Development of the PathoYeastract database: aiming the study of transcriptional regulation in pathogenic yeasts

Sauvagya Manna
Department of Biotechnology
Instituto Superior Técnico
Lisbon, Portugal
Email: sauvagya.manna@tecnico.ulisboa.pt

***Abstract*— The PATHOgenic YEAst Search for Transcriptional Regulators And Consensus Tracking (PathoYeastract - www.pathoyeastract.org) database is a tool for the analysis and prediction of transcription regulatory associations at the gene and genomic levels in pathogenic yeasts. In the currently available version it enlists information focused on the two most prevalent of pathogenic yeasts of the *Candida* genus: *C. albicans* and *C. glabrata*.**

This MSc thesis is a contribution to the construction of PathoYeastract. It includes the development of scripts for the automatic retrieval of data for each *C. albicans* and *C. glabrata* genes encountered in the Candida Genome Database, including coding and promoter sequences, description, associated Gene Ontology terms and orthologs in *Saccharomyces cerevisiae* and in other *Candida* species.

Given the overall lack of experimental data, the newly constructed PathoYeastract database has been developed to predict regulatory associations in 2 *Candida* species, based on the known transcription regulation of orthologous transcription factors and target genes in *Saccharomyces cerevisiae*, a model Yeast organism which has been widely studied for the last few decades. With the aid of this tool, a comprehensive comparison is expected to bring light into the mechanisms underlying the evolution of transcription regulatory networks among related yeasts.

The incorporation of regulatory data on other closely related *Candida* species to widen the scope of this database to the study of inter-species regulatory network is envisaged for the near future with expected impact in the understanding of the development of pathogenesis and antifungal drug resistance.

Keywords*— *Candida glabrata, Candida albicans, Regulatory associations, Evolution, PathoYeastract, Transcription regulation prediction, Yeast, Transcription regulatory network.Introduction

I. INTRODUCTION

I.I Motivation

The total number of eukaryotic species on earth has been estimated nearly about 8.7 million (as of 2013), amongst which 7% (611,000 species) are fungi and only 600 of them had been identified as human pathogens till date [1]. Besides posing severe infections in human hosts, Candida species are the fourth most common cause of life-threatening systemic human infections, with mortality rates as high as 50% according to the United States hospital records of patients in the United States of America [2]. *Candida* infections in humans can be of two different types: 1) superficial, oral or vaginal candidiasis or 2) systemic infections which poses severe threat and prolonged effects. Colonization of *Candida sp.* in the oral cavity of 75% of the entire human population has been recorded [3]. Though it remains in a commensal state in healthy individuals, there is a fair chance of oral candidiasis to be developed even if the individual is mildly immunocompromised which affects the person's oropharynx and/or esophagus. Persons suffering from dysfunction of adaptive immune system e.g. HIV or cancer are at high risk to be exposed to *Candida* pathogenesis. Another common infection caused by *Candida sp.* in females is vulvovaginal candidiasis; 75% of women suffer from it at least once in their lifetime. Despite their abundancy, superficial *Candida* infections are non-lethal. In contrast, systemic infections caused by *Candida sp.* can be lethal, even after first line antifungal therapy. Damage of the gastrointestinal mucosa and neutropenia are the risk factors for developing experimental systemic candidiasis. Further risk factors include central venous catheters, which allow direct access of the fungus to the bloodstream, the application of broad-spectrum antibacterial, which enable fungal overgrowth, and trauma or gastrointestinal surgery, which disrupts mucosal barriers. In both cases the pathogenicity of *Candida* is triggered by several virulence factors. In a study done by Klempp-Selb, Rimek and Kappe [4] has shown that the two specific species *Candida glabrata* and *Candida albicans* resemble each other at a genomic level resulting in similar pattern of genetic regulation and drug resistance which makes the study of the gene regulatory network of the two species together.

Several virulence factors include morphological transition between yeast and hyphal form (polymormphism), in the case of *Candida albicans*, expression of adhesins and invasins on the cell surface, thigmotropism, formation of biofilms, phenotypic switching and the secretion of hydrolytic enzymes, and fitness attributes such as adaptation to different pH helps the colonization of *Candida sp.* in a wide range of host niches.

Some Candida species have the ability to undergo a morphogenetic switch from budding yeast to hyphal growth in response to stimulus and growth conditions. Recent virulence studies of filamentation regulatory mutants argue that both yeast and filamentous forms have roles in infection. The pathogenicity of Candida is linked to its capacity to switch among different growth forms [5]. Cph2 regulates hyphal development in C. albicans, as cph2/cph2 mutant strains show medium-specific impairment in hyphal development and in the induction of hypha-specific genes. However, many hyphae-specific genes do not have potential Cph2 binding sites in their upstream regions. Interestingly, upstream sequences of all known hyphae-specific genes are found to contain potential binding sites for Tec1, a regulator of hyphal development.

Multi-drug resistance (MDR), a phenomenon of acquiring non-susceptibility to a wide range of structurally and functionally distinct cytotoxic compounds, is ubiquitously presented in living organisms, from bacteria to mammals [6]. The emergence of MDR is becoming a challenging problem in the treatment of infectious diseases, food preservation and in crop protection [7]. The ABC transporter genes *CDR1* and *CDR2* are typically upregulated in *Candida albicans* when developing resistance to azoles or can be upregulated by exposing cells transiently to drugs such as fluphenazine [8]. The upregulation of *CDR1* and *CDR2* by fluphenazine in *C. albicans* is controlled by the TF Tac1, which is a major contributor to azole-resistance. Another transcription factor involved in this process in *C. albicans* is Mrr1 which controls the expression of the Mdr1 efflux pump and mediates multidrug resistance in *C. albicans* [9] [10]. In another study carried out at IST, Lisbon and University of Lisbon in recent days which has proved the Clotrimazole Drug Resistance in *Candida glabrata* Clinical Isolates Correlates with Increased Expression of the Drug:H^+ Antiporters CgAqr1, CgTpo1_1, CgTpo3, and CgQdr2 [11].

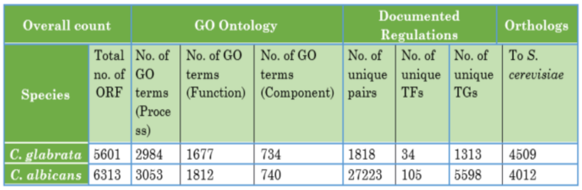
I.II Approach

Given the importance of transcriptional control in pathogenesis-related phenotypes, as in all cellular processes, it is vital to understand the global mechanisms of transcriptional control in Candida species. With that aim, this work is focused on the development of the PATHOYEASTRACT (Pathogenic Yeast Search for Transcriptional Regulators And Consensus Tracking) database. Its goal is to: 1) make a publicly available online resource for Transcription Regulation Factors and target genes and the relation between TFs and DNA binding sites in Candida species, 2) to predict the gene regulatory network for Candida species, based on the data available for the model yeast Saccharomyces cerevisiae and 3) to study gene regulatory network evolution through the development of a cross-species comparison tool.

The new database was built using a similar architecture to that of the YEASTRACT database [12] [13] [14] [15]. Its structure was organized considering the basic functioning of transcription factors. TFs are proteins which are involved in the process of transcribing DNA into RNA (1). The presence of DNA-binding domains in TFs enables them to bind to specific sequences of DNA called promoter sequences (2). Some TFs bind to the DNA promoter sequences near the transcription initiation sites to form the transcription initiation complex while other TFs can bind to regulatory sequences e.g. enhancer sequences to either stimulate or repress the regulation and expression of the related gene. Regulatory sequences can be hundreds of base pairs upstream from the transcription site. Once they are attached to the specific DNA, they trigger or suppress transcription, making genes up or downregulated.

The YEASTRACT database had been introduced in 2006 for the first time. Since then it had been highly praised among the researchers in the field with around 170,000 queries (2011-2014) and 8500 scientists accessing around the world. There had been a huge development and expansion on the available data since the beginning with a 425% more transcription regulation data. It presently contains 206,299 (as of 2014) regulatory associations between genes and TFs. YEASTRACT contains 41.693 regulatory associations based on DNA binding evidence and 172.814 on expression evidence, with some overlap [12]. Other databases reporting information on the transcriptional control in Saccharomyces species include MYBS [16], TRANSFAC [17], RSAT [18], YPA [19] and YeTFaSCo [20]. However, their focus is mainly on the prediction and analysis of the promoter regions.

After this achievement with YEASTRACT, building PATHOYEAST database was a challenge on its own. Besides providing the tools for promoter analysis in Candida, PATHOYEASTRACT will also provide a complete integration of all experimentally validated transcription regulation data ever published for Candida. Given that it is focused in multiple-species it will further enable the cross comparison between transcriptional networks from different species, providing interesting clues on the evolution of those networks. So far the regulation data from the 2 most prevalent of *Candida* species; *Candida glabrata and Candida albicans* has been updated to the database.



Overall count		GO Ontology			Documented Regulations			Orthologs
Species	Total no. of ORF	No. of GO terms (Process)	No. of GO terms (Function)	No. of GO terms (Component)	No. of unique pairs	No. of unique TFs	No. of unique TGs	To *S. cerevisiae*
C. glabrata	5601	2984	1677	734	1818	34	1313	4509
C. albicans	6313	3053	1812	740	27223	105	5598	4012

Table 1: Documented regulation for Candida glabrata and* Candida*albicans for unique Transcription Factors against each other and Saccharomyces cerevisiae

Comparative transcriptomics has been used as a state of the art technology for the past few years to study gene regulation, enabling the comparison of transcriptome-wide responses to the same environmental cues in yeast species of the same family. Saccharomyces cerevisiae has been used as a model organism in that sense. As such, it became a key aspect of this work to develop a tool to use the huge amount of information gathered for S. cerevisiae to predict transcription regulation in the far less well studied Candida species.

II. MATERIALS AND METHODS

II.I Data Extraction

The information residing in different sources focusing on geneontology (http://www.geneontology.org/), geneassociation(ftp://ftp.geneontology.org/pub/go/gene-associations /submission/gene_association.cgd.gz), [both CGD [21](http://www.candidagenome.org/) and SGD [22] (http://www.yeastgenome.org/)], orfidcodin sequence were imported into the tables of the new integrated database using PHP scripts.

