Biopatternsg: A logical and ontological framework for knowledge discovery on gene regulatory networks.

Description, installation, execution, consultation methods and standard results.

The Researcher's Guide

(Version 2.0)

López José[1,2], Ramírez Yacson[1], Jacinto Dávila[2]
LCAR: Laboratorio de Computación de Alto Rendimiento, Universidad Nacional Experimental del Táchira, San Cristóbal, Táchira, Venezuela.
jlopez@unet.edu.ve, yacson.ramirez@gmail.com.

CESIMO: Centro de Simulación y Modelos, Fac. Ingeniería. Universidad de Los Andes, Mérida, Venezuela. jacinto@ula.ve.

Table of contents:

Introduction.

Gene regulatory networks (GRNs) are very complex systems with very diverse sets of biological objects. To carry out the modeling of such networks, the scientific community uses and develops various computer services available through portals available on the Internet. Some of these portals are: GeneOntology, PDB, HGNC, Pathway Commons, UniProt, PubMed, among others. The mentioned sites provide services that make it possible to access such services automatically and therefore, these can be used to organize knowledge bases that integrate their resources. Our team has developed a system that allows such integration and the analysis of the information obtained in different modalities. Services such as automatic identity modeling (e.g. receptor, enzyme, etc.), molecular functions, and biological processes for the objects in a network and their protein-protein interactions, have been implemented. Our work describes an ontological and logical alternative for the discovery of biological signaling pathways and regulatory subnetworks within an GRN. This document describes the system's capabilities and modes of use, which we hope will facilitate the definition of adjustments and new requirements, aimed at improving the utility of the system we have called bioPatternsg (biopathways searching). For this purpose, an example of modeling and analysis is used, in which it is desired to explore possible links between the regulatory processes inherent in SARS-COV, SARS-COV-2 and HIV. This can be experienced directly if desired, since a server has been set up to give access to interested researchers. The example here only illustrates the use of the system, so its results only have an academic purpose. This document includes two sections. The first section focuses on the interaction that the researcher typically makes with the system. The second section shows what is required to execute a modeling experiment in the context of some network, plus a trace of its execution, which describes all the resources and knowledge bases that the system organizes automatically. It will be seen that the user can with her intervention improve the results that the system provides. This document includes two annexes. The first describes how the information used by the system is collected and organized, and the second, which provides details about its download and installation. Download the complementary material available at this address, to assist the description made about the products of the system. Here is a description of the content of this guide.

Section 1. Modeling a GRR. SARS-COV and SARS-COV-2 case.
It describes the main menus of the system and the results for the experiment that the user will replicate later. Useful to get a general idea about what the system offers.

Section 2. Biopatterns. Running an experiment.

> Section 2.1. Execution trace and knowledge bases generated.
> Section 2.1.1. Detail about the steps to follow to replicate the experiment described in section 1.
>
> Sections 2.1.2 and 2.1.3. Adjusting knowledge bases.
> It describes how the user can adjust the knowledge automatically organized, in order to improve the results that the system offers.
>
> Section 2.1.4. Products associated with an experiment.
> It describes the different stages that the system goes through when collecting information from the Internet and the knowledge bases produced. Since an experiment can last for days, the trace keeps the user informed about the stage their experiment is in.

Section 2.2. The role of ontologies in biopatternsg and how to take advantage of them.
Useful to understand the role of ontologies in the system and how to take advantage of them, in order to achieve experiments more adjusted to the interests of the researcher.

Section 2.3. How could the user scale the experiment described here.
The system provides various resources that can guide the expansion of the scope of an experiment. This section gives some examples of this.

Annex A. Biopatternsg. About how the information is collected.
Useful to know the sources consulted on the Internet and to propose new functionalities to the system that take advantage of them.

Annex B. Biopatternsg: Installation and execution.
This appendix guides the installation of the system and describes its components; in addition to the way to execute it. The Windows user is guided on the tools to use to access the server.

1. Modeling a GRN. SARS-COV/SARS-COV-2 case.

This section describes how to model a genetic regulatory network (GRN), given an initial collection of biological objects suggested for it. An experiment is run in which we want to explore possible pathways and subnets that could suggest some kind of relationship between SARS-COV and SARS-COV-2. Table 1.1 shows an extract of the KB (Knowledge Base) obtained in this experiment (see mc / minery / networks / COVID19 / COVID19-IMMUNOLOGY / kBase.pl). In addition to the KB mentioned, the system provides a documented version of it, which allows to verify the validity of the automatically modeled regulatory events (see kBaseDoc.txt, same location). The aforementioned pair of KBs are complemented by other knowledge bases, which makes it possible to define criteria according to which to explore the network; for example, explore objects that shape pathways that end with an object of interest (SARS-COV-2, for instance).

1.1. On the execution of biopatternsg and the search for information.

In biopatternsg, the execution of an experiment requires initial information consisting of: 1) a list of biological objects inherent to the network, in this case objects belonging to the mentioned virus, defined in the file expert_objects.txt file (see Table 1.2 and mc / data / COVID-19 / COVID-19-IMMUNOLOGY /); 2) the possible regulatory region to study; in this example the SARS-COV-2's UTR region (see covid19RegProm file, same location); 3) an optional list of complementary objects; for example, homologous proteins (not used in this case); and 4) an optional list of PubMed IDs that the user wishes to be considered (pubmed_IDsExp file). Later we will see how this data is provided to the system. Assume that the system has already been run and has generated outputs. Here are some details about the operation of the system, the modeling process and the type of analysis possible, accompanied by some details about the outputs that the system provides.

According to Figure 1, the system organizes searches by levels. For each group of objects defined at a modeling level, the system proceeds to query both the Protein Data Bank and the object identification services available through the HGNC. In the case of PDB, this service responds by informing about complexes that contain the object on which it is queried. Since PDB responds with complexes, their analysis allows obtaining new possible objects for the network; objects that become the new objects of the next level of construction of the GRN (see Figure 1). On the other hand, HGNC provides metadata about each queried object, which includes official name and related synonyms. Both queries (PDB and HGNC) offer the possibility that the list of initially provided objects is extended according to criteria defined by the user. For the time being, the option just described only allows us to obtain complexes according to the order of relevance defined by the PDB itself. Biologists and physicians are expected to propose other criteria to better exploit this functionality. For the purposes of this example, the query to PDB is not used, but it is a functionality that the user could exploit by providing (the programmers) her own search criteria. Other portals are also consulted to inquire about information related to the objects of a network under construction, such as the case of GeneOntology and MeSH. More details regarding the description of the system in the paper accessible at this link. Annex A provides an introduction about this.

Figure 1 shows that a KB on modeling goes through automatic analysis processes, which eventually leads to possible regulation pathways of the style presented in Table 1.3. Table 1.3 illustrates the results in the search for pathways available in the modeled BC (see Table 1.1 and pathways.txt file, mc / minery / networks / COVID19 / COVID19-IMMUNOLOGY /). Table 1.3 shows one of several possible regulation pathways. The same table describes the sentences that support the

discovered pathway. On the other hand, the system provides the documented version of the KB of events, which allows locating such sentences in the general summary of the abstracts downloaded by the system. Using those sentences, one can go to PubMed and retrieve the PubMed IDs corresponding to the events of a regulation pathway.

Table 1.1. Partial view of the Knowledge base (KB) for SARS-COV / SARS-COV-2.

```
base([
...
event('MHC',reveal,'PSMB7'),
event('MHC',develop,'PSMB7'),
event('STAT',relate,'CCR5'),
event('CCR5',relate,'STAT'),
event('CCR5',lead,'CCR5'),
event('Interferon',associate,'CCR5'),
event('STAT',activate,'CCR5'),
event('CCR5',activate,'STAT'),
….
event('ORF8',reveal,'SARS-CoV'),
event('SARS-CoV',reveal,'ORF8'),
event('SARS-CoV-2',associate,'ORF8'),
event('ORF8',associate,'SARS-CoV-2'),
event('SARS-CoV',associate,'ORF8'),
event('ORF8',emerge,'SARS-CoV-2'),
event('ORF8',emerge,'SARS-CoV'),
...
]).
```

Note: The KB was modeled automatically and contains 3908 events in total. As usual, the KB includes false positives. Comments about that in section 2 of this guide.

Table 1.3 includes only one of the possible pathways, relative to the possible relationship of the objects ORF6, CD4, JAK3, STAT and MHC in the regulation of SARS-CoV-2. All the pathways found must be analyzed manually to determine which of them contain only positive events. We say "positive event", in the sense that the sentence related with the event turns out to be well represented by the event itself. For the example used here, the system returns 401 possible pathways from which we have chosen one at random. Since many pathways have at least one false positive event, they must be considered wrong; however, it may happen that when reading the related chain of events, the user discovers something of interest. It will be seen later that in scenarios like this, the user can correct or add information and then proceed to re-run the system at some stage of interest, thus improving the results' quality.

1.2. Analysis options.

Biopatternsg can be downloaded and installed or be used remotely on a server (details in Annex B). Table 1.4 shows a summary of the navigation that is typically performed when running the system. As the table shows, the networks on modeling are listed first. Once we choose a network, the system lists the different processes that have been executed or are being executed for it. As soon as we choose any of the listed processes, the system lists the experiment's configuration, accompanied by a summary of the results obtained, both in relation to modeling and searching for pathways. Finally, the system displays the analysis options it provides for the process under consideration.

Table 1.2. Initial identifiers for SARS-COV / SARS-COV-2.

