Università
Ca'Foscari
Venezia

DEPT. ENVIRONMENTAL SCIENCES, INFORMATICS AND STATISTICS
Master's degree programme in Computer Science

Final thesis

A game theoretic approach to Disease Gene Prediction

Supervisor:
Chiar.mo Prof. Marcello Pelillo
Ca' Foscari University of Venice

Co-supervisor:
Chiar.mo Prof. Alberto Paccanaro
Royal Holloway University of London

Graduand:
Michele Nacucchi
Student Number 855539

Academic Year:
2016 – 2017

Abstract

Nowadays there are several tools aimed to help biologists study undiscovered phenomenons by exploiting the wide set of available data shared in the research community. One of these phenomenons requires the creation of a list of genes ranked by their likeliness to be involved in the emergence of a certain genetic disorder: in bioinformatics this problem is called "Disease Gene Prediction". We propose a semi supervised learning method to the DGP formulated in terms of Evolutionary Game Theory. In order to explore different approaches, we've built an evolutionary model based of a two players game and a multiplayer (polymatrix) game. The first one exploits a very recent variation of a well known graph-based clustering technique (Dominant Sets[25]), introducing the possibility to set some constraints (Constrained Dominant Sets[41]). This graph-based technique extracts the best cluster which contains at least a non-empty subset of the constraints nodes. In our application those are the already known genes of a particular disease and the extracted cluster contains the best unknown candidates. In the multi-population scenario this is formalized as a polymatrix non-cooperative game using the connection between the so called Relaxation Labeling Processes[11] and game theory (Miller and Zucker[21]). In a relaxation labeling process we have a distribution of the objects among all the possible labels, so in the end each object is assigned to the best label according to the concept of global consistency. For our purposes there are only two possible labels: "good candidate" or "bad candidate" and obviously by objects we mean genes. For both approaches we define a score in order to rank all the possible candidates for a given disease as expected by the purpose of the problem. We applied those methods on a gene network, a weighted graph based on HIPPIE[30] dataset, starting from some seeds genes present in the 2012 version of OMIM Morbid Map[20]. In order to evaluate the performance of our results we verified the position of recently discovered genes present in the current version of OMIM Morbid Map. Experimental results show that, in case of diseases with high modularity, we have a good precision in predicting unknown genes. Unfortunately some diseases seem to affect different areas of the network, therefore it's difficult to handle with our approach that can only exploit local informations. We propose a possible future development to solve these cases.

Chapter 1

Introduction

In the study of genetic disorders the knowledge of involved genes has an important role. Indeed, today we know a large number of hereditary diseases for which researchers have demonstrated the inheritance but were unable to identify what the involved genes are, or only a few known but are believed to be others. To have an idea of the complexity of the domain we remark that there are an estimated 20.000 - 25.000 human protein-coding genes. Each gene might encode more than one proteins and each protein might interact with others (activating or inhibiting processes, forming protein complexes, etc...) In biomedical research the hypothesis validation requires time and costly trials which slow down the new discoveries, therefore in order to help researchers to focus only on good research targets, they use bioinformatic tools that leverage the shared data in the scientific community. A modern approach to support these studies is called "network medicine": it consists of using network theory results in order to obtain new knowledge from those medical datasets available in form of graphs. Suppose, given a certain graph that some labels are assigned to one or more of its nodes: it is then possible to spread this information to the closer nodes in order to estimate if they might be also labelled the same way (diffusion method) Performing this propagation in a consistent way relies on a common a priori assumption known as "the cluster assumption" (Zhou, Bousquet, Lal, Weston, & Scholkopf, 2004; Chapelle et al., 2006), which is reminiscent of the homophily principle used in social network analysis (Easley & Kleinberg, 2010). The assumption states that neighboring points and/or points in the same cluster are expected to have the same label. A well known diffusion method applied to Disease Gene Prediction is Zhou's Consistency Method[42], a semi-supervised learning method used to rank disease genes[12] that allows the preservation of initial labelling through a regularisation parameter. The ranking can be found according to the labelling after a label diffusion process in the graph. The diffusion process follows the minimisation of a given cost function. We propose a novel game-

theoretic perspective to the problem, from two different point of view: an evolutionary non-cooperative two-players game and an evolutionary non-cooperative multiplayer game. We model the diffusion dynamics, how this spread is performed among the nodes, as an evolutionary game in which we follow the emergence of the dominant strategy. This framework it has been already used mainly in computer vision, see [27] about the general approach. For other examples about a game theory applied to bioinformatics see [3] or in different fields see [34] (word sense disambiguatio)
The input of our algorithm is a weighted graph in which the weights on the edges correspond to a similarity between connected vertices - the genes - given by an abstraction on a PPI network (HIPPIE dataset). The information that we want to diffuse is given by the OMIM dataset in which there are known pairs (gene-disease) that tell us whether the presence of mutations in a specific gene can lead to the manifestation of certain hereditary disease. The output is the rank of the best candidates not yet discovered, obtained starting from an already known (small) subset of disease genes of a certain hereditary disease. The following chapters show the basic concepts that underlie our method: in chapter 1 and 2 we review the basic concepts of biology and game theory. in chapter 3 we present our method using the techniques of Constrained Dominant Sets and Relaxation Labelling Process to solve the problem of Disease Gene Prediction. This chapter ends with some results of the application of our method to a small toy graph. The chapter 4 we show the applications on the real dataset, pointing out the experimental setup and the evaluation metrics. The last chapter presents some consideration about the results and some interesting future works.

Chapter 2

Biological background

Biology can be studied in many different scales, as far as this work is concerned, the scope will be at a cellular component level. Most cellular components perform their function interacting with other cellular components either from the same cell or from other cells. Between the coding genes and the proteins they produce, metabolites, functional RNA molecules, there are over 100,000 different entities and orders of magnitude larger amount of functionally relevant interactions.

2.1 Genes and PPI

Genes do not technically interact themselves, since genes are the definition of proteins to be produced, and proteins actually perform the tasks. Without getting into many details we can consider that two genes interact if the proteins they code dock together. Also, a single gene may code for one or multiple proteins, as no one to one matching is available and the terms cannot be used interchangeably. From a Systems Biology point of view, cellular components and their interactions are seen like networks or graphs, where the cellular entities are the nodes and the interactions edges. Commonly studied cellular networks are: Protein-protein interaction networks(PPI), where the nodes are proteins and edges represent their physical interaction. Metabolic networks, where nodes are metabolites and edges indicate their participation in a common biochemical reaction (also known as pathways). Regulatory networks, where a directed edge represents a regulatory relationship between a transcription factor and a gene and RNA networks, where nodes can be microRNAs or interfering RNAs linked to each other (RNA-RNA edges) or to DNA (RNA-DNA edges) Probably human PPIs have the largest amount of different manually curated databases, such as the Münich Information Center for Protein Sequence (MIPS), protein interaction database, the Biomolecular Interaction Network Database (BIND), the Database of Interacting Proteins (DIP), the Molecular Interaction database(MINT), and

the protein Interaction database (IntAct), the Biological General Repository for Interaction Datasets (BioGRID) and the Human Protein Reference Database (HPRD), the Human Integrated Protein-Protein Interaction rEference (Hippie), among others [2]. The Kyoto Encyclopedia of Genes and Genomes (KEGG) is the main referral for metabolic networks, but the Biochemical Genetic and Genomics (BIGG), the Human Metabolome Database (HMDB) and Reactome also are commonly used databases. Regulatory networks are available through the Universal Protein Binding Microarray Resource for Oligonucleotide Binding Evaluation (UniPROBE) and JASPAR, but this type of networks are considered to still be the most incomplete. Some microRNA-gene networks are available in databases such as TargetScan, PicTar, microRNA, miRBase and miRDB. Other manually curated databases are available, and are very useful in disease gene research: The Online Mendelian Inheritance in Man (OMIM) provides known disease and gene mappings[20]. The Gene Ontology (GO) provides an ontology of gene functions, and association files to map genes of different organisms into the ontology [1].

2.1.1 PPI network properties

Biological networks present some interesting topological properties, that reveal some evolutionary organizing principles which randomly linked networks do not present:

- **Modules**: a high degree of modularity is present in most of the networks, which represent local regions of the networks with a significant amount of interactions.
- **Scale free distribution**: the edge degree distribution of PPIs and metabolic networks follow a power law, defined as $P(k) = k^{-\gamma}$, where k is the edge degree and γ is the degree exponent of the network. A consequence of this distribution is the existence of hubs (highly connected nodes) that dominate the structure of the network.
- **Small world**: the longest shortest-path between any two nodes in the networks is relatively short. The main visible consequence in a biological sense, is that the perturbation in any node may be reflected in the behaviour of most of the network.
- **Motifs**: they are subgraphs of the network appear more (or less) frequently than expected by the degree distribution of the network. These motifs are likely to be associated with some meaningful biological process.
- **Network bridges**: nodes with a high betweenness centrality appear in most networks. Biologically, those nodes tend to correlate with

essential genes or transcription factors.

2.2 Disease Gene Prediction

In the scope of this work a genetic disease is defined as an observable phenotype(performs an adverse function) that happens when one or more genes present abnormalities (i.e. mutations, deletion or duplication). Diseases may be caused by a single or multiple genes. The high degree of connectivity that the cellular components present, show that no single component acts isolated, so the effect of the perturbation will be extended to the functionally related components that would characterise a disease, defining the *disease module*[2] (see figure 2.1)

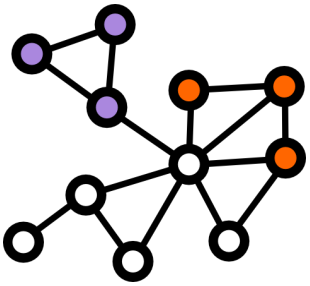


Figure 2.1: Sketch of two disease modules in a PPI

2.2.1 The modules in a PPI

One goal in network medicine is to discover every disease module, while for Disease Gene Prediction the goal is to unveil the genes within those disease modules whose alteration causes the disease. Network based methods for disease gene prediction tend to use the networks that are more complete. Both PPIs and Metabolic Pathways are being extensively analysed, and have a lot of physical evidence that support the network data. The underlying assumption for methods that use PPIs is guilt by association, in which the modular nature of the network is explained by the functional relation of the components. Hence, the distance between candidate genes and known disease genes in the PPI is key for multiple approaches. Common measures for the proximity of elements in a PPI are direct connection, shortest path, and diffusion kernels (idea that can initially be expanded to use random walkers with restart, and further expanded as propagation flow). Modern approaches based solely on PPI networks were proposed by Oti et al [24], which only uses direct neighbours, Köhler et al [13] , which includes random walkers with restart, and Navlakha et al [23], which includes propagation flow and clustering techniques. Another approach is to include phenotypic

information in addition to the PPI networks as proposed by Lage et al [14], Wu et al [39, 40], Care et al [7], Chen et al [8], Li et al [15], and Vanunu et al [35]. The last approach integrates multiple networks mainly by ranking the genes in different contexts, and then combining the results. Genetic approaches that study complex human diseases have changed from family based linkage studies, that mapped genetic diseases to population based association studies, which are mostly only able to explore single genes. The availability of high throughput gene analysis platforms lead to large scale Genome-Wide Association Studies (GWAS), which serve as basis for multiple complex disease analysis, such as pathway analysis. Modern approaches are multivariate, such as the two-stage approach by Wang et al [36] and Luo et al [17], supervised PCA by Chen et al [9], logistic kernel machines by Liu et al [16] and Wu et al [38], and topological based analysis by Massa et al [19], Martini et al [18] and Tarca et al [33] There are currently 7633 known mendelian disorders, for the sake of simplicity, in this document we consider each of the disorders as a different genetic disease. From these diseases, 5712 have some known molecular basis, which leaves 1921 diseases without any known molecular basis (orphan diseases), as 5th of April of 2016 (data available publicly through OMIM's morbidmap database [20]). Most network based methods for disease prediction are unable to predict genes for orphan diseases, and can only predict with a reasonable accuracy genes for diseases with many known genes[23]. Effective gene prediction techniques can help in the identification of the molecular basis of genetic diseases. The discovery of molecular basis of mendelian diseases allows diagnosis, prognosis, and therapy development for these diseases[4].

Chapter 3

Game Theory background

In this chapter, we briefly introduce the basic concepts of classical and evolutionary game theory, for more detail about these topics the reader is referred to (Weibull[37]; Leyton-Brown and Shoham[31]; Sandholm[29]).
A few cornerstones in game theory:

- 1921-1928: John von Neumann and Emile Borel give the first formulation of a mixed strategy along with the idea of finding minimax solutions of normal-form games
- 1944, 1947: John von Neumann and Oskar Morgenstern published "Theory of Games and Economic Behavior"
- 1950-1953: In four papers John Nash made important contributions to both non-cooperative game theory and to bargaining theory
- 1972-1982: John Maynard Smith applies game theory to biological problems thereby founding "evolutionary game theory"
- late 1990: Development of algorithmic game theory

3.1 Classical Game Theory

Game theory is a mathematical science that studies and analyzes the individual decisions of a subject in situations of conflict or strategic interaction with other rivals, aimed to maximum gain of each subject, such that the decisions of one can affect the results obtainable from the others and vice versa. It has been introduced by Von Neumann and Morgenstern(1944) in order to develop a mathematical framework able to model the essentials of decision making in interactive situations. In its normal form representation it consists of a finite set of players $I = \{1, .., n\}$, a set of pure strategies for each player $S_i = \{s_1, ..., s_m\}$, and a utility function $u_i : S_1 \times ... \times S_n \to \mathbb{R}$, which associates strategies to payoffs. Each player can adopt a strategy in

order to play a game and the utility function depends on the combination of strategies played at the same time by the players involved in the game, not just on the strategy chosen by a single player. An important assumption in game theory is that the players are rational and try to maximize the value of u_i

Furthermore, in non-cooperative games the players choose their strategies independently, considering what the other players can play and try to find the best strategy profile to employ in a game. A strategy si is said to be dominant if and only if:

$$u_i(s_i^*, s_{-i}) > u_i(s_i, s_{-i}), \qquad \forall s_{-i} \in S_{-i}$$

where S_{-i} represents all strategy sets other than player i's. As an example, we can consider the famous *Prisoner's Dilemma*, whose payoff matrix is shown in Table 1. Each cell of the matrix represents a strategy profile, where the first number represents the payoff of *Player 1* and the second is the payoff of *Player 2*, when both players employ the strategy associated with a specific cell(see Table 3.1).