Every gene existing in the database is denoted with a unique Open Reading Frame ID (orfid) for the identification of the ORF when searched by the user. The information in the concept orfid was drawn from a reference file provided by Dr. Miguel Nobre Parreira Cacho Teixeira and Dr. Pedro Tiago Monteiro in the Biological Sciences Research Group (BSRG), Instituto Superior Técnico (IST) and INESC-ID. All those biologically relevant features in the reference file were extracted from different databases, e.g. CGD and SGD.

All the GO terms and their hierarchical associations were extracted from the Gene Ontology Consortium, while the mapping between GO terms and the *orfs* were obtained from **Figure 3** on the table processlist. For all the orfs and gene encoded in the genomes of *Candida glabrata*, *Candida albicans* and their related species, the information of names (systematic orf, gene and orthologs[http://www.candidagenome.org/download/homology/orthologs/]) were drawn from the model organism databases CGD and SGD, and the information on the corresponding bibliographic references was extracted from PubMed. The tables *orfgene* (loaded from CGD databank; the geneassociation.cgd.gz compressed file[http://www.candidagenome.org/download/go/]) and *regulation* were fetched from the file provided by Dr. Miguel Teixeira where the results had been recorded from the wet lab experiments. For the table *orthologs*, the information on the gene names are only available for those genes which are orthologous in the genomes of *Candida glabrata, Candida albicans* and them related species.

II.II Data Structure

The entity–relationship model (ER model) is used to describe the data. This model allows the information representing the real world to be implemented in a database in terms of concepts and their relations. This section describes the identification of five principle entities (Orf, Protein, Orthologs, Regulation, Gene Ontology) and the mapping between them for the definition of the structure of the integrated database.

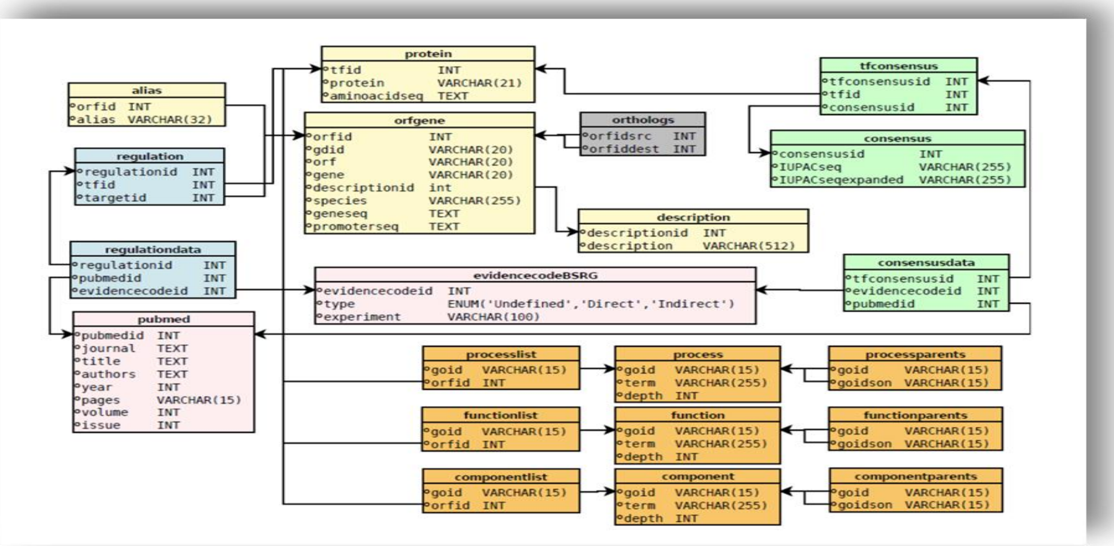


Figure 3: Physical model of the database

II.II.I Concept of ORF

In molecular biology, an orf is the part of a reading frame which can potentially be translated into a protein or peptide or RNA. It is a continuous sequence of nucleotides beginning with a start codon and ending with a stop codon (Brown, 2010). Each gene has a corresponding orfid, but the reverse is not necessarily true, which means that the presence of an orfid is not conclusive evidence for the presence of a gene.

II.II.I.I Attributes

For each unique transcription factor (tfid) and target gene (targetid); a unique orf (orfid) is generated. Based on the provided genes/TFs, the mapping between the GO terms and the orfs is done where all the information related to the specific set of genes/TFs are generated. In addition to the standard transcription factor name, there might be several alternative names given by different research groups.
Apart from the basic biologically relevant features of each transcription factor, it provides data to aid in the evolutionary studies of genes. Since the information of the orfid was extracted from different databases, a link (URL) to the source database was provided and can be accessed by users as a description ID (descriptionid). The concept of orfid was created containing the Orfid (orfid), Gene Ontology ID (goid), gene name (gene), description of the orfid (descriptionid), species name (species), gene sequence (geneseq) and promoter sequence (promoterseq).

II.II.I.II Relations

This concept has three associated relations: Translation, Gene Ontology Annotation and Transcription Factor Regulation. The relation Translation is a relation between orfid concept and the Protein concept. The relation Gene Ontology Annotation is the mapping between orfid and Gene Ontology terms, while the relation regulation annotation is the correspondence between the transcription factor and target id with the orfid. This two relations will be explained in the Gene Ontology concept and the Regulation concept, respectively.

II.II.II Concept of Protein

Proteins are large biological molecules, composed of one or more chains of amino acids, and they are the fundamental components and functional units of the cell.

II.II.II.I Attributes

In this system, one protein is described using the following attributes: a transcription factor id (tfid), a protein name (protein) and a sequence of amino acid (aminoacidsequence).

II.II.III Concept of Orthologs

Orthologs are the genes which belong to different species but share the same e ancestral root and have the same function in species. They are an important component to derive the functional relation between transcription factors cross-species.

II.II.III.I Attributes

In this system, the orfid description and the destination of the orfid relates back to a unique orfid. So, if the description of an orfids found in different species, it would point back to the specific orfid on the database.

II.II.IV Concept of Regulation

For each transcription regulation there is a unique regulation ID is assigned (regulationid) which relates to the regulationrefs field, it correlates to every documents relates to that specific Transcription regulation.

II.II.IV.I Attributes

In the regulationrefs table three attributes are assigned which are regulationid, pubmedid and evidencecodeid. The pubmedid is related to the pubmed field which has the information of all the literature/research papers about the specific regulation id. It redirects to the PubMed website for the referencing.

II.II.V Concept of Gene Ontology

After genome sequencing, there are much information needed to be interpreted. It is necessary to assign terms to represent the biological process and functions in which the objective proteins and genes are involved in a synthetic and standard form. Initially, there was no universal standard form for these terms, which means that the usage of terms may be different according to research areas or even research groups. This leads to increasing difficulties in communication and sharing data.

In order to overcome this problem, the Gene Ontology (GO) consortium was founded in 1998 by three model organism databases, FlyBase (Drosophila), the Saccharomyces Genome Database (SGD) and the Mouse Genome Database (MGD), aiming at defining consistent Ontology terms of gene product properties. Candida Genome Database (CGD) also follows the convention of GO convention which had been followed in the PATHOYEAST database. The GO project has developed three ontologies describing gene products in three different domains: cellular component, molecular function and biological process.

- Cellular component: a part of a cell or its extracellular environment.
- Molecular function: the “abilities” that the gene product has.
- Biological process: a set of molecular events with a defined beginning and end.

Each GO term has a term name, a unique alphanumeric identifier (goid), a definition with cited sources, a namespace indicating the domain which it belongs to, and several other elements describing its characteristics. These three ontologies are organized hierarchically through part of (composition) and is a (inheritance) relations. The “son” terms are more specialized than their “parent” terms, but unlike a strict hierarchy, a GO term may have multiple parent terms. There is a root, the most general term and corresponding to the top of the hierarchy, for each of the three ontologies cellular_component (GO:0005575), molecular_function (GO:0003674), biological_process (GO:0008150). All other terms in a domain can trace their parentage to the corresponding root term.

II.II.V.I Attributes

Since the concepts are hierarchically organized, there is an attribute to indicate the distance between the objective GO term and the corresponding root term, called depth. The attribute goidsons is used to show the GO terms which are directly under the object GO term. In summary, the concept of each ontology contains three attributes: a unique identifier (goid), a term name (term), depth and goidson.

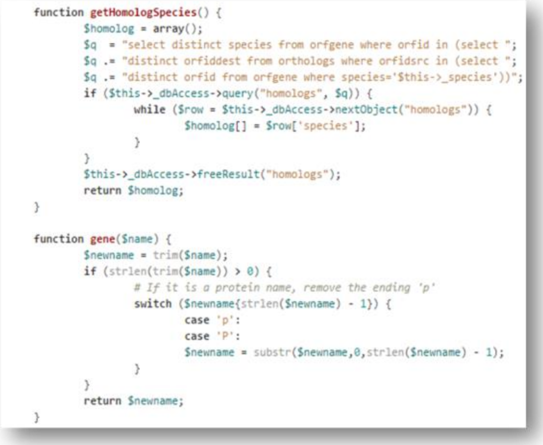
II.II.V.II Relations

This concept has a type of relation--Gene Ontology Annotation, which is the mapping between orfids and the corresponding GO terms.