JAK
JAK3
MHC
ORF6
ORF8
importin
STAT
ACE2
CD4+
CD4
CD8+
CD8
CCR5
CXCR4
SARS-CoV-2
SARS-CoV
MICA
MICB
MICC
HLA-A
HLA-B
HLA-C

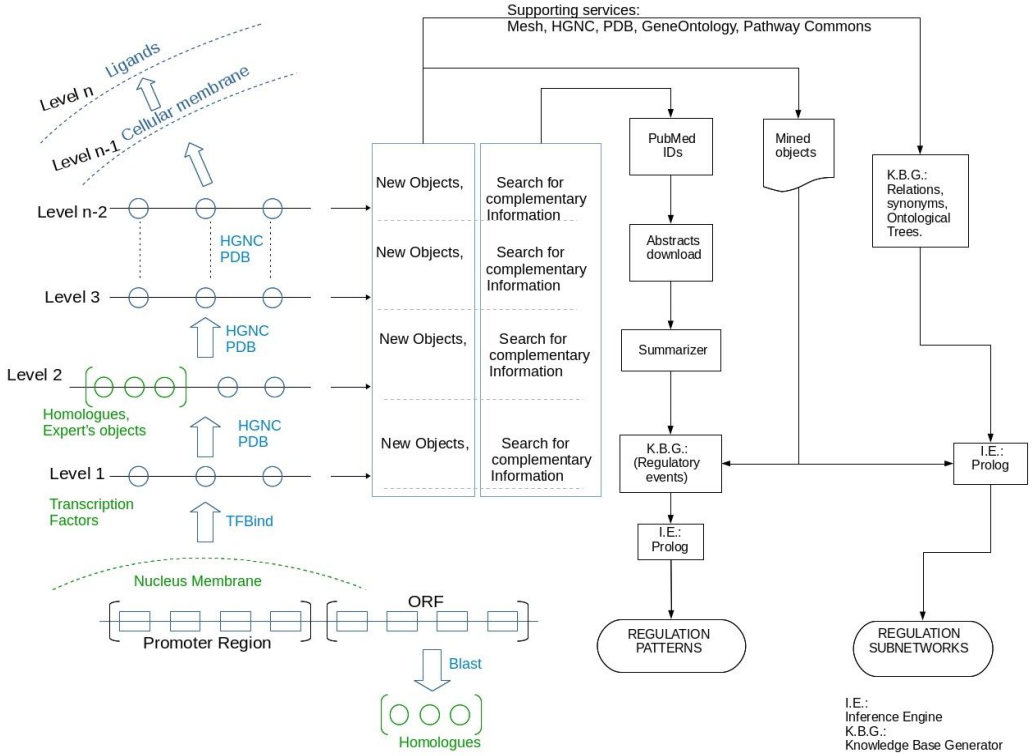

Figure 1. System architecture and work-flow for the semantic modeling and analyzing of a GRN.

Table 1.4 shows that 39 objects were analyzed for the COVID-19-IMMUNOLOGY-I experiment. We can also see the number of combinations of the objects' identifiers (names and synonyms), according to which the PubMed queries were made; which led to 76881 PubMed IDs for the ongoing experiment. Once the related abstracts have been downloaded and

the regulatory sentence identification process has been carried out, the system automatically builds the BC shown in Table 1.1. Table 1.4 reports that 3908 regulatory events were modeled for this experiment.

Table 1.3. Inference of patterns related to the regulation of SARS-CoV-2 and STAT (an example of 401 possible).

P = 'ORF6',bind,'STAT';'STAT',associate,'JAK3';'JAK3',associate,'MHC';'MHC',associate,'CD4';'CD4',regulate,'SARS-CoV-2'

----> event: 'ORF6',bind,'STAT'

We mapped the region of ORF6, which binds karyopherin alpha 2 , to the C terminus of ORF6 and show that mutations in the C terminus no longer bind karyopherin alpha 2 or block the nuclear import of STAT1.

We also show that N-terminal deletions of karyopherin alpha 2 that no longer bind to karyopherin beta 1 still retain ORF6 binding activity but no longer block STAT1 nuclear import.

----> event: 'STAT',associate,'JAK3'

These results imply that JAK/STAT activation is associated with replication of leukemic cells and that therapeutic approaches aimed at JAK/STAT inhibition may be considered to halt neoplastic growth.

Multiple cytokine receptors signal through Janus kinases (JAKs) and associated signal transducers and activators of transcription (STATs).

----> event: 'JAK3',associate,'MHC'

These include HLA-B27 and the aminopeptidases (ERAP1, ERAP2, and LNPEPS) , which are involved in antigen processing and presentation to T-cells, and several genes (IL23R, IL6R, STAT3, JAK2, IL1R1/2, IL12B, and IL7R) involved in IL23 driven pathways of inflammation.

Compared with MHCC-97H-ROCK2, the DEmRNAs in MHCC-97H-ROCK1 were involved in the JAK-STAT cascade, the Akt signaling pathway and the activity of several different peptidases.

----> event: 'MHC',associate,'CD4'

In contrast, the increase in HLA-ABC expression by CD8+ lymphocytes was associated with transition from 2 H4+ to 2 H4int status, which suggests that increased HLA-ABC expression occurs at an earlier stage in the acquisition of CD45RO in CD8+ cells than for CD4+ cells.

After treatment, we found that the upregulation of PD-1 and T cell immunoglobulin mucin-3 (Tim-3) expression on CD4+ and CD8+ T cells was significantly associated with a poor clinical outcome in the HLA-A,2402-matched group (p = 0.033 , 0.0282 , 0.0046 , and 0.0068 , respectively).

----> event: 'CD4',regulate,'SARS-CoV-2'

Importantly, using this system, we functionally identified the CD4+ and CD8+ peptide epitopes targeted during SARS-CoV-2 infection in H2b restricted mice.

CD4+ T\240\cell responses to spike, the main target of most vaccine efforts, were robust and correlated with the magnitude of the anti-SARS-CoV-2 IgG and IgA titers.

|**Note**: The documentation of any pathway can be accessed by consulting the pathwaysDoc.txt file (located at mc/minery/networks / COVID19 / COVID19-IMMUNOLOGY/. Open the file and search, for instance, the pathway pointed out by **P** in the table.

Once an experiment finishes its modeling phase, the system proposes the exploration of regulatory pathways. Table 1.4 shows that such a step has been executed at least once, indicating 401 available pathways. Once the pathways searching is complete, the main menu is available to start the network analysis (see Table 1.4, bottom part). They stand out there, in order: 1) the possibility of executing the modeling again; 2) access the general analysis menu; 3) infer again the regulation pathways; 4) repeating the modeling process from a particular stage (rebuilding the BC, for example). This last option makes it easier for the user to modify any of the knowledge bases, subsequently requesting that the modeling and analysis stages of their interest be repeated.

Three analysis options are illustrated in this section: 1) the general search for regulatory pathways present in the BC of events (see Table 1.3); 2) given a ligand and a receptor, determine the type of pathways to which they could lead (inhibitory or stimulatory) (see Table 1.5); and 3) the search for pathways that first stimulate the presence of an object in the network,

connected to other pathways that could later inhibit it (see Table 1.6). The tables mentioned are representative of the type of outputs that the system produces and only show part of the corresponding content. Each of the pathways presented in such tables requires user validation of its regulatory events. For now, the system proposes events, pathways and subnets, according to biological restrictions that can be extended at the user's request. When considering the examples, it should be kept in mind that since the events modeling is automatic, it is normal to see false positives in both pathways and binding events.

Table 1.4. A summary of the initial bioPatternsg options.

```
Select an option:

==== Networks available ====
1.- BAXS
2.- COVID-19
I.-  Go to integrated networks
0.- Exit
2

Network: COVID-19
Select an option
==== Processes available ====
1.- COVID-19-IMMUNOLOGY
2.- COVID-19-Farmacos
3.- COVID-19
4.- COVID-19-IMMUNOLOGY-I

I. Do network integration
0.- Go back
4

BioPatternsg
minery/networks/COVID-19/COVID-19-IMMUNOLOGY-I

Initial configuration:

Regulatory region file:               COVID-19RegProm
Number of PDB objects:          0
Number of searching levels (Number of levels to build the network) :              0
Confidence for TFbind:          99
Number of PubMed IDs for every pair of combined symbols:   2000
Expert's objects from:  data/COVID-19/COVID-19-IMMUNOLOGY/pubmed_IDsExp

Results:

Consulted objects:            39
Number of pairs of objects combined:    10521
Pubmed IDs:       76881
Regulatory events:         3908
Patterns (Pathways):        401

Select an option:
1.- New process
2.- Go to the GRN analysis menu.
3.- Go to infer pathways.
4.- Updating knowledge base of events.
5.- Generating pathways' documentation.
6.- Run the system from
0.- Go back
```

As mentioned before, the system provides the possibility to reactivate any of its modeling stages (see option 6, lower part of Table 1.4); the Table 1.7 shows the result of accessing this option. Since any of the knowledge bases can be adjusted manually the mentioned option allows the active participation of the user. For example, suppose that the system has finished organizing the knowledge bases and even that pathways have already been inferred. Suppose now that when reviewing the sentences of some pathway of interest, the importance of a new object not considered at the beginning of modeling became

clear. In this scenario the user can add the new identifier to the list of names and synonyms; as well as manually including the identity of the object where it is appropriate. Once this has been done, the stage 6 of the system's pipeline can be re-executed (Table 1.7), thus redefining the knowledge base related to regulation events. This last step will possibly add new events to the knowledge base, events in which the indicated object participates. The scenario just described is one where the user wants to take advantage of the collection of abstracts he has already collected while, perhaps, he is already running an experiment from scratch in which the new object has already been incorporated.

Table 1.5. Type of standards available for a given ligand / receptor (ORF6 / STAT case).

```
'ORF6',bind,'STAT';'STAT',associate,'JAK3';'JAK3',associate,'MHC';'MHC',associate,'CD8';'CD8',regulate,'SARS-CoV-2'
['ORF6', 'STAT', 'JAK3', 'MHC', 'CD8', 'SARS-CoV-2']
Tipo: Estimulatorio

'ORF6',bind,'STAT';'STAT',associate,'JAK3';'JAK3',associate,'MHC';'MHC',associate,'CD8';'CD8',inhibit,'SARS-CoV-2'
['ORF6', 'STAT', 'JAK3', 'MHC', 'CD8', 'SARS-CoV-2']
Tipo: Inhibitorio

'ORF6',bind,'STAT';'STAT',associate,'JAK3';'JAK3',associate,'MHC';'MHC',associate,'CD8';'CD8',bind,'SARS-CoV-2'
['ORF6', 'STAT', 'JAK3', 'MHC', 'CD8', 'SARS-CoV-2']
Tipo: Estimulatorio
....
```

Note: In this example, different pathways are proposed that could lead to the stimulation or inhibition of the regulation of SARS-CoV-2. It is agreed here that ORF6 is a ligand, so that pathways can be initiated from it

Once the user is fine about the knowledge bases and has explored pathways, then it is possible to take advantage of the facilities offered by the general analysis menu; available through option 2 of the main menu (see lower part of Table 1.4). Table 1.8 shows the result of accessing this option. Such a menu offers several options, including those described in Tables 1.5 and 1.6, corresponding to options 4 (search for pathways for ligands indicated by the user) and 8 (search for chains of connected pathways). The other options will be explained in future versions of this guide, although each one has a small text that describes them when they are activated.

Table 1.4 shows that the system displays a first group of analysis options. Such options include the number 2, designed for the general analysis of GRNs. Table 1.8 shows the result of accessing such an option. Options 4 and 8 visible in Table 8 lead to results of the type illustrated in Tables 1.5 and 1.6, respectively. In general, the options presented in Table 8 provide the following query modes: 1) given a ligand, which receptors recognize it; 2) to identify complexes in the network and propose their possible regulatory roles; 3) given a receptor, to which complexes it binds; 4) given a ligand, to which pathways does it lead and of what regulatory type are they; 5) Given a receptor, to which DNA motifs it binds and to which pathways it leads; 6) given a receptor, which ligands recognize it; 7) given a receptor, which of its ligands could be considered agonists, antagonists or mixed function; 8) given an object, find groups of connected pathways of the stimulatory / inhibitory type that could be associated with it; and 9) given an object, display its names, synonyms, MESH and GeneOntology trees (textual, non-graphic representations).