Table 3.1: The Prisoner's dilemma

P1 / P2	*confess*	*don't confess*
confess	-5,-5	0,-6
don't confess	-6,0	-1,-1

In this game the strategy confess is a dominant strategy for both players and this strategy combination is the Nash equilibrium of the game.

3.1.1 Nash equilibria

Nash equilibria represent the key concept of game theory and can be defined as those strategy profiles in which each strategy is a best response to the strategy of the co-player and no player has the incentive to unilaterally deviate from his decision, because there is no way to do better. In many games, the players can also play mixed strategy, which are probability distributions over their pure strategies. Within this setting, the players can be defined as vector $x = (x_1, ..., x_m)$, where m is the number of pure strategies and each component x_h denotes the probability that player i chooses its h-th pure strategy.
For each player its strategy set is defined as a standard simplex:

$$\Delta = \{x \in \mathbb{R}^n : \sum_{h=1}^{m} x_h = 1, \; and \; x_h \geq 0 \; for \; all \; h \in x\}$$

Each mixed strategy corresponds to a point on the simplex and its corners correspond to pure strategies. In a two-players game we can define a strategy profile as a pair (p, q) where $p \in \Delta_i$ and $q \in \Delta_j$.
The expected payoff for this strategy profile is computed as follows: $u_i(p, q) = p \cdot A_i q$ and $u_j(p, q) = q \cdot A_j p$ where A_i and A_j are the payoff matrices of player i and player j, respectively.

The Nash equilibrium is computed in mixed strategies in the same way of pure strategies. It is represented by a pair of strategies such that each is a best response to the other. The only difference is that, in this setting, the strategies are probabilities and must be computed considering the payoff matrix of each player.
A game theoretic framework can be considered as a solid tool in decision making situations since a fundamental theorem by Nash[22] states that any normal-form game has at least one mixed Nash equilibrium, which can be employed as the solution of the decision problem.

3.2 Evolutionary Game Theory

Evolutionary game theory has been introduced by Smith and Price[32] overcoming some limitations of traditional game theory, such as the hyperrationality imposed on the players. In fact, in real life situations the players choose a strategy according to heuristics or social norms (Szabó and Fath 2007). It has been introduced in biology to explain the evolution of species. In this context, strategies correspond to phenotypes(traits or behaviors), payoffs correspond to offsprings, allowing players with a high actual payoff (obtained thanks to their phenotype) to be more prevalent in the population. This formulation explains natural selection choices among alternative phenotypes based on their utility function. This aspect can be linked to rational choice theory, in which players make a choice that maximizes their utility, balancing cost against benefits (Okasha and Binmore 2012).

This intuition introduces an inductive learning process, in which we have a population of agents which play games repeatedly with their neighbors. The players, at each iteration, update their beliefs on the state of the game and choose their strategy according to what has been effective and what has not in previous games. The strategy space of each player i is defined as a mixed strategy profile x_i, as defined in the previous section, which lives in the mixed strategy space of the game, given by the Cartesian product:

$$\Theta = \times_{i \in I} \Delta_i$$

The expected payoff of a pure strategy e^h in a single game is calculated as in mixed strategies. The difference in evolutionary game theory is that a player can play the game with all other players, obtaining a final payoff which is the

sum of all the partial payoff obtained during the single games. We have that the payoff relatives to a single strategies is: $u_i(e_i^h) = \sum_{j=1}^{n}(A_{ij}x_j)_h$ and the average payoff $u_i(x) = \sum_{j=1}^{n} x_i^T A_{ij} x_j$ where n is the number of players with whom the games are played and A_{ij} is the payoff matrix between player i and j.
Another important characteristic of evolutionary game theory is that the games are played repeatedly. In fact, at each iteration a player can update its strategy space according to the payoffs gained during the games. He can allocate more probability to the strategies with high payoff until an equilibrium is reached. In order to find those states that correspond to the Nash equilibria of the games, the replicator dynamic equation is used (Taylor and Jonker 1978):

$$\dot{x} = [u(e^h, x) - u(x, x)] \cdot x^h \qquad \forall x \in x$$

which allows better than average strategies (best replies) to grow at each iteration. The following theorem states that with equation 5 it is always possible to find the Nash equilibria of the games (see Weibull 1997 for the proof).

Theorem 1. *A point $x \in \Theta$ is the limit of a trajectory of equation (?) starting from the interior of Θ if and only if x is a Nash equilibrium.*
Further, if point $x \in \Theta$ is a strict Nash equilibrium, then it is asymptotically stable, additionally implying that the trajectories starting from all nearby states converge to x.

3.3 Evolutionary Games and Constrained Clustering

The *"Classical"* Clustering Problem: given a set of n objects and a nxn matrix A of pairwise similarities (we can consider this information as the description of a weighted graph), the goal is to partition the input objects (the vertices of the graph) into clusters.
For the widely accepted (informal)definition of a *"cluster"*, it should satisfy two criteria:

- External criterion: all "objects" outside a cluster should be highly dissimilar to the ones inside
- Internal criterion: all "objects" inside a cluster should be highly similar to each other

A game-theoretic approach to identify the best cluster in a dataset, available in form of graph, is the **Dominant Sets** technique. The formal definition of cluster proposed, called *"dominant-set"*, generalises the classical

graph-theoretic notion of a maximal clique to edge-weighted graphs.
The advantages of this game-theoretic perspective are many, making Dominant Sets a very general framework, which can be applied in a wide range of scenarios and allows the handling weighted and unweighted data.

3.3.1 Dominant set

In this section, we briefly introduce the basic concepts of dominant set, for a more detailed analysis of these topics referred to [25]

Let $S \subseteq V$ be a non-empty subset of vertices, and $i \in S$.
The *(average) weighted degree* of i w.r.t. S is defined as:

$$awdeg_S(i) = \frac{1}{|S|} \sum_{j \in S} a_{ij}$$

Observe that $awdeg_i(i) = 0$ for any $i \in V$. Moreover, if $j \notin S$, we define:

$$\phi_S(i,j) = a_{ij} - awdeg_S(i)$$

Intuitively, $\Phi_S(i,j)$ measures the similarity between vertices j and i, with respect to the (average) similarity between vertex i and its neighbors in S.

The weight of i w.r.t. S is defined as:

$$w_S(i) = \begin{cases} 1, & if\ |S| = 1 \\ \sum_{j \in S \setminus \{i\}} \ \phi_{S \setminus \{i\}}(i,j)\ w_{S \setminus \{i\}}(j), & otherwise \end{cases}$$

Further, the total weight of S is defined as: $W(S) = \sum_{i \in S} w_S(i)$

Intuitively, $w_S(i)$ gives us a measure of the overall *(relative) similarity* between vertex i and the vertices of $S \setminus \{i\}$ with respect to the overall similarity among the vertices in $S \setminus \{i\}$.
This allows us to get to the formal definition of Dominant Set:

Definition 1. *(Pavan and Pelillo, 2003, 2007)*

A non-empty subset of vertices $S \subseteq V$ such that $W(T) > 0$ for any non-empty $T \subseteq S$, is said to be a dominant set if:

1. *$w_S(i) > 0$, for all $i \in S$ (internal homogeneity)*
2. *$w_{S \cup \{i\}}(i) < 0$, for all $i \notin S$ (external homogeneity)*

Dominant set $\equiv$ Cluster

We conclude this brief presentation of this framework with the following important result:

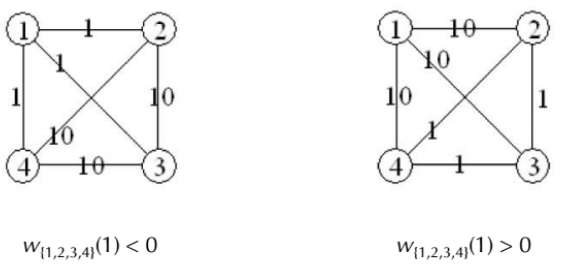


Figure 3.1: Two example of edge-weigthed graph with different value of $w_S(1)$

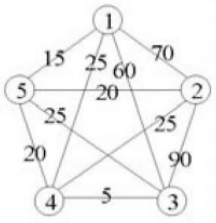


Figure 3.2: Example of Dominat Set

Theorem 2. *(Torsello, Rota Bulò and Pelillo, 2006)*
Evolutionary stable strategies of the clustering game with affinity matrix A are in one-to-one correspondence with dominant sets.

3.3.2 Constrained Dominant set

As we saw a dominant set is the best cluster, which is a subset of the vertices of a (possibly weighted) graph with specific properties, but we have no prior information about membership(or not) of a vertex in the selected cluster.
In some applications could be required that a specific vertex have to belong to the cluster, or better, that we are looking for the best cluster in which this vertex belongs.
In this most recent work[41] it has been proposed a new approach based on some properties of dominant sets. In particular it shows that by properly controlling a regularization parameter, which determines the structure and the scale of the underlying problem, it is possible to extract groups of dominant-set(clusters) which are constrained to contain user-selected elements(vertices).

If we consider the previous formulation of dominant set equivalent to the following linearly-constrained quadratic optimization problem

$$\begin{aligned} maximize \;\; & f(x) = x^T A x \\ subject\ to \;\; & x \in \Delta \end{aligned} \tag{3.1}$$

where Δ is the standard simplex of $\mathbb{R}^n$.

In [41] a connection is established between dominant sets and the local solutions of 3.1. In particular, it is shown that if S is a dominant set then its *"weighted characteristics vector"* which is the vector of Δ defined as

$$x_i = \begin{cases} \dfrac{w_s(i)}{W(S)}, & if \;\; i \in S \\ 0, & otherwise \end{cases}$$

is a *strict local solution* of 3.1. Conversely, under mild conditions, it turns out that if x is a (strict) local solution of program 3.1 then its *"support"*:

$$\sigma(x) = \{i \in V \;:\; x_i > 0\}$$

is a dominant set.
By virtue of this result, we can find a dominant set by first localizing a solution of program 3.1 with an appropriate continuous optimization technique, and then picking up the support set of the solution found.

Replicator Dynamics

A simple and effective optimization algorithm to extract a dominant set from a graph is given by the so-called *replicator dynamics*, developed and studied in evolutionary game theory, which are defined as follows:

$$x_i^{(t+1)} = x_i^{(t)} \frac{(Ax^{(t)})_i}{(x^{(t)})^T A x^{(t)}} \qquad for \;\; i = 1 \ldots n$$

In case of *constrained dominant set*, we want to specify an initial subset of vertices so in this work[41] the problem3.1 it has been reformulated as following:

Given a subset of vertices $S \subseteq V$ and a parameter $\alpha > 0$, define the following new parameterized family of quadratic programs:

$$\begin{aligned} maximize \;\; & f_S^{\alpha}(x) = x^T (A - \alpha \hat{I}_S) x \\ subject\ to \;\; & x \in \Delta \end{aligned} \tag{3.2}$$

where $\hat{I}_S$ is the $n \times n$ diagonal matrix whose diagonal elements are set to 1 in correspondence to the vertices contained in $V \setminus S$ and to zero otherwise. In other words, assuming for simplicity that S contains, say, the first k vertices of V, we have:

$$\hat{I}_S = \begin{pmatrix} 0 & 0 \\ 0 & I_{n-k} \end{pmatrix}$$

where I_{n-k} denotes the $(n-k) \times (n-k)$ principal submatrix of the $n \times n$ identity matrix I indexed by the elements of $V \setminus S$. Accordingly, the function f_α^S can also be written as follows

$$f_\alpha^S = x^T A x - \alpha x_S^T x_S$$

x_S being the $(n-k)$-dimensional vector obtained from x by dropping all the components in S.
Thanks to this new formulation of the optimization problem, without going into further details, if we consider a new matrix $\hat{A}$ equal to $A - \alpha \hat{I}_S$ and α greater than the largest eigenvalue of $A_{V \setminus S}$, the original matrix without the rows and columns related to the initial vertices S: it has been proved that the dominant set of the graph having A as adjacency matrix, contains at least one vertex of S. By re-applying the algorithms excluding the already extracted vertices, we obtain all the cluster with at least one element of S.

Infection and immunization

In order to overcome computational problems of which replicator dynamics are affected, we review a new class of evolutionary dynamics inspired by *infection and immunization* processes. For a complete overview of the topic we refer to [5]

These dynamics are built upon a central paradigm of evolutionary game theory called invasion barrier, the main concept is the following: Consider the set of all the populations not in equilibrium. For any of these x there exists at least one mixed strategy y that is a better response to x than x to itself. In this case we say that x has no invasion barrier against y, therefore if small share of mutant agents *"infect"* the current population, namely play an *"infective strategy"* y, they will spread until the invasion barrier against them becomes positive. This until the new population turns out to be *"immune"* against the "infective strategy" y.
This process remembers how a vaccine works: a small share of virus is introduced in a body in order to lead its immune system to prevent future infections. The authors of this recent class of evolutionary game dynamics propose to iterate this process of infection and immunization in order to obtain a population for which no infective strategy can be found anymore, because in that case a Nash equilibrium has been reached. In their work they provide a formal proof that fixed points of these dynamics are Nash equilibria and vice versa, independently from the way they select infective strategies at each iteration.