II.III Functions

II.III.I Sorting by Homologous Species

In this function the script has been written to pick up the unique homologs so the regulations can be reclassified based on distinct orfids in different species which helps to sort out the regulations of the genes of interest based on the species.



```
function getHomologSpecies() {
        $homolog = array();
        $q  = "select distinct species from orfgene where orfid in (select ";
        $q .= "distinct orfiddest from orthologs where orfidsrc in (select ";
        $q .= "distinct orfid from orfgene where species='$this->_species'))";
        if ($this->_dbAccess->query("homologs", $q)) {
                while ($row = $this->_dbAccess->nextObject("homologs")) {
                        $homolog[] = $row['species'];
                }
        }
        $this->_dbAccess->freeResult("homologs");
        return $homolog;
}

function gene($name) {
        $newname = trim($name);
        if (strlen(trim($name)) > 0) {
                # If it is a protein name, remove the ending 'p'
                switch ($newname{strlen($newname) - 1}) {
                        case 'p':
                        case 'P':
                        $newname = substr($newname,0,strlen($newname) - 1);
                }
        }
        return $newname;
}
```

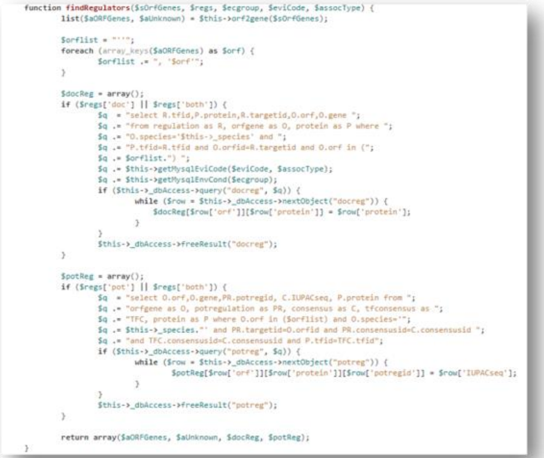
Figure 1: Added filter to the query based on Documented Regulations by homology

This change in functionality in the form helps to filter the queries to retrieve the dataset from the database and show the results on the query page on the user interface.

II.III.II Search by TF

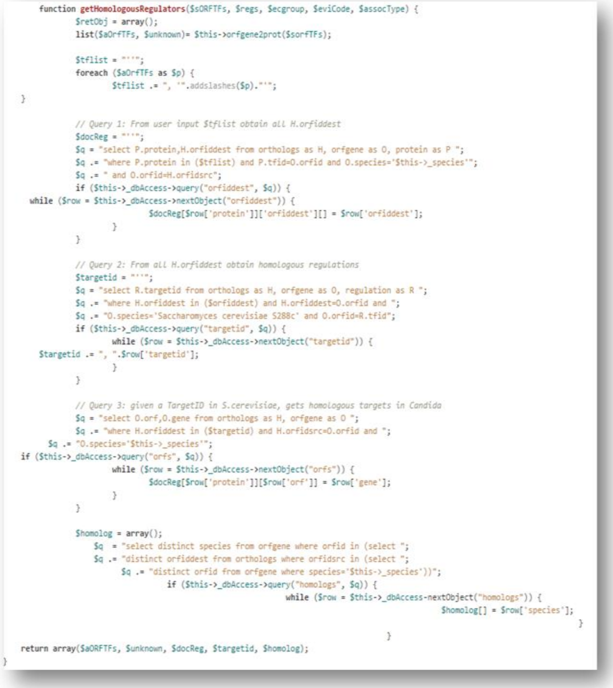
This function helps to call the internal database layer to look for the specific regulators supplied as a query by the user and then returns the name of the regulators as an array. It then filters the query based on the criteria e.g. environmental conditions or association type.

When the regulators are found it returns the results as ORFGenes while the query input is an array of targetids, which calls the targetids from the potregulation table and return an array of orfid which corresponds to the consensusids of the consensus table. If there is no match for the specific set of TFs, it prints 'No target genes found'.By this function the possibility of returning the same regulator ids (double copy) is eliminated, so it returns only the unique ids on the same page.



```
function findRegulators($sOrfGenes, $regs, $ecgroup, $eviCode, $assocType) {
        list($aORFGenes, $aUnknown) = $this->orf2gene($sOrfGenes);

        $orflist = "''";
        foreach (array_keys($aORFGenes) as $orf) {
                $orflist .= ", '$orf'";
        }

        $docReg = array();
        if ($regs['doc'] || $regs['both']) {
                $q  = "select R.tfid,P.protein,R.targetid,O.orf,O.gene ";
                $q .= "from regulation as R, orfgene as O, protein as P where ";
                $q .= "O.species='$this->_species' and ";
                $q .= "P.tfid=R.tfid and O.orfid=R.targetid and O.orf in (";
                $q .= $orflist.") ";
                $q .= $this->getMysqlEviCode($eviCode, $assocType);
                $q .= $this->getMysqlEnvCond($ecgroup);
                if ($this->_dbAccess->query("docreg", $q)) {
                        while ($row = $this->_dbAccess->nextObject("docreg")) {
                            $docReg[$row['orf']][$row['protein']] = $row['protein'];
                        }
                }
                $this->_dbAccess->freeResult("docreg");
        }

        $potReg = array();
        if ($regs['pot'] || $regs['both']) {
                $q  = "select O.orf,O.gene,PR.potregid, C.IUPACseq, P.protein from ";
                $q .= "orfgene as O, potregulation as PR, consensus as C, tfconsensus as ";
                $q .= "TFC, protein as P where O.orf in ($orflist) and O.species='";
                $q .= $this->_species."' and PR.targetid=O.orfid and PR.consensusid=C.consensusid ";
                $q .= "and TFC.consensusid=C.consensusid and P.tfid=TFC.tfid";
                if ($this->_dbAccess->query("potreg", $q)) {
                        while ($row = $this->_dbAccess->nextObject("potreg")) {
                                $potReg[$row['orf']][$row['protein']][$row['potregid']] = $row['IUPACseq'];
                        }
                }
                $this->_dbAccess->freeResult("potreg");
        }

        return array($aORFGenes, $aUnknown, $docReg, $potReg);
}
```

Figure 2: Script to search for TFs based on documented homology

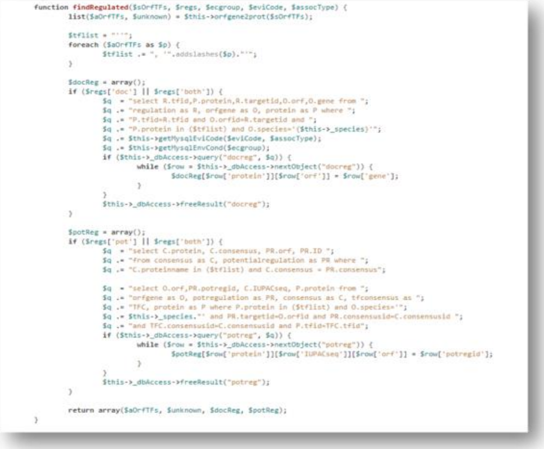


```
    function getHomologousRegulators($sORFTFs, $regs, $ecgroup, $eviCode, $assocType) {
            $retObj = array();
            list($aOrfTFs, $unknown)= $this->orfgene2prot($sorfTFs);

            $tflist = "''";
            foreach ($aOrfTFs as $p) {
                    $tflist .= ", '".addslashes($p)."'";
}

            // Query 1: From user input $tflist obtain all H.orfiddest
            $docReg = "''";
            $q = "select P.protein,H.orfiddest from orthologs as H, orfgene as O, protein as P ";
            $q .= "where P.protein in ($tflist) and P.tfid=O.orfid and O.species='$this->_species'";
            $q .= " and O.orfid=H.orfidsrc";
            if ($this->_dbAccess->query("orfiddest", $q)) {
  while ($row = $this->_dbAccess->nextObject("orfiddest")) {
                            $docReg[$row['protein']]['orfiddest'][] = $row['orfiddest'];
                    }
            }

            // Query 2: From all H.orfiddest obtain homologous regulations
            $targetid = "''";
            $q = "select R.targetid from orthologs as H, orfgene as O, regulation as R ";
            $q .= "where H.orfiddest in ($orfiddest) and H.orfiddest=O.orfid and ";
            $q .= "O.species='Saccharomyces cerevisiae S288c' and O.orfid=R.tfid";
            if ($this->_dbAccess->query("targetid", $q)) {
                    while ($row = $this->_dbAccess->nextObject("targetid")) {
   $targetid .= ", ".$row['targetid'];
                    }
            }

            // Query 3: given a TargetID in S.cerevisiae, gets homologous targets in Candida
            $q = "select O.orf,O.gene from orthologs as H, orfgene as O ";
            $q .= "where H.orfiddest in ($targetid) and H.orfidsrc=O.orfid and ";
     $q .= "O.species='$this->_species'";
 if ($this->_dbAccess->query("orfs", $q)) {
                    while ($row = $this->_dbAccess->nextObject("orfs")) {
                            $docReg[$row['protein']][$row['orf']] = $row['gene'];
                    }
            }

            $homolog = array();
                $q  = "select distinct species from orfgene where orfid in (select ";
                $q .= "distinct orfiddest from orthologs where orfidsrc in (select ";
                    $q .= "distinct orfid from orfgene where species='$this->_species'))";
                            if ($this->_dbAccess->query("homologs", $q)) {
                                            while ($row = $this->_dbAccess-nextObject("homologs")) {
                                                            $homolog[] = $row['species'];
                                                                    }
                                            }
 return array($aORFTFs, $unknown, $docReg, $targetid, $homolog);
}
```

