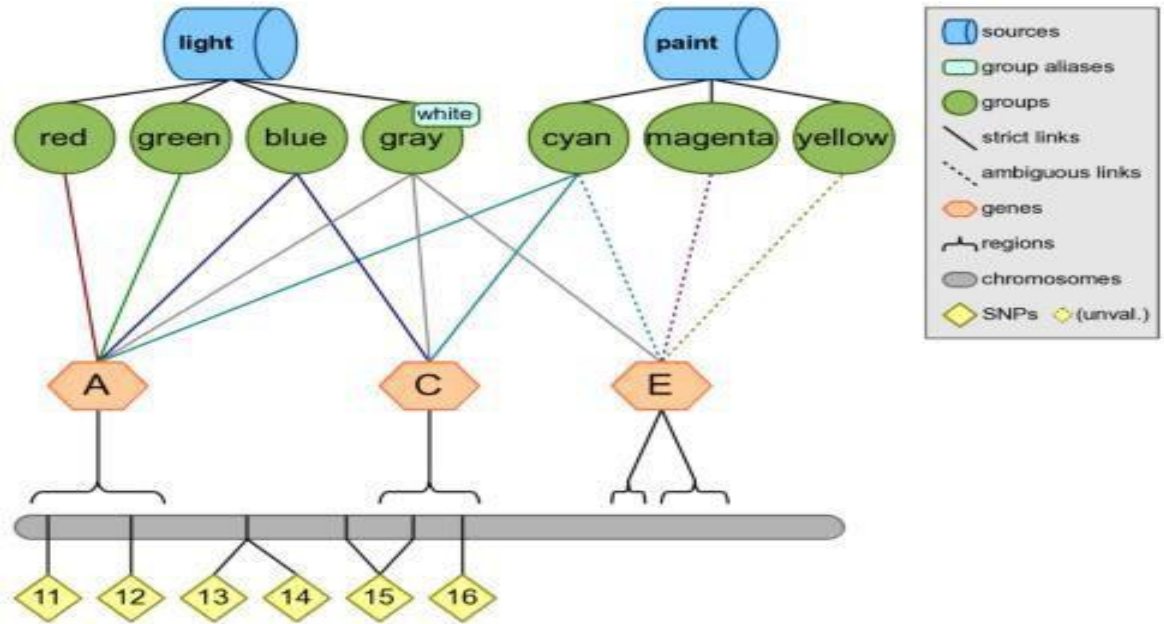
SNP\_FILE input1

GENE\_FILE input2

FILTER snp

• run “ biofilter.py Test.config”

• What is expected output ???



Can you make inference by looking

# Annotation mode : a SNP with gene region information

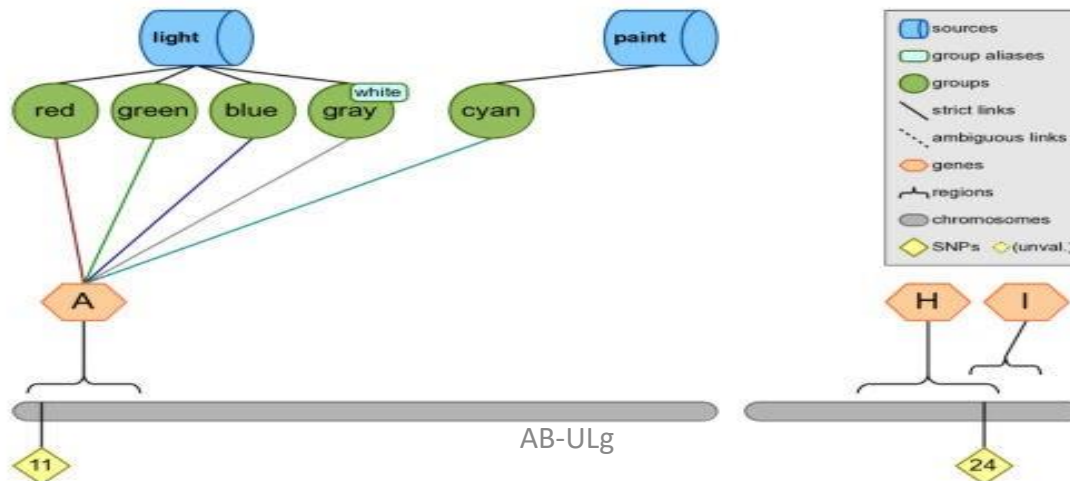
## •Test.config

```
KNOWLEDGE test.db SNP rs11 rs24 rs99
ANNOTATE snp region
```

## •Biofilter.py test.config

## •Output

#snp	chr	region	start	stop
rs11	1	A	8	22
rs24	2	H	22	42
rs24	2	I	38	48
rs99				





# Pair wise Gene-Gene and SNP-SNP interaction

## Step 1

Map the input list of SNPs to genes within Biofilter.