Below we describe an example of the system's execution, accompanied by a detailed description of the different knowledge bases that it generates.

Table 1.6. Example of a possible stimulatory / inhibitory subnetwork for SARS-CoV-2.

Pathway=> 'CD4',bind,'importin';'importin',associate,'CD8';'CD8',associate,'JAK3';'JAK3',associate,'MHC';'MHC',associate,'STAT';'STAT',regulate,'SARS-CoV-2' Linking events: 'SARS-CoV-2',affect,'MHC'; 'SARS-CoV-2',mediate,'MHC'; 'SARS-CoV-2',change,'MHC'; 'SARS-CoV-2',bind,'MHC'; 'SARS-CoV-2',recognize,'MHC'; Pathway=> 'MHC',bind,'STAT';'STAT',bind,'STAT' Linking events: 'STAT',express,'ORF6'; Pathway=> 'ORF6',bind,'STAT';'STAT',inhibit,'SARS-CoV-2'

Note: This example suggests that FT STAT1 could be involved in both stimulation and inhibition of SARS-CoV-2 replication. The possible binding events suggest how the two scenarios could be connected.

Table 1.7. Options available for the re-execution of the modeling stages.

Select an option 1.- Keywords generation 2.- Search for PubMed IDs 3.- Download of abstracts 4.- Formatting names and ontologies 5.- Generating summaries 6.- Generating knowledge base of events 7.- Generating objects' identities 0.- Go back

Table 8. Options available for the analysis of the information of a GRN.

Queries for the GRN analysis. 1.- Receptors searching 2.- Complexes searching 3.- Receptor/complexes searching 4.- Ligands/pathways searching 5.- Motifs/pathways searching 6.- Ligands searching 7.- Defining ligand type 8.- Search of connected pathways 9.- Querying an object 0.- Go back

2. Biopatternsg. Running an experiment.

This section describes how to get access to biopatternsg and run an experiment. The system's products (knowledge bases) and the stages in which they are generated are also briefly described. Biopatrones is freely accessible and can be downloaded from: https://github.com/biopattersg/biopatternsg. Instructions for installing the system are available in Annex B of this document. We will use the system from a server. Please do not hesitate to email us if you want our support (josesmooth@gmail.com or jacinto.davila@gmail.com). There is also a series of demos that describe the general characteristics of the system and its modes of use. They can be accessed at Biopatternsg: Demos - Playlist. If you have a way to access the system, we recommend doing it now. In the case of Windows environments, the following programs are recommended: 1) puTTY, to access the server in terminal mode and 2) WinSCP, to access the server in files explorer mode. Terminal mode is used in this guide. Explorer mode will allow you from Windows to access the different folders on the server in graphic mode.

Biopatternsg requires some initial ***data*** to run an experiment in the context of a network (in this example COVID-19). The way to provide such data is to place it in a folder called ***data***, which is the folder containing the different data folders for the networks under study. The data folder is located under the system folder, named biopatternsg. For this example, the data folder contains multiple networks, one of which is COVID-19; under the latter another three are listed, named: COVID-19, COVID-19-Drugs and COVID-19-IMMUNOLOGY. The data/COVID-19/COVID-19 folder refers to an experiment in which the only interest is to model interactions of objects related to COVID-19, while data/COVID-19/COVID-19-Drugs refers to an experiment in which we want to see possible relationships between objects linked to COVID-19 and specific drugs. Finally, the data/COVID-19/COVID-19-IMMUNOLOGY experiment contains names of COVID-19 objects interacting with objects related with the immune system. Each folder, let's say data/COVID-19/ COVID-19-IMMUNOLOGY, will contain the following files (an example of the data folder is included in the supplemental material):

1. Regulatory region. This file must contain the piece of DNA that is considered to define the transcriptional regulatory region for the object of interest (SARS-CoV-2, in this experiment). The name of the file in this example is "COVID19RegProm".
2. Expert objects. This file lists the initial set of objects for the network on modeling. The file is called "expert_objects.txt" (see example).
3. Homologous proteins (optional). List the identifiers of the homologous proteins related with the object of interest. The file is called "homologous" (see example in the previous file). In this experiment the file is empty.
4. Expert literature (optional). Corresponds to the list of PubMed IDs of the abstracts that the user/expert wants to be included in the network on modeling, adding it to the abstracts that the system automatically collects. The file must be named "pubmed_IDsExp" (see example).

2.1 An experiment, its tuning and its outputs.

Table 2.1 shows how to run experiments on the system. As shown in Table 2.1, the system displays the list of networks for which data have been provided, or experiments that have already been developed. Once we select a network (say COVID-19), the system lists the different modeling processes/experiments that may have been developed for it, offering the opportunity to 1) integrate them or 2) select a particular experiment. If you choose to integrate (option I), then all the available knowledge bases are integrated, generating a single model for the entire network. In case of choosing a particular process (say 3), then the corresponding results for that experiment are displayed. In the event that no modeling has been developed for the selected experiment, the system requests the necessary parameters for this purpose. Table 2.1 corresponds to the latter case.

2.1.1 Running an experiment.

The first step when we run an experiment is to get access to the server. If you have done a local installation of the system on your Linux machine, then it is a good time to go to the folder where the installation took place. In case of accessing a server, you must have received the corresponding access details by email.

Once the requirements for an experiment have been configured in the data folder, the system is invoked by accessing the biopatternsg folder. From there, the command indicated below must be executed. It is recommended to run the screen command first, useful to later retrieve the experiment in progress. The following steps should lead us to the view indicated in Table 2.1, which shows how to configure a new experiment. The order of the listed processes may differ, be sure to select the <COVID-19-IMMUNOLOGY> option. The intention is that you provide the data illustrated in Table 2.1.

```
$ screen
$ java -jar biopatternsg.jar
```

Once you choose a language, you will see a screen similar to the one presented in Table 2.1. The Table 2.1 shows those parameters that are required for a new experiment. Such parameters correspond to:

- Promoter region. It corresponds to the DNA sequence that the user considers to be the regulatory region of the biological object on study. In this case, it is provided in the file COVID-19RegProm. (see mc / data / COVID-19 / COVID-19-IMMUNOLOGY / COVID-19RegProm). Look too, at the contents of the data folder on the server.
- Maximum number of complexes from PDB. It corresponds to the amount of protein complexes for each object on the network, to be downloaded from PDB. In this case, it has been chosen not to use this option. This option allows adding automatically new objects to the network, related to those that already are part of it.
- Number of search levels. It corresponds to the number of levels that will be explored in the construction of the network model (see Figure 1). In this case, it has been chosen to keep the level at zero and that means only to work with the expert's objects and the transcription factors coming from TFBIND.
- Reliability index in TFBind. It corresponds to the reliability threshold from which the FTs detected in the regulatory region will be selected. This parameter indicates the probability of anchoring on the supplied DNA, according to which the FTs will be chosen to add them to the objects of the network. In this case, only FTs with probability equal to or greater than 0.98 are chosen to be added to the network under construction.
- Maximum number of PubMed IDs for each search. It corresponds to the number of PubMed IDs that will be taken for each pair of keywords consulted. For instance, if the combination JAK3 and STAT is searched in PubMed, then Table 2.1 shows that only the first 100 PubMed IDs will be recorded and downloaded later.
- References PubMed IDs from the expert. It allows you to indicate if a PubMed IDs' file will be provided by the user. The name of the file that contains them must be indicated (see mc / data / COVID-19 / COVID-19-IMMUNOLOGY / pubmed_IdsExp).
- Short or long names. It defines whether short or long names for the object's identifiers will be used in the searching for PubMed IDs. It is normal for objects to have acronyms or mnemonics associated; answering yes implies that only short names must be used.

Once the data is provided the experiment will start and it will take some time to finish. The time an experiment requires will depend on the number of objects and the parameters indicated; time that can vary from a few hours to several days. If your experiment is stopped for any reason, the system is able to recover it from the closest stable point to the time of the interruption.

Table 2.1. Configuring and running an experiment.

```
BioPatternsg

Select an option
==== Networks available ====
1.- BAXS
2.- COVID-19

I.-Go to integrated networks
0.- Exit

Network: COVID-19
Select an option
==== Processes available ====
1.- COVID-19-IMMUNOLOGY
2.- COVID-19-Farmacos
3.- COVID-19
4.- COVID-19-IMMUNOLOGY-I

I. Do network integration
0.- Go back
1

Biopatternsg

------------------------------
New data mining process
------------------------------

Provide configuration data:

Regulatory region file name: COVID-19RegProm
Confidence value for TFbind (0-100): 99
Number of PDB objects (The amount of PDB objects that must be downloaded): 0
Number of searching levels (Number of levels to build the network): 0
Do you want to specify your own set of PubMed IDs? ..(Y/N): y
Set the name of the file with the PubMed IDs: pubmed_IDsExp
Number of PubMed IDs for every pair of combined symbols: 100
Only short names to combine and generate paired symbols... (Y/N): y

BioPatternsg
minery/networks/COVID-19/COVID-19-IMMUNOLOGY

Searching complementary information for expert's objects

search JAK.....
search JAK3.....
```

Execute the following command to leave your experiment running on the server (without releasing the <ctrl> + <a> keys).

```
$ <Ctrl> + <a> + <d>
```

Execute the following command to recover it.

```
$ screen -r
```

Re-execute `<ctrl> + <a> + <d>` to leave the experiment running in the background.

The *screen* command allows you to have multiple experiments running and retrieve them at any time. If you have more than one experiment running, the *screen -r* command will return the identifiers associated with each one of them. If you run *screen -r <identifier>*, the command will return the terminal to the corresponding experiment.

Run the system again, but now to explore an example of what you will get at the end of the experiment that you just started.

```
$ screen
$ java -jar biopatternsg.jar
```

Select the option <COVID-19-IMMUNOLOGY-I> from the output illustrated in Table 2.1. Note that this experiment requests up to 2000 abstracts for each pair of objects queried to PubMed; this experiment took between 4 to 7 days to finish. Your experiment in the background should take no more than a couple of days but this depends too on how fast your Internet connection is.