Figure 3: Script for getting homologous regulators

II.III.III Search by Genes and filter by homology

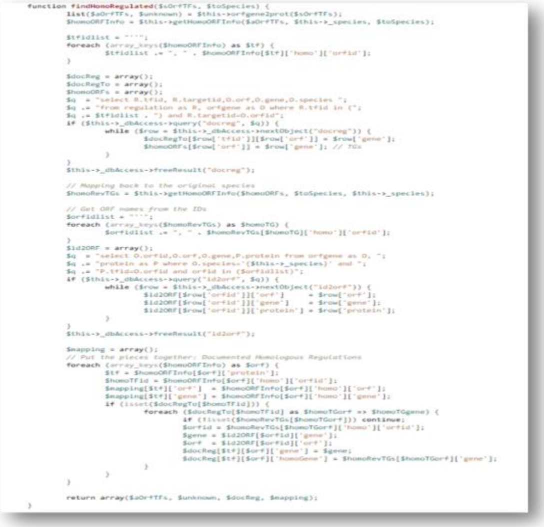
This function helps to find the regulations based on homology in the internal database when there is a specific regulation between two or multiple genes. The purpose of this function is to aid in mapping the based on regulation to construct the gene regulatory network.



```
function findRegulated($sOrfTFs, $regs, $ecgroup, $eviCode, $assocType) {
    list($aOrfTFs, $unknown) = $this->orfgene2prot($sOrfTFs);

    $tflist = "''";
    foreach ($aOrfTFs as $p) {
        $tflist .= ", '".addslashes($p)."'";
    }

    $docReg = array();
    if ($regs['doc'] || $regs['both']) {
        $q  = "select R.tfid,P.protein,R.targetid,O.orf,O.gene from ";
        $q .= "regulation as R, orfgene as O, protein as P where ";
        $q .= "P.tfid=R.tfid and O.orfid=R.targetid and ";
        $q .= "P.protein in ($tflist) and O.species='{$this->_species}'";
        $q .= $this->getMysqlEviCode($eviCode, $assocType);
        $q .= $this->getMysqlEnvCond($ecgroup);
        if ($this->_dbAccess->query("docreg", $q)) {
            while ($row = $this->_dbAccess->nextObject("docreg")) {
                $docReg[$row['protein']][$row['orf']] = $row['gene'];
            }
        }
        $this->_dbAccess->freeResult("docreg");
    }

    $potReg = array();
    if ($regs['pot'] || $regs['both']) {
        $q  = "select C.protein, C.consensus, PR.orf, PR.ID ";
        $q .= "from consensus as C, potentialregulation as PR where ";
        $q .= "C.proteinname in ($tflist) and C.consensus = PR.consensus";

        $q  = "select O.orf,PR.potregid, C.IUPACseq, P.protein from ";
        $q .= "orfgene as O, potregulation as PR, consensus as C, tfconsensus as ";
        $q .= "TFC, protein as P where P.protein in ($tflist) and O.species='";
        $q .= $this->_species."' and PR.targetid=O.orfid and PR.consensusid=C.consensusid ";
        $q .= "and TFC.consensusid=C.consensusid and P.tfid=TFC.tfid";
        if ($this->_dbAccess->query("potreg", $q)) {
            while ($row = $this->_dbAccess->nextObject("potreg")) {
                $potReg[$row['protein']][$row['IUPACseq']][$row['orf']] = $row['potregid'];
            }
        }
        $this->_dbAccess->freeResult("potreg");
    }

    return array($aOrfTFs, $unknown, $docReg, $potReg);
}
```

Figure 4: Script to search for Genes based on documented homology

If there is a documented regulation found, it returns the regulation with the documentation source. If not, the genes are characterized as ‘Uncharacterized’ meaning there is no documented regulation found in the database. If there is no regulation found amongst the input genes, it returns ‘Not Found’ as a result meaning there is neither a documented nor a potential regulation is found in the database.

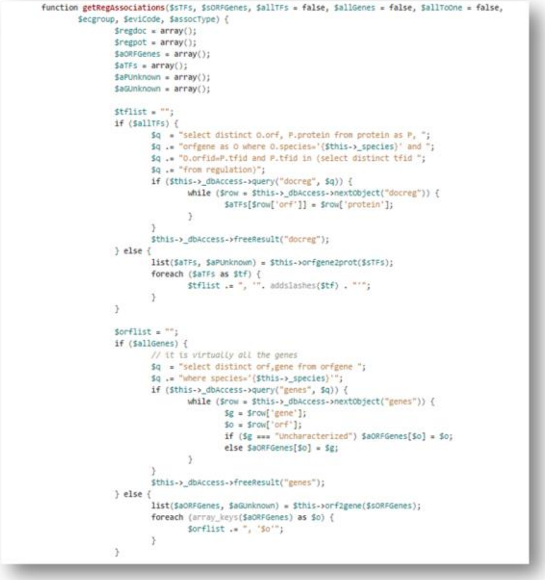


```
function findHomoRegulated($sOrfTFs, $toSpecies) {
    list($aOrfTFs, $unknown) = $this->orfgene2prot($sOrfTFs);
    $homoORFInfo = $this->getHomoORFInfo($aOrfTFs, $this->_species, $toSpecies);

    $tfidlist = "''";
    foreach (array_keys($homoORFInfo) as $tf) {
        $tfidlist .= ", " . $homoORFInfo[$tf]['homo']['orfid'];
    }

    $docReg = array();
    $docRegTo = array();
    $homoORFs = array();
    $q  = "select R.tfid, R.targetid,O.orf,O.gene,O.species ";
    $q .= "from regulation as R, orfgene as O where R.tfid in (";
    $q .= $tfidlist . ") and R.targetid=O.orfid";
    if ($this->_dbAccess->query("docreg", $q)) {
        while ($row = $this->_dbAccess->nextObject("docreg")) {
            $docRegTo[$row['tfid']][$row['orf']] = $row['gene'];
            $homoORFs[$row['orf']] = $row['gene']; // TGs
        }
    }
    $this->_dbAccess->freeResult("docreg");

    // Mapping back to the original species
    $homoRevTGs = $this->getHomoORFInfo($homoORFs, $toSpecies, $this->_species);

    // Get ORF names from the IDs
    $orfidlist = "''";
    foreach (array_keys($homoRevTGs) as $homoTG) {
        $orfidlist .= ", " . $homoRevTGs[$homoTG]['homo']['orfid'];
    }
    $id2ORF = array();
    $q  = "select O.orfid,O.orf,O.gene,P.protein from orfgene as O, ";
    $q .= "protein as P where O.species='{$this->_species}' and ";
    $q .= "P.tfid=O.orfid and orfid in ($orfidlist)";
    if ($this->_dbAccess->query("id2orf", $q)) {
        while ($row = $this->_dbAccess->nextObject("id2orf")) {
            $id2ORF[$row['orfid']]['orf']     = $row['orf'];
            $id2ORF[$row['orfid']]['gene']    = $row['gene'];
            $id2ORF[$row['orfid']]['protein'] = $row['protein'];
        }
    }
    $this->_dbAccess->freeResult("id2orf");

    $mapping = array();
    // Put the pieces together: Documented Homologous Regulations
    foreach (array_keys($homoORFInfo) as $orf) {
        $tf = $homoORFInfo[$orf]['protein'];
        $homoTFid = $homoORFInfo[$orf]['homo']['orfid'];
        $mapping[$tf]['orf']  = $homoORFInfo[$orf]['homo']['orf'];
        $mapping[$tf]['gene'] = $homoORFInfo[$orf]['homo']['gene'];
        if (isset($docRegTo[$homoTFid])) {
            foreach ($docRegTo[$homoTFid] as $homoTGorf => $homoTGgene) {
                if (!isset($homoRevTGs[$homoTGorf])) continue;
                $orfid = $homoRevTGs[$homoTGorf]['homo']['orfid'];
                $gene = $id2ORF[$orfid]['gene'];
                $orf  = $id2ORF[$orfid]['orf'];
                $docReg[$tf][$orf]['gene'] = $gene;
                $docReg[$tf][$orf]['homoGene'] = $homoRevTGs[$homoTGorf]['gene'];
            }
        }
    }

    return array($aOrfTFs, $unknown, $docReg, $mapping);
}
```

Figure 5: Script for getting homologous regulators

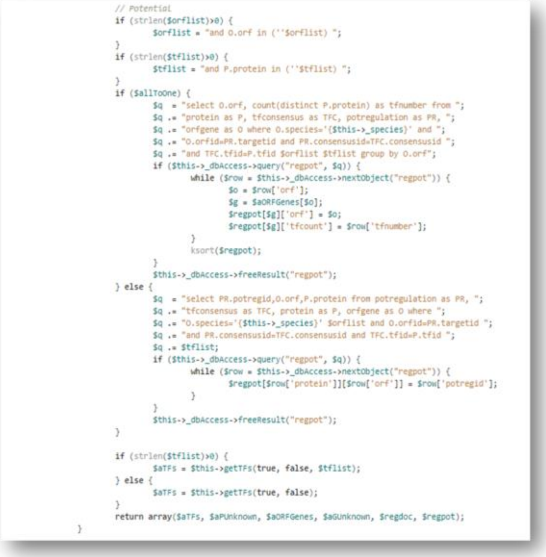
II.III.IV Search for Regulatory associations

This function is used to find the regulatory association among user input TFs and target genes. The results are filtered based on several conditions which are called as a function, so it can provide the user with the suitable results defined by the search parameters.



```
function getRegAssociations($sTFs, $sORFGenes, $allTFs = false, $allGenes = false, $allToOne = false,
        $ecgroup, $eviCode, $assocType) {
            $regdoc = array();
            $regpot = array();
            $aORFGenes = array();
            $aTFs = array();
            $aPUnknown = array();
            $aGUnknown = array();

            $tflist = "";
            if ($allTFs) {
                $q  = "select distinct O.orf, P.protein from protein as P, ";
                $q .= "orfgene as O where O.species='{$this->_species}' and ";
                $q .= "O.orfid=P.tfid and P.tfid in (select distinct tfid ";
                $q .= "from regulation)";
                if ($this->_dbAccess->query("docreg", $q)) {
                    while ($row = $this->_dbAccess->nextObject("docreg")) {
                        $aTFs[$row['orf']] = $row['protein'];
                    }
                }
                $this->_dbAccess->freeResult("docreg");
            } else {
                list($aTFs, $aPUnknown) = $this->orfgene2prot($sTFs);
                foreach ($aTFs as $tf) {
                    $tflist .= ", '". addslashes($tf) . "'";
                }
            }

            $orflist = "";
            if ($allGenes) {
                // it is virtually all the genes
                $q  = "select distinct orf,gene from orfgene ";
                $q .= "where species='{$this->_species}'";
                if ($this->_dbAccess->query("genes", $q)) {
                    while ($row = $this->_dbAccess->nextObject("genes")) {
                        $g = $row['gene'];
                        $o = $row['orf'];
                        if ($g === "Uncharacterized") $aORFGenes[$o] = $o;
                        else $aORFGenes[$o] = $g;
                    }
                }
                $this->_dbAccess->freeResult("genes");
            } else {
                list($aORFGenes, $aGUnknown) = $this->orf2gene($sORFGenes);
                foreach (array_keys($aORFGenes) as $o) {
                    $orflist .= ", '$o'";
                }
            }
```