## Step 2

Connect, pairwise, the genes that contain SNPs in the input list of SNPs.

## Step 3

Break down the gene-gene models into all pairwise combinations of SNPs across the genes within sources

# Step 1 : Pair wise Gene-Gene and SNP-SNP interaction

- we will use all of the SNPs on the first chromosome.
- Test.config

KNOWLEDGE test.db

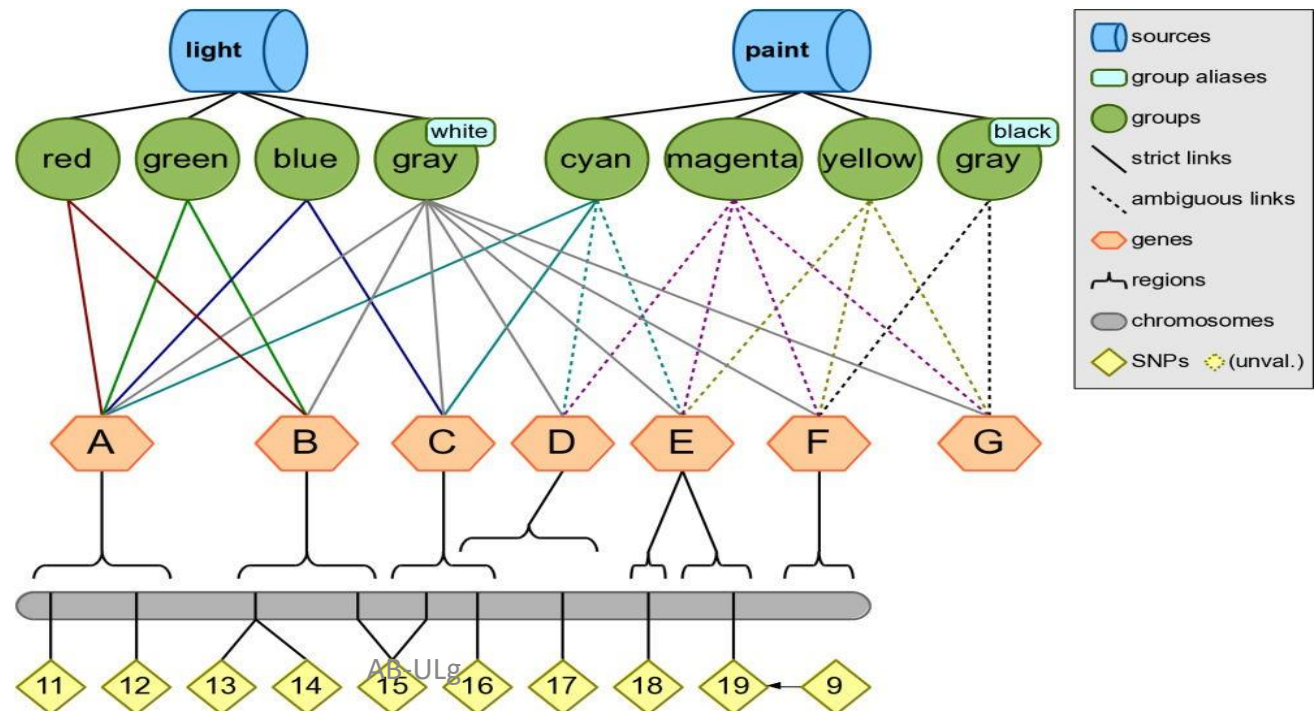
SNP 11 12 13 14 15 16 17 18 19

FILTER gene

• Output:

#gene

A B C D E



# Step 2 : Connect, pairwise, the genes that contain SNPs in the input list of SNPs.

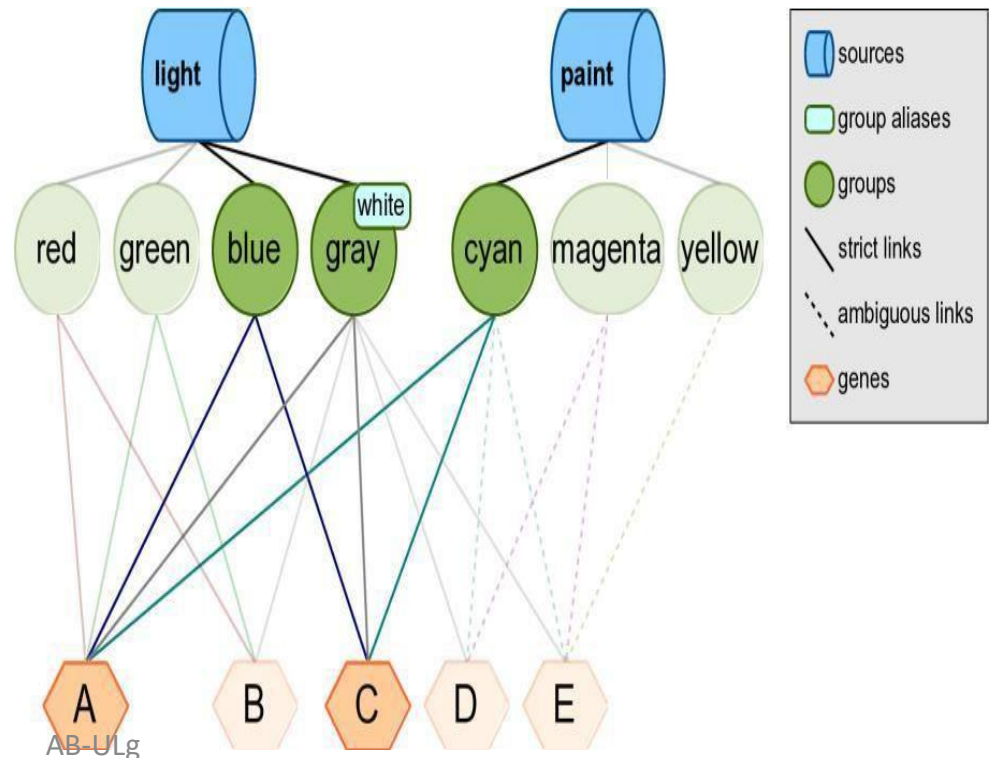
- Test.config

```
KNOWLEDGE test.db
GENE A B C D E
MODEL gene
```

- biofilter.py test.config

- output

#gene1	#gene2	score
A	C	2-3



# Step 3 : Break down the gene-gene models into all pairwise combinations of SNPs

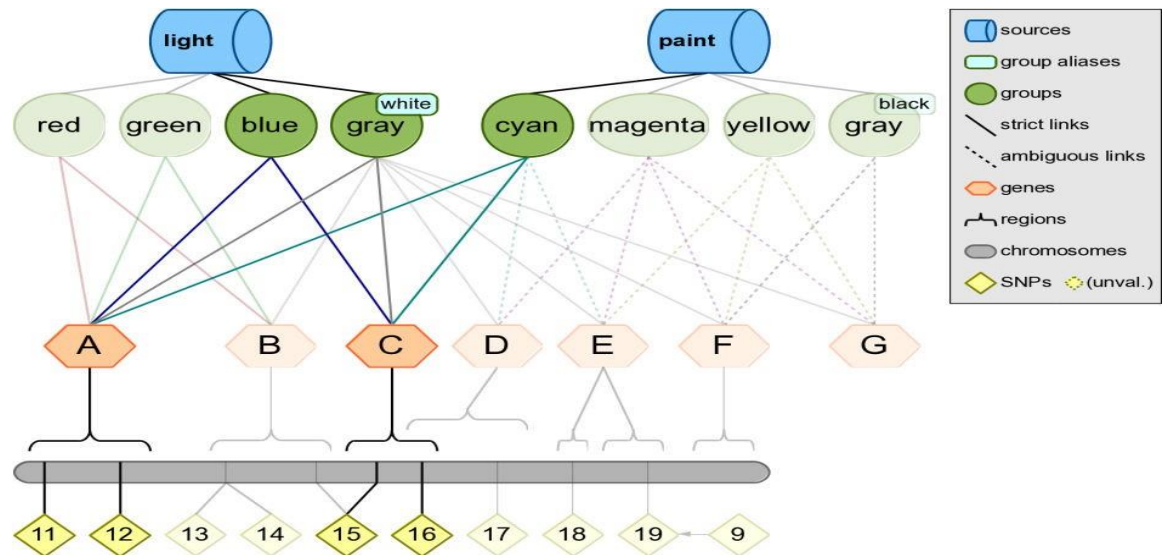
Configuration:

- biofilter.py test.config

```
KNOWLEDGE test.db  
SOURCE light paint  
MODEL snp
```

Output:

```
#snp1  snp2  score(src-grp)  
rs11   rs15   2-3  
rs11   rs16   2-3  
rs12   rs15   2-3  
rs12   rs16   2-3
```



# **LD Profiles : GWAS information**

**Biofilter and LOKI allow for gene regions to be adjusted by the linkage disequilibrium (LD) patterns in a given population.**

**When comparing a known gene region to any other region or position (such as CNVs or SNPs), areas in high LD with a gene can be considered part of the gene, even if the region lies outside of the gene's canonical boundaries.**