3.3.3 Polymatrix Games and Consistent Labeling

In a polymatrix game there are n players each of whom can use m pure strategies and for each pair (i, j) of players there is an $m \times m$ payoff matrix

A_{ij}. Therefore we are able to model situations in which each player is playing, at the same time, in n different games with his mixed strategy (among m strategies).
The payoff of player i for the strategy combination $s_1, \ldots, s_n$ is given by

$$u_i(s_1, \ldots s_n) = \sum_j A^{ij}_{s_i s_j}$$

The number of payoff values required to represent such a game is $O(n^2m^2)$ and the problem of finding a Nash equilibrium in a polymatrix game is PPAD-complete, so we adopted a slightly different approach to model this polymatrix game.

3.3.4 Relaxation Labeling Processes

A large class of problems can be formulated in terms of labels to objects: problems involving involve a set of *objects* $B = \{b_1, \ldots, b_n\}$ and a set of *labels* $\Lambda = \{\lambda_1, \ldots, \lambda_n\}$. The purpose is to label each object of B with one label of Λ.

In their work of R.Hummel and S.Zucker[11] have defined a process to solve problems like these showing that the problem of finding *consistent labeling* is equivalent to solving a *variational inequality*.
The general structure of relaxation labeling is motivated by the decomposition of complex computations into a network of simpler ones and the usage of context in resolving ambiguities.

To do this, two sources of information are exploited:

1. Local measurements which capture the salient features of each object viewed in isolation.

2. Contextual information, expressed in terms of a real-valued $n^2 \times m^2$ matrix of compatibility coefficients $R = \{r_{ij}(\lambda, \mu)\}$.

Where the coefficient $r_{ij}(\lambda, \mu)$ measures the strength of compatibility between the two hypotheses: "b_i is labeled λ" and "b_i is labeled μ".
It is generally possible to construct, for each object b_i, a vector $p_i = (p_i(1), \ldots, p_i(m))^T$ with $p_i(\lambda) > 0 \ \ \forall\lambda$ and $\sum_\lambda p_i(\lambda) = 1$.
Each p_i can thus be interpreted as the a priori (non-contextual) probability distribution of labels for b_i. By simply concatenating $p_1, \ldots, p_n$ we obtain an *initial weighted labeling assignment* for the objects of B that will be denoted by $p^{(0)} \in \mathbb{R}^{nm}$. The space of weighted labeling assignments is:

$$\mathbb{IK} = \underbrace{\Delta \times \ldots \times \Delta}_{m-times}$$

where each Δ is the standard simplex of $\mathbb{R}^n$. Vertices of $\mathbb{IK}$ represent unambiguous labeling assignments.
A relaxation labeling process takes the initial labeling assignment $p^{(0)}$ as input and iteratively updates it taking into account the compatibility matrix R.
The *compatibility matrix* is represented by a four-dimensional matrix of real-valued nonnegative compatibility coefficients $r_{ij}(\lambda, \mu)$. High values correspond to compatibility and low values correspond to incompatibility of the hypotheses.

Rosenfeld, Hummel, and Zucker update rule

In order to eventually achieve global consistency, Rosenfeld, Hummel and Zucker introduced[28] heuristically the following update rule:

$$p_i^{(t+1)}(\lambda) = \frac{p_i^{(t)}(\lambda) q_i^{(t)}(\lambda)}{\sum_\mu p_i^{(t)}(\mu) q_i^{(t)}(\mu)}$$

where the denominator is simply a normalization factor, and

$$q_i^{(t)}(\lambda) = \sum_j \sum_\mu r_{ij}(\lambda, \mu) p_i^{(t)}(\mu)$$

represents a "contribution" function that measures the strength of support that context gives at time t to the hypothesis "b_i is labeled λ". See Pelillo[26] for a rigorous derivation of this rule in the context of a formal theory of consistency.

The process is stopped when some termination condition is met (e.g. when the distance between two successive labelings becomes negligible or, more commonly, after a fixed number of iterations) and the final labeling is then used to label the objects according to a maxima selection criterion.
The preceding formulas are those originally proposed by Rosenfeld, Hummel and Zucker[28] which, despite their completely heuristic derivation, have recently been shown to possess interesting dynamical properties and to be intimately related to Hummel and Zucker's[11] theory of consistency.

Hummel and Zucker's Consistency

In 1983, Bob Hummel and Steve Zucker developed[11] an elegant theory of consistency in labeling problem. By analogy with the unambiguous case, which is easily understood, they define a weighted labeling assignment $p \in \mathbb{IK}$ **consistent** if:

$$\sum_\lambda p_i(\lambda) q_i(\lambda) \geq \sum_\lambda v_i(\lambda) q_i(\lambda) \qquad i = 1, \ldots, n$$

for all labeling assignments $v \in \mathbb{IK}$. If strict inequalities hold for all $v \neq p$, then p is said to be **strictly consistent**. Generalization of classical constraint satisfaction problems.

Theorem 3. *Theorem (Hummel and Zucker, 1983)*
A labeling $p \in \mathbb{IK}$ is consistent if and only if, for all $i = 1, \ldots, n$, the following conditions hold:

1. $q_i(\lambda) = c_i$ *whenever* $p_i(\lambda) > 0$
2. $q_i(\lambda) \leq c_i$ *whenever* $p_i(\lambda) = 0$ *for some constants* $c_1, \ldots, c_n$

Average Local Consistency

The notion of consistency suggests a measure for guiding the updating of a nearly consistent labeling into a consistent one: the *"average local consistency"* of a labeling $p \in \mathbb{IK}$ is defined as:

$$A(p) = \sum_i \sum_\lambda p_i(\lambda) q_i(\lambda)$$

Each of terms in the sum represents the local consistency of p from the viewpoint of an object weighted by the labeling weight. Since this measure should increase at each application of the update rule, meaning that we have a better("more consistent") labeling respect the previous one, it would seem natural to attempt to maximize it. The following results have led to a formalization of this concept:

Theorem 4. *Theorem (Hummel and Zucker, 1983)*
If the compatibility matrix R is symmetric, i.e., $r_{ij}(\lambda, \mu) = r_{ij}(\mu, \lambda)$, then any local maximizer $p \in \mathbb{IK}$ of A is consistent.

Theorem 5. *Theorem (Pelillo, 1997)*
The RHZ relaxation operator is a "growth transformation" for the average local consistency A, provided that compatibility coefficients are symmetric.

In other words, the algorithm strictly increases the average local consistency on each iteration, $A(p_{(t+1)}) > A(p_{(t)})$ for $t = 0, 1, \ldots$ until a fixed point is reached.

Theorem 6. *Theorem (Elfving and Eklundh, 1982; Pelillo, 1997) Let $p \in \mathbb{IK}$ be a strictly consistent labeling. Then p is an asymptotically stable equilibrium point for the RHZ relaxation scheme, whether or not the compatibility matrix is symmetric.*

3.3.5 Relaxation Labeling and Polymatrix Games

Referring back to the original wording in terms of evolutionary game theory, as observed by Miller and Zucker[21] the *consistent labeling problem* is equivalent to a *polymatrix game*.

Indeed, in such formulation we have:

- Objects = players
- Labels = pure strategies
- Weighted labeling assignments = mixed strategies
- Compatibility coefficients = payoffs

and:

- Consistent labeling = Nash equilibrium
- Strictly consistent labeling = strict Nash equilibrium

Further, the RHZ update rule corresponds to discrete-time multi-population "replicator dynamics".

Chapter 4

Our Method

4.1 Constrained Dominant Set as a Disease Module

As we pointed out in the definition of Disease Gene Prediction(sec.2.2), the goal is to unveil the genes in which the alteration of the disease modules cause the disease.
So we might observe disease modules as a subset of vertices in the genes network in which the cluster assumption holds.
It easily follows that if we have a network of genes whose interaction is represented by weighted edges, which are interpreted as similarity between them, then knowing some genes of a specific disease we can apply the Constrained Dominant Sets(sez.3.3.2) approach to identify its module.

4.1.1 Ranking score

In their work[41] it is explained that the components of the converged vector give us a measure of the participation of the corresponding vertices in the cluster. In our point of view this give us a measure to rank the selected genes.
Note that genes not belonging to constrained cluster have zero valued components, instead the selected ones have a positive values.

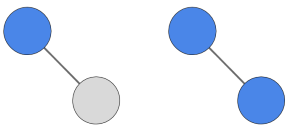
4.2 Relaxation Labeling as a diffusion method for DGP

Unlike the previous approach, there isn't a direct approach to map Disease Gene Prediction as a Relaxation Labelling Process(sez.3.3.4).
Noting that RLP requires objects, labels and a compatibility matrix, we again define genes as objects. Introducing also the only two possible labels:

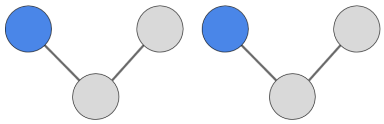
"*good candidate*" and "*bad candidate*", it remains to be defined the compatibility matrix.

Compatibility matrix generates the diffusion behaviour

The key point of this process is that the initial weighted label assignment might embody the a priori knowledge of the problem (e.g. it is more likely that an object is labeled in a specific way) and the compatibility matrix embodies the context in which the process is taking place.
In our formulation we don't consider previous knowledge to set up initial assignments (see definition of p vector sez.3.3.4).
Regarding the context, we define a scenario in which similar objects - genes - have to induce each other in order to have the same label. Supposing that a gene is already labeled, it should induce neighboring genes to became labeled:



But in the meantime, unlabeled genes should induce neighboring genes to remain unlabeled



Considering the proximity due to the similarity between genes, the predisposition to change label should take this into account.

This propagation mechanism is similar to the problem of the graph transduction, in which to propagate the information available at the labeled nodes to unlabeled ones in a *"consistent"* way, it has been formulated as a (polymatrix) non-cooperative game in this work by Erdem and Pelillo[10]

Recalling that the compatibility matrix R is a block matrix, whereby for each pair of objects we have, in our case with only two labels, a 2×2 matrix.

We denote by λ the label *"good candidate"* and with $\overline{\lambda}$ the alternative label *"bad candidate"* and with R_{ij} the block of the compatibility matrix related to the pair of objects i and j.
In the same manner as Erdem and Pelillo's [10], where i and j are different, R_{ij} is defined in the following way:

$$R_{ij} = \begin{cases} w_{ij} & 0 \\ 0 & w_{ij} \end{cases}$$

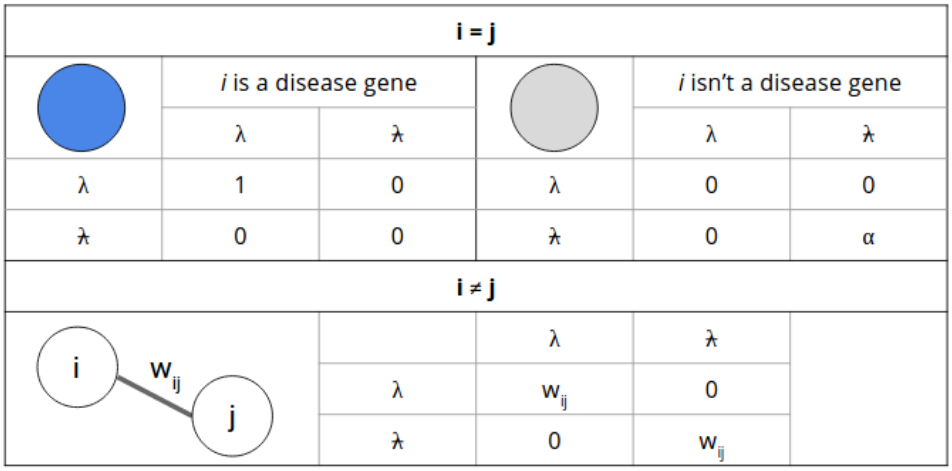
Where w_{ij} is the similarity between the i-th and j-th gene.
For our purposes we consider the case in which i and j are equal, so we are modelling the *"self-conviction"* of an object, we might have two situations: we are studying a disease for which it is known the implication of the gene i or otherwise we have no assumption about that. R_{ii} is defined in the following way:

$$R_{ii} = \begin{cases} \begin{matrix} 1 & 0 \\ 0 & 0 \end{matrix} & if\ i\ IS\ adiseasegene \\ \\ \begin{matrix} 0 & 0 \\ 0 & \alpha \end{matrix} & if\ i\ ISNOT\ adiseasegene \end{cases}$$

Notice that we have a parameter α, that we call *dissipation factor*, it regulates the predisposition of a gene to being labeled as *"good candidate"*. This is important because it could be different, depending on the disease, that there are other genes involved. In this case we can increase this parameter in order to slow down the spreading of the label *"good candidate"* in the network. This shall be clarified with the example in the following section with the toy graph.

The construction of R is summarized in the following table:

To conclude this section we define the *initial weighted label assignment*: this is a vector of n vectors of 2 components, so one vector for each object



(gene). $P^{(0)} = (P^{(0)}(\lambda), P^{(0)}(\overline{\lambda}))$ so then $P_i^{(0)}(\lambda)$ and $P_i^{(0)}(\overline{\lambda})$ show which label is assigned to i.

This vector is updated at each step by the RHZ rule(sez. 3.3.4) starting from $P^{(0)}$ with each $P_i^{(0)}(\lambda) = P_i^{(0)}(\overline{\lambda}) = 1/2$ meaning that we are not assuming a specific label at the starting point. This is the key difference between this approach and graph transduction formulation[10] in which the aim is to model the dynamics of the competition between two labels contending unlabeled objects.

Diffusion execution and optimization

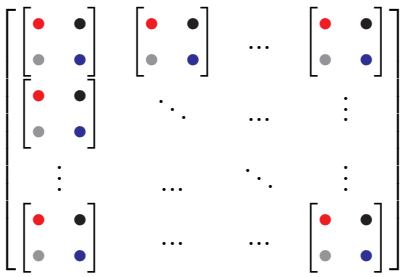
Defined the vector $P^{(0)}$ and the block matrix R we can execute the following algorithm:

1. Compute $Q^{(t)}$ (support vector) as defined in (sez.3.3.4) in function of $P^{(t)}$
2. Apply RHZ update rule (sez. 3.3.4) in function of $P^{(t)}$
3. Update $P^{(t+1)}$

The consistency theory ensures that this procedure converges into a consistent solution.
This means that for every object i: $(P_i^{(t)}(\lambda) = 1$ and $P_i^{(t)}(\overline{\lambda}) = 0))$ or $(P_i^{(t)}(\lambda) = 0$ and $P_i^{(t)}(\overline{\lambda}) = 1)$.
In this way we know which are the good candidates for a given disease.