Figure 6: Script for finding regulatory associations among the target genes



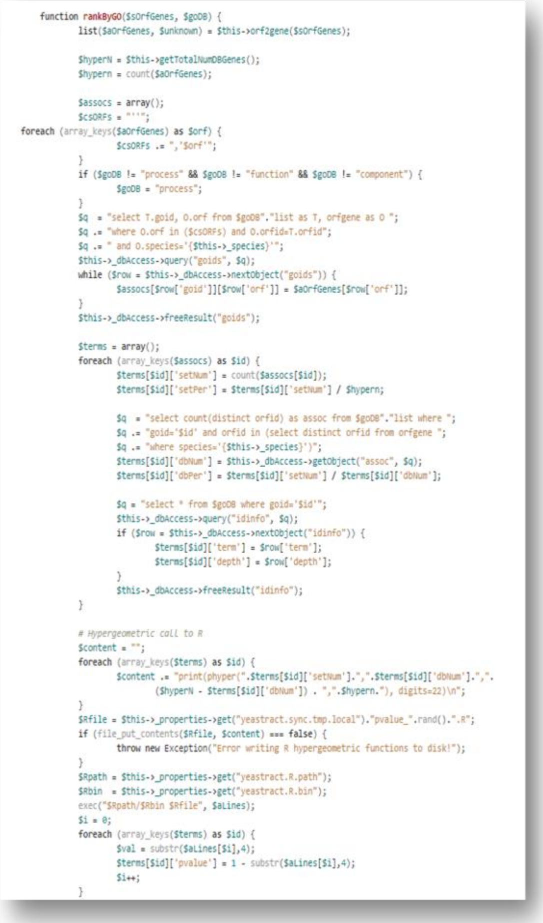
```
            // Potential
            if (strlen($orflist)>0) {
                $orflist = "and O.orf in (''$orflist) ";
            }
            if (strlen($tflist)>0) {
                $tflist = "and P.protein in (''$tflist) ";
            }
            if ($allToOne) {
                $q  = "select O.orf, count(distinct P.protein) as tfnumber from ";
                $q .= "protein as P, tfconsensus as TFC, potregulation as PR, ";
                $q .= "orfgene as O where O.species='{$this->_species}' and ";
                $q .= "O.orfid=PR.targetid and PR.consensusid=TFC.consensusid ";
                $q .= "and TFC.tfid=P.tfid $orflist $tflist group by O.orf";
                if ($this->_dbAccess->query("regpot", $q)) {
                    while ($row = $this->_dbAccess->nextObject("regpot")) {
                        $o = $row['orf'];
                        $g = $aORFGenes[$o];
                        $regpot[$g]['orf'] = $o;
                        $regpot[$g]['tfcount'] = $row['tfnumber'];
                    }
                    ksort($regpot);
                }
                $this->_dbAccess->freeResult("regpot");
            } else {
                $q  = "select PR.potregid,O.orf,P.protein from potregulation as PR, ";
                $q .= "tfconsensus as TFC, protein as P, orfgene as O where ";
                $q .= "O.species='{$this->_species}' $orflist and O.orfid=PR.targetid ";
                $q .= "and PR.consensusid=TFC.consensusid and TFC.tfid=P.tfid ";
                $q .= $tflist;
                if ($this->_dbAccess->query("regpot", $q)) {
                    while ($row = $this->_dbAccess->nextObject("regpot")) {
                        $regpot[$row['protein']][$row['orf']] = $row['potregid'];
                    }
                }
                $this->_dbAccess->freeResult("regpot");
            }

            if (strlen($tflist)>0) {
                $aTFs = $this->getTFs(true, false, $tflist);
            } else {
                $aTFs = $this->getTFs(true, false);
            }
            return array($aTFs, $aPUnknown, $aORFGenes, $aGUnknown, $regdoc, $regpot);
        }
```

Figure 7: Script for finding potential regulatory associations among the target genes

The ‘allTFs’ parameter lists all the possible regulatory associations for the TFs and the ‘allGenes’ which are given as inputs as target genes. The ‘allToOne’ parameter checks a specific gene against all the documented regulators based on homology. Furthermore, the results can be filtered based on direct and indirect regulation as well and DNA binding and/plus expression evidence.

II.III.V Ranking genes by Gene Ontology

With the help of this function ranking of genes can be done by the three gene ontologies: Biological Process, Molecular Function or Cellular Component. This query returns a table organized by GO terms, ordered by the percentage of the genes (from the input list) associated with each GO term, or by the enrichment of the user’s dataset in that specific GO term, when compared to the genome (evaluated by p-value). For each GO term the table contains the names of the genes associated with it.



```
    function rankByGO($sOrfGenes, $goDB) {
            list($aOrfGenes, $unknown) = $this->orf2gene($sOrfGenes);

            $hyperN = $this->getTotalNumDBGenes();
            $hypern = count($aOrfGenes);

            $assocs = array();
            $csORFs = "''";
foreach (array_keys($aOrfGenes) as $orf) {
                    $csORFs .= ",'$orf'";
            }
            if ($goDB != "process" && $goDB != "function" && $goDB != "component") {
                    $goDB = "process";
            }
            $q  = "select T.goid, O.orf from $goDB"."list as T, orfgene as O ";
            $q .= "where O.orf in ($csORFs) and O.orfid=T.orfid";
            $q .= " and O.species='{$this->_species}'";
            $this->_dbAccess->query("goids", $q);
            while ($row = $this->_dbAccess->nextObject("goids")) {
                    $assocs[$row['goid']][$row['orf']] = $aOrfGenes[$row['orf']];
            }
            $this->_dbAccess->freeResult("goids");

            $terms = array();
            foreach (array_keys($assocs) as $id) {
                    $terms[$id]['setNum'] = count($assocs[$id]);
                    $terms[$id]['setPer'] = $terms[$id]['setNum'] / $hypern;

                    $q  = "select count(distinct orfid) as assoc from $goDB"."list where ";
                    $q .= "goid='$id' and orfid in (select distinct orfid from orfgene ";
                    $q .= "where species='{$this->_species}')";
                    $terms[$id]['dbNum'] = $this->_dbAccess->getObject("assoc", $q);
                    $terms[$id]['dbPer'] = $terms[$id]['setNum'] / $terms[$id]['dbNum'];

                    $q = "select * from $goDB where goid='$id'";
                    $this->_dbAccess->query("idinfo", $q);
                    if ($row = $this->_dbAccess->nextObject("idinfo")) {
                            $terms[$id]['term'] = $row['term'];
                            $terms[$id]['depth'] = $row['depth'];
                    }
                    $this->_dbAccess->freeResult("idinfo");
            }

            # Hypergeometric call to R
            $content = "";
            foreach (array_keys($terms) as $id) {
                    $content .= "print(phyper(".$terms[$id]['setNum'].",".$terms[$id]['dbNum'].",".
                            ($hyperN - $terms[$id]['dbNum']) . ",".$hypern."), digits=22)\n";
            }
            $Rfile = $this->_properties->get("yeastract.sync.tmp.local")."pvalue_".rand().".R";
            if (file_put_contents($Rfile, $content) === false) {
                    throw new Exception("Error writing R hypergeometric functions to disk!");
            }
            $Rpath = $this->_properties->get("yeastract.R.path");
            $Rbin  = $this->_properties->get("yeastract.R.bin");
            exec("$Rpath/$Rbin $Rfile", $aLines);
            $i = 0;
            foreach (array_keys($terms) as $id) {
                    $val = substr($aLines[$i],4);
                    $terms[$id]['pvalue'] = 1 - substr($aLines[$i],4);
                    $i++;
            }
```

Figure 8: Script for ranking genes by gene ontology

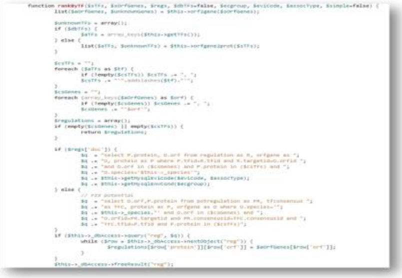
II.III.VII Ranking genes by Transcription Factors

This function allows the user to execute the query to enable the user to group a given list of genes (e.g., a set of co-activated genes from a microarray experiment) according to the TFs which are their documented or potential regulators, ranking those TFs according to two possible criteria: % of genes associated to each TF; or enrichment on the TF targets in the dataset, evaluated based on a p-value when a list of genes and optionally a list of TFs are used as input.

The TFs considered in this query can be either those in the input list, or all the TFs in the PathoYeastract database (by selecting the option Check for all TFs). Either documented or potential regulations can be considered (by selecting Documented Regulations or Potential Regulations, respectively). Furthermore, for each established regulon, the percentage value representing the proportion of genes regulated by each TF can be calculated relative to:

I. The total number of genes in the input list; this may identify the TFs involved in the regulation of the given gene list;
II. The number of genes, in the whole yeast genome, documented as being regulated by the same TF; this may identify the TF networks predominantly involved in the regulation of the referred gene list.
III. A p-value the represents the enrichment of the TF regulon in our dataset, when compared to the genome.

Furthermore, the query enables the user to restrict its’ search to the regulatory associations identified based on direct or indirect evidences. Depending on the options selected, the output is a table containing the input genes, grouped by TF and ordered by the percentage of genes regulated by the respective TF or a downloadable image of the regulatory network.