**This step require use of additional tool**



# Get the genes statistic of loki.db

Open terminal and type

```
biofilter.py --knowledge loki.db --report-gene-name-stats yes
```

This indicates number of genes belongs to each category

#type	names	unique	ambiguous
symbol	117857	115238	2619
entrez_gid		81664	81664 0
uniprot_pid		32983	32668 315
ensembl_gid		75453	75369 84
pharmgkb_gid		26650	26650 0
refseq_gid		189201	189201 0
refseq_pid		117448	117448 0
ensembl_pid		41732	41732 0
hgnc_id	41163	41163	0
mim_id	17130	17130	0
vega_id	19138	19123	15
mirbase_id		1879	1879 0
unigene_gid		29152	27997 1155

# Get the groups statistic of loki.db

Open terminal and type

**biofilter.py --knowledge loki.db --report-group-name-stats yes**

**This indicates no of entries represented by each group**

#type	names	unique	ambiguous
biogrid_id		371104	371104 0
go_id	44957	44957	0
ontology	44957	44957	0
kegg_id	323	323	0
pathway	2605	2588	17
netpath_id		28	28 0
oreganno	23393	23393	0
pfam_id	16718	16718	0
proteinfamily		32501	32164 337
pharmgkb_id		108	108 0
reactome_id		2163	2163 0
ucsc_ecr	77858	77858	0

# Comparison of Two SNPs list

**Download snp1 and snp2 from course website**  
**Create file name test1.config**

```
KNOWLEDGE loki.db  
SNP_FILE snp1  
SNP_FILE snp2  
FILTER snp
```

**Run biofilter.py test1.config**

**you will get two output files named as**  
**biofilter1.log , biofilter1.snp**

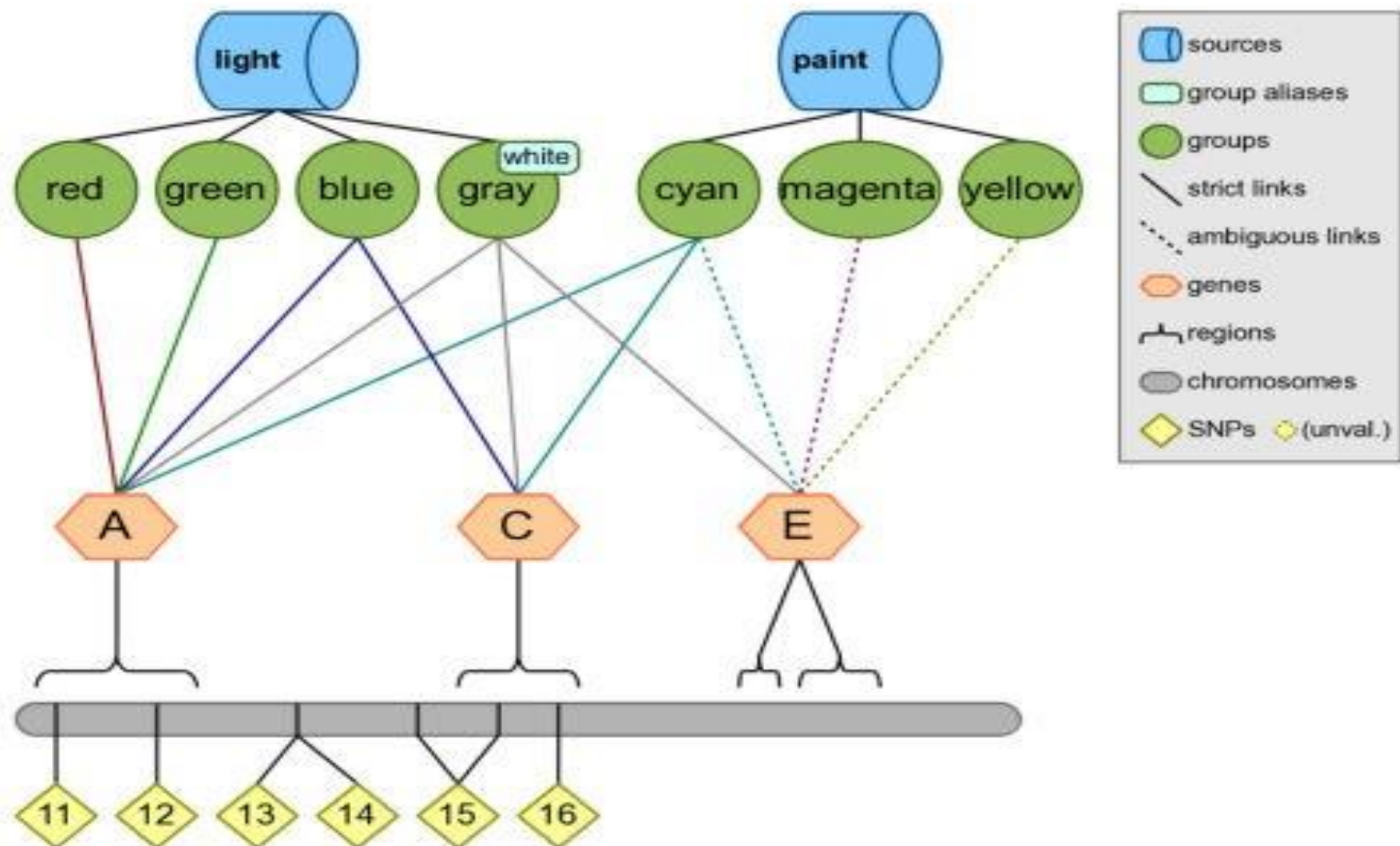
# biofilter1.log

```
loading knowledge database file '/usr/local/bin/loki.db' ...  
... OK  
knowledge database genome build: GRCh38 / UCSC hg38  
adding to main SNP filter ...  
... OK: added 4 SNPs (1 RS#s merged) reducing main SNP  
filter ...  
... OK: kept 1 SNPs (3 dropped, 0 RS#s merged) writing  
'snp' filter to 'biofilter.snp' ...  
... OK: 1 results
```