Let's say you want to reproduce the results shown in Tables 1.3, 1.5 and 1.6, using the experiment <COVID-19-IMMUNOLOGY-I>. The procedure would be as follows:

To reproduce Table 1.3, select option 3 from the bottom menu in Table 1.4. Such an option corresponds to pathways inference. Provide the parameters indicated in Table 2.2. Once the pathways searching is finished, the system will display a summary of the experiment's parameters and the number of pathways found (in this case 401). The searching results generates three files, in this case: 1) pathways.txt, which contains the 401 patterns found; 2) eventsDoc.txt, which contains the different events that shape the pathways, accompanied by each one with the different sentences that give them conceptual support from the abstracts, and 3) pathwaysDoc.txt, which lists the pathways accompanied by the events and sentences that shape each pathway. The aforementioned files are placed by the system in the path minery / networks / COVID-19 / COVID-19-IMMUNOLOGY-I. The files can also be viewed at mc / minery / networks / COVID-19 / COVID-19-IMMUNOLOGY. Table 1.3 corresponds to one of the 401 pathways described in pathwaysDoc.txt. We recommend copying the pathway and searching for it in such a file. Table 1.3 is a simplified version of the result that you will find by searching. Please note that some regulatory events have their documentation in another pathway, therefore the pathway's number where to go is indicated.

To reproduce Table 1.5, choose option 3 from the lower menu in Table 2.2 and get access to the general menu for the analysis of a network (Table 1.8). Once there, select option 4, corresponding to the search for pathways for a ligand and a receptor. Supply ORF6 as the ligand and STAT as the receptor. In this example we have agreed for exploration purposes that the FT STAT fulfills the role of being the receptor with which ORF6 could start regulation pathways. This option shows your results only on screen, soon your results will be exported to a file named ligand-receptor-pathways.txt.

To reproduce the Table 1.6, return to the menu shown at the bottom of Table 1.8 and choose option 8, corresponding to the searching for connected pathways. From Table 1.6, copy the pathway shown there and supply it as the starting point to explore subnetworks. This pathway was originally chosen from the pathways.txt file, suggesting that the user found

something of interest in it; which motivates you to use it as a starting pathway when searching for subnets. Once you provide the initial pathway, the system will ask about how many objects you will allow to be present in the pathways in the subnetworks. It is usually a number between 3 to 8, but it can be any number; it is recommended to gradually increase the number until you get some result. The output of this option is saved in a file called chainsPathways.txt. We recommend that you rename it so that you do not lose it in future executions of this system's option. Here we have chosen to name it using the object that closes the pathways; in this case SARS-CoV-2 (see the file SARS-CoV-2_chainsPathways.txt, supplementary material). Note that these are pathways that apparently first stimulate the virus replication and then inhibit it.

Table 2.2. Searching of pathways.

```
BioPatternsg
minery/networks/COVID-19/COVID-19-IMMUNOLOGY-I

Pathways Inference
Do you want to limit the objects in the pathways to a specific list?  ..(Y/N): y
Provide a list of objects' symbols separated by (,): for example: EGF,EGFR,Ras,CREB,SST
JAK3,STAT,MHC,SARS-CoV-2,ORF6,importin,CD4,CD8
Do you want to provide a symbol object to finish the pathways? ..(Y/N): y
Provide the object name: STAT
Do you want to provide another object to finish the pathways?.. (Y/N): y
Provide the object name: SARS-CoV-2
Do you want to provide another object to finish the pathways?.. (Y/N): n
Do you want to use a reduced knowledge base?  .. (Y/N): y

BioPatternsg
minery/networks/COVID-19/COVID-19-IMMUNOLOGY-I

Pathways Inference
.....

Initial configuration:

Regulatory region file:               COVID-19RegProm
Number of PDB objects:          0
Number of searching levels (Number of levels to build the network) :              0
Confidence for TFbind:          99
Number of PubMed IDs for every pair of combined symbols:   2000
Expert's objects from:  data/COVID-19/COVID-19-IMMUNOLOGY/pubmed_IDsExp

Results:

Consulted objects:           39
Number of pairs of objects combined:    10521
Pubmed IDs:       76881
Regulatory events:         3908
Patterns (Pathways):       401

Select an option
1.- New process
2.- Go to the GRN analysis menu.
3.- Go to infer pathways.
4.- Updating knowledge base of events.
5.- Generating pathways' documentation.
6.- Run the system from
0.- Go back
```

2.1.2 Adjusting the information collected to obtain results.

Before we get results like those seen above, some adjustments must be made to the information that the system organizes automatically. In this regard, the two main settings correspond to the minedObjects.txt and pathwaysObjects.pl files.

The minedObjects.txt file contains the names and synonyms separated by ';', associated with the objects of interest that are listed in Table 1.2 (see an example of the file in mc / minery / networks / COVID19 / COVID19-IMMUNOLOGY). The adjustment about this file relates to ensure that each object has only one description line associated with it. Such a line should include the first the main name for the object, followed by the synonyms that could have been automatically organized for it. Note that Table 1.2 includes names like JAK3 and JAK. This is usually done when there are two names for the same object, which are equally important to the user. When collecting the information from the internet, this will generate two descriptive lines in minedObjects.txt. It is up to the user to ensure that both lines are organized into one, so that the knowledge base does not have redundant names and events. If you specify no more than one identifier per object in the expert_objects.txt file (Table 2), then this setting is not necessary. Another situation for tuning minedObjects.txt is the case where the system has failed to find synonyms for some of the objects. In this case, you must include them by indicating the main name first, followed by possible synonyms separated by ';' without line breaks. Another reason the minedObjects.txt file should be checked is to make sure no synonyms appear more than once. If this is the case, it is up to the user to choose where to leave it. In case the system fails to find synonyms for an identifier, it will repeat the name of the object as the only synonym (please, allow that exception). Any change that the user makes to minedObjects.txt means that the kBase.pl event knowledge base must be rebuilt. This task must be carried out by the user through the option 6 of the menu described in the lower part of Table 2.2.

Regarding the pathwaysObjects.pl file, the user must verify if the roles assigned to the objects in the network are correct (see the file in the location indicated above). Roles such as ligand, protein, transcription factor or receptor can be seen in the mentioned file. These roles guide the search for pathways. A pathway has to start with a ligand that binds to one receptor or through a receptor that binds to another receptor; following this first event for a set of intermediate events corresponding to proteins connecting with each other. Such Intermediate events end with a closing event in which a receptor, or an FT, binds to an object of interest. The user can play with the roles, thus defining which objects can start pathways and which can close them. Additional roles are expected to be added as users suggest new restrictions to explore pathways.

Once the aforementioned adjustments have been made, the user is able to explore the inference of pathways and subnets, as has been shown so far.

2.1.3 Debugging the kBase.pl knowledge base.

A task that typically should be considered when analyzing the searching of pathways and subnetworks, is the debugging of the regulatory events' knowledge base. The purpose of such debugging is to eliminate the false positives. Generally you will be interested in a subset of objects working in the network and the pathways and the subnets that connect

them, so you may well focus on the events related with those objects. In Table 2.2 it can be seen that the interest goes around the objects JAK3, STAT, MHC, SARS-CoV-2, ORF6, importin, CD4, CD8; let's see how to proceed if we want the related pathways not containing false positives.

Table 2.3. Some examples of how to tag throttling events described in eventsDoc.txt.

event('MHC',bind,'STAT'):F
Both isoforms can downregulate MHC class II, however they differ in a number of other immunomodulatory properties, such as the ability to bind the IL10 receptor and induce signaling through STAT3.

event('MHC',recognize,'STAT'):F
Along with human leukocyte antigen gene encoding B,51 (HLA-B,51) and areas including the major histocompatibility complex class I, genome-wide association studies have recognized numerous other BD susceptibility genes including those encoding interleukin (IL)-10 , IL-12 receptor β 2 (IL-12RB2) , IL-23 receptor (IL-23R) , C-C chemokine receptor 1 gene, signal transducer and activator of transcription 4 (STAT4) , endoplasmic reticulum aminopeptidase (ERAP1) , and genes encoding killer cell lectin-like receptor family members (KLRC4-KLRK1).

event('MHC',recognize,'KLRC4'):U
Along with human leukocyte antigen gene encoding B,51 (HLA-B,51) and areas including the major histocompatibility complex class I, genome-wide association studies have recognized numerous other BD susceptibility genes including those encoding interleukin (IL)-10 , IL-12 receptor β 2 (IL-12RB2) , IL-23 receptor (IL-23R) , C-C chemokine receptor 1 gene, signal transducer and activator of transcription 4 (STAT4) , endoplasmic reticulum aminopeptidase (ERAP1) , and genes encoding killer cell lectin-like receptor family.

event('MHC',detect,'STAT'):F
Statistical studies of associated alleles detected on each microsatellite locus showed that the pathogenic gene for Behçet disease is most likely found within a 46-kb segment between the MICA and HLA-B genes.

Note: The labeling in this example has not been done by a biologist.

Each time the system is requested to infer pathways, it generates a file called eventsDoc.txt, which contains all the events that are present in the collection of inferred pathways (see the eventsDoc.txt file in the supplementary material). The mentioned file breaks down the events present in the pathways and the sentences from which they are modeled. To debug eventsDoc.txt, your job as a user is to label each event as positive (: P), false (: F), or user added (: U). When the new events come from the user, it is enough to take the event and the corresponding sentence she wants to correct, copy it, and then proceed to modify the structure of the related regulatory event. Afterwards the tag: U must be added indicating so to the system that such an event was modeled manually. The third event in the Table 2.3 shows an example; there an event modeled by the user is shown, derived from the second event listed there. Note that the second event is one that the user tagged as false by adding the tag: F. It can happen that the user sees new events in events that she labeled positive. The procedure is the same.

Once all the events have been tagged in the eventsDoc.txt file, the system is requested to update the regulatory events knowledge base (kBase.pl file). Such action is carried out using option 4 of the menu visible at the bottom of Table 2.2. Then, you must proceed again to the inference of pathways (option 3 of the same menu). It is normal thet new pathways emerge when making these changes, so the eventsDoc.txt file may contain new events that require manual annotation. The procedure so far must be repeated until the system indicates in the eventsDoc.txt file that all events have already been properly labeled. This ends the kBase.pl debugging process about your subset of objects. The system keeps a history of the

events that you have already tagged. The idea is that the user should not repeat labeling that he has already made. The tagged event history is stored in the eventsDoc-History.txt file.

When the kBase.pl update is carried out, and it is verified that eventsDoc.txt no longer contains new events to tag, it is possible to generate the documentation of the regulation pathways, in such a way that they only contain true positive events. Such action can be executed with option 5 of the menu visible at the bottom of Table 2.2. The documentation of the pathways is stored in pathwaysDoc.txt. If you do not want to label the events at the moment and still generate the documentation of the pathways, then you just have to label all the events as positive, save and proceed as already indicated. Keep in mind that in this case the history will save false positives. Therefore, you will need to remove the eventsDoc-History.txt file, when you decide to tag the events; so the system will start with you from scratch. We recommend keeping the kBase.pl file backed up. If required, such a file can be generated again from option 6 of the menu visible at the bottom of Table 2.2.