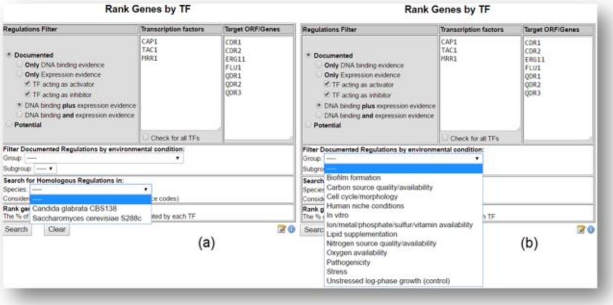
```
function rankByTF($sTFs, $sOrfGenes, $regs, $dbTFs=false, $ecgroup, $evicode, $assocType, $simple=false) {
        list($aOrfGenes, $unknownGenes) = $this->orf2gene($sOrfGenes);

        $unknownTFs = array();
        if ($dbTFs) {
                $aTFs = array_keys($this->getTFs());
        } else {
                list($aTFs, $unknownTFs) = $this->orfgene2prot($sTFs);
        }

        $csTFs = "";
        foreach ($aTFs as $tf) {
                if (!empty($csTFs)) $csTFs .= ", ";
                $csTFs .= "'".addslashes($tf)."'";
        }
        $csGenes = "";
        foreach (array_keys($aOrfGenes) as $orf) {
                if (!empty($csGenes)) $csGenes .= ", ";
                $csGenes .= "'$orf'";
        }
        $regulations = array();
        if (empty($csGenes) || empty($csTFs)) {
                return $regulations;
        }

        if ($regs['doc']) {
                $q  = "select P.protein, O.orf from regulation as R, orfgene as ";
                $q .= "O, protein as P where P.tfid=R.tfid and R.targetid=O.orfid ";
                $q .= "and O.orf in ($csGenes) and P.protein in ($csTFs) and ";
                $q .= "O.species='$this->_species'";
                $q .= $this->getMysqlEvicode($evicode, $assocType);
                $q .= $this->getMysqlEnvCond($ecgroup);
        } else {
                // PZX potential
                $q  = "select O.orf,P.protein from potregulation as PR, tfconsensus ";
                $q .= "as TFC, protein as P, orfgene as O where O.species='";
                $q .= $this->_species."' and O.orf in ($csGenes) and ";
                $q .= "O.orfid=PR.targetid and PR.consensusid=TFC.consensusid and ";
                $q .= "TFC.tfid=P.tfid and P.protein in ($csTFs)";
        }
        if ($this->_dbAccess->query("reg", $q)) {
                while ($row = $this->_dbAccess->nextObject("reg")) {
                        $regulations[$row['protein']][$row['orf']] = $aOrfGenes[$row['orf']];
                }
        }
        $this->_dbAccess->freeResult("reg");
```

Figure 9: Script for ranking genes by TFs

III. RESULTS

III.I Predicting gene and genomic regulation

The query “Rank by TF” enables the user to automatically select and rank the transcription factors potentially involved in the regulation of the genes of interest. The TFs are represented in a descending order of relevance score calculated for each transcription factor which are cross checked against the regulation and regulatory data available on the PATHOYEASTRACT database. There are several filters available for the user to choose from so they can narrow down the search depending on the environmental conditions or species to check the regulation against.



Rank Genes by TF

Regulations Filter	Transcription factors	Target ORF/Genes
● Documented ○ Only DNA binding evidence ○ Only Expression evidence ☑ TF acting as activator ☑ TF acting as inhibitor ● DNA binding plus expression evidence ○ DNA binding and expression evidence ○ Potential	CAP1 TAC1 MRR1 □ Check for all TFs	CDR1 CDR2 ERG11 FLU1 QDR1 QDR2 QDR3

Filter Documented Regulations by environmental condition:
Group: -----
Subgroup: -----

Search for Homologous Regulations in:
Species: -----
Consider ... e codes)

Candida glabrata CBS138
Saccharomyces cerevisiae S288c

Rank ge... The % of ... ted by each TF

Search Clear

(a)

Rank Genes by TF

Regulations Filter	Transcription factors	Target ORF/Genes
● Documented ○ Only DNA binding evidence ○ Only Expression evidence ☑ TF acting as activator ☑ TF acting as inhibitor ● DNA binding plus expression evidence ○ DNA binding and expression evidence ○ Potential	CAP1 TAC1 MRR1 □ Check for all TFs	CDR1 CDR2 ERG11 FLU1 QDR1 QDR2 QDR3

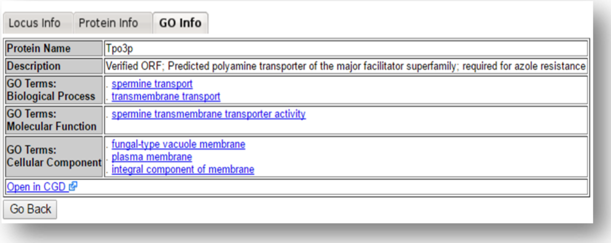
Filter Documented Regulations by environmental condition:
Group: -----

Biofilm formation
Carbon source quality/availability
Cell cycle/morphology
Human niche conditions
In vitro
Ion/metal/phosphate/sulfur/vitamin availability
Lipid supplementation
Nitrogen source quality/availability
Oxygen availability
Pathogenicity
Stress
Unstressed log-phase growth (control)

(b)

Figure 10: (a) Rank by TF filtered by species; (b): Rank by TF filtered by environmental conditions

To explain the functionalities, the example of TPO3 has been used which is a homolog of the gene PdR1 in Candida glabrata. Using the tool quick search on the homepage the user can gather all the Locus, Protein and GO information for the gene of interest.



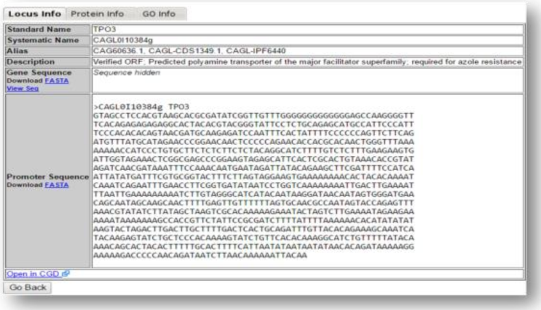
Locus Info | Protein Info | GO Info

Protein Name	Tpo3p
Description	Verified ORF; Predicted polyamine transporter of the major facilitator superfamily; required for azole resistance
GO Terms: Biological Process	. spermine transport . transmembrane transport
GO Terms: Molecular Function	. spermine transmembrane transporter activity
GO Terms: Cellular Component	. fungal-type vacuole membrane . plasma membrane . integral component of membrane

Open in CGD

Go Back

(a)



Locus Info | Protein Info | GO Info

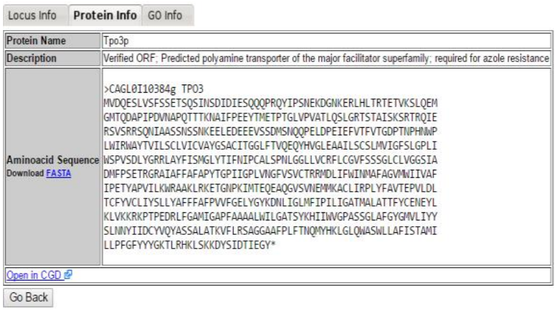
Standard Name	TPO3
Systematic Name	CAGL0I10384g
Alias	CAG60636.1, CAGL-CDS1349.1, CAGL-IPF6440
Description	Verified ORF; Predicted polyamine transporter of the major facilitator superfamily; required for azole resistance
Gene Sequence Download FASTA View Seq	Sequence hidden
Promoter Sequence Download FASTA	

```
>CAGL0I10384g TPO3
GTAGCCTCCACGTAAGCACGCGATATCGGTTGTTTGGGGGGGGGGGGGAGCCAAGGGGTT
TCACAGAGAGAGAGGCACTACACGTACGGGTATTCCTCTGCAGAGCATGCCATTCCCATT
TCCCACACACAGTAACGATGCAAGAGATCCAATTTCACTATTTTCCCCCCAGTTCTTCAG
ATGTTTATGCATAGAACCCGGAACAACTCCCCCAGAACACCACGCACAACTGGGTTTAAA
AAAAACCATCCCTGTGCTTCTCTCTTCTCTACAGGCATCTTTTGTCTCTTTGAAGAAGTG
ATTGGTAGAAACTCGGCGAGCCCGGAAGTAGAGCATTCACTCGCACTGTAAACACCGTAT
AGATCAACGATAAATTTCCAAACAATGAATAGATTATACAGAAGCTTCGATTTTCCATCA
ATTATATGATTTCGTGCGGTACTTTCTTAGTAGGAAGTGAAAAAAAACACTACACAAAAT
CAAATCAGAATTTGAACCTTCGGTGATATAATCCTGGTCAAAAAAAAATTGACTTGAAAAT
TTAATTGAAAAAAAAAATCTTGTAGGGCATCATACAATAAGGATAACAATAGTGGGATGAA
CAGCAATAGCAAGCAACTTTTGAGTTGTTTTTTAGTGCAACGCCAATAGTACCAGAGTTT
AAACGTATATCTTATAGCTAAGTCGCACAAAAAGAAATACTAGTCTTGAAAATAGAAGAA
AAAATAAAAAAAGCCACCGTTCTATTCCGCGATCTTTTATTTTAAAAAACACATATATAT
AAGTACTAGACTTGACTTGCTTTTGACTCACTGCAGATTTGTTACACAGAAAGCAAATCA
TACAAGAGTATCTGCTCCCACAAAAGTATCTGTTCACACAAAGGCATCTGTTTTTATACA
AAACAGCACTACACTTTTTGCACTTTTCATTAATATAATAATATAACACAGATAAAAAGG
AAAAAGACCCCCAACAGATAATCTTAACAAAAAATTACAA
```

Open in CGD

Go Back

(b)



Locus Info | Protein Info | GO Info

Protein Name	Tpo3p
Description	Verified ORF; Predicted polyamine transporter of the major facilitator superfamily; required for azole resistance
Aminoacid Sequence Download FASTA	

```
>CAGL0I10384g TPO3
MVDQESLVSFSSETSQSINSDIDIESQQQPRQYIPSNEKDGNKERLHLTRTETVKSLQEM
GMTQDAPIPDVNAPQTTTKNAIFPEEYTMETPTGLVPVATLQSLGRTSTAISKSRTRQIE
RSVSRRSQNIAASSNSSNKEELEDEEEVSSDMSNQQPELDPEIEFVTFVTGDPTNPHNWP
LWIRWAYTVILSCLVICVAYGSACITGGLFTVQEQYHVGLEAAILSCSLMVIGFSLGPLI
WSPVSDLYGRRLAYFISMGLYTIFNIPCALSPNLGGLLVCRFLCGVFSSSGLCLVGGSIA
DMFPSETRGRAIAFFAFAPYTGPIIGPLVNGFVSVCTRRMDLIFWINMAFAGVMWIIVAF
IPETYAPVILKWRAAKLRKETGNPKIMTEQEAQGVSVNEMMKACLIRPLYFAVTEPVLDL
TCFYVCLIYSLLYAFFFAFPVVFGELYGYKDNLIGLMFIPILIGATMALATTFYCENEYL
KLVKKRKPTPEDRLFGAMIGAPFAAAALWILGATSYKHIIWVGPASSGLAFGYGMVLIYY
SLNNYIIDCYVQYASSALATKVFLRSAGGAAFPLFTNQMYHKLGLQWASWLLAFISTAMI
LLPFGFYYYGKTLRHKLSKKDYSIDTIEGY*
```

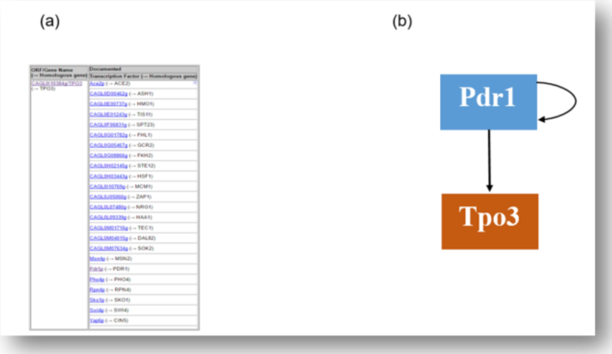
Open in CGD

Go Back

(c)

Figure 11: The gene information of TPO3- (a) GO information; (b) Locus information; (c) Protein information

For example, TPO3 shows no regulation in Candida glabrata based on the PathoYeastract prediction tool, while compared with the Candida albicans it shows PDR1 plays a role in the transcription of TPO3.