## biofilter1.snp

```
#snp rs62653571
```

# Let us find the SNPs falling on genes (1)



# Let us find the SNPs falling on genes (2)

Create file name test2.config , define snp1 as input

```
KNOWLEDGE loki.db
SNP_FILE snp1
GENE_FILE gene
FILTER snp
```

← Type of filter

Run as **biofilter.py test2.config**

As a result you will get two files :

**biofilter.log**

**biofilter.snp**



# biofilter.log

```
loading knowledge database file '/usr/local/bin/loki.db' ...  
... OK  
knowledge database genome build: GRCh38 / UCSC hg38 adding to main  
SNP filter ...  
... OK: added 4 SNPs (1 RS#s merged) adding to main gene filter ...  
... OK: added 2 genes  
writing 'snp' filter to 'biofilter.snp' ...  
... OK: 4 results
```

## biofilter.snp

```
#snp rs62653571  
rs2071569  
rs2075596  
rs6533526
```



# Let us find the groups contains specific “regions”

create regions.config which contain group  
information “R-HSA-5083635”

```
KNOWLEDGE loki.db
GROUP R-HSA-5083635
FILTER region
```

Run as `biofilter.py regions.config`

As a result you will get two files :

`biofilter.log`

`biofilter.region`

# biofilter.region :It will consist of all regions belongs to group

#chr	region	start	stop
X	CFP	47624213	47630305
15	THBS1	39581079	39598918
6	THBS2	169215780	169254114
5	SEMA5A	9035026	9546121
1	ADAMTS4	161189725	161199080
4	ADAMTS3	72280969	72568799
5	ADAMTS2	179110851	179345430
21	ADAMTS1	26836287	26845409
9	ADAMTSL2		133532164 133575519
4	SPON2	1166932	1208962
11	SPON1	13962637	14268133
9	ADAMTS13		133414339 133459403
11	ADAMTS8	130404923	130428993
21	ADAMTS5	26917912	26967120
15	ADAMTS7	78759203	78811464
5	ADAMTS6	65148736	65482014
13	B3GLCT	31199975	31332276
7	SSPO	149776042	149833965

# Output a list of all genes within a data source

Create `group.config` which contain source information

```
KNOWLEDGE loki.db  
SOURCE biogrid pfam  
FILTER gene
```

Run as `biofilter.py group.config`

As a result you will get two files : `biofilter.log`  
`biofilter.gene`

**biofilter.gene : All genes belong to group PFAM and biogrid**

```
#gene TRIM54 HDGF EXOSC10 JMJD6 MC4R NEDD8 COPS7A  
COG1  
SIAH1 ..... SO ON
```

**Can you count the number of genes belong to “PFAM” only ????**

# Let us find genes associated with a pathway or group

Create `tt.config` which contains the Genes detail

```
KNOWLEDGE loki.db  
GENE THSD7A COG8 UBC  
FILTER gene snp region group source
```

Run as `biofilter.py tt.config`

As a result , you will get two files `biofilter.log`  
`biofilter.snp.region.group.source`

## **biofilter.log**

```
loading knowledge database file '/usr/local/bin/loki.db' ...  
... OK  
knowledge database genome build: GRCh38 / UCSC hg38 adding  
to main gene filter ...  
... OK: added 3 genes  
writing 'gene snp region group source' filter to 'biofilter.gene-snp-  
region-group-source' ...  
... OK: 1668612 result
```