Due to the spatial complexity of R we had to optimize the algorithm to run it: notice that R is a $n^n \times 2^2$ matrix with n equal of the number of genes($\backsim$ 16000).

The first consideration is that R is a sparse matrix, seeing the table 4.2 it is clear that R_{ij} is always a *diagonal matrix*. Furthermore in addition we can split R in four sub matrices containing only one element of the 2×2 block matrix R_{ij}(for every block).



We define R_L the submatrix of R with all the top-left elements of each block (red dots), R_N the submatrix of R with all the bottom-right elements of each block (blue dots) and both R_{NL} and R_{LN} the submatrices with top-right and bottom-left elements (gray and black dots). Notice that those are 4 $n \times n$ submatrices of R and, thanks to the previous consideration, we know that R_{NL} and R_{LN} are *null matrices*.
Recalling the formula to compute the support vector Q we notice that, for every object i the component related to λ is equal to

$$Q_i^{(t)}(\lambda) = \sum_j \sum_\mu r_{ij}(\lambda, \mu) P_j^{(t)}(\mu) \qquad \mu \in \{\lambda, \overline{\lambda}\}$$

Notice that we have only two labels: λ and $\overline{\lambda}$

$$Q_i^{(t)}(\lambda) = \sum_j r_{ij}(\lambda, \lambda) P_j^{(t)}(\lambda) + \sum_j r_{ij}(\lambda, \overline{\lambda}) P_j^{(t)}(\overline{\lambda})$$

But $r_{ij}(\lambda, \lambda)$ and $r_{ij}(\lambda, \overline{\lambda})$ are respectively R_L and R_{LN}

$$Q_i^{(t)}(\lambda) = \sum_j R_L(i, j) P_j^{(t)}(\lambda) + \sum_j R_{LN}(i, j) P_j^{(t)}(\overline{\lambda})$$

With $R_{LN}(i, j) = 0 \qquad \forall i, j$, so in the end:

$$Q_i^{(t)}(\lambda) = \sum_j R_L(i, j) P_j^{(t)}(\lambda)$$

that we can compute for every i as a vector operation:
$Q^{(t)}(\lambda) = R_L(i, j) \cdot P^{(t)}(\lambda)$
The same goes for $Q^{(t)}(\overline{\lambda}) = R_N(i, j) \cdot P^{(t)}(\overline{\lambda})$.

We finally note that, unless the main diagonal, R_L and R_N are equal to W the similarity matrix between pairs of genes. This allows us to optimize the allocation of R and the calculation of the update rule.

Ranking score

After having performed the optimized algorithm, we obtain the genes labeled as "good candidates", these with $P_i^{(t^*)}(\lambda) = 1$ at convergence. They can all also be depending of the known genes and of the value of the dissipation factor λ(e.g. if $\lambda = 0$ and there is at least one known gene, the spread will reach every gene).
The Disease Gene Prediction problem requires to rank the obtained genes. In order to have a score for each of them, we sum the values of $P_i^{(t)}(\lambda)$ at each iteration of updating process. Rather than just check $P_i^{(t)}(\lambda)$ at convergence, we are more interested in how it reaches the final value.
This shall be clarified with the example in the following section with the toy graph.

4.3 Examples with a toy graph

This section clarified the application of the two presented methods by a small graph with 20 nodes and 32 weighted edges (max: 0.2 min: 0.9 mean: 0.42) and several examples with simulated known genes.

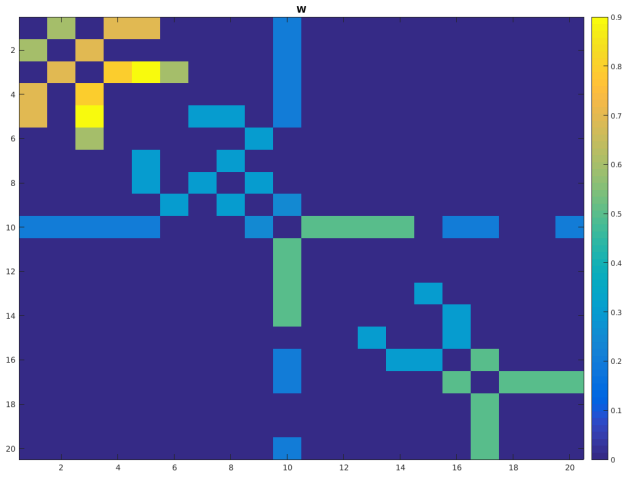


Figure 4.1: (weighted) adjacency matrix of the toy graph

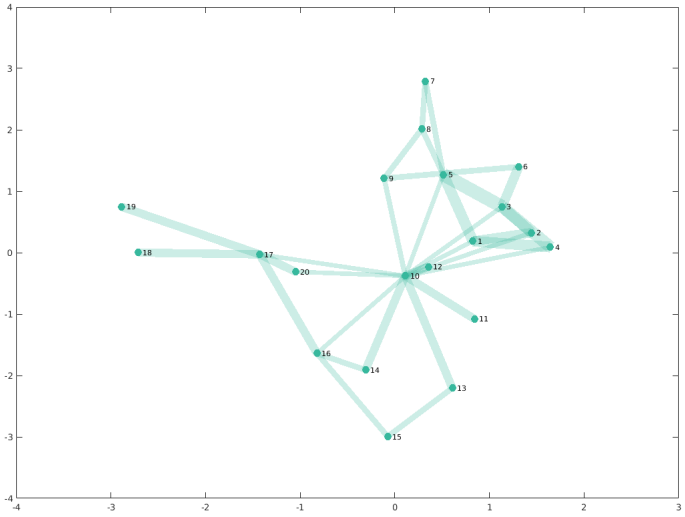


Figure 4.2: toy graph plotted by MATLAB with force-directed layout uses attractive forces between adjacent nodes and repulsive forces between distant nodes. Edges thickness is proportional to the similarity between nodes

Dissipation factor α for RLP

To understand the impact of the parameter α on the values of P during the relaxation labelling process, we see the values of $P^{(t)}(\lambda)$ at each step considering the toy graph and the vertex 1 as a known gene. The following images show the execution, until the convergence, for different values of α. We are interested only in the $P(\lambda)$ component of each object, because we know that $P^{(t)}(\overline{\lambda})$ is exactly $1 - P^{(t)}(\lambda)$ every time.
On the x axis there are the iteration steps. For each step we see the values of vector $P^{(t)}(\lambda)$ in a color scale in which dark blue means 0, yellow 1 and water green means 0.5.
In practise, they are concatenations of the different $P^{(t)}(\lambda)$ vectors.

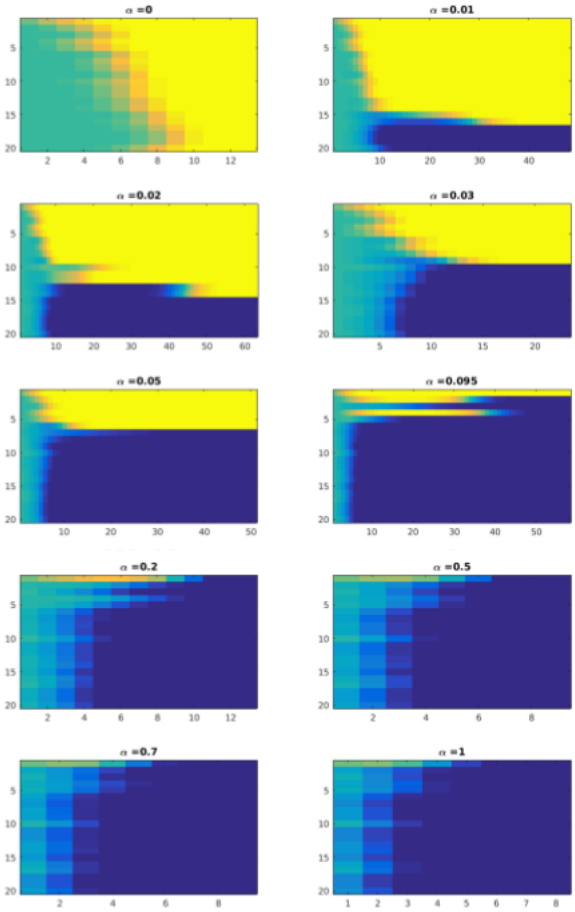
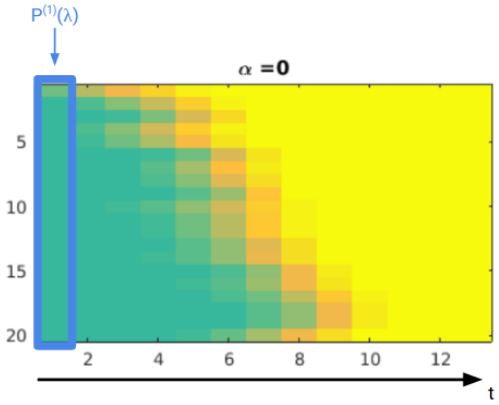


Figure 4.3: Diffusion dynamics on toy graph starting from vertex 1

Notice that different values of α involve different values of final $P^{(t^*)}(\lambda)$, and also different number of iterations needed to converge.

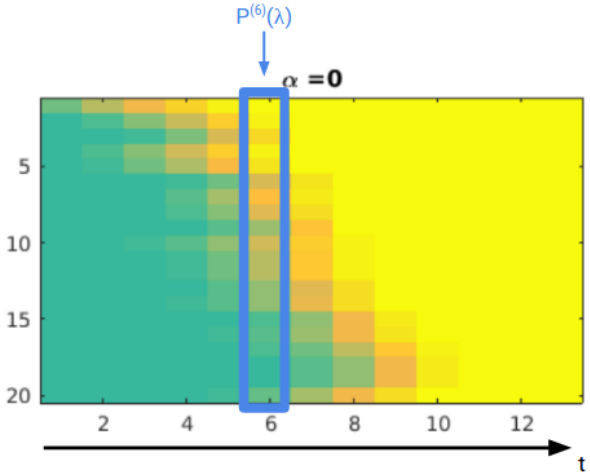
Ranking score of RL

Analyzing in detail the first case($\alpha = 0$) we notice that from the first update, $P^{(t)}(\lambda)$ increases and from the second onwards it starts to influence its neighbors to increase. see the following image:



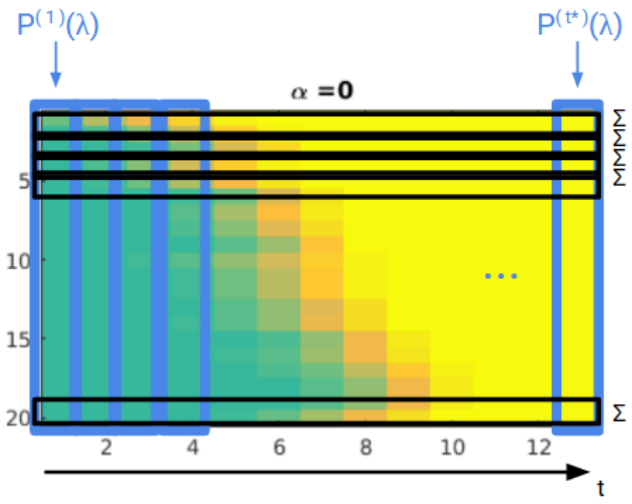
	λ	ƛ
1	0.59	0.41
2	0.50	0.50
3	0.50	0.50
4	0.50	0.50
5	0.50	0.50
6	0.50	0.50
7	0.50	0.50
8	0.50	0.50
9	0.50	0.50
10	0.50	0.50
11	0.50	0.50
12	0.50	0.50
13	0.50	0.50
14	0.50	0.50
15	0.50	0.50
16	0.50	0.50
17	0.50	0.50
18	0.50	0.50
19	0.50	0.50
20	0.50	0.50

After some steps:



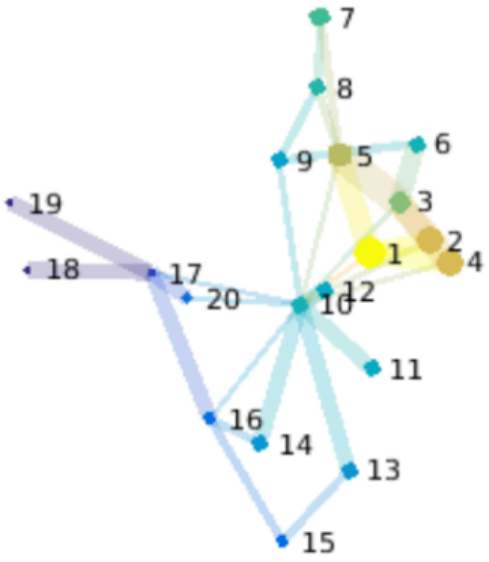
	λ	ƛ
1	0.97	0.03
2	0.88	0.12
3	0.75	0.25
4	0.88	0.12
5	0.81	0.19
6	0.59	0.41
7	0.66	0.34
8	0.62	0.38
9	0.55	0.45
10	0.63	0.37
11	0.58	0.42
12	0.58	0.42
13	0.55	0.45
14	0.55	0.45
15	0.51	0.49
16	0.52	0.48
17	0.51	0.49
18	0.50	0.50
19	0.50	0.50
20	0.53	0.47

In the last step each object came out as labeled. In order to clarify the ranking system for the relaxation labeling process described before [ref], this is the result obtained in this case:



RL score

1	12.46
2	11.73
3	11.29
4	11.74
5	11.53
6	10.80
7	11.04
8	10.88
9	10.56
10	10.81
11	10.66
12	10.66
13	10.40
14	10.41
15	9.97
16	9.97
17	9.70
18	9.47
19	9.47
20	9.99



