UNIVERSITY OF NAPLES FEDERICO II

SCHOOL OF MEDICINE

PhD Thesis

in

Neuroscience and Pathology of Brain Aging

Antipsychotic induced Gene Expression in the Rat Brain: a Computational and Experimental Approach

Tutor:

Prof. Andrea de Bartolomeis, MD, PhD.

PhD student:

Alberto Ambesi Impiombato, MD.

ACCADEMIC YEAR 2005-2006

Table of Contents

list of figures

list of tables

Acknowledgements

This work would not have been possible without the support and encouragement of Prof. Andrea de Bartolomeis, under whose supervision I chose this topic and began the work for this thesis. He has introduced me to the world of neuroscience and has motivated and supported me throughout my years at University. It is difficult to overstate my gratitude to Dr. Diego Di Bernardo, my advisor in the final stages of my academic itinerary. He has introduced me to computational and systems biology, and always provided the most stimulating working environment. Both have the special gift of understanding the appropriate time to put me under pressure, or to give me confidence.

I am grateful also to my advisor prof. Silvestro Formisano for his academic and life guidance since the beginning of my University studies, and to prof. Giovanni Muscettola for his support and role in my post-graduate education in neuroscience. I also thank Dr. Pietro Liò for teachings in computational biology at the Computer Lab in Cambridge, UK, as well as for his kind support.

Special thanks to go all my friends and colleagues that have helped me, intellectually and by providing a cheerful everyday environment including Mukesh, Santosh, Giusy, and Rossella at the Tigem Institute, as well as Fabio, Marilena, Felice, Germano and Daniela at Policlinico, University of Naples.

Mukesh has been especially helpful for his technical advice sharing his knowledge in computer science.

Thanks are due also to my brothers Massimo and Riccardo who have often been an example because of their good judgment and successful lives, as well as all of my friends, including my best and authentic friends Emilio and Giordano who gave me moral support at times when it was most needed. Last but not least, big thank you to my beloved Elisabeth for her everyday caring and faithfulness.

Thank you Steve Jobs for ‘thinking different’ and making the computer world a better place. I can’t even think of how I could have done all this work on a PC.

I cannot end without thanking my parents, whose constant encouragement and love I have relied on, throughout my life.

Dedication

This thesis is dedicated to my family, without whom none of this would have been even possible. Of course my family includes my uncles Claudio and Mariolina, as well as my cousins Roberta and Marco. This work is especially dedicated to my three wonderful nephews Francesco, Cristina, and Giorgia, whom I love so much and to whom I wish everyday happiness.

"As far as the laws of mathematics refer to reality, they are not certain; and as far as they are certain, they do not refer to reality."
Albert Einstein

"If you want happiness for an hour – take a nap.
If you want happiness for a day – go fishing.
If you want happiness for a month – get married.
If you want happiness for a year – inherit a fortune.
If you want happiness for a lifetime – help someone else."
Chinese proverb

"Stay hungry, Stay foolish."
Steve Jobs

Chapter 1

Introduction

1.1 Aim of the thesis

This thesis aims to investigate the gene response patterns to different classes of psychoactive compounds, with a special focus on antipsychotic agents and drugs that affect dopaminerigc transmission. The first part of the thesis presents the experimental results of drug induced gene expression in rat brain. Those results were obtained by means of *in situ* hybridization and image analysis, allowing us to detect specific patterns of gene expression by different classes of drugs. We show that *Homer 1* gene is differentially expressed by typical and atypical

antipsychotics, which has important implications for the clinical distinctive features of those two classes of drugs. Drug target identification is also explored in the second part of this thesis by Systems and Computational Biology approaches, which offer novel strategies to analyze the ever increasing amounts of biological data resulting form the high throughput technologies that are becoming commonplace in the post-genomic era. We are now facing the challenge to make sense of large amounts of data in order to better understand the complexity of cellular mechanisms, such as the response to drugs. An assessment of the current approaches to drug target identification is outlined in Chapter 3, based on my review that was recently published on *Current Bioinformatics* (Ambesi-Impiombato and Di Bernardo, 2006). Finally the feasibility of computational predictions of gene regulation as an aid to drug response investigation are discussed in Chapter 6, with the presentation of a computational framework for binding site predictions validated on *in silico* promoter sequence simulations as well as on 'real' promoters of genes for which regulation is known by literature. After validation of our computational framework on those datasets predictions of the transcription factor that regulate the *Homer 1* gene are presented. The computer code of this computational framework was written in Java (http://java.sun.com) and Matlab® (www.mathworks.com), with examples shown in Appendices A and B.