**Open file and check information types**



# biofilter.snp.region.group.source : It consist of following entries (top 15 lines)

#gene	snp	chr	region	start	stop	group	source
THSD7A	rs983143041	7	THSD7A	11370435	11832198	biogrid:612143	biogrid
THSD7A	rs1015982900	7	THSD7A	11370435	11832198	biogrid:612143	biogrid
THSD7A	rs962840243	7	THSD7A	11370435	11832198	biogrid:612143	biogrid
THSD7A	rs558399301	7	THSD7A	11370435	11832198	biogrid:612143	biogrid
THSD7A	rs539304291	7	THSD7A	11370435	11832198	biogrid:612143	biogrid
THSD7A	rs974293917	7	THSD7A	11370435	11832198	biogrid:612143	biogrid
THSD7A	rs921514204	7	THSD7A	11370435	11832198	biogrid:612143	biogrid
THSD7A	rs571860735	7	THSD7A	11370435	11832198	biogrid:612143	biogrid
THSD7A	rs932952501	7	THSD7A	11370435	11832198	biogrid:612143	biogrid
THSD7A	rs750203807	7	THSD7A	11370435	11832198	biogrid:612143	biogrid
THSD7A	rs945700395	7	THSD7A	11370435	11832198	biogrid:612143	biogrid
THSD7A	rs1042762941	7	THSD7A	11370435	11832198	biogrid:612143	biogrid
THSD7A	rs114612380	7	THSD7A	11370435	11832198	biogrid:612143	biogrid
THSD7A	rs370567942	7	THSD7A	11370435	11832198	biogrid:612143	biogrid
THSD7A	rs912527472	7	THSD7A	11370435	11832198	biogrid:612143	biogrid
THSD7A	rs934667503	7	THSD7A	11370435	11832198	biogrid:612143	biogrid

# Let us find a list of genes falling within a group

- Let us create ge-gr.config file which contains the list of genes and group

**KNOWLEDGE loki.db**

**GENE HIST1H3A KIAA2013 PQBP1 DCAF8**

**GROUP R-HSA-5173214**

**FILTER gene group**

- Run as **biofilter.py test.config**
- output : **biofilter.log , biofilter.gene-group**

# biofilter.log

loading knowledge database file '/usr/local/bin/loki.db' ...  
... OK  
knowledge database genome build: GRCh38 / UCSC hg38  
adding to main gene filter ...  
... OK: added 4 genes  
adding to main group filter ...  
... OK: added 1 groups  
writing 'gene group' filter to 'biofilter.gene-group' ...  
... OK: 0 results



**Open file and check information types**

# Annotating a SNP with gene region information

- Let us create `annotate.config` file which contains the list of snps

```
KNOWLEDGE loki.db
SNP rs11 rs24 rs99
ANNOTATE snp region
```

- Run as `biofilter.py annotate.config`
- output : `biofilter.log` , `biofilter.gene-group`

## **biofilter.log ,**

```
loading knowledge database file '/usr/local/bin/loki.db' ...  
... OK  
knowledge database genome build: GRCh38 / UCSC hg38  
adding to main SNP filter ...  
... OK: added 3 SNPs (0 RS#s merged)  
writing 'snp : region' annotation to 'biofilter.snp.region' ...  
... OK: 3 results
```

## **biofilter.snp-region**

**Open file and check information types**

# Annotating SNPs with location information

Let us create annotate2.config file which contains the list of snps

KNOWLEDGE loki.db

SNP rs11 rs24 rs99

ANNOTATE snp position

Run as biofilter.py annotate2.config

output : biofilter.log , biofilter.snp-position

loading knowledge database file '/usr/local/bin/loki.db' ...

... OK

knowledge database genome build: GRCh38 / UCSC hg38 adding to main  
SNP filter ...

... OK: added 3 SNPs (0 RS#s merged)

writing 'snp : position' annotation to 'biofilter.snp.position' ...

... OK: 3 results

## **biofilter.snp-region**

- **Open file and check chromosomal number where SNPs are present and also define position .**
- **Can you calculate distance among SNPs (in base pairs)**

# Map a SNP to the groups and sources where the SNP is present

Let us create annotate3.config file which contains  
the list of snps

```
KNOWLEDGE loki.db SNP rs11  
rs24 rs99  
ANNOTATE snp group source
```

Run as `biofilter.py annotate3.config`

output : `biofilter.log` , `biofilter.snp-group-source`



## biofilter.log

loading	knowledge	database	file	'/usr/local/bin/loki.db'	...
...					OK

**Open file and check information types**

# **Annotating a base pair region with the list of SNPs in that region**

**Let us create annotate4.config file which contains genome position and chromosomal number**

```
KNOWLEDGE loki.db  
REGION 1:30000:40000  
ANNOTATE snp region
```