RL score

1	12.46
4	11.74
2	11.73
5	11.53
3	11.29
7	11.04
8	10.88
10	10.81
6	10.80
11	10.66
12	10.66
9	10.56
13	10.40
14	10.41
20	9.99
15	9.97
16	9.97
17	9.70
18	9.47
19	9.47

RL vs CDS

In this section are shown some examples of executions on the toy graph of the Relaxation Labeling Process as Diffusion Method(RL) and Constrained Dominant Set as a Disease Module(CS) as previously described(sez. 3.3.4 and 3.3.2). For RL algorithm α is fixed and equal to 0 whereas for CDS is computed as described in [41].

```
RLdiffusion      L=[1]
(15 iterations)

ConstrainedDominatSet    S=[1]
alpha = 1.7227
(81 iterations of RD)    cluster 1 : 1    2    3    4    5
10
```

RelaxationLabelling		ConstrainedDominatSet	
1	12.46	1	0.59969
4	11.744	5	0.12994
2	11.727	4	0.12703
5	11.534	2	0.090267
3	11.293	3	0.039628
7	11.039	10	0.012042
8	10.879	6	0
10	10.805	7	0
6	10.799	8	0
11	10.656	9	0
12	10.656	11	0
9	10.558	12	0
14	10.412	13	0
13	10.399	14	0
20	9.9854	15	0
15	9.9747	16	0
16	9.9742	17	0
17	9.6957	18	0
18	9.4705	19	0
19	9.4705	20	0

```
RLdiffusion      L=[1   20]
(12 iterations)
```

```
ConstrainedDominatSet   S=[1   20]
alpha = 1.7139
(83 iterations of RD)   cluster 1 : 1   2   3   4   5
10
(108 iterations of RD)  cluster 2 : 16  17  18  19  20
```

RelaxationLabelling		ConstrainedDominatSet	
20	10.047	20	0.76515
1	9.4642	1	0.59922
17	8.7953	17	0.18318
4	8.764	5	0.13014
2	8.7495	4	0.12725
18	8.5662	2	0.090313
19	8.5662	3	0.039546
5	8.5559	16	0.015463
3	8.3189	18	0.013756
10	8.3107	19	0.013756
16	8.1561	10	0.012067
11	8.1136	6	0
12	8.1136	7	0
7	8.0538	8	0
14	8.0255	9	0
8	7.9207	11	0
13	7.8997	12	0
6	7.8477	13	0
9	7.747	14	0
15	7.7461	15	0

```
RLdiffusion     L=[1   16   20]
(12 iterations)

ConstrainedDominatSet   S=[1   16   20]
alpha = 1.7018
(86 iterations of RD)   cluster 1 : 1   2   3   4   5
10
(182 iterations of RD)  cluster 2 : 14  15  16  17  20
```

RelaxationLabelling		ConstrainedDominatSet	

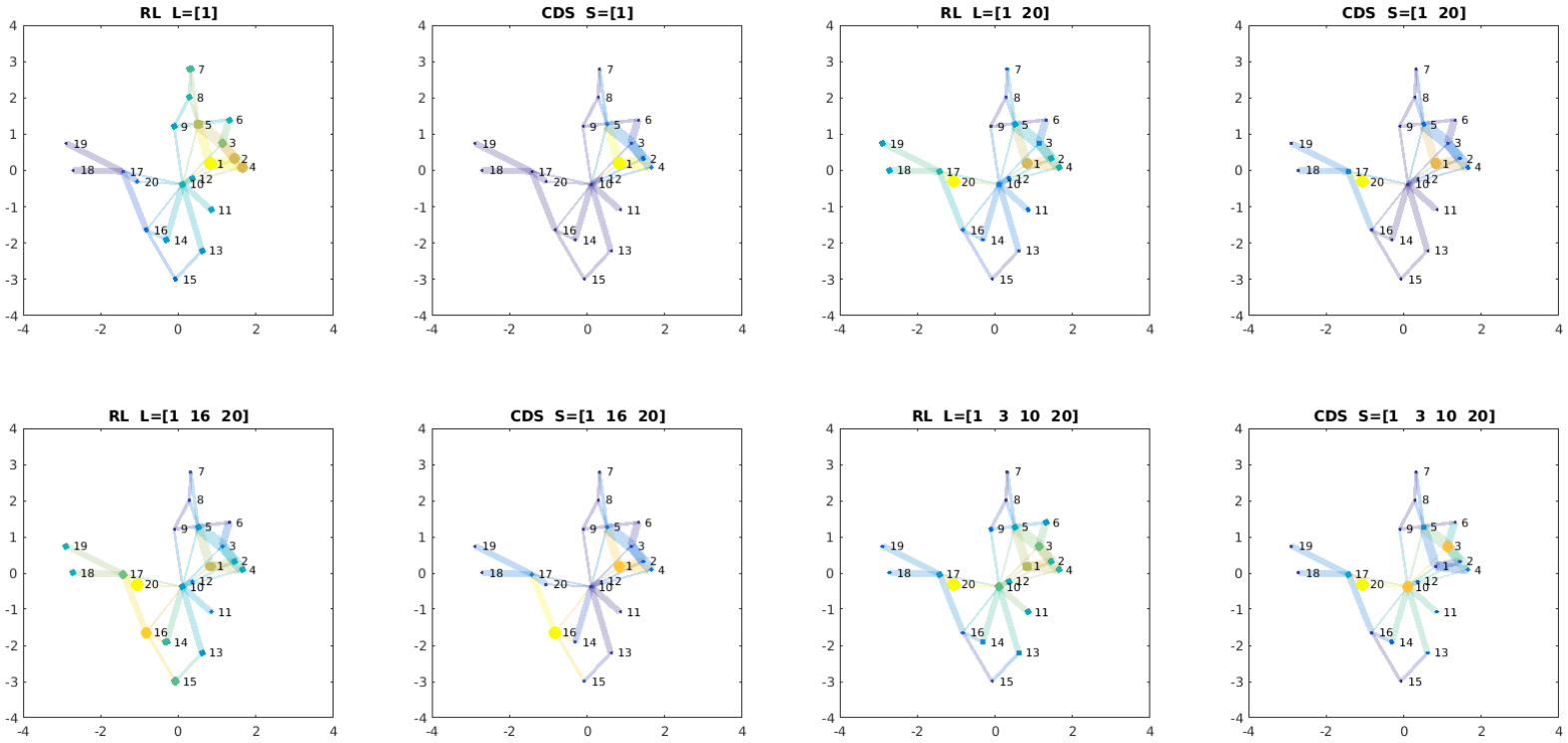
20	10.051	16	0.69091
16	9.793	1	0.59834
1	9.4678	17	0.15292
17	9.1583	5	0.13038
15	9.0911	4	0.12751
14	8.9893	2	0.090352
18	8.8207	15	0.05594
19	8.8207	14	0.055905
4	8.7808	20	0.0409
2	8.7686	3	0.039481
10	8.6356	10	0.012277
5	8.5747	6	0
13	8.5472	7	0
11	8.3617	8	0
12	8.3617	9	0
3	8.3408	11	0
7	8.0653	12	0
8	7.9536	13	0
6	7.886	18	0
9	7.8855	19	0

RLdiffusion L=[1 3 10 20]
(11 iterations)

ConstrainedDominatSet S=[1 3 10 20]
alpha = 0.9185
(78 iterations of RD) cluster 1 : 1 2 3 4 5
(65 iterations of RD) cluster 2 : 10 11 12 13 14
(80 iterations of RD) cluster 3 : 16 17 18 19 20

RelaxationLabelling		ConstrainedDominatSet	
20	9.0589	20	0.68374
1	8.5576	10	0.58594
3	8.3708	3	0.57278
10	8.3212	17	0.26131
4	8.2059	5	0.19849
2	8.2034	4	0.13689
11	8.0963	14	0.10213
12	8.0963	13	0.098128
5	8.0393	11	0.097398
6	7.9498	12	0.097398

17	7.916	2	0.074184
14	7.8705	16	0.018776
13	7.8348	18	0.01804
9	7.6574	19	0.01804
18	7.6372	1	0.012644
19	7.6372	6	0
16	7.6084	7	0
7	7.4456	8	0
8	7.4169	9	0
15	7.4088	15	0


RL L=[1]
CDS S=[1]
RL L=[1 20]
CDS S=[1 20]
RL L=[1 16 20]
CDS S=[1 16 20]
RL L=[1 3 10 20]
CDS S=[1 3 10 20]

Chapter 5

Experimental Results

5.1 Data sources

In order to apply the theoretical model on real data, in this section will be specified the used datasets. Basically, we have a *gene network* obtained by an abstraction of a PPI network, this PPI network(HIPPIE[30]) has the feature of having a scoring system that gives us the possibility to build a *weighted graph*. The information about known genes is given by the OMIM dataset[20], in which are shown pairs of disease-gene based on latest researches.
We take an older version of this dataset (2012) and a more recent one (2016), to validate the model. For the sake of simplicity we consider only diseases that had some known genes in 2013 and the same in 2016 plus at least a new one.
Given a disease, we extract the genes for which it was already known (in 2013) the involvement in the emergence of this disease and we use this information to run our methods on a gene network. Then we check the position of the new genes discovered in 2016.

5.1.1 HIPPIE - Human Integrated Protein-Protein Interaction rEference

For our purposes we can see HIPPIE[30] as a graph whose nodes are human proteins connected by edges if there is an interaction between them, as in other PPI datasets. The main feature of this one is , on the contrary, the presence of a scoring system that tells us how this interaction is relevant from the biological point of view.
More in details, general PPIs can be detected through different experimental approaches and are collected in several expert curated databases. In many analyses the reliability of the characterization of the interactions becomes important and it might be necessary to select sets of PPIs of different confidence levels. To this goal HIPPIE(Human Integrated Protein-Protein

Interaction rEference), provides a human PPI dataset with a normalized scoring scheme that integrates multiple experimental PPI datasets. HIPPIE's scoring scheme has been optimized by human experts and a computer algorithm to reflect the amount and quality of evidence for a given PPI and we show that these scores correlate to the quality of the experimental characterization.

Data are available via web tool or text file with the following fields:

1. UniProt identifier (first protein)
2. Entrez Gene identifier (gene that express the first protein)
3. UniProt identifier (second protein)
4. Entrez Gene identifier (gene that express the second protein)
5. Score
6. Comment field (summarizing the origin of the evidence)

The first and the third ones are unique ids related to the *interacting proteins*, whereas the second and the fourth ones indicate uniquely the genes that have coded them.
As we saw, a gene can encode more than one protein and this is the reason why there could be cases in which different interactions (among different proteins) have the same genes ids. E.g.: gene A encodes proteins $P1$ and $P2$ whereas gene B encodes $P3$ and $P4$, supposing $P1$ interacts with $P3$ and $P2$ with $P4$, and supposing the respective score 0.3 and 0.9, we should have the following entries:

$$A,\ P1,\ B,\ P3,\ 0.3$$

$$A,\ P2,\ B,\ P4,\ 0.9$$

From our point of view, we are interested on *"gene interaction"* so we should easily deduce that gene A and B interact, but with what score?
For the sake of simplicity we can say that higher score means higher biological importance, so we are interested on higher scores and for our abstraction these two entries are considered simply as:

$$A,\ B,\ 0.9$$

That is equivalent to say: an edge with weight of 0.9 between nodes A and B.

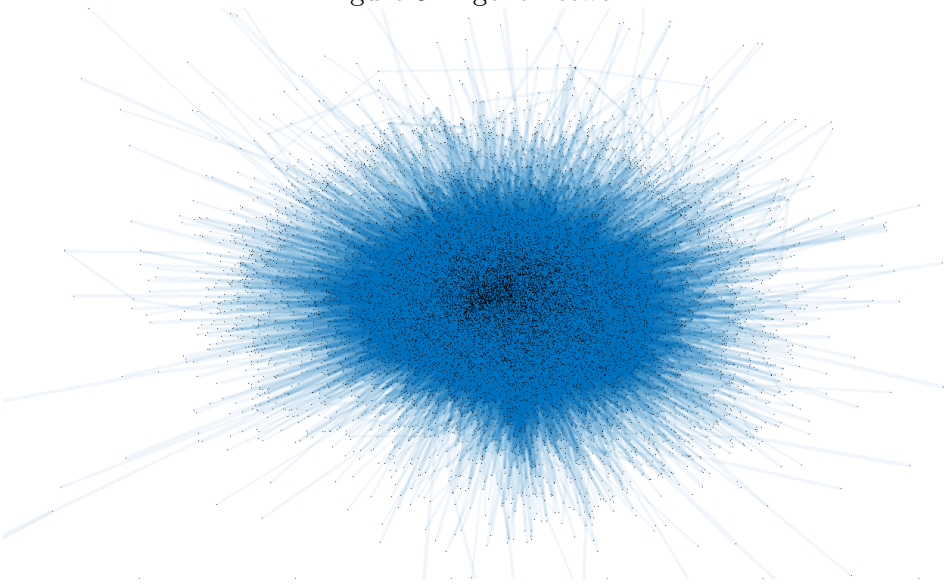
There may even be *self-loops*, in case of different proteins encoded by the same gene that interact (e.g. previous $P1$ and $P2$). These entries are discarded.

Data preparation

Taking into account the last considerations, we've built our gene network from HIPPIE removing invalid entries and dealing with repeated occurrences. The version of HIPPIE in the beginning (v.1.8 of 09/01/2015) has 239684 entries (Protein-Protein interactions), we removed 9250 self-loop, 690 repeated interactions (eventually with fewer score) and obtained 229731 interactions with 16480 unique ids (genes)

This means that our gene network is composed by 16480 nodes and 229731 weighted edges (min: 0.03 max: 1 mean: 0.65)

Figure 5.1: gene network



5.1.2 OMIM Morbid Map

OMIM (Online Mendelian Inheritance in Man)[20] is a continuously updated catalog of human genes and genetic disorders and traits, with particular focus on the molecular relationship between genetic variation and phenotypic expression. It is thus considered to be a phenotypic companion to the Human Genome Project.
The OMIM Morbid Map present the cytogenetic locations of genes and disorders that are described in OMIM. Only OMIM entries for which a cytogenetic location has been published in the cited references are represented in the Morbid Map. This dataset allows us to map each hereditary disease to one or more genes known to be involved in the emergence of that disease. For our purposes it provides labels for nodes of the gene network previously seen.