1.2 Antipsychotics and gene expression

Antipsychotic drugs are the mainstay of the treatment of schizophrenia. The new class of drugs referred to as 'atypical' antipsychotics is now extensively adopted as pharmacological therapy of psychotic patients (Lieberman et al., 2005). Compared to the 'typical' antipsychotics these newer medications are equally effective in reducing the positive symptoms like hallucinations and delusions, and have a lower incidence of extrapyramidal side effects (EPSEs) (Arnt, 1998). Arguably atypical antipsychotics may also be more effective at relieving the negative symptoms of the schizophrenia, such as withdrawal and flattened affect (Beasley et al., 1996; Borison et al., 1996). Neuronal expression of immediate-early genes (IEGs) such as *c-fos* in response to antipsychotics (Morgan and Curran, 1991) may provide a better tool for the screening of their pharmacological profile, and for understanding the mechanisms that underlie the distinctive clinical features of atypical antipsychotics. Sampling the response to chronic treatments in animal models may more accurately resemble what is required in order to obtain the pharmaceutical effects in clinical practice, and it may help investigating the long-term mechanisms involved in stimulus-induced neuronal plasticity.

Typical and atypical antipsychotics have been demonstrated to affect differently neuronal gene expression in several preclinical paradigms (Angulo et al., 1990; Merchant and Dorsa, 1993; Robertson and Fibiger, 1992; Semba et al., 1996). The identification of new preclinical predictors of 'atypicality' in animal models can provide a powerful tool for investigation, as well as a potential means of preclinical characterization of putative novel antipsychotic agents and may shed

light on neurotransmitters and trasductional systems involved. Typical and atypical antipsychotics have already been demonstrated to differently affect neuronal expression of early genes in several preclinical paradigms (Angulo et al., 1990; Merchant and Dorsa, 1993; Robertson and Fibiger, 1992; Semba et al., 1996). Recently, a differential expression pattern of postsynaptic density protein homer 1a has been reported for typical and atypical antipsychotics and has been proposed as a putative preclinical characterization of antipsychotic agents (de Bartolomeis et al., 2002).

The D_2 dopamine receptor blockade is shared by virtually all antipsychotic agents, thus it is considered a crucial mechanism for their clinical efficacy, but it also affects the liability to extrapyramidal side effects (EPSEs) due to impairment of the striato-nigral dopaminergic system. The introduction of atypical antipsychotics and analysis of experimental in vitro and in vivo (by PET) data proved that the two effects may be separated using certain drugs at appropriate dosages (Kapur et al., 2000a). Their lower propensity to induce EPS is thought to depend on their ability to preferentially affect mesolimbic dopaminergic system as opposed to the typical antipsychotics which inhibit both mesolimbic and mesostriatal systems (Scatton and Sanger, 2000). Other mechanisms may explain the clinical difference of the two classes: the $5HT_2/D_2$ receptor affinity ratio (Meltzer et al., 1992), the multiple receptor targeting (Bymaster et al., 2003) and the fast dissociation from D_2R (Kapur and Seeman, 2001; Tauscher et al., 2004). However, the molecular mechanisms involved in the distinctive clinical and pharmacodynamic properties of atypical antipsychotics remain not fully understood.