**Run as biofilter.py annotate4.config**

**output : biofilter.log , biofilter.snp-regions**

# biofilter.log

```
loading knowledge database file '/usr/local/bin/loki.db' ...  
... OK  
knowledge database genome build: GRCh38 / UCSC hg38  
WARNING: UCSC hg# build version was not specified for region input;  
assuming it matches the knowledge database  
adding to main region filter ...  
... OK: added 1 regions  
writing 'snp : region' annotation to 'biofilter.snp.region' ... calculating  
main region zone coverage ... OK  
... OK: 88 results
```



**88 SNPs falling in that regions**

## **biofilter.snp-regions**

**#snp chr region start stop**

**rs534702355 1 chr1:30000-40000 30000 40000**

**rs867282737 1 chr1:30000-40000 30000 40000**

**rs62028215 1 chr1:30000-40000 30000 40000**

**rs778316262 1 chr1:30000-40000 30000 40000**

**rs28688489 1 chr1:30000-40000 30000 40000**

**rs28628742 1 chr1:30000-40000 30000 40000**

**rs28594168 1 chr1:30000-40000 30000 40000**

**rs558169846 1 chr1:30000-40000 30000 40000**

**..... SO ON**

**Count SNPs falling from 6000-10000 genomic regions of chromosome 1, 2, 3, 4 and 5**

**Create 5 different config files. Analyse result**

**Develop the bar graph based on SNPs count .**

# Mapping regions to genes based on percent of overlap

Let us create annotate5.config file which contains genome position and chromosomal number

```
KNOWLEDGE loki.db REGION  
1:3000:40000  
REGION_MATCH_PERCENT 50  
FILTER gene
```

- Run as `biofilter.py annotate5.config`
- output : `biofilter.log` , `biofilter.gene`

# biofilter.log

```
loading knowledge database file '/usr/local/bin/loki.db' ...  
... OK  
knowledge database genome build: GRCh38 / UCSC hg38  
WARNING: UCSC hg# build version was not specified for region input;  
assuming it matches the knowledge database  
adding to main region filter ...  
... OK: added 1 regions  
writing 'gene' filter to 'biofilter.gene' ... calculating main region zone  
coverage ... OK  
... OK: 6 results
```

## biofilter.gene

```
#gene DDX11L1  
MIR1302-2  
MIR6859-1  
MIR1302-2HG  
WASH7P  
FAM138A
```

# Mapping regions to genes based on base pair overlap

Let us create annotate6.config file which contains genome position and chromosomal number

KNOWLEDGE loki.db

REGION 1:4000:10000

REGION\_MATCH\_BASES 10

FILTER gene

- Run as biofilter.py annotate6.config
- output : biofilter.log , biofilter.gene

**Open biofilter.gene and check genes count**



# Pair wise Gene-Gene and SNP-SNP interaction

## Step 1

Map the input list of SNPs to genes within Biofilter.

## Step 2

Connect, pairwise, the genes that contain SNPs in the input list of SNPs.

## Step 3

Break down the gene-gene models into all pairwise combinations of SNPs across the genes within sources

# Step 1

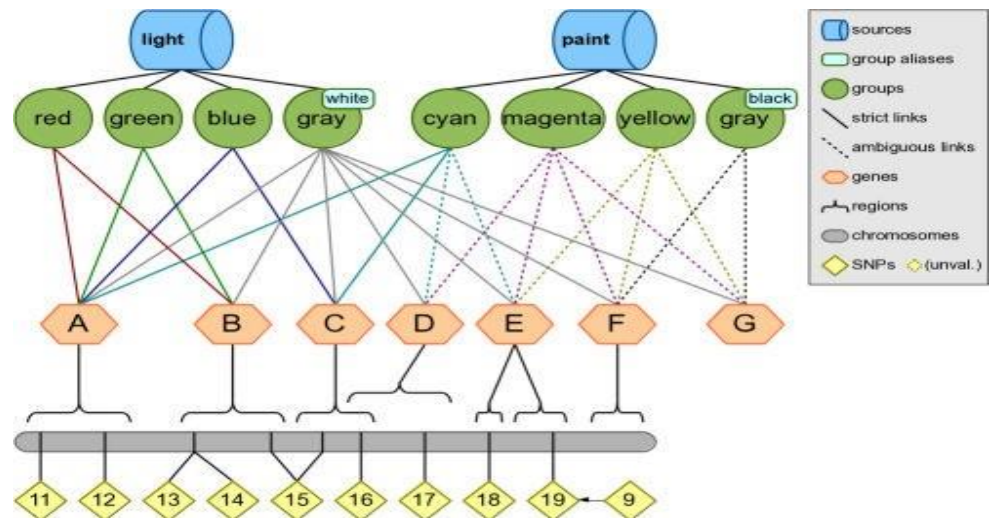
Let us create mod1.config file which contains snp list