We use two versions of this dataset, the current one (2016) and an older one (2012).
The older one gives us the starting label that we exploit in our methods, in order to predict unknown genes, we compare these results with the updated version.

Data preparation

The 2012 version of OMIM Morbid Map contains 4285 pairs of gene-disease, we removed (214) pairs that refer to genes not present in our network. Current version contains 5251 pairs, of which we've removed for the same reasons 287.

From these associations pairs, we have :
2643 unique genes in 2012, 3253 (+ 610) in 2016
3346 unique diseases in 2012, 4258 (+ 912) in 2016

Where with unique genes we mean node of our gene network, and with unique diseases we mean possible labels.
From 2012 to 2016, on average, the number of genes related to a specific disease is increased of 0.2 genes (max +7) and it's decreased even of 13 genes. About this, it happens that a set of diseases it is discovered to be the same and so that it becomes a new unique id, or gene associations are arranged in order to reflect new disease nomenclature. Among diseases we consider only those with the same genes of 2012, still present in 2016, with at least a new one.
Only for the following 40, indicated by their OMIM ids, this consideration holds:
105200, 106210, 109400, 114480, 114550, 133100, 133200, 133239, 137580, 162091, 162900, 163200, 168600, 182940, 187500, 187950, 193530, 202400, 208150, 219000, 219700, 252010, 252011, 254450, 260350, 601001, 605027, 607174, 607785, 607831, 607948, 608089, 608133, 609135, 609423, 609821, 611162, 612376, 613065, 613254

5.2 Metrics

For each of these 40 diseases we take the known genes in 2013, we use them as seeds of our methods and run the algorithms on the gene network. We obtain a ranked list of genes that should be sorted depending on the probability of being unknown genes of that disease. To measure the performance of this result we verify the position of new genes(discovered in 2016), not used as seeds, in this list.

5.2.1 AUC n

The standard measure used to have a numerical value that immediately shows us the quality of this result is the *AUC (Area Under the Curve)*: the integral of the ROC curve, given by the *true positive rate (TPR)* and *false positive rate (FPR)*). The TPR defines how many correct positive results occur among all positive samples available during the test. FPR, on the other hand, defines how many incorrect positive results occur among all negative samples available during the test.

Considering a vector with the same length of the genes list with values equal to 1 in correspondence of target genes and 0 otherwise (ground truth), we have the same output of a binary classifier. Of course we removed the rows corresponding to seeds gene.
Values equal to 1 are equivalent to a TP (true positives) and 0s are FP(false positives).
The AUC value is the sum, starting from the first position, of the ratios between TPR and FPR updated by scrolling the list. When is equal to 1 means that there aren't FP. In our case we stop the evaluation of AUC after n step.
The rationale behind it is that we usually have few target genes and we hope that they are on the top of the list, differently we wouldn't have got a good result. So it's useless to go through the entire list.

5.3 Results

The tables below show the results of our approaches on selected diseases. The Known column shows the ones which are the seeds gene($Known = 1$) and targets($Known = 0$). *RL_pos* and *CDS_pos* show the positions obtained in the ranking by *RL* and *CDS* algorithms:

Disease 73 MIM: 105200

Entrez	Known	RL_pos	CDS_pos
335	1	10	2
4069	1	28	1
2243	1	4	3
567	0	822	426

RL_AUC100	RL_AUC1000	CDS_AUC100	CDS_AUC1000
0	0.179	0	0.575

Disease 96 MIM: 106210

Entrez	Known	RL_pos	CDS_pos
5080	1	2	1
26610	0	7739	8519

RL_AUC100	RL_AUC1000	CDS_AUC100	CDS_AUC1000
0	0	0	0

Disease 163 MIM: 109400

Entrez	Known	RL_pos	CDS_pos
5727	1	9	1
8643	0	4	5574
51684	0	9040	9513

RL_AUC100	RL_AUC1000	CDS_AUC100	CDS_AUC1000
0.485	0.4985	0	0

Disease 261 MIM: 114480

Entrez	Known	RL_pos	CDS_pos
5002	1	1	2
207	1	469	11
8438	1	2	1
999	1	196	17
5290	1	70	18
580	1	344	22
11200	1	64	15
9821	1	55	7
5888	1	27	21
3161	1	12	6
472	1	140	14
7251	1	203	8
841	1	180	16
7517	1	4	9
7157	1	1259	20
4835	1	3	4
8493	1	5	3
5245	1	210	12
79728	1	17	10
3845	1	53	13
83990	1	10	5
675	1	48	19

2099	0	5531	181

RL_AUC100	RL_AUC1000	CDS_AUC100	CDS_AUC1000
0	0	0	0.82

Disease 263 MIM: 114550

Entrez	Known	RL_pos	CDS_pos
841	1	82	4
5290	1	22	2
8312	1	39	8
324	1	59	6
7157	1	1934	7
1499	1	186	5
4233	1	52	3
5157	1	1	1
3482	0	9097	2577

RL_AUC100	RL_AUC1000	CDS_AUC100	CDS_AUC1000
0	0	0	0

Disease 570 MIM: 133100

Entrez	Known	RL_pos	CDS_pos
10019	1	1	1
2057	1	3	2
3717	0	199	5

RL_AUC100	RL_AUC1000	CDS_AUC100	CDS_AUC1000

0	0.802	0.96	0.996

Disease 572 MIM: 133200

Entrez	Known	RL_pos	CDS_pos
2707	1	1	1
2697	0	458	1726

RL_AUC100	RL_AUC1000	CDS_AUC100	CDS_AUC1000
0	0.543	0	0

Disease 573 MIM: 133239

Entrez	Known	RL_pos	CDS_pos
6049	1	2	2
50514	1	5	3
11178	1	1	1
7048	1	22	4
51741	1	232	5
1630	0	5858	1453

RL_AUC100	RL_AUC1000	CDS_AUC100	CDS_AUC1000
0	0	0	0

Disease 657 MIM: 137580

Entrez	Known	RL_pos	CDS_pos
1.148e+05	1	1	1
3067	0	10716	2125

RL_AUC100	RL_AUC1000	CDS_AUC100	CDS_AUC1000
0	0	0	0

Disease 1083 MIM: 162091

Entrez	Known	RL_pos	CDS_pos
4771	1	1	1
6598	0	8657	4388

RL_AUC100	RL_AUC1000	CDS_AUC100	CDS_AUC1000
0	0	0	0

Disease 1099 MIM: 162900

Entrez	Known	RL_pos	CDS_pos
2261	1	9	1
3265	1	38	3
5290	1	28	2
4893	0	123	3182

RL_AUC100	RL_AUC1000	CDS_AUC100	CDS_AUC1000

0	0.878	0	0

Disease 1102 MIM: 163200

Entrez	Known	RL_pos	CDS_pos
3265	1	11	2
3845	1	3	1
4893	0	39	3263

RL_AUC100	RL_AUC1000	CDS_AUC100	CDS_AUC1000
0.62	0.962	0	0

Disease 1200 MIM: 168600

Entrez	Known	RL_pos	CDS_pos
2629	1	4	2
6908	1	28	4
126	1	1	1
4137	1	22	3
6311	0	5640	4273

RL_AUC100	RL_AUC1000	CDS_AUC100	CDS_AUC1000
0	0	0	0

Disease 1450 MIM: 182940

Entrez	Known	RL_pos	CDS_pos

80199	1	1	1
81839	1	6	2
6347	1	4	3
57216	0	776	10845

RL_AUC100	RL_AUC1000	CDS_AUC100	CDS_AUC1000
0	0.225	0	0

Disease 1536 MIM: 187500

Entrez	Known	RL_pos	CDS_pos
182	1	2	3
23414	1	5	4
2627	1	1	1
1482	1	3	2
2626	0	9	7
6899	0	10353	4611

RL_AUC100	RL_AUC1000	CDS_AUC100	CDS_AUC1000
0.46	0.496	0.47	0.497

Disease 1548 MIM: 187950

Entrez	Known	RL_pos	CDS_pos
7066	1	1	1
10019	1	2	2
811	0	4677	565

RL_AUC100	RL_AUC1000	CDS_AUC100	CDS_AUC1000
0	0	0	0.436

Disease 1661 MIM: 193530

Entrez	Known	RL_pos	CDS_pos
2121	1	1	1
1.3288e+05	0	11681	13950

RL_AUC100	RL_AUC1000	CDS_AUC100	CDS_AUC1000
0	0	0	0

Disease 1723 MIM: 202400

Entrez	Known	RL_pos	CDS_pos
2244	1	2	2
2243	1	1	1
2266	0	4	3

RL_AUC100	RL_AUC1000	CDS_AUC100	CDS_AUC1000
0.97	0.997	0.98	0.998

Disease 1803 MIM: 208150

Entrez	Known	RL_pos	CDS_pos

5913	1	2	2
2.8549e+05	1	1	1
4593	0	4	5

RL_AUC100	RL_AUC1000	CDS_AUC100	CDS_AUC1000
0.97	0.997	0.96	0.996

Disease 2007 MIM: 219000

Entrez	Known	RL_pos	CDS_pos
80144	1	2	1
3.4164e+05	1	1	2
23426	0	13426	7979

RL_AUC100	RL_AUC1000	CDS_AUC100	CDS_AUC1000
0	0	0	0

Disease 2022 MIM: 219700

Entrez	Known	RL_pos	CDS_pos
7040	1	12	1
1080	1	158	2
2212	0	5079	1773

RL_AUC100	RL_AUC1000	CDS_AUC100	CDS_AUC1000
0	0	0	0

Disease 2564 MIM: 252010

Entrez	Known	RL_pos	CDS_pos
79133	1	7	6
4723	1	18	12
91942	1	4	4
4694	1	2	2
4724	1	11	16
4726	1	20	14
80224	1	1	1
4719	1	32	13
4709	1	6	5
4720	1	19	11
25915	1	8	8
51103	1	10	7
4729	1	12	10
1.2633e+05	1	9	15
29078	1	5	9
55572	1	3	3
4715	0	71	3091
4722	0	26	73

RL_AUC100	RL_AUC1000	CDS_AUC100	CDS_AUC1000
0.53	0.953	0.14	0.464

Disease 2565 MIM: 252011

Entrez	Known	RL_pos	CDS_pos
6389	1	4	1
6392	0	2	4244

RL_AUC100	RL_AUC1000	CDS_AUC100	CDS_AUC1000
0.99	0.999	0	0

Disease 2613 MIM: 254450

Entrez	Known	RL_pos	CDS_pos
3717	1	14	2
10019	1	3	1
4352	1	1	3
811	0	3944	597

RL_AUC100	RL_AUC1000	CDS_AUC100	CDS_AUC1000
0	0	0	0.404

Disease 2738 MIM: 260350

Entrez	Known	RL_pos	CDS_pos
7157	1	360	2
3845	1	1	1
4089	0	693	3120
6794	0	317	232

RL_AUC100	RL_AUC1000	CDS_AUC100	CDS_AUC1000
0	0.4965	0	0.3845

Disease 3779 MIM: 601001

Entrez	Known	RL_pos	CDS_pos
3861	1	2	1
3852	0	51	2

RL_AUC100	RL_AUC1000	CDS_AUC100	CDS_AUC1000
0.5	0.95	0.99	0.999

Disease 4313 MIM: 605027

Entrez	Known	RL_pos	CDS_pos
8438	1	1	1
5551	1	2	2
843	1	5	3
8915	0	687	5749
25788	0	621	8187

RL_AUC100	RL_AUC1000	CDS_AUC100	CDS_AUC1000
0	0.3475	0	0

Disease 4598 MIM: 607174

Entrez	Known	RL_pos	CDS_pos
5728	1	32	4
51684	1	2	2
4330	1	1	1
4771	1	27	3
5155	0	5051	3572

6605	0	2325	4574

RL_AUC100	RL_AUC1000	CDS_AUC100	CDS_AUC1000
0	0	0	0

Disease 4721 MIM: 607785

Entrez	Known	RL_pos	CDS_pos
4763	1	10	2
23092	1	1	1
5781	1	41	3
10962	0	15163	7272

RL_AUC100	RL_AUC1000	CDS_AUC100	CDS_AUC1000
0	0	0	0

Disease 4729 MIM: 607831

Entrez	Known	RL_pos	CDS_pos
54332	1	1	1
56704	0	7133	10638

RL_AUC100	RL_AUC1000	CDS_AUC100	CDS_AUC1000
0	0	0	0

Disease 4752 MIM: 607948

Entrez	Known	RL_pos	CDS_pos
3459	1	10	8
6347	1	15	4
1154	1	50	5
3431	1	12	2
3458	1	2	7
4159	1	1	1
30835	1	11	3
1.1461e+05	1	97	6
7097	0	240	90

RL_AUC100	RL_AUC1000	CDS_AUC100	CDS_AUC1000
0	0.761	0.11	0.911

Disease 4776 MIM: 608089

Entrez	Known	RL_pos	CDS_pos
2956	1	30	2
27030	1	3	1
999	1	55	3
5728	1	23	4
4437	0	87	111

RL_AUC100	RL_AUC1000	CDS_AUC100	CDS_AUC1000
0.14	0.914	0	0.89

Disease 4788 MIM: 608133

Entrez	Known	RL_pos	CDS_pos
5961	1	1	1
6094	0	2	2

RL_AUC100	RL_AUC1000	CDS_AUC100	CDS_AUC1000
0.99	0.999	0.99	0.999

Disease 5005 MIM: 609135

Entrez	Known	RL_pos	CDS_pos
3458	1	1	1
4683	0	10520	3001
5551	0	4194	3611
51119	0	10698	9143