1.3 Homer genes as antipsychotics targets

Proteins of the Homer family are products of three distinct genes in mammals. They are localized at the postsynaptic density (PSD) of excitatory synapses and interact, through a conserved amino-terminal EVH1 domain which binds to a proline rich sequence, with the C-terminal intracellular tail of group 1 metabotropic glutamate receptors (mGluRs), inositol 1-4-5-triphosphate receptor (IP3R), ryanodine receptors (RyRs), transient receptors potential canonical-1 (TRPC-1) ion channels, and the NMDA glutamate receptor scaffolding protein *shank* (Tu et al., 1998; Xiao et al., 1998). Through its C-terminus coiled-coil (CC) domain Homer proteins multimerize, creating a reticular machinery at the PSD. Previous studies have demonstrated that homer gene is regulated as an immediate early gene (IEG) and can be induced by dopaminergic modulations, light exposure, and maximal electroconvulsive seizures (Brakeman et al., 1997). *Homer 1* gene encodes for a number of transcriptional variants some of which, such as *Homer 1a* and *ania-3*, are induced as IEGs and play a relevant and direct role in the modulation of glutamate synaptic plasticity at the level of the PSD. Both stimulus-responsive isoforms contain the EVH1 domain but lack the CC motif required for dimerization (Bottai et al., 2002), acting as natural 'dominant negatives' by disrupting CC-Homer interactions with EVH1-bound proteins (Xiao et al., 1998). Their overexpression ultimately results in a modification of synaptic architecture (Sala et al., 2003), a redistribution of CC-Homer expression (Inoue et al., 2004), and an alteration in excitatory synaptic transmission (Hennou et al., 2003; Minami et al., 2003). Homer 1a and ania-3 proteins differ only in their C-terminal 10 and 41 amino acids respectively, and it is not known whether they are

differentially regulated, nor if this difference in their amino acid sequence has any functional consequence. Homer 1 gene family is implicated in several behavioral disorders (de Bartolomeis and Iasevoli, 2003; Lominac et al., 2005; Szumlinski et al., 2006b) such as schizophrenia (Norton et al., 2003; Szumlinski et al., 2005a) fragile X syndrome (Giuffrida et al., 2005) alcohol dependence (Szumlinski et al., 2005b) and cocaine addiction (Dahl et al., 2005; Kalivas et al., 2004; Szumlinski et al., 2006a)as well as in motor dysfunction (Tappe and Kuner, 2006).

In previous studies, we have shown that *Homer 1* is strongly upregulated in caudate-putamen and nucleus accumbens by haloperidol and only in accumbens by atypical antipsychotics such as clozapine and olanzapine (de Bartolomeis et al., 2002; Polese et al., 2002). Thus we have proposed the regulation of *Homer 1* gene expression in rat striatum as a novel preclinical characterization of antipsychotics.

1.4 Systems Biology and Drug Discovery

Systems and Computational Biology are emerging as the new disciplines of the post-genomic era that could help predict drug targets through the analysis of the increasingly available data obtained by high throughput technologies. Computational Biology can be defined as the development and application of data-analytical and theoretical methods, mathematical modeling and computational simulation techniques to the study of biological, behavioral, and social systems. Systems Biology on the other hand focuses on measuring and monitoring biological systems on the system level (Quantitative Systems Biology), as well as on mapping, explaining and predicting systemic biological

processes and events through the building of computational and visualization models (Systems Biology Modeling).

The drug discovery process is complex, time consuming and very expensive. Typically, the time to develop a candidate drug is about 5 years, while the clinical phases leading, possibly, to the commercial availability of the drug are even longer (>7 years) for a total cost of more than 700 Million dollars (DiMasi et al., 2003). The drug discovery process begins from the identification of an area of "unmet medical needs" and then proceeds by identifying "druggable" biological targets that could relief the symptoms of the disease, or, as in the recent years, that are involved in the causative process of the disease. The pharmaceutical industry is moving from a symptomatic relief focus towards a more pathology-based approach where a better understanding of the pathophysiology should help deliver drugs whose targets are directly involved in the causative processes underlying the disease (Ratti and Trist, 2001). Identification of drug novel targets will help the industry in the discovery on novel drugs that are more effective and with lower toxicity.