ORF/Gene Name (→ Homologous gene)	Documented Transcription Factor (→ Homologous gene)
CAGL0I10384g/TPO3 (→ TPO3)	Ace2p (→ ACE2)
	CAGL0D00462g (→ ASH1)
	CAGL0E00737g (→ HMO1)
	CAGL0E01243g (→ TIS11)
	CAGL0F06831g (→ SPT23)
	CAGL0G01782g (→ FHL1)
	CAGL0G05467g (→ GCR2)
	CAGL0G08866g (→ FKH2)
	CAGL0H02145g (→ STE12)
	CAGL0H03443g (→ HSF1)
	CAGL0I10769g (→ MCM1)
	CAGL0J05060g (→ ZAP1)
	CAGL0L07480g (→ NRG1)
	CAGL0L09339g (→ HAA1)
	CAGL0M01716g (→ TEC1)
	CAGL0M04015g (→ DAL82)
	CAGL0M07634g (→ SOK2)
	Msn4p (→ MSN2)
	Pdr1p (→ PDR1)
	Pho4p (→ PHO4)
	Rpn4p (→ RPN4)
	Sko1p (→ SKO1)
	Swi4p (→ SWI4)
	Yap6p (→ CIN5)

Figure 12: Regulation of the Tpo3 gene in (a) in *Saccharomyces cerevisiae*; (b) *Candida albicans*

It is interesting to observe that there is no documented regulation found for Tpo3 in Candida glabrata whilst there has been documented the regulation of Tpo3/Pdr1 regulations in both S. cerevisiae and C. albicans. With this regulatory documentation, the prediction could be made that though Pdr1 is present in both of this species and regulates the transcription of Tpo3; Pdr1 being the main regulator of azole drug resistance in this yeast species. It might be the case that the gene Tpo3 is not present in Candida glabrata. If that is the case, maybe some other gene e.g. Qdr2; which is a homolog of Tpo3 or Tac1 may play a role in transcription regulation in Candida glabrata. The transcriptional control of these homologous genes suggests, not only that indeed they play no direct role in drug resistance, but also strongly point to the hypothesis that they may serve more than paralogous roles.

Alternatively, using the “Search for Genes” query it is possible to pinpoint all the genes that have been shown to be regulated by a given transcription factor. Keeping to the drug resistance case study, using this tool it is possible to observe that there are 399 genes known to be regulated by the *C. glabrata* Pdr1 TF, whereas there are only 46 genes known to be regulated by the *C. albicans* Tac1 TF in (Figure 23).

Interestingly, the list of genes whose expression is controlled by Pdr1 or Tac1 is also quite diverse in terms of associated functions, far beyond the classical targets, the multidrug efflux pumps of the ATP-Binding Cassette superfamily Cdr1 and Cdr2. Both lists include shared genes and functions, such as Hsp12, a stress resistance related protein chaperone, the Erg11 gene, the target of azole antifungal drugs, but also genes associated to central metabolic pathways. For example, ADH1 and SNZ1 were identified as targets of the Tac1 TF, being related to central carbon metabolism and vitamin B synthesis, respectively. This observation, raises the possibility of either Tac1 playing additional roles in C. albicans biology or that Adh1 and Snz1 contribute somehow to drug resistance.

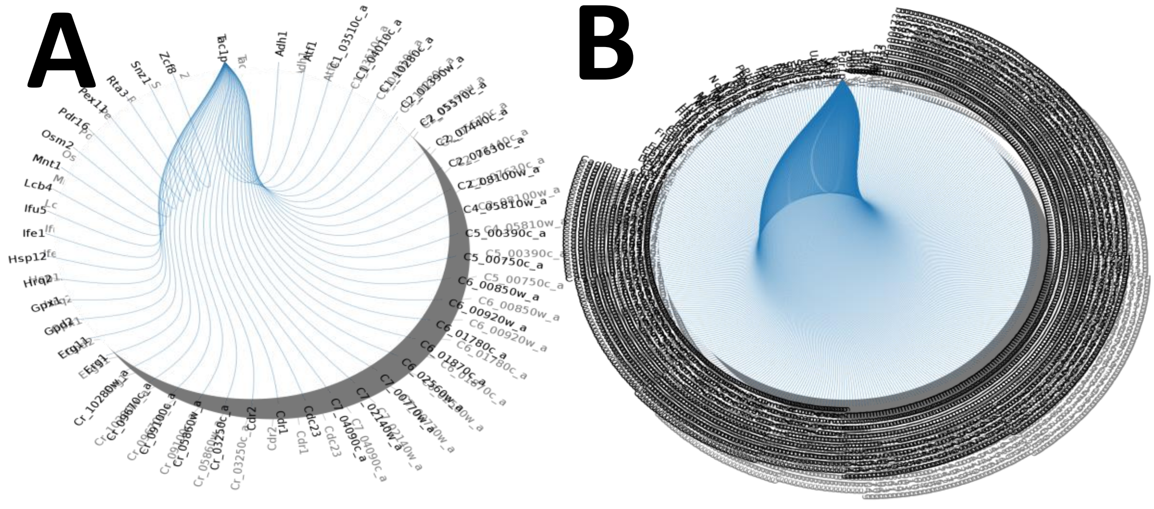
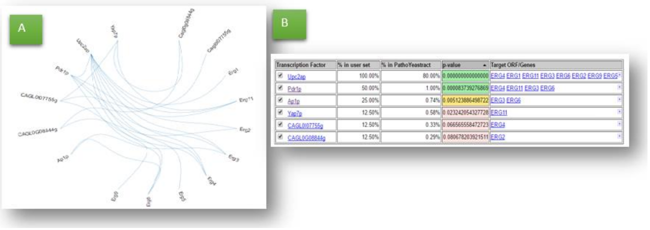


Figure 13 - The Tac1 / Pdr1 regulon in C. albicans (A) / C. glabrata (B) as obtained using the “Search for target genes” in the PathoYeastract database.

III. II Analyzing Genome-wide regulation

One of the key features of the genome-wide regulation study in the PathoYestract is aimed to study the regulatory network of a given sets of genes/TFs which controls a certain genome-wide expression remodeling. To use this feature, the query “Rank by TF” is used. By using this tool, the genome-wide regulation of the target genes can be predicted. The user provides a list of genes which are upregulated under certain environmental condition, the program searches for all the predicted TFs which are responsible for regulating the genes provided by the user and rank them by importance. For example, in Candida glabrata, Upc2ap has regulation for 100%, whereas Pdr1 is 50%. It supports the hypothesis that UPC2A is required for high-level azole antifungal resistance in Candida glabrata [23].



Transcription Factor	% in user set	% in PathoYeastract	p-value ▲	Target ORF/Genes
☑ Upc2ap	100.00%	80.00%	0.000000000000000	ERG4 ERG1 ERG11 ERG3 ERG6 ERG2 ERG9 ERG5
☑ Pdr1p	50.00%	1.00%	0.000083739276869	ERG4 ERG11 ERG3 ERG6
☑ Ap1p	25.00%	0.74%	0.005123886498722	ERG3 ERG6
☑ Yap7p	12.50%	0.58%	0.023242054327728	ERG11
☑ CAGL0I07755g	12.50%	0.33%	0.066565558472723	ERG4
☑ CAGL0G08844g	12.50%	0.29%	0.080678203921511	ERG2

Figure 14: Genome-wide study based on regulatory network (A) in C. glabrata; (B) Rank by TF according to importance

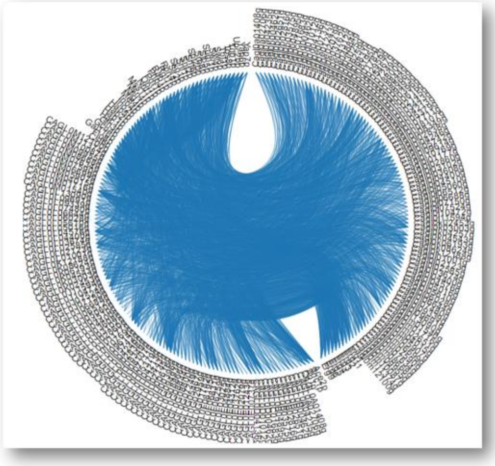


Figure 15: Regulation of genes in Saccharomyces cerevisiae predicted by the ‘’Rank by TF’’ tool in PathoYestract database

Similarly, for the same set of genes, Pdr1 is a common regulator in Saccharomyces cerevisiae with an importance/coverage of 62.50%, whereas there is no documented regulation is found in Candida albicans for those two regulators. It can be predicted from this study that some other regulator; e.g. Tac1 could be responsible for the regulations of a set of upregulated genes under the specific environmental conditions.