KNOWLEDGE loki.db


SNP rs983143041 rs101598290 rs962840243 rs558399301 rs539304291  
rs974293917 rs921514204 rs571860735 rs932952501 rs750203807  
rs945700395 rs1042762941 rs114612380

FILTER gene

- Run as biofilter.py  
mod1.config
- output : biofilter.log  
, biofilter.gene



```
loading knowledge database file '/usr/local/bin/loki.db' ...  
... OK  
knowledge database genome build: GRCh38 / UCSC hg38  
adding to main SNP filter ...  
... OK: added 13 SNPs (0 RS#s merged) writing 'gene'  
filter to 'biofilter.gene' ...  
... OK: 2 results
```



## 2 SNPs falling in gene regions

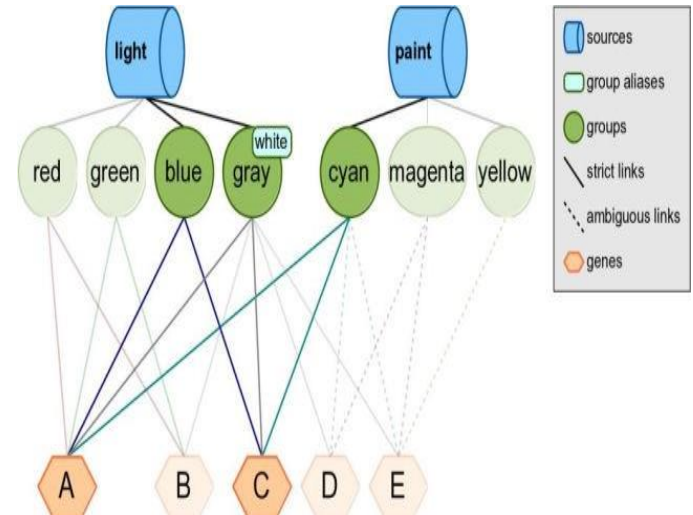
```
#gene LOC105375153  
THSD7A
```

## Step 2

- Let us create mod2 .config file which contains gene list observed in step 1

```
KNOWLEDGE loki.db  
GENE LOC105375153 THSD7A  
MODEL gene
```

- Run as biofilter.py  
mod2.config
- output : biofilter.log ,  
biofilter.gene



## biofilter.log

```
loading knowledge database file '/usr/local/bin/loki.db' ...  
... OK  
knowledge database genome build: GRCh38 / UCSC hg38  
adding to main gene filter ...  
... OK: added 2 genes  
writing 'gene' models to 'biofilter.gene.models' ...  
identifying main model candidates ... OK: 2 candidates  
identifying candidate model groups ... OK: 161848 groups  
calculating baseline models ... OK: 0 models  
... OK: 0 results
```

**0 genes validated in modelling step .**  
**These genes have no interaction .**

# Let us create mod11.config file which contains snp list

KNOWLEDGE loki.db


SNP rs268 rs316 rs326 rs328 rs333 rs334 rs544 rs551 rs567 rs662 rs669 rs671 rs683  
rs684 rs688 rs689 rs690 rs693 rs694 rs695 rs696 rs698 rs699 rs700 rs703 rs705 rs712  
rs715 rs835 rs868 rs900 rs910 rs958 rs1124 rs1164 rs1182 rs1183 rs1208  
rs1303 rs1321 rs1421 rs1442 rs1506 rs1510 rs1545 rs1547 rs1590 rs1748 rs2506  
rs2566 rs2688 rs2689 rs2765 rs2767 rs2942 rs2962

FILTER gene

**Let us create mod22.config file which contains gene list from step 1 .**

**Check output . Did you get any output ? YES**

#gene1	gene2	score(src-grp)
LIPC	LPL	4-12
INS	INSR	3-16
APOB	LDLR	3-6
APOB	LPL	3-5
ADH1C	ALDH2	3-4
APOB	LIPC	3-4
LDLR	LIPC	3-4
MKKS	CCT5	3-4
SNRPN	SNURF	2-8
GH1	INS	2-3
INS	PAX6	2-2
INS	HNF1B	2-2



These genes are  
interacting in pair  
wise manner

**Download SNP from NCBI (like 100 snps),  
check their gene information.**

**Analyse which genes are interacting and  
back trace the SNPs based on that interaction  
result.**

**Analyse their genomic position.**

**Calculate distance among SNPs.**