1.5 Computational prediction of gene regulation

Control of gene expression is essential to the establishment and maintenance of all cell types, and is involved in pathogenesis of several diseases, possibly including many complex diseases, such as mental disorders (Hong et al., 2005). Neuronal gene expression regulation is expected to be more complex than other cell types. It is largely orchestrated by transcription factors (TFs) that activate and repress

specific cohorts of genes in both neural and non-neural cells, required for differentiation of adult neural stem cells and is implicated in several neuropathologies including Huntington's disease, epilepsy and ischemia. Possibly all mental disorders including schizophrenia and mood disorders, for which a biological component is strongly supported by evidence, may be caused by a dysregulation of neural gene expression during development or adulthood, rather than by structural variations in proteins (Hong et al., 2005). The identification of genes that encode novel targets of neural-specific transcription factor will provide insights into the pathogenesis of mental disorders and in the identification of clinically relevant drug-induced gene expression patterns. Although the possibility of predicting the regulation of gene expression is appealing, the underlying biological mechanisms are not completely understood, and the development of bioinformatics tools capable of accurate predictions is far from trivial. It is known that the mechanisms of regulation of gene expression involve the binding of TFs to regulatory elements on gene promoters, known as Transcription Factor Binding Sites (TFBSs), but attempts to computationally predict such elements in DNA sequences of gene promoters typically yield an excess of false positives.

Computational identification of cis-Regulatory Elements (CREs) is currently based mainly on three different approaches: (i) identification of conserved motifs using interspecies sequence global alignments (Pennacchio and Rubin, 2001); (ii) motif-finding algorithms that identify previously unknown motifs that are overrepresented in the promoters of co-expressed genes (Bailey and Elkan, 1994; Bussemaker et al., 2001; Eskin and Pevzner, 2002; Fujibuchi et al., 2001; Hughes et al., 2000; Palin et al., 2002; Sudarsanam et al., 2002); (iii) computational

detection of previously known motifs in promoters of genes for which regulating TFs are unknown (Kel et al., 2003). Limitations of the first approach are caused by the high mutation, deletion and insertion rates in gene promoter regions (Ludwig, 2002) that prevent a correct alignment of the promoter region, and several other reasons, including rearrangements of binding sites within the non-coding regions or changes in regulation of the ortholog genes. The second approach requires a large number of sequences containing a highly overrepresented motif. The third approach seems promising since the quality of the motif models of each TF is increasing, allowing for more accurate predictions of unknown target genes.

Accurate predictions require the use of an appropriate statistical background model of DNA sequence and integration of several sources of data, such as genomic sequence of gene promoters, as well as genomic sequence of ortholog genes, and gene expression data. Different strategies have been proposed to improve the accuracy of predictions, such as using a statistical background model or the information vector of a position weight matrix (PWM) (Kel et al., 2003), or, more recently, motif co-occurrence (Bulyk et al., 2004). A promising approach was recently shown to successfully predict TFBSs in higher eukaryotic genomes by considering overrepresented combinations of motifs in phylogenetically conserved regions and correlate them with expression profiles (Zhu et al., 2005).

Tadesse et al. (Tadesse et al., 2004) could successfully improve specificity of the identification of DNA regulatory motifs by fitting a linear regression model to microarray data in yeast. A novel computational tool was recently released by Hallikas et al. (Hallikas et al., 2006) for the prediction of distal enhancer elements

in mammalian genomes, based on both genomic sequence and conservation. This method tries to detect highly conserved sequences containing clusters of TFBSs by aligning large stretches (50kb) of genomic DNA from two species. Our focus is somewhat complimentary, as we try to detect TFBSs in the proximal promoter of vertebrate genes as opposed to distal enhancers. Proximal promoters cannot be easily aligned with promoters of ortholog genes, however, our method takes conservation into account in a way that does not require alignment. Conlon et al. (Conlon et al., 2003) showed recently that integration of gene expression profiles and PWM scores through a linear regression analysis can indeed improve the prediction accuracy.