RL_AUC100	RL_AUC1000	CDS_AUC100	CDS_AUC1000
0	0	0	0

Disease 5080 MIM: 609423

Entrez	Known	RL_pos	CDS_pos
6356	1	5	2
3811	1	4	3
3566	1	85	11
6347	1	19	10
3586	1	1	4
6349	1	3	1
6348	1	9	6

3458	1	2	5
3107	1	179	12
6387	1	10	7
3577	1	15	8
30835	1	31	9
7098	0	3200	4810

RL_AUC100	RL_AUC1000	CDS_AUC100	CDS_AUC1000
0	0	0	0

Disease 5164 MIM: 609821

Entrez	Known	RL_pos	CDS_pos
64805	1	4	1
5023	0	16467	3275

RL_AUC100	RL_AUC1000	CDS_AUC100	CDS_AUC1000
0	0	0	0

Disease 5429 MIM: 611162

Entrez	Known	RL_pos	CDS_pos
2213	1	14	7
3383	1	640	12
948	1	10	5
1154	1	36	8
2993	1	1	1
2995	1	2	2
1.1461e+05	1	92	9
6521	1	8	6

2532	1	3	3
1378	1	4	4
7124	1	169	10
4843	1	634	11
2212	0	317	81
2539	0	4805	1863
2994	0	18	2166
3043	0	2647	2199

RL_AUC100	RL_AUC1000	CDS_AUC100	CDS_AUC1000
0.2075	0.417	0.05	0.23

Disease 5696 MIM: 612376

Entrez	Known	RL_pos	CDS_pos
5914	1	10	1
4926	0	3833	3308

RL_AUC100	RL_AUC1000	CDS_AUC100	CDS_AUC1000
0	0	0	0

Disease 5934 MIM: 613065

Entrez	Known	RL_pos	CDS_pos
613	1	8	1
581	0	4235	470
2322	0	143	1636
4683	0	13408	3048
6886	0	10434	4609
6887	0	4908	4610

8021	0	10625	5213

RL_AUC100	RL_AUC1000	CDS_AUC100	CDS_AUC1000
0	0.143	0	0.0885

Disease 6027 MIM: 613254

Entrez	Known	RL_pos	CDS_pos
3458	1	1	1
7249	0	6485	4820

RL_AUC100	RL_AUC1000	CDS_AUC100	CDS_AUC1000
0	0	0	0

Disease 6177 MIM: 613659

Entrez	Known	RL_pos	CDS_pos
324	1	15	1
843	0	9847	754
1316	0	12206	1075
2064	0	1599	1547
2263	0	384	1685
3659	0	12511	2499
4595	0	320	3056
5290	0	1224	3543

RL_AUC100	RL_AUC1000	CDS_AUC100	CDS_AUC1000

0	0.18557	0	0.035286

Disease 6581 MIM: 614470

Entrez	Known	RL_pos	CDS_pos
4893	1	1	1
3845	0	49	2540

RL_AUC100	RL_AUC1000	CDS_AUC100	CDS_AUC1000
0.52	0.952	0	0

Disease 6606 MIM: 614519

Entrez	Known	RL_pos	CDS_pos
1284	1	7	2
1636	1	1	1
1282	0	25	5

RL_AUC100	RL_AUC1000	CDS_AUC100	CDS_AUC1000
0.76	0.976	0.96	0.996

In a visual way, following the same results:

Figure 5.2: AUC with n = 100. on the x-axis diseases and y-axis AUC values

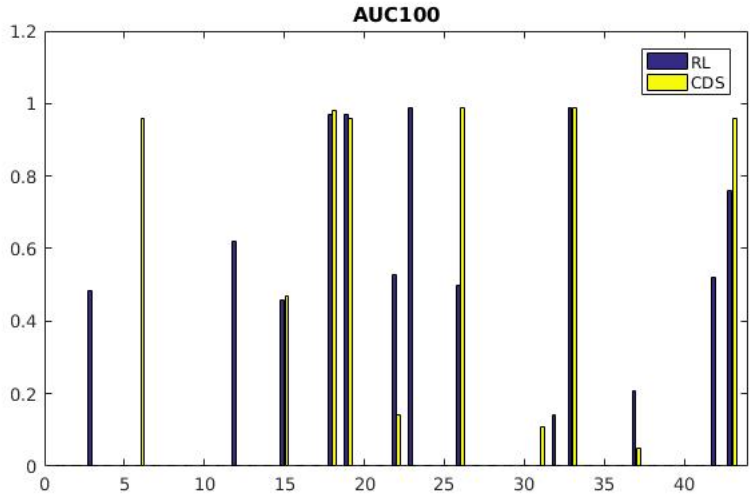
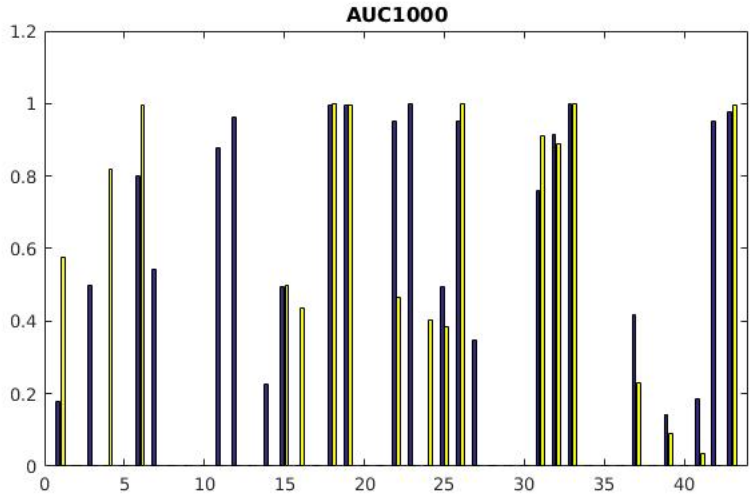


Figure 5.3: AUC with n = 1000. on the x-axis diseases and y-axis AUC values



As we can see the results are not uniform for all diseases, the reasons may be different and related to some biological assumptions. We are using the same methods for all. In the case of RL method, I remark that we fixed alpha to 0, but this value might need to be increased for some diseases. We see in detail some results to analyze their impacts on the performances:

5.3.1 Considerations on main results

MIM 608133 (RETINITIS PIGMENTOSA)

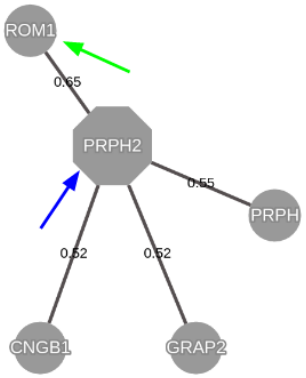
http://www.omim.org/entry/608133

Considering this hereditary disease that affects the retina, we know by the OMIM Morbid Map that in 2013 there was only one gene discovered (ID 5961) pointed by a blue arrow in the following figure. In 2016 it has been added a new gene (ID 6094) pointed by a green arrow. The figure shows the subnetwork of the starting gene (5961) with its own neighbors (including also the target gene).

MIM 608133 (RETINITIS PIGMENTOSA)

HIPPIE network query: 5961

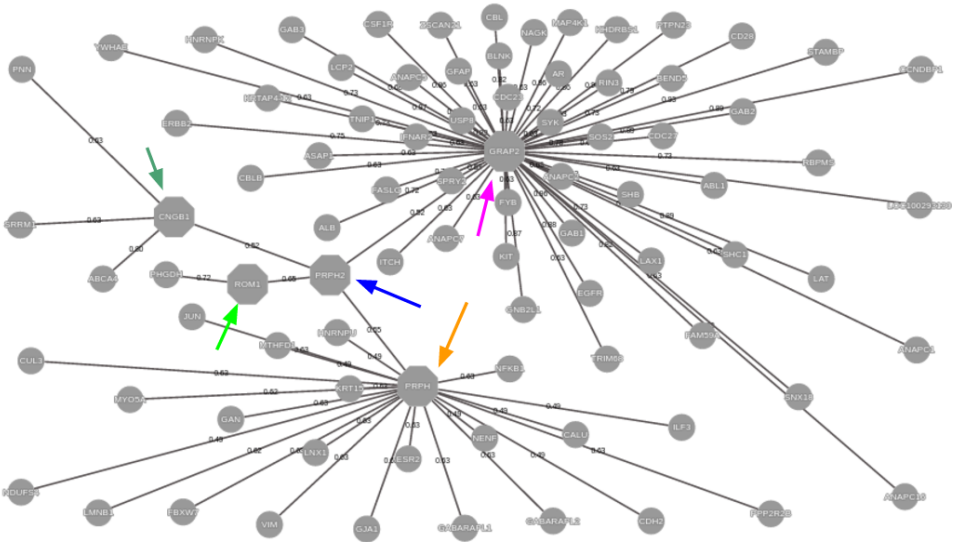
A - Entrez gene id	A - gene symbol	B - Entrez gene id	B - gene symbol	score
5961	PRPH2	6094	ROM1	0.65
5961	PRPH2	5630	PRPH	0.55
5961	PRPH2	9402	GRAP2	0.52
5961	PRPH2	1258	CNGB1	0.52



As we see by the following results the target gene is the first unknown gene in the ranking:

Entrez	Known	RL_pos	CDS_pos
5961	1	1	1
6094	0	2	2
1258	0	3	4
5630	0	5	3
9402	0	9	5

It's interesting to notice that also the other neighbors are in the top of the list, but with different positions. Although they are all directly related, some of them have other attached genes which dissuade them from becoming labeled. In the following figure those genes are included:



In this case and others like this, it seems that the methods follow the assumptions of exploiting the local information to reword possible good candidates.

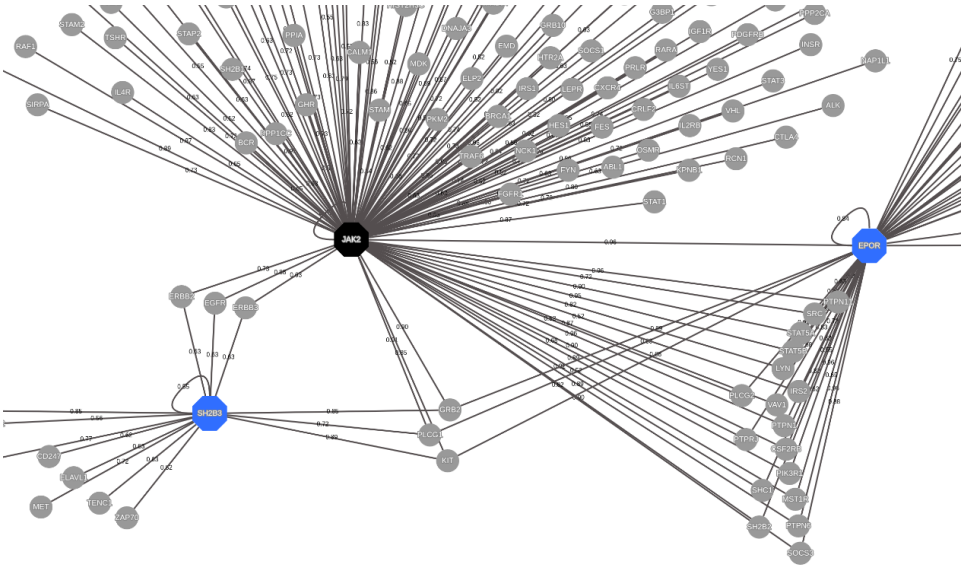
MIM 133100 (ERYTHROCYTOSIS, FAMILIAL, 1; ECYT1)

http://www.omim.org/entry/133100

This example shows a big difference between CDS and RL results, so much that CDS has an AUC100 of 0.96 (target gene is at 5th position) whereas RL ranks it at 199th place.

Entrez	Known	RL_pos	CDS_pos
10019	**1**	**1**	**1**
2057	**1**	**3**	**2**
3717	**0**	**199**	**5**

We might supposing that the high degree of the target gene (the black hexagon in the following figure) dissipates the diffusion during the RL process. It is instead clear that those three genes form a (generalization of a) clique that we might consider a disease module. This is very well exploited by CDS technique.