Let's go back to the experiment you started earlier. To do this, exit the current experiment by going back in the menus. Close the terminal in which the current experiment is running, using the command <ctrl> + <d>. Now retrieve the experiment for <COVID-19-IMMUNOLOGY> with screen -r. If you disconnect from the server by mistake, reconnect again. Your experiment will not be lost.

Your experiment may not be over yet, so you must be in one of the phases corresponding to gathering information. Below we describe a typical trace that will help you to locate which stage you are in and the type of product that stage generates. If when entering the experiment you find it interrupted due to a server failure, rerun the experiment; the system will resume it at the stage closest to the point of failure. In what follows, all the steps are described and supported by the supplementary material, just as if you were observing your experiment after it was finished.

2.1.4. Products generated in the exploration of knowledge.

Table 2.1 shows that the information searching process starts right after indicating the necessary parameters; Table 2.4 on the other hand, shows a simplified trace of the scanning process that follows for the COVID-19-IMMUNOLOGY experiment. Table 2.4 illustrates the different modeling stages as they unfold. The content of this table is commented below and the outputs of each stage performed are indicated. The system products obtained in this example are available in the supplementary material that accompanies this document. Such products are also available on the server under the path minery / networks / COVID-19 / COVID-19-IMMUNOLOGY-I /.

- Consulting homologous objects. Corresponds to the search for complementary information (names, synonyms, MESH and GeneOntology ontologies), relative to the homologues associated with the main object for the experiment (in this case SARS-COV-2). This option is not used in the present example, so Table 2.4 does not show information about it.

- Consulting expert objects. It corresponds to the search for complementary information for the objects that the expert provides. The complementary information collected is available in mc / minery / networks / COVID-19 /

COVID-19-IMMUNOLOGY / (minedObjects.txt and minedObjects.pl). Note: The complementary information relates to current names, synonyms, tissues presence, among other details. The complementary information is collected for all the objects in the network and corresponds to the search for additional information described in Figure A.2. annex A, of this document.

- Consulting FTs from TFBind. Corresponds to the search for complementary information, referring to each of the FTs defined using TFBind. Note: Every time complementary information is consulted for any object on the network, its current standard name, synonyms, MESH ontologies and GeneOntology are defined. Details regarding the MESH and GeneOntology ontologies can be seen in the files ontologyMESH.pl and ontologyGO.pl, available at mc / minery / networks / COVID-19 / COVID-19-IMMUNOLOGY /.

- Consulting new PDB objects. In this case, for each iteration in the network construction, complementary information is consulted for the new objects defined from PDB. This option is not used in the present example, so the trace does not show information about it.

- Generating keyword combinations. Once the network objects are organized, all the names and synonyms found are used to generate the collection of keywords used in the searching for PubMed Ids.

- Search for PubMed IDs. Previously defined keyword combinations are used to define and to record PubMed IDs.

- Downloading abstract collections. Once the PubMed IDs are defined, the related abstracts are downloaded (see mc / minery / networks / COVID-19 / COVID-19-IMMUNOLOGY / abstracts). Several files with html extension can be seen in the indicated folder; each of them keeps a maximum of 700 abstracts.

- Formatting ontologies, names and synonyms. All the information collected for all the network's objects, defined in minedObjects.txt and in the MESH and GeneOntology ontologies, is represented in prolog format. This generates the files: minedObjects.pl, pathwaysObjects.pl, ontologiaGO.pl, ontologiaMESH.pl, well_know_rules.pl (see mc / minery / networks / COVID-19 / COVID-19-IMMUNOLOGY /). The mentioned files describe for each object: 1) names and synonyms; 2) basic identity information (receptor, enzyme, FT, ligand, etc.); 3) the trees of molecular function, biological processes and cellular component; 4) the definition and families, according to MESH; and, 5) the representation of identity information in the form of rules to support automatic analysis. This last file comes from the first four, using inference processes supported by prolog.

- Generating summaries from the abstracts. A summary is generated from the downloaded abstracts. Such a summary contains only sentences related to regulatory events (see mc / minery / networks / COVID-19 / COVID-19-IMMUNOLOGY / abstracts). A summary is generated for each set of abstracts.

- Generating knowledge base. This step collects regulatory events from the summary files and a related KB is constructed; KB which is used to infer pathways later (see mc / minery / networks / COVID-19 /

COVID-19-IMMUNOLOGY / kBase.pl). The system also generates a documented knowledge base (kBaseDoc), which makes it possible to identify the text lines in the summaries, associated with each event in kBase.pl.

Table 2.4. Trace of the system execution (COVID-19-IMMUNOLOGY experiment).

```
BioPatternsg
minery/networks/COVID-19/COVID-19-IMMUNOLOGY

Searching complementary information for expert's objects

search JAK.....
search JAK3.....

==== Level 0 ====
* Searching complementary information for the objects indicated by the expert... ok

* Looking for FT records from TFBIND....
* 5 Transcription factors found

Searching complementary information.. ok

* Generating keywords combinations..ok

* PubMed ID search ... ok

* Downloading abstracts' collections .. ok

* Formatting ontologies and objects' names.. ok

* Generating summaries from abstracts .. ok

* Generating knowledge base.. ok

Pathways Inference:
Do you want to limit the objects in the pathways to a specific list?  ..(Y/N): y
Provide a list of objects' symbols separated by (,): for example: EGF,EGFR,Ras,CREB,SST
JAK3,STAT,MHC,SARS-CoV-2,ORF6,importin,CD4,CD8
Do you want to provide a symbol object to finish the pathways? ..(Y/N): y
Provide the object name: STAT
Do you want to provide another object to finish the pathways?.. (Y/N): y
Provide the object name: SARS-CoV-2
Do you want to provide another object to finish the pathways?.. (Y/N): n
Do you want to use a reduced knowledge base?  .. (Y/N): y


Pathways Inference
..... ok.

Results:

Consulted objects:            39
Number of pairs of objects combined:    10521
Pubmed IDs:       76881
Regulatory events:         3908
Patterns (Pathways):        401

Select an option
1.- New process
2.- Go to the GRN analysis menu.
3.- Go to infer pathways.
4.- Updating knowledge base of events.
5.- Generating pathways' documentation.
6.- Run the system from
0.- Go back
```

- Pathways inference. It allows (firstable) to restrict the search of pathways to a particular set of the objects (optional). This option allows us to define (secondly) particular closure objects for the pathways. It is possible to indicate the use of a reduced version of the KB of events; reduced in the sense that synonym events are removed from the KB, which reduces the searching space.

- Inferring pathways. Pathways are inferred according to the criteria defined by the user (see mc / minery / networks / COVID-19 / COVID-19-IMMUNOLOGY / pathways.txt).

- General information about the modelling and inference processes is reported and the general menu of the system is displayed (see Table 2.4, bottom).

So far what is related to the presentation of the system and the way you can perform a modeling experiment. As can be seen, option 2 of the menu visible at the bottom of Table 2.4 leads to the general analysis menu provided by the system. From there, options 4 and 8 were illustrated (see Tables 1.5 and 1.6); other options will be described in future versions of this document, but when accessing them they include short texts that describe them. We encourage you to explore them and send us your questions or comments. Below we present two sections that will guide you on how you could take better advantage of the potentialities of the system developing a knowledge management better aligned to what you may require.

2.2 The role of ontologies in biopatternsg and how to take advantage of them.

In general, a knowledge representation strategy has been followed that seeks to facilitate its automatic and semi-automatic analysis; what we hope will facilitate the incorporation of new restrictions, according to new requirements. If a user proposes a new restriction to navigate the knowledge bases, then such a strategy should be relatively easy to incorporate into the system options (we hope so). About this, consider the following example.

Figure 2 shows the result of a MESH query regarding the Somatostatin hormone (SST). There you can see the six taxonomic branches, which according to MESH, correspond to the aforementioned protein. To take advantage of the knowledge provided in Figure 2, we have opted for a representation that favors the interrelation of definitions such as those indicated there. bioPatternsg uses internal definitions to represent what is described in Figure 2, in the following style: is_a ('Somatostatin-28', 'Somatostatin'), is_a ('Somatostatin', 'Pancreatic Hormones') and is_a ('Pancreatic Hormones', 'Hormones'). Such definitions (plus others of that style) allow the system to deduce that somatostatin is a family of pancreatic hormones, of which somatostatin-28 turns out to be one of its members. This approach makes it easy for the system to automatically determine if somatostatin is a hormone or not; which can lead in a searching process to a solution in which such protein participates. Note that in order to deduce that somatostatin belongs to the protein family, one must observe the different taxonomic branches that MESH provides. In bioPatternsg, such a task is performed using automatic inference processes. Figure 3 illustrates how bioPatternsg internally represents taxonomies received from MESH.

In bioPatternsg, the internal representation of the knowledge described in Figure 5, allows the definition of what we have named the identity tree for the SST protein. Figure 6 shows an example, elaborated manually that corresponds to the graphic representation of the taxonomic knowledge, represented in bioPatternsg as a set of relationships of the style “is-a”. Since the

system makes those trees available to all objects in a network, then the user might see opportunities from them to indicate various ways to restrict and explore the objects in her searching processes.

Figure 2. MESH response to consultation inherent to Somatostatin.

The knowledge trees described above are also available for GeneOntology. Figure 4 shows part of the GeneOntology tree for somatostatin. Figure 4 only shows six of the possible twenty immediate nodes to somatostatin at the bottom; most of them associated with biological processes. We invite the reader to imagine the breadth and complexity of the corresponding graph. In bioPatternsg, trees are available logically, not graphically; however, all the potential that these graphs suggest is within the user's reach to define search criteria (or restrictions). Consider all trees for all objects in a network. If criteria of interest are defined to explore this “forest”, then possible interrelationships between objects in the network could be discovered, leading to possible solutions; for example, very specific types of pathways or subnets, restricted to very specific types of objects. The working dynamic that we propose here is one in which the user knows the potentialities of the system explained so far and, based on them, she could define new requirements and restrictions, in order to guide explorations supported by the knowledge trees described above.