Chapter 2

The *in situ* Hybridization Method

2.1 Abstract

Based on a previously published methodology paper (Ambesi-Impiombato et al., 2003) I describe the detailed method for quantitative in situ hybridization histochemistry adopted in the experiments described in this thesis.

2.2 Animals

Male Sprague-Dawley rats of approx. 250g were obtained from Harlan Laboratories (Udine, Italy). The animals were housed and let to adapt to human handling in a temperature and humidity controlled colony room with 12/12h light/dark cycle (lights on from 6:00 a.m. to 6:00 p.m.) with *ad libitum* access to lab chow and water. All procedures were conducted in accordance with the NIH *Guide for Care and Use of Laboratory Animals* (NIH Publication N0.85-23, revised, 1985) and were approved by local Animal Care and Use Committee.

2.3 Special equipment

2.3.1 Histology and molecular biology equipment

- Refrigerator (4°C), freezer (-20°C) and deep freezer (-70°C).
- Cryostat, OTF (Bright Instrument Co., Ltd., Cambridgeshire, UK).
- Microcentrifuge 5415D (Eppendorf S.r.l., Milan, Italy).
- ProbeQuant G-50 Micro Columns (Amersham Biosciences; Milan, Italy).
- Scintillation counter, LS3801 (Beckman Coulter S.p.A., Milan, Italy).
- Variable-temperature waterbath and variable-temperature incubator (Delchimica Scientific Glassware; Naples, Italy).

2.3.2 Imaging and data analysis

- Kodak-Biomax MR Autoradiographic film (Amersham Biosciences; Milan, Italy), Kodak X-Omatic light-tight cassettes (Kodak; USA) and darkroom facilities.

- Light box (Northern Light), camera (Sierra Scientific, Imaging Research Inc., St. Catherine's, Ontario), video interface (QuickCapture, Data Translation, Inc., Marlboro, MA), Radius PrecisionColor Display/20 (Radius, Inc., San Jose, CA), transparency film scanner Umax PowerLook 1100PRO (Umax UK Ltd., Gomshall Surrey, UK), Apple PowerPC G3 and ImageJ 1.28 (W. Rasband, NIMH, Bethesda, MD).

2.4 Chemicals and reagents

2.4.1 Chemicals

- Dextran sulfate, gelatin, chromium potassium sulfate (Chrome alum), triethanolamine (TEA), heparin sulphate, H_2O depc-treated and autoclaved, ethylenediaminetetraacetic acid (EDTA), acetic anhydride, Tris HCl, sodium dodecyl sulphate (SDS) (Sigma-Aldrich; Milan, Italy).

- Formamide, SSC, ethanol (100%), dithiothreitol (DTT), chloroform, formaldehyde, phosphate saline buffer (PBS), sodium pyrophosphate, sodium chloride, potassium phosphate monobasic, sodium phosphate dibasic, sodium hydroxide, hydrochloric acid (Delchimica Scientific Glassware; Naples, Italy).

- Terminal deoxynucleotidyl transferase kit (TdT, 15 units/ml, terminal transferase buffer, cobalt chloride) (Roche; Milan, Italy).

- ^{35}S-dATP (Specific Activity >1000 Ci/mmol; Amersham Biosciences; Milan, Italy).

- Killik-Frozen section medium (Bio-Optica; Milan, Italy).

- Kodak x-ray developer and fixer (Kodak; Chalon S/Saone, France).

2.4.2 Drugs

- Olanzapine, powder (Ely-Lilly, Indianapolis, IN).

- Haloperidol injectable solution (Lusofarmaco, Milan, Italy).

2.4.3 Oligonucleotide probes

- The Homer probe was a 45-base oligodeoxyribonucleotide complementary to bases 805-849 of the rat Homer mRNA (GenBank # U92079) (MWG Biotech; Florence, Italy).

- The PSD-95 probe was a 45 base pair oligodeoxyribonucleotide complementary to bases 225-269 of the rat PSD-95 mRNA (GenBank # M96853) (MWG Biotech; Florence, Italy).