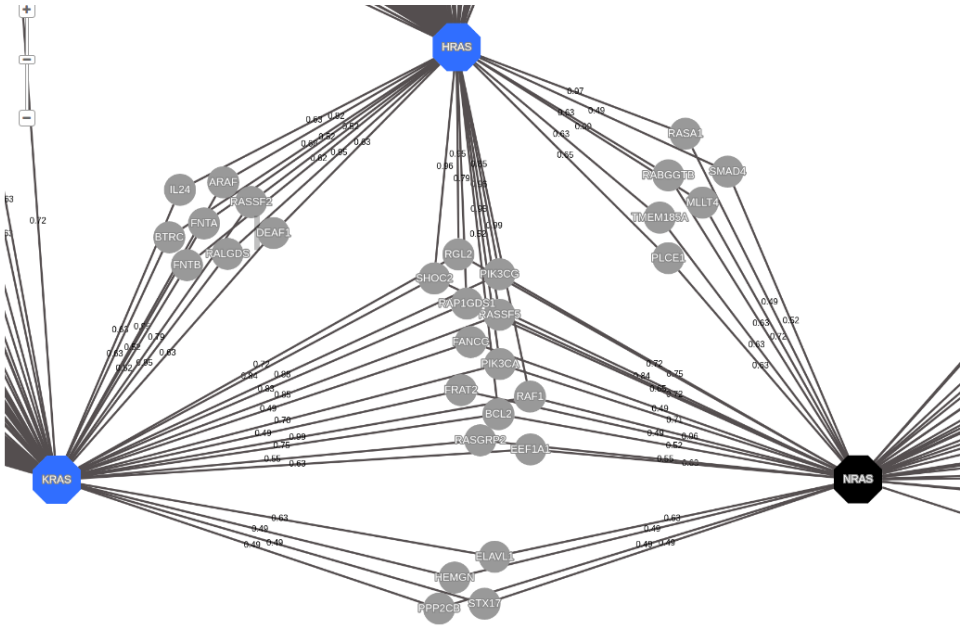


MIM 163200 (SCHIMMELPENNING-FEUERSTEIN-MIMS SYNDROME; SFM)

http://www.omim.org/entry/163200

Opposite situation is the following, in which RL gets 0.62 of AUC100 whereas CDS gets 0.0:

Entrez	Known	RL_pos	CDS_pos
3265	**1**	**11**	**2**
3845	**1**	**3**	**1**
4893	**0**	**39**	**3263**



MIM 613659 (GASTRIC CANCER, INTESTINAL, INCLUDED)

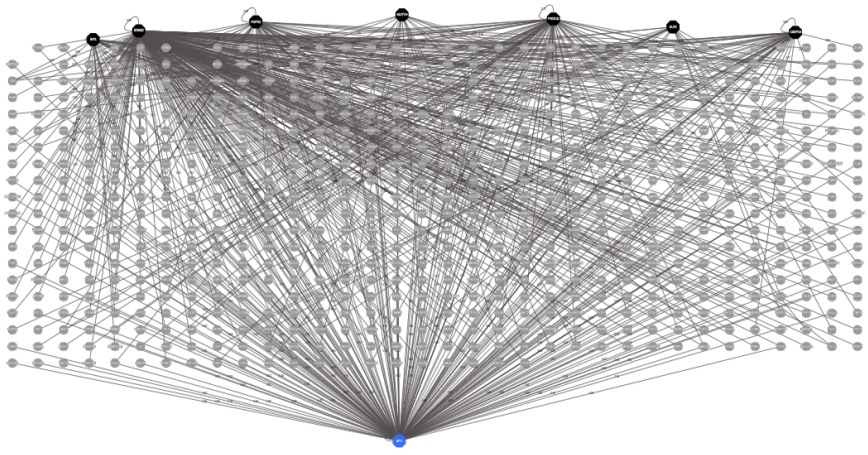
http://www.omim.org/entry/613659

If we want to consider a bad result for both, we use this type of cancer as an example. The only known gene in 2012 was ID: 324, then in 2016 7 new genes were added (324, 4595, 2263, 5290, 2064, 843, 1316)

Entrez	Known	RL_pos	CDS_pos	Zhou	Zhou+DSim
324	**1**	**15**	**1**	–	–
843	**0**	**9847**	**754**	**4592**	**6**
1316	**0**	**12206**	**1075**	**7605**	**11**
2064	**0**	**1599**	**1547**	**241**	**1**
2263	**0**	**384**	**1685**	**288**	**3**
3659	**0**	**12511**	**2499**	**4066**	**10**
4595	**0**	**320**	**3056**	**404**	**9**
5290	**0**	**1224**	**3543**	**745**	**4**

RL_AUC1000	0.18557
CDS_AUC1000	0.035286
Zhou_AUC1000	0.33314
Zhou+Dsim_AUC1000	0.99771

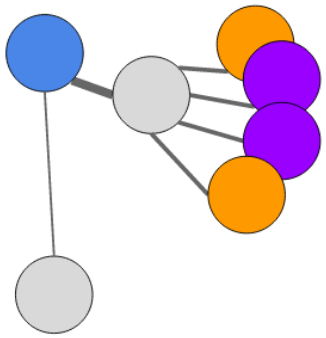
The reason why we have this bad performance, is due to the high degree of this gene in the subnetwork. Thus the spreading is dissipated before arriving to the target genes that are not even directly connected to the seed. See figure below(ID: 324 is the one on the bottom)



Chapter 6

Conclusions and future Implementations

In the last example analyzed we compare our results to two novel techniques: Zhou and Zhou+DSim. Those results are from Juan Caceres's work on DGP, he is a PhD student of Paccanaro Lab at Royal Holloway University of London and in his 2nd year annual review[12] he shows how to exploit diseases similarities, in addition of the genes network, in order to get better results. He applies the consistency method of Zhou on the same data used in this work, getting comparable results to our approach. Major improvements are obtained by taking into account the similarity between diseases. This information it is given by DSim dataset: DSim[6] is a MeSH-based method that accurately quantifies similarity between heritable diseases at molecular level 7.574 hereditary diseases (96% of OMIM).
The rationale behind this is that in our approach we consider one disease at time, therefore we run our algorithm starting from few nodes that are already known, that usually are less than 10 on over 16 thousand. So we are assuming that there are no difference between remaining nodes, except for the gene similarity. It is indeed important to also control the diffusion towards areas of the network with genes that, although not directly related to the emergence of the disease, are related with similar diseases.



	-	0	3.58
	0	-	1.2
	3.58	1.2	-

As we can see in the previous figure knowing the similarity between diseases allows us to add fundamental informations e.g. if we are interested in new blue genes, we notice that the purple ones are labeled with very similar disease(in [6] 3.58 is the largest score), consequently we should drive the spread of the blue label towards them and not to the bottom. For this reason we think that as a future implementation of our approach it's necessary to introduce this additional informations.

To recap, in this thesis we applied a game theoretical framework based on Constrained Dominant Sets and Relaxation Labelling Process in order to prioritize genes which are most likely to be involved in the emergence of an hereditary disease.
We introduced the concepts that underlie those frameworks and the biological background, then we presented how to model CDS and RL for the purposes of Disease Gene Prediction. In the last chapter we showed the experimental setup and the obtained results on a gene network based on HIPPIE dataset and OMIM Morbid Map labelling. This investigation has allowed us to discover that gene network combined with game theoretical methods that exploit locality in the data, are good enough to unveil good candidates if some assumptions hold. This assumption are related to the fact that unknown genes are strictly close to known ones and gather in clusters. Some diseases have showed, instead, that they can involve different areas of the network and thus we need other information to manage them.

Bibliography

[1] Michael Ashburner, Catherine A Ball, Judith A Blake, David Botstein, Heather Butler, J Michael Cherry, Allan P Davis, Kara Dolinski, Selina S Dwight, Janan T Eppig, et al. Gene ontology: tool for the unification of biology. *Nature genetics*, 25(1):25–29, 2000.

[2] Albert-László Barabási, Natali Gulbahce, and Joseph Loscalzo. Network medicine: a network-based approach to human disease. *Nature Reviews Genetics*, 12(1):56–68, 2011.

[3] Jona Boscolo Cappon. Detective overlapping protein complexes in protein-protein interaction network using dominant sets. Master's thesis, Università Ca' Foscari Venezia, Itay, 2013.

[4] Kym M Boycott, Megan R Vanstone, Dennis E Bulman, and Alex E MacKenzie. Rare-disease genetics in the era of next-generation sequencing: discovery to translation. *Nature Reviews Genetics*, 14(10):681–691, 2013.

[5] Samuel Rota Bulò and Immanuel M Bomze. Infection and immunization: a new class of evolutionary game dynamics. *Games and Economic Behavior*, 71(1):193–211, 2011.

[6] Horacio Caniza, Alfonso E Romero, and Alberto Paccanaro. A network medicine approach to quantify distance between hereditary disease modules on the interactome. *Scientific reports*, 5, 2015.

[7] MA Care, JR Bradford, CJ Needham, AJ Bulpitt, and DR Westhead. Combining the interactome and deleterious snp predictions to improve disease gene identification. *Human mutation*, 30(3):485–492, 2009.

[8] Jing Chen, Bruce J Aronow, and Anil G Jegga. Disease candidate gene identification and prioritization using protein interaction networks. *BMC bioinformatics*, 10(1):73, 2009.

[9] Xi Chen, Lily Wang, Bo Hu, Mingsheng Guo, John Barnard, and Xiaofeng Zhu. Pathway-based analysis for genome-wide association

studies using supervised principal components. *Genetic epidemiology*, 34(7):716–724, 2010.

[10] Aykut Erdem and Marcello Pelillo. Graph transduction as a noncooperative game. *Neural Computation*, 24(3):700–723, 2012.

[11] Robert A Hummel and Steven W Zucker. On the foundations of relaxation labeling processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (3):267–287, 1983.

[12] Juan Josè Càceres. 2nd year phd report. 2016.

[13] Sebastian Köhler, Sebastian Bauer, Denise Horn, and Peter N Robinson. Walking the interactome for prioritization of candidate disease genes. *The American Journal of Human Genetics*, 82(4):949–958, 2008.

[14] Kasper Lage, E Olof Karlberg, Zenia M Størling, Páll I Olason, Anders G Pedersen, Olga Rigina, Anders M Hinsby, Zeynep Tümer, Flemming Pociot, Niels Tommerup, et al. A human phenome-interactome network of protein complexes implicated in genetic disorders. *Nature biotechnology*, 25(3):309–316, 2007.

[15] Yongjin Li and Jagdish C Patra. Genome-wide inferring gene–phenotype relationship by walking on the heterogeneous network. *Bioinformatics*, 26(9):1219–1224, 2010.

[16] Dawei Liu, Debashis Ghosh, and Xihong Lin. Estimation and testing for the effect of a genetic pathway on a disease outcome using logistic kernel machine regression via logistic mixed models. *BMC bioinformatics*, 9(1):292, 2008.

[17] Li Luo, Gang Peng, Yun Zhu, Hua Dong, Christopher I Amos, and Momiao Xiong. Genome-wide gene and pathway analysis. *European Journal of Human Genetics*, 18(9):1045–1053, 2010.

[18] Paolo Martini, Gabriele Sales, M Sofia Massa, Monica Chiogna, and Chiara Romualdi. Along signal paths: an empirical gene set approach exploiting pathway topology. *Nucleic acids research*, 41(1):e19–e19, 2013.

[19] Maria Sofia Massa, Monica Chiogna, and Chiara Romualdi. Gene set analysis exploiting the topology of a pathway. *BMC Systems Biology*, 4(1):121, 2010.

[20] Victor A McKusick. Mendelian inheritance in man and its online version, omim. *The American Journal of Human Genetics*, 80(4):588–604, 2007.

[21] Douglas A Miller and Steven W Zucker. Copositive-plus lemke algorithm solves polymatrix games. *Operations Research Letters*, 10(5):285–290, 1991.

[22] John Nash. Non-cooperative games. *Annals of mathematics*, pages 286–295, 1951.

[23] Saket Navlakha and Carl Kingsford. The power of protein interaction networks for associating genes with diseases. *Bioinformatics*, 26(8):1057–1063, 2010.

[24] Martin Oti, Berend Snel, Martijn A Huynen, and Han G Brunner. Predicting disease genes using protein–protein interactions. *Journal of medical genetics*, 43(8):691–698, 2006.

[25] Massimiliano Pavan and Marcello Pelillo. Dominant sets and pairwise clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(1):167–172, 2007.

[26] Marcello Pelillo. The dynamics of nonlinear relaxation labeling processes. *Journal of Mathematical Imaging and Vision*, 7(4):309–323, 1997.

[27] Marcello Pelillo. Game theory in graph-based computer vision and pattern recognition. Summer School on Graphs in Computer Graphics, Image and Signal Analysis Bornholm, Denmark, 2011.

[28] Azriel Rosenfeld, Robert A Hummel, and Steven W Zucker. Scene labeling by relaxation operations. *IEEE Transactions on Systems, Man, and Cybernetics*, 6(6):420–433, 1976.

[29] William H Sandholm. *Population games and evolutionary dynamics*. MIT press, 2010.

[30] Martin H. Schaefer, Jean-Fred Fontaine, Arunachalam Vinayagam, Pablo Porras, Erich E. Wanker, and Miguel A. Andrade-Navarro. Hippie: Integrating protein interaction networks with experiment based quality scores. *PLoS ONE*, 7(2):1–8, 02 2012.

[31] Yoav Shoham and Kevin Leyton-Brown. *Multiagent systems: Algorithmic, game-theoretic, and logical foundations*. Cambridge University Press, 2008.

[32] J Maynard Smith and GR Price. Lhe logic of animal conflict. *Nature*, 246:15, 1973.

[33] Adi Laurentiu Tarca, Sorin Draghici, Purvesh Khatri, Sonia S Hassan, Pooja Mittal, Jung-sun Kim, Chong Jai Kim, Juan Pedro Kusanovic, and Roberto Romero. A novel signaling pathway impact analysis. *Bioinformatics*, 25(1):75–82, 2009.

[34] Rocco Tripodi, Marcello Pelillo, and Rodolfo Delmonte. Word sense disambiguation. 2014.

[35] Oron Vanunu, Oded Magger, Eytan Ruppin, Tomer Shlomi, and Roded Sharan. Associating genes and protein complexes with disease via network propagation. *PLoS Comput Biol*, 6(1):e1000641, 2010.

[36] Kai Wang, Mingyao Li, and Maja Bucan. Pathway-based approaches for analysis of genomewide association studies. *The American Journal of Human Genetics*, 81(6):1278–1283, 2007.

[37] Jörgen W Weibull. *Evolutionary game theory*. MIT press, 1997.

[38] Michael C Wu, Peter Kraft, Michael P Epstein, Deanne M Taylor, Stephen J Chanock, David J Hunter, and Xihong Lin. Powerful snp-set analysis for case-control genome-wide association studies. *The American Journal of Human Genetics*, 86(6):929–942, 2010.

[39] Xuebing Wu, Rui Jiang, Michael Q Zhang, and Shao Li. Network-based global inference of human disease genes. *Molecular systems biology*, 4(1):189, 2008.

[40] Xuebing Wu, Qifang Liu, and Rui Jiang. Align human interactome with phenome to identify causative genes and networks underlying disease families. *Bioinformatics*, 25(1):98–104, 2009.

[41] Eyasu Zemene and Marcello Pelillo. Interactive image segmentation using constrained dominant sets. In *European Conference on Computer Vision*, pages 278–294. Springer International Publishing, 2016.

[42] Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. *Advances in neural information processing systems*, 16(16):321–328, 2004.