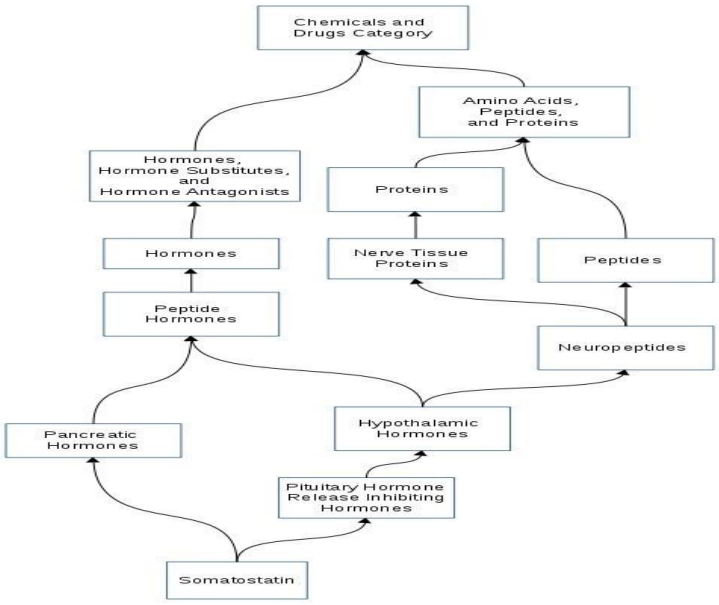

Figure 3. MESH response for Somatostatin from biopatterns (manually elaborated).

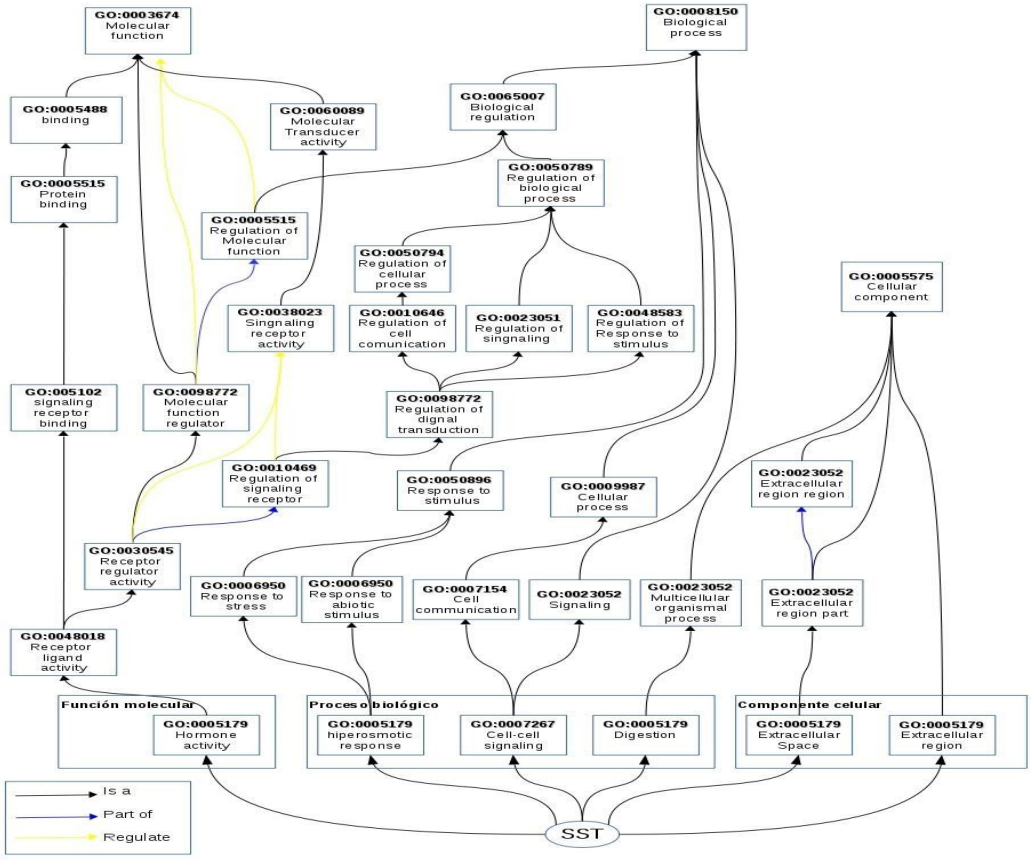

Figure 4. Partial view of the GeneOntology tree for somatostatin from biopatternsg (elaborated manually).

2.3. How to scale the experiment described so far.

Table 1.2 indicates a modeling in which several of the objects that characterize the SARS-COV and SARS-COV-2 participate. On the other hand, Table 1.4 shows that the TFBind reliability index corresponds to 0.98, which implies a restricted participation of FTs in the experiment. It is also possible to see that PDB services are not being used and also that homologous of the supplied DNA sequence are not used either. The various parameters mentioned can be incorporated in new experiments, depending on the user, and these could be used to execute processes with particular orientations. Consider a scenario in which we choose to run three additional modeling processes. The first process could be focused on an expanded list of the SARS-COV and SARS-COV-2 objects, which includes ORFx accessory proteins. In the second experiment, we could keep the list shown in Table 1.4, but choose to decrease the TFBind reliability index. The third modeling process could focus on a detailed list of proteases, but including a very specific group of drugs, that the user considers may inhibit their activity. Once we complete the modeling processes, they could be integrated to use the corresponding advantages. Each experiment in itself is demanding in terms of computing resources, so it is convenient to design independent experiments that are integrated afterwards. Also consider that when we analyze results, one can carry out the experiments separately and, after having "matured" the results obtained in each case, we can proceed to use the possible advantages from the integration.

We have exposed the functionalities of a system that allows the organization of knowledge bases, starting from a set of identifiers of interest. Once the knowledge is available, it becomes possible to access a set of options that explore possible regulatory pathways and subnets, among other possibilities. The dynamics described here is one in which the systematization of processes is sought, but that leaves different possibilities open, so that the user actively participates. Our intention with this document is to expose to potential users the facilities of bioPatternsg, in order to invite them to team up with us and explore possible working scenarios.

Annex A. Biopatternsg. About how the information is collected.

Gene Regulatory Networks (GRNs) define a very active area in the computer modeling of biological systems, and are suggesting innovative solutions to very diverse problems (Emmert-Streib et al, 2014). In such networks, a particular product (a transcript) is related to the regulation of other objects present in the same network or in different networks. Therefore, a regulatory event can activate a product that in turn participates in an event, which activates or inhibits another product. It is normal, then, that in such networks the complexity of the interrelationships grows very rapidly when we try to model them. To manage the current knowledge inherent in GRNs, computer strategies have been developed for the description, organization, interrelation and analysis of the elements that constitute them. Among such strategies are ontologies (Demir E. et al, 2010; Aditya and Babita, 2017; Muñoz-Torres and Carbon, 2017) and process diagrams (Kitano et al, 2005; Kitano,2015; Kitano, 2016); the first oriented to the semantic analysis of networks and the second to the simulation of their molecular dynamics. It is of our particular interest here, modeling and analysis systems based on inferential knowledge base processing (Rougny et al, 2018). Elaborating on such trends we have proposed the automatic construction of knowledge bases (KBs), based on the way they are described in the scientific literature. Specifically, we have developed a system that reads scientific summaries of documents and produces representations of regulatory events, for a GRN of interest, that can be computationally analyzed later. To do this, we have proposed to automatically collect the information that may be required about the objects present in a network and their possible modes of interaction.

We aim in this work: 1) the automatic construction of knowledge bases that describe GRNs (molecular species and their possible interactions) and 2) the development of exploratory analysis strategies that could guide new findings, or new conclusions, from what has already been published. Regarding 1) the developed system collects information from various repositories and services available on the Internet, necessary to answer various queries, of which we have implemented some prototypes. Regarding 2), we have designed a set of representations for the collected information, which we hope will facilitate its automatic and semi-automatic analysis. Our purpose is to answer queries of the following style: given a ligand, what kinds of protein-protein interactions result when that ligand binds to a known receptor, and which lead to activation (or inhibition) of the transcriptional response of some gene?. We also consider questions such as: for a given pair of proteins, is there a subnet in this GRN that describes interconnected regulatory pathways, in which such proteins stimulate and inhibit their transcription in any way?.

For the purposes described above, we have developed the schematic strategy expressed in Figure A.1. The modeling of a GRN starts from the region of transcription regulation of some biological object, advancing by levels, until reaching extracellular objects (a drug, for example). Our methodology begins by proposing transcription factors that could recognize the regulatory region of a protein, or other biological object (a virus, for example). From there, the modeling moves upwards integrating other objects, provided by the user or consulted on the Internet. Figure 1 shows that other objects can be added to the network from services such as those provided by the Protein Data Bank (PDB) (Rose et al, 2017) (Berman et al, 2014).

Our searching for regulatory pathways is guided by definitions like this: a regulatory pathway is one in which a ligand recognizes a receptor, which triggers a cascade of regulatory events in which different types of proteins may be involved; cascade of events that is closed by one in which a transcription factor recognizes a response element, and therefore activates or inhibits the transcription of a specific product (RNA or protein). In our case, the finding of the solution consists in determining a collection of regulatory events that satisfy the restrictions implicit in the previous definition. Having organized knowledge automatically and semi-automatically, as indicated in Figures A.1 and A.2, then events that satisfy specific constraints can be explored in a knowledge base like the one depicted in Table A.1. Our goal is to develop a logical and ontological framework, in which it may be possible to receive definitions from the user (like the previous one), program them and build knowledge bases, robust enough to help in their resolution.

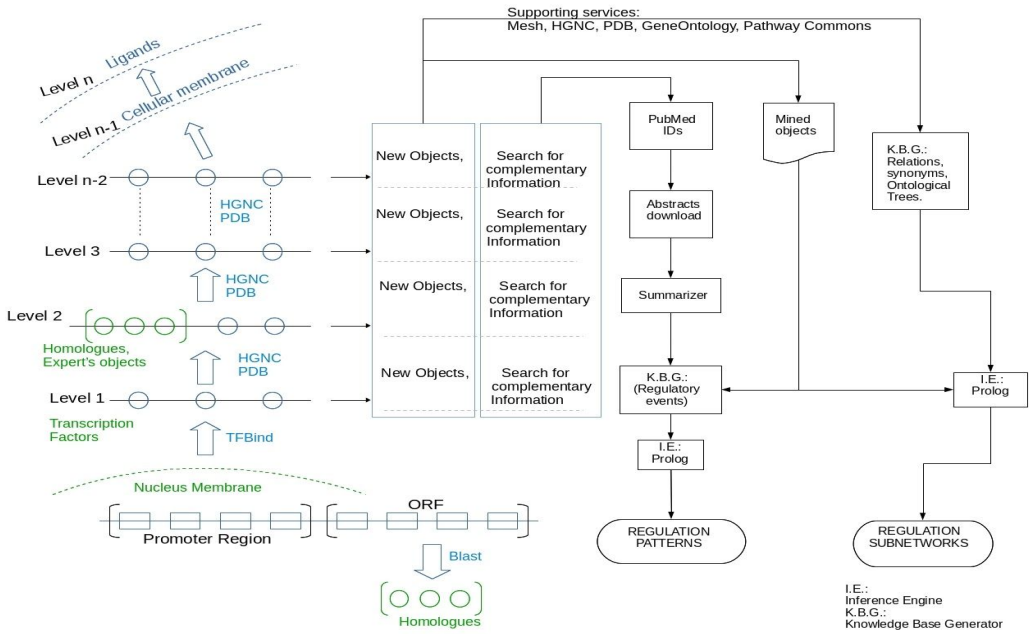

Figure A.1. System architecture and work-flow for the semantic modeling and analyzing of a GRN.