2.4.4 Solutions

- Subbing solution: gelatin 0.25% (w/v), Chrome alum 0.025% (w/v) added just before use.

- Probe labeling and purification: oligonucleotide (stock solution 100 pmol/µl; working solution 5 pmol/µl); reagents supplied in TdT kit (15 units/ml terminal deoxynucleotidyl transferase in 200mM potassium cacodylate, 200mM KCl, 1mM EDTA, 4mM 2-mercaptoethanol, 50% glycerol (v/v), pH 6.5; 5X tailing buffer; 25mM cobalt chloride); 10µCi/µl of [α^{35}S] dATP; STE solution (100mM NaCl, 20mM TRIS-HCl, 10mM EDTA); 1M DTT.

- Tissue fixation: 1.5% Formaldehyde (v/v), 0.01M phosphate saline buffer (PBS), pH 7.4.

- Prehybridization washes: 0.01M phosphate saline buffer (pH 7.4;PBS), 0.25% acetic anhydride (v/v), 0.1M TEA, 0.9% NaCl (w/v), pH 8.0; 80%, 95%, 100% Ethanol (v/v); 100% Chloroform.

- Hybridization: sterile hybridization buffer (0.1% sodium pyrophosphate (w/v), 0.2% sodium dodecylsulphate (w/v), 0.02% heparin sulphate (w/v), 4mM EDTA, 80mM TRIS-HCl, 600mM NaCl, 50% formamide (v/v), 10% dextran sulphate (v/v), 100mM DTT); ^{35}S-probe diluted to 1-2x10^6 cpm/100µl in hybridization buffer.

- Post-hybridization washes: 1X SSC; 2X SSC, 50% formamide (v/v); 70% ethanol (v/v).

2.4.5 Radioactive standards

ARC-146 (CG) slide containing a scale of sixteen known amounts of ^{14}C standards ranging from 0.00 to 35.00 µCi/g (American Radiolabeled Chemicals, Inc., St. Louis, MO, USA).

2.5 Detailed procedure

2.5.1 Drug treatment and tissue preparation

On the day of the experiment rats are randomly assigned to the treatment groups. The brains are rapidly removed, quickly frozen on powdered dry ice and stored at -70^{O}C prior to sectioning.

2.5.2 Tissue sectioning

Serial coronal sections of 12µm are cut on a cryostat at -18°C, through the forebrain using the rat brain atlas of Paxinos and Watson (Paxinos and Watson, 1997) as an anatomical reference (approx. from bregma 1.20mm to 1.00mm). Care is taken to select (Paxinos and Watson, 1997) identical anatomical levels of treated and control sections using thionin-stained reference slides. Sections are thaw-mounted onto gelatin-coated slides, and stored at -70^{O}C for subsequent analysis.

2.5.3 Radiolabeling and purification of oligonucleotide probes

DNA probes of 45-base oligodeoxyribonucleotide complementary to the transcripts of interest are selected. For each probe a 50µl labeling reaction mix is prepared on ice using depc treated water, 1X tailing buffer, 1.5mM $CoCl_2$, 7.5pmol/µl of oligo, 125 Units of TdT and 100 µCi ^{35}S-dATP. The mix is incubated 20 min at 37°C. The unincorporated nucleotides are separated from radiolabeled DNA using ProbeQuant G-50 Micro Columns (Amersham Biosciences; Milan, Italy).

2.5.4 In situ hybridization

All solutions are prepared with sterile double distilled water. The sections are fixed in 1.5% formaldehyde in 0.12 M sodium-phosphate buffered saline (PBS, pH 7.4), quickly rinsed three times with 1xPBS, and placed in 0.25% acetic anhydride in 0.1 M triethanolamine/0.9% NaCl, pH 8.0, for 10 minutes. Next, the sections are dehydrated in 70%, 80%, 90% and 100% ethanol, delipidated in chloroform for 5 minutes, rinsed again in 100% and 95% ethanol and air dried.