IV. CONCLUSION

IV.I Summary of achievements

With the radical change and development in the field of Systems Biology research in the past decade, the study of gene regulatory networks has become the inevitable state of the art technology. This cutting edge approach is not only fast, it is accurate in most of the cases. The greatest motivation to develop the PathoYeastract website and database is to aid the scientific community all around the world with a reliable and complete tool to focus and study on the transcription regulations of pathogenic yeasts on the molecular basis of candidaemia and its prophylaxis and treatment.

I have achieved so far to develop the working tool to integrate the both the documentation of the regulation data and constructing the gene regulatory network based on the search as well as represent them visually for the sake of the usability at the external user level. At present the web portal (http://pathoyeastract.org/) is fully functioning being publicly available and freely accessible. If someone is interested to study a specific set of TFs/genes for either *Candida glabrata* or *Candida albicans*, for the easement there had been two different tabs assigned on the homepage to redirect them to the databank and the webpage of the specific species. As there are regulation data for all the three species i.e. *Candida glabrata*, *Candida albicans* and *Saccharomyces cerevisiae* available on the database, the user can easily compare the gene regulations interspecies easily which provides the scope to study the gene regulatory network amongst different species resulting wider and more apprehensive study at a cross-species level.

However, as the size of the gene regulatory network increases, it makes it difficult to document and study the gene regulations. When the size of the network exceeds a certain size, it makes it eventually impossible to identify the interesting regulations and verify the predictions against the experimental results while the automated simulation using the computational tool can perform the tasks seamlessly. The curation of the website and the periodical update has been an important aspect from the very beginning of the development of the website. I was finally able to make the architecture of the database to adapt to the system so that although the verification of an updated source is manual, the update is automated with a script, but it needs human supervision for the unpredictable cases that always appear.

It enables the website to be always updated with new documented data relevant to the database. Though the curators of the website are always keen to provide the user with most updated and current data, it is important to empower the database system to perform the task on its own without the need of somebody performing the update manually which can be possibly done through further work on the database development.

IV.II Future Direction

Although the PathoYestract database and website is up and running in a fully functional state at this moment, there is still scope for further development and advancement. For example, at this moment there are documented regulations, potential regulations and TF binding sites for two *Candida* species and the model organism *Saccharomyces cerevisiae* have been documented in the website. There is a window of opportunity to expand the database with other closely related *Candida* species such as; *Candida parapsilosis* and *C. dubliniensis*, *C. orthopsilosis*, *C. krusei* and *C. lusitaniae* which can be proven significant to document genetic regulations similar to that in the species which have already been documented. The study of gene regulatory network in pathogenic yeast is emerging all over the world as researchers are focusing on different organisms. In the future the expansion of the database is possible especially through the development of more complex dedicated tools.

ACKNOWLEDGMENT

- Supervisors: Prof. Miguel Teixeira and Dr. Pedro Monteiro
- ERASMUS MUNDUS Association
- Portuguese Foundation for Science and Technology (contracts: PTDC/BBB-BIO/4004/2014 and UID/BIO/04565/2013)

BIBLIOGRAPHY

1. Mayer FL, Wilson D, Hube B. *Candida albicans* pathogenicity mechanisms. *Virulence*. 2013;4(2):119-128.
2. Pfaller MA, Diekema DJ. Epidemiology of Invasive Candidiasis: a Persistent Public Health Problem. *Clinical Microbiology Reviews*. 2007;20(1):133-163.
3. Caggiano G, Puntillo F, Coretti C, et al. Candida Colonization Index in Patients Admitted to an ICU. *International Journal of Molecular Sciences*. 2011;12(10):7038-7047.
4. Klempp-Selb B, Rimek D, Kappe R (2000) Karyotyping of Candida albicans and Candida glabrata from patients with Candida sepsis. Mycoses 43:159–163.
5. H. J. Lo, J. R. Köhler, B. DiDomenico, D. Loebenberg, A. Cacciapuoti, G. R. Fink. Nonfilamentous C. albicans mutants are avirulent. Cell. 1997 September 5; 90(5): 939–949.
6. Dias, P. J., & Sá-Correia, I. The drug: H+ antiporters of family 2 (DHA2), siderophore transporters (ARN) and glutathione: H+ antiporters (GEX) have a common evolutionary origin in hemiascomycete yeasts. BMC genomics, (2013), 14(1), 901.
7. Sá-Correia, I., dos Santos, S. C., Teixeira, M. C., Cabrito, T. R., & Mira, N. P. Drug: H+ antiporters in chemical stress response in yeast. *Trends in microbiology*, (2009). *17*(1), 22-31.
8. Coste AT, *et al.* TAC1, transcriptional activator of CDR genes, is a new transcription factor involved in the regulation of Candida albicans ABC transporters CDR1 and CDR2. *Eukaryot Cell* (2004) 3(6):1639-52.
9. Morschhauser J, *et al.* The transcription factor Mrr1p controls expression of the MDR1 efflux pump and mediates multidrug resistance in Candida albicans.*PLoS Pathog* (2007),3(11):e164.
10. Clarissa J. Nobile *et. al.*, A Recently Evolved Transcriptional Network Controls Biofilm Development in Candida albicans, Cell, (2012) Volume 148, Issues 1-2, p126– 138, 20.
11. Catarina Costa, Jonathan Ribeiro, Isabel M. Miranda, Ana Silva-Dias, Mafalda Cavalheiro,

Sofia Costa-de-Oliveira, Acácio G. Rodrigues and Miguel C. Teixeira. Clotrimazole Drug Resistance in *Candida glabrata* Clinical Isolates Correlates with Increased Expression of the Drug:H$^+$ Antiporters CgAqr1, CgTpo1_1, CgTpo3, and CgQdr2, *Frontiers in microbiology,* (2016).

12. Miguel C. Teixeira, Pedro T. Monteiro, Joana F. Guerreiro, Joana P. Gonçalves, Nuno P. Mira, Sandra C. dos Santos, Tânia R. Cabrito, Margarida Palma, Catarina Costa, Alexandre P. Francisco, Sara C. Madeira, Arlindo L. Oliveira, Ana T. Freitas, Isabel Sá-Correia. The YEASTRACT database: an upgraded information system for the analysis of gene and genomic transcription regulation in *Saccharomyces cerevisiae,* Nucl. Acids Res., (2014) 42: D161-D166, Oxford University Press.
13. Dário Abdulrehman, Pedro T. Monteiro, Miguel C. Teixeira, Nuno P. Mira, Artur B. Lourenço, Sandra C. dos Santos, Tânia R. Cabrito, Alexandre P. Francisco, Sara C. Madeira, Ricardo S. Aires, Arlindo L. Oliveira, Isabel Sá-Correia, Ana T. Freitas. YEASTRACT: providing a programmatic access to curated transcriptional regulatory associations in *Saccharomyces cerevisiae* through a web services interface Nucl. Acids Res., (2011) 39: D136-D140, Oxford University Press.
14. Pedro T. Monteiro, Nuno Mendes, Miguel C. Teixeira, Sofia d'Orey, Sandra Tenreiro, Nuno Mira, Hélio Pais, Alexandre P. Francisco, Alexandra M. Carvalho, Artur Lourenço and Isabel Sá-Correia, Arlindo L. Oliveira, Ana T. Freitas. YEASTRACT-DISCOVERER: new tools to improve the analysis of transcriptional regulatory associations in *Saccharomyces cerevisiae*. Nucl. Acids Res., (2008) 36: D132-D136, Oxford University Press.
15. Miguel C. Teixeira, Pedro Monteiro, Pooja Jain, Sandra Tenreiro, Alexandra R. Fernandes, Nuno P. Mira, Marta Alenquer, Ana T. Freitas, Arlindo L. Oliveira, and Isabel Sá-Correia. The YEASTRACT database: a tool for the analysis of transcription regulatory associations in *Saccharomyces cerevisiae.* Nucl. Acids Res., (2006) 34: D446-D451, Oxford University Press.
16. Tsai,H.K., Chou,M.Y., Shih,C.H., Huang,G.T., Chang,T.H. and Li,W.H. MYBS: a comprehensive web server for mining transcription factor binding sites in yeast. Nucleic Acids Res., (2007) 35, W221–W226.
17. Wingender,E., Chen,X., Fricke,E., Geffers,R., Hehl,R., Liebich,I., Krull,M., Matys,V., Michael,H., Ohnhauser,R. et al. The TRANSFAC system on gene expression regulation. Nucleic Acids Res., (2001) 29, 281–283.
18. van Helden,J. Regulatory sequence analysis tools. Nucleic Acids Res., (2003) 31, 3593–3596.
19. Chang,D.T., Huang,C.Y., Wu,C.Y. and Wu,W.S. YPA: an integrated repository of promoter features in Saccharomyces cerevisiae. Nucleic Acids Res., (2011) 39, D647–D652.
20. de Boer,C.G. and Hughes,T.R. YeTFaSCo: a database ofevaluated yeast transcription factor sequence specificities. Nucleic Acids Res., (2012) 40, D169–D179.
21. Inglis DO, Arnaud MB, Binkley J, Shah P, Skrzypek MS, Wymore F, Binkley G, Miyasato SR, Simison M, Sherlock G. The Candida genome database incorporates multiple Candida species: multispecies search and analysis tools with curated gene and protein information for Candida albicans and Candida glabrata. *Nucleic Acids Res* (2012) 40(Database issue): D667-74.
22. Cherry JM[1], Hong EL, Amundsen C, Balakrishnan R, Binkley G, Chan ET, Christie KR, Costanzo MC, Dwight SS, Engel SR, Fisk DG, Hirschman JE, Hitz BC,Karra K, Krieger CJ, Miyasato SR, Nash RS, Park J, Skrzypek MS, Simison M, Weng S, Wong ED. Saccharomyces Genome Database: the genomics resource of budding yeast. Nucleic Acids Res. 2012 Jan;40(Database issue): D700-5.
23. Whaley SG, Caudle KE, Vermitsky JP, Chadwick SG, Toner G, Barker KS, Gygax SE, Rogers PD. UPC2A is required for high-level azole antifungal resistance in Candida glabrata. Antimicrob Agents Chemother. 2014 Aug;58(8):4543-54.