Table A.1. A subset of regulatory events in the BAXS System GRN.

```
base([
event('cholesterol',regulate,'oxysterols'),
event('oxysterols',bind,'LXRa'),
event('oxysterols',activate,'LXRa'),
event('LXRa',associate,'RXR'),
event('LXRa',bind,'LXRE'),
event('LXRa',activate,'CYP7A1'),
event('CYP7A1',increase,'ba'),
event('ba',bind,'FXR'),
event('ba',activate,'FXR'),
event('FXR',associate,'RXR'),
event('FXR',bind,'FXRE'),
event('FXR',regulate,'SHP'),
event('SHP',associate,'LRH'),
event('LRH',bind,'LRHRE'),
event('LRH',inhibit,'SHP'),
event('LRH',inhibit,'CYP7A1')
]).
```

Nota: This KB has been modeled from the literature and corresponds to what is described in Figure A.3.

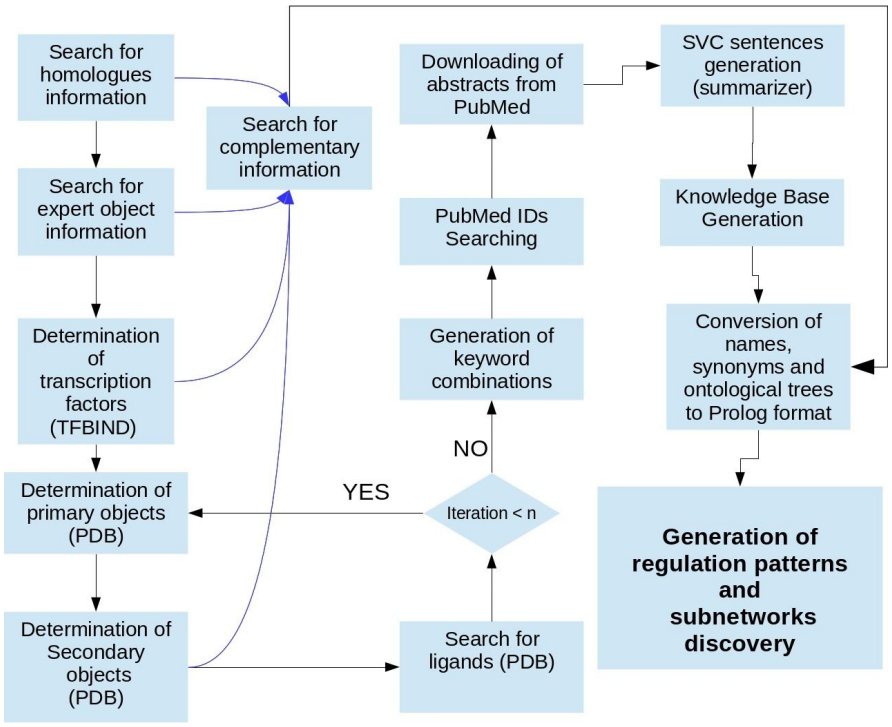

Figure A.2. Sequence of processes proposed to implement the semantic modeling of a GRN.

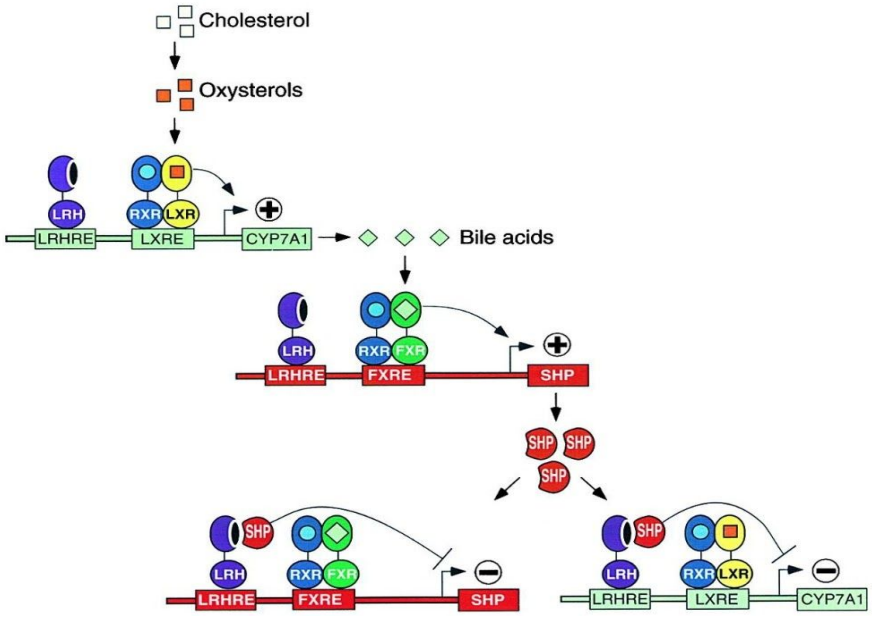

Figure A.3. A regulatory subnet of the BAXS system. Source: Lu TT, et al (2000).

1.1. A chain of processes

BioPatternsg basically consist in a set of sequentially connected programs (or processes) (see Figure A.2). Figure A.2 shows exploration levels that provide new objects accompanied by its corresponding supplementary information. Once

the modeling levels indicated by the user are completed, the system proceeds to define possible PubMed IDs related to the names and synonyms for the objects in the network. The abstracts related to those PubMed IDs are then downloaded and a general summary is obtained from then, containing only sentences related to regulatory events. Once the overall summary is defined, an event KB of the style described in Table A.1 is automatically generated. Subsequently, the modeled KB is delivered to an inference system, making it possible to discover possible pathways and subnets. Such pathways should ideally show routes from the cell membrane (region close to ligands) to the nuclear membrane (region close to transcription factors) (see Figure A.1).

We mentioned before that our aim is also to determine possible subnets of closely linked objects. Figure A.3 shows an example taken from the BAXS system (Lu TT et al, 2000). There we see two main proteins, CYP7A1 and SHP. To discover subnets like the one shown, our approach is guided by determining regulatory pathways that are connected through linking events; which offers to those who explore a GRN, possible alternatives between particularly interesting interrelated pathways. The main idea is to determine the subnets in which the proteins of interest are stimulated and inhibited. For example, if a subnet is found in which CYP7A1 is stimulated (first) and inhibited (later), and it is possible to observe in it some other protein that mediates between both scenarios, then other subnets could be determined for such a mediating protein, which suggests the conditions that govern its stimulation or inhibition. In the example, CYP7A1 is stimulated and inhibited depending on the presence or absence of SHP, therefore it is important to find subnets in which SHP is stimulated or inhibited. Two subnets coexist in Figure A.3. One that shows the conditions that stimulate and inhibit the presence of CYP7A1, and another that describes the same for SHP. Our system offers a semi-automatic scanning option to discover scenarios like the one described above.

1.2. About the knowledge gathered.

Figures A.1 and A.2 illustrate the processes that guide the development of various knowledge bases, necessary to describe the different objects collected for a GRN. Table 1 shows an example, corresponding to a set of events like the one illustrated in Figure A.3 (Lu TT et al, 2000). Table A.1 shows events in which a subject somehow affects an object and the syntax to represent this is: event (subject, relationship, object). The subjects and objects visible in Table A.1 illustrate the molecular species collected in a modeling process. Considering each of those objects, bioPatternsg builds four additional KBs in order to define: 1) name, synonyms and other basic definitions, from PubMed (Fiorini et al, 2017), Protein Data Bank (PDB) (Rose et al, 2017) (Berman et al, 2014), HGNC (HUGO Gene Nomenclature Committee) (Gray et al, 2016), GeneOntology (Thomas, 2017), Uniprot (Pundir, 2017), Mesh (Baumann, 2016) and Pathway Commons (Rodchenkov et al, 2019); 2) MESH ontology, 3) GeneOntology ontology (molecular function, biological processes and cellular component); and 4) logical facts, which establish that an object satisfies the constraints that guide the analysis of pathways and subnets. The next section describes how to build those KBs and proceed in the modelling and analyzing of a GRN.

Annex B. BioPatternsg: downloading, installation and running.

Biopatternsg runs on Linux environments so far, but once installed on a server, it can also be accessed from Windows. The installation described here was performed on a machine running Debian 10. Java 8 or higher, and Prolog version 7.2.3 or higher are required. Here we describe the installation of prolog using apt, which implies the version 8.0.2. About Java, OpenJDK 11 has been used, which is the version that Debian 10 includes by default. If you are a Windows user, the following programs are recommended to access a server: PuTTY, to access the server in terminal mode and 2) WinSCP, to access the server in files explorer mode. Terminal mode is used in this guide. The explorer mode will allow you (from Windows) to access the different folders on the server in graphic mode.

If you have access to our server then run the system and follow the researcher's guide from section 2. We also recommend section 1 of the guide if you do not have a previous reference about how the system works. The Section 3 and 4 below, describes how to run the system but prolog and java must be correctly installed first. Use the section 3 of this wiki from the line "--> List the contents of the ..." to know how to track your experiments using the screen command.

If you are a developer, please follow the instructions at the end of this page. An example using NetBeans will show you how to get access to the code and run the system. We recommend to follow the instructions 1, 2, and 3, to be sure that everything is well configured.

Steps for installing Biopatternsg:

1. Java installation.

Get access as root or use the sudo command, and run the following commands: How to Cite

```
$ sudo apt update

$ sudo apt install default-jdk
```

Once the installation has finished, check your java version:

```
$ java -version

Output: openjdk version "11.0.9" 2020-10-20 OpenJDK Runtime Environment (build 11.0.9+11-post-Debian-1deb10u1)
OpenJDK 64-Bit Server VM (build 11.0.9+11-post-Debian-1deb10u1, mixed mode, sharin
```

2. Prolog and JPL installation:

Get access as root or use the sudo command, and run the following commands:

```
$ sudo apt-get update

$ sudo apt-get install swi-prolog

$ sudo apt-get install swi-prolog-java
```

Edit the your ./.bashrc file, including the following lines (be sure that each export occupies only one line):

```
export
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/lib/jvm/java-11-openjdk-amd64/lib/:/usr/lib/jvm/java-11-openjdk-amd64/lib/s
erver/:/usr/lib/swi-prolog/lib/:/usr/lib/swi-prolog/lib/x86_64-linux/
```

```
export LD_PRELOAD=/usr/lib/swi-prolog/lib/x86_64-linux/libjpl.so
```

3. Installing biopatternsg (git option):

Since the modeling processes often require time and computational resources, it is recommended to run the system on a server. The system can be run on a personal computer, but it must be a well-resourced one. Machines with at least 8GB of RAM and latest generation processors are recommended.