Sections are hybridized with 0.4-0.6x10^6 cpm of radiolabeled oligonucleotide in buffer containing 50% formamide, 600mM NaCl, 80mM Tris-HCl (pH 7.5), 4mM EDTA, 0.1% pyrophosphate, 0.2mg/ml heparin sulfate, and 10% dextran sulfate. Slides are covered with coverslips and incubated at 37^{O}C in a humid chamber for 20 hours. After hybridization the coverslips were removed in 1X SSC and the sections are washed 4x15 minutes in 2xSSC/50% formamide at 40^{O}C, followed by two 30 min washes with 1xSSC at 40^{O}C. The slides are rapidly rinsed in distilled water and then in 70% ethanol.

2.5.5 Autoradiography

The sections are dried and exposed to Kodak-Biomax MR Autoradiographic film (Amersham Biosciences; Milan, Italy) for 3-30 days. A slide containing a scale of 16 known amounts of ^{14}C standards are co-exposed with the samples. The optimal time of exposure is chosen to maximize signal to noise ratio but to prevent optical density from approaching the limits of saturation. Film

development protocol included a 1.5 min dip in the developer solution and 3 min in the fixer.

2.5.6 Image analysis

The quantitation of the autoradiographic signal is performed using a computerized image analysis system including: a transparency film scanner (Microtek Europe B. V., Rotterdam, The Netherlands), an Apple PowerPC G4, and ImageJ software (v. 1.36, Rasband, W.S., http://rsb.info.nih.gov/ij). Sections on film are captured individually. Each experimental group contained 4-6 animals. Each slide contained 3 adjacent sections of a single animal. All hybridized sections to be compared are exposed on the same sheet of X-ray film. Forebrain sections are analyzed in the regions of interest (ROIs) including the following (Figure 2-1): parietal and frontal cortex, caudate-putamen subregions (dorsolateral, dorsomedial, ventromedial, and ventrolateral), and nucleus accumbens (core and shell). ROIs were outlined on digitized autoradiograms using an oval template tool of ImageJ software and the mean optical density is measured within each ROI. Sections are quantitated blind to the treatment conditions. In order to test for inter-observer reliability an independent quantitation is performed by a second investigator. Only quantitatively comparable results, in terms of consistency of statistically significant effects obtained by the two investigators, are considered reliable.

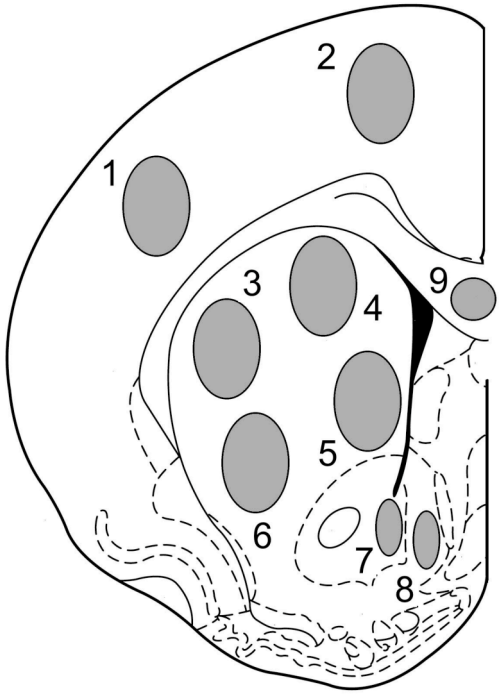


Figure 2-1:diagram of regions of interest (ROIs) quantitated on autoradiographic film images in rat forebrain. 1 = Frontal cortex (FC); 2 = Parietal cortex (PC); 3 = dorsolateral caudate-putamen (DL); 4 = dorsomedial caudate-putamen (DM); 5 = ventromedial caudate-putamen (VM); 6 = ventrolateral caudate-putamen (VL); 7 = core of accumbens (AcbCo); 8 = shell of accumbens (AcbSh); 9 corpus callosum. Modified from Paxinos and Watson Rat Brain Atlas (Paxinos and Watson, 1997).