Make a directory named biopatternsg:

```
$ mkdir biopatternsg
```

Change to the biopatternsg directory:

```
$ cd biopatternsg
```

Run the following commands:

```
$ git init

$ git remote add origin https://github.com/biopatternsg/biopatternsg.git

$ git fetch origin

$ git checkout runnable

Output: Branch 'runnable' set up to track remote branch 'runnable' from 'origin'. Switched to a new branch
'runnable'
```

The output above means that you are already synchronized with the runnable version of the system.

--> List the contents of the biopatternsg folder:

```
$ ls  -l
```

It must shows something like:

```
biopatternsg.jar % The system.

data % Folder containing the data required for the different experiments related to each GRN.

language % Texts messages used in the system menus in different languages.

lib % Libraries supporting the system.

minery % It will contain the models for each GRN and the models integration for each one of these.

README.TXT % Includes this installation guide.

bioinformant_summarizer % Summarizing system that allows the processing of abstracts.

scripts % scripts for the development of queries and regulatory pathways searches.
```

Run the following command to track the terminal in which the system will run.

```
$ screen
```

Run the system using the following command; then choose a language.

```
$ java -jar biopatternsg.jar
```

Proceed as described in this guide.

Once your session has finished you can end the tracking of your terminal using <Ctrl> + <d> . If you want to leave your experiment running, then use the <Ctrl> + <d> + <a> keys. You can leave the server at that moment and later, recover your session using screen -r.

4. Installing biopatternsg (zip option):

To install biopatterns, you only need to download it, unzip it and run it. Get access to download the systems' zip file and use the CODE icon. Since modeling processes often require time and computational resources, it is recommended to run the system on a server. The system can be run on a personal computer, but it must be a well-resourced one. Machines with at least 8GB of RAM and latest generation processors are recommended.

Unzip the system using the following command:

```
$ unzip biopatternsg-runnable.zip
```

Go to the biopatternsg-runnable folder and list its contents.

```
$ ls -l
```

The mentioned folder should contain the following items:

```
biopatternsg.jar % The system.

data % Folder containing the data required for the different experiments related to each GRN.

language % Texts messages used in the system menus in different languages.

lib % Libraries supporting the system.

minery % It will contain the models for each GRN and the models integration for each one of these.

README.TXT % Includes this installation guide.

bioinformant_summarizer % Summarizing system that allows the processing of abstracts.

scripts % scripts for the development of queries and regulatory pathways searches
```

Run the following command to track the terminal in which the system will run.

```
$ screen
```

Run the system using the following command; then choose a language.

```
$ java -jar biopatternsg.jar
```

Proceed as described in this guide.

Once your session has finished you can end the tracking of your terminal using <Ctrl> + <d>. If you want to leave your experiment running, then use the <Ctrl> + <a> + <d> keys. You can leave the server at that moment and later, recover your session using screen -r.

5. Running the system using NetBeans.

Follow the instructions 1, 2 and 3. They will ensure you that your machine is well configured to run the code in your framework; besides offering you the data and minery folders, required later to run the system properly using NetBeans.

Now, proceed to define your local version of the system's sources:

Make a directory named biopatternsg:

```
$ mkdir biopatternsg
```

Change to the biopatternsg directory:

```
$ cd biopatternsg
```

Run the following commands:

```
$ git init

$ git remote add origin https://github.com/biopatternsg/biopatternsg.git

$ git fetch origin master

Output: From https://github.com/biopatternsg/biopatternsg

branch master -> FETCH_HEAD

[new branch] master -> origin/master
```

Run the following command to pull the system's code:

```
$ git pull origin master

Output: From https://github.com/biopatternsg/biopatternsg

branch master -> FETCH_HEAD
```

Copy the local folders data and minery, from the local runnable branch, to the local master branch. This will offer you the data and the minery examples need it to run the system properly.

Open up the project.

If some jar files seem to be lost, please add them. Usually the jpl.jar and GSON.jar could require this step. You will locate those jars in the lib folder of the project.

The main class of the system is pipeline.BioPattern. Go to the Properties of the project, select the Run tab, and make sure that the mentioned class is declared as the main one.

In the Properties>Run tab, section VM options, set up the following line:

```
-Djava.library.path="/usr/lib/swi-prolog/lib/x86_64-linux/"
```

Run the system, you should see the following initial menu:

```
BioPatternsg

1.- ENGLISH

2.- ESPAÑOL
```

Please, follow the researcher's guide.

If you have any problem do not hesitate to contact us at this email address: josesmooth@gmail.com.

References

Aditya Khamparia, Babita Pandey (2017). Comprehensive analysis of semantic web reasoners and tools: a survey. Education and Information Technologies. November 2017, Volume 22, Issue 6, pp 3121–3145.

Baumann N (2016). How to use the medical subject hea dings (MeSH). Int J Clin Pract. 2016 Feb;70(2):171-4. doi: 10.1111/ijcp.12767. Epub 2016 Jan 13.

Berman, H.M., Kleywegt, G.J., Nakamura, H., and Markley, J.L. (2014). The Protein Data Bank archive as an open data resource. Journal of Computer-Aided Molecular Design. October, Volume 28, Issue 10, pp 1009–1014.

Bhalla U (2003). Understanding complex signaling networks through models and metaphors. Prog Biophys Mol Biol. 2003 Jan;81(1):45-65.

Demir E., Cary MP, Paley S., Fukuda K., Lemer C., Vastrik I., Wu G., D'Eustachio P., Schaefer C., Luciano J., Schacherer F., Martinez-Flores I., ..., Sander C., and Bader G.D., (2010). The BioPAX community standard for pathway data sharing. Nat Biotechnol. 2010 Sep;28(9):935-42. doi: 10.1038/nbt.1666. Epub 2010 Sep 9.

Emmert-Streib F., Dehmer M., and Haibe-Kains Benjamin (2014). Gene regulatory networks and their applications: understanding biological and medical problems in terms of networks. Front Cell Dev Biol. 2014 Aug 19;2:38. DOI: 10.3389/fcell.2014.00038.

Fiorini, N., Lipman, D. J., & Lu, Z. (2017). Towards PubMed 2.0. eLife, 6, e28801. doi:10.7554/eLife.28801.

Gevaert O, Van Vooren S, De Moor B (2007). A framework for elucidating regulatory networks based on prior information and expression data. Ann N Y Acad Sci. 2007 Dec;1115:240-8. Epub 2007 Oct 9.

Gray, K.A., Seal, R.L., Tweedie, S., Wright, M.W., and Bruford, E.A. (2016). A review of the new HGNC gene family resource. Human Genomics. Feb 3;10(1):6. DOI: 10.1186/s40246-016-0062-6. PMID:26842383.

Kitano H., Funahashi A., Matsuoka Y., and Oda K. (2005). Using process diagrams for the graphical representation of biological networks. Nat Biotechnol. 2005 Aug;23(8):961-6. DOI: 10.1038/nbt1111.

Kitano, H. (2015). Accelerating systems biology research and its real world deployment. NPJ Syst Biol Appl. 2015 Sep 28;1:15009. DOI: 10.1038/npjsba.2015.9.

Kitano, H. (2016). Artificial Intelligence to Win the Nobel Prize and Beyond: Creating the Engine for Scientific Discovery. AI Magazine, 37(1), 39-49. https://doi.org/10.1609/aimag.v37i1.2642.

Lu TT, Makishima M, Repa JJ, Schoonjans K, Kerr TA, Auwerx J, Mangelsdorf DJ (2000). Molecular basis for feedback regulation of bile acid synthesis by nuclear receptors. Mol Cell. 2000 Sep;6(3):507-15. DOI: 10.1016/s1097-2765(00)00050-2.

Munoz-Torres M, Carbon S (2017). Get GO! Retrieving GO Data Using AmiGO, QuickGO, API, Files, and Tools. Methods Mol Biol. 2017;1446:149-160. PMID: 27812941.

Pundir, S., Martin, M. J., & O'Donovan, C. (2017). UniProt Protein Knowledgebase. Methods in molecular biology (Clifton, N.J.), 1558, 41–55. doi:10.1007/978-1-4939-6783-4_2.

Rodchenkov I, Babur O, Luna A, Aksoy BA, Wong JV, Fong D, Franz M, Siper MC, Cheung M, Wrana M, Mistry H, Mosier L, Dlin J, Wen Q, O'Callaghan C, Li W, Elder G, Smith PT, Dallago C, Cerami E, Gross B, Dogrusoz U, Demir E, Bader GD, Sander C (2019). Pathway Commons 2019 Update: integration, analysis and exploration of pathway data. Nucleic Acids Res. 2019 Oct 24. pii: gkz946. doi: 10.1093/nar/gkz946. PMID: 31647099.

Rose P.W., Prlić A., Altunkaya A., Bi C., Bradley A.R., Christie C.H., Costanzo L.D., Duarte J.M., Dutta S., Feng Z., Green R.K., Goodsell D.S., Hudson B., Kalro T., Lowe R., Peisach E., Randle C., Rose A.S., Shao C., Tao Y.P., Valasatava Y.,

Voigt M., Westbrook J.D., Woo J., Yang H., Young J.Y., Zardecki C., Berman H.M., and Burley S.K., (2017). The RCSB protein data bank: an integrative view of protein, gene and 3D structural information. Nucleic Acids Res. 2017 Jan 4;45(D1):D271-D281. DOI: 10.1093/nar/gkw1000. Epub 2016 Oct 27.

Rougny, A., Gloaguen, P., Langonné, N. et al (2018). A logic-based method to build signaling networks and propose experimental plans. Sci Rep 8, 7830 (2018) doi:10.1038/s41598-018-26006-2.

T.Tsunoda, and T.Takagi (1999).Estimating Transcription Factor Bindability on DNA. BIOINFORMATICS, Vol.15, No.7/8, pp.622-630, 1999.

Thomas P. D. (2017). The Gene Ontology and the Meaning of Biological Function. Methods in molecular biology (Clifton, N.J.), 1446, 15–24. doi:10.1007/978-1-4939-3743-1_2.