2.6 Data processing

Measurements of optical density (OD) within ROIs are converted a calibration curve based on the standard scale co-exposed to the sections. Standard values from 4 through 12 have been previously cross-calibrated to ^{35}S brain paste standards, in order to assign a dpm/mg tissue wet weight value to each OD measurement through a calibration curve. For this purpose a "best fit" 3rd degree polynomial is used. For each animal, measurements from the 2-4 adjacent sections

are averaged. Data are analyzed for treatment effects by a One Way Analysis of Variance, (ANOVA). Student-Neuman-Keuls *post hoc* test is used to determine the locus of effects in any significant ANOVA.

2.7 Troubleshooting

2.7.1 High overall background signal

- As we experienced technical problems (high and non-uniform background signal) using commercially gelatin-coated slides, we suggest to pre-treat slides with a subbing solution made fresh each time.

- Use fresh dextran in hybridization mix;

- It is crucial to check the homogeneity of the final hybridization mix in order to avoid unequal distribution of probe and mixing it gently if necessary. It may be useful repeat this step several times when pipetting the mix over a large number of slides.

- If adopting the CCD camera acquisition method warm up the lightboard for aprox. 30 min before use in order to stabilize light intensity.

2.7.2 Probe specificity

When designing a novel oligo probe not found to be adopted previously in literature, we follow the following strategy. We select a sequence such to maximize the following parameters: GC percent close to 60%; least number of

secondary structures as oligo duplexes or hairpin stem formations in the antisense sequence. The sequence is then tested for specificity through a BLAST search (www.ncbi.gov). The hybridization specificity is finally tested by means of an in situ hybridization experiment comparing the antisense signal distribution with the sense oligonucleotide.

2.8 Alternative and support protocols

2.8.1 Slide subbing protocol

Soak uncoated slides in distilled water and soap for 1 hr, rinse thoroughly, dip slides into subbing solution (gelatin 0.25% (w/v), Chrome alum 0.025% (w/v) added just before use) twice, allowing to drain on paper towel for 1hr between cycles. Store when thoroughly dry (leave overnight loosely covered).

2.8.2 Data processing

The statistical analysis of data could benefit from the application of optional preprocessing steps in order to reduce unspecific signal variability. Such variability could arise for several reasons: heterogeneity of different parts of film, overall signal intensity variation across different hybridized slides and preanalytic variability due to the precision limits of the image acquisition system. Those preprocessing options include: smoothing algorithm, histogram based background subtraction, optical density ratio between the region of interest and corpus callosum of the same section.

Finally, we suggest a method for quantitation of small ROIs difficult to outline manually, based on the application of a lower density level threshold filter. Such filter is a tool provided by ImageJ software that limits the measurement of any ROI selection to precisely those pixels whose density level exceeds the arbitrarily chosen threshold. Application of this filter is very useful when measuring small regions with a high density level compared to its surroundings, such as Ammon's Horn (CA) and the Dentate Gyrus (DG) of the hippocampus. An appropriately chosen threshold allows a consistent region delimitation across all sections, devoid of bias due to manual outline.

2.9 Quick procedure

i. Tissue preparation: randomly assign rats to treatment group, inject, decapitate after 3 hours. Remove and rapidly freeze brains with powdered dry ice. Store at –70°C.

ii. Tissue sectioning: cut serial coronal sections (12µm) and mount onto gelatin-coated slides. Store at –70°C until hybridized.

iii. Radiolabeling and purification of oligonucleotide probes: label oligonucleotide DNA probes with ^{35}S-dATP by means of a terminal transferase reaction. Purify probe using ProbeQuant G-50 Micro Columns.

iv. In situ hybridization: sections are fixed, dehydrated, and delipidated. Hybridize sections with radiolabeled oligonucleotide probe in hybridization buffer, cover slides with coverslips and incubate at 37^{O}C in

a humid chamber for 20 hours. Remove coverslips, wash slides, allow to air dry and place in imaging cassette with film and ^{14}C standard slides. Expose for 3-11 days.

v. Signal detection and quantitation: capture sections, measure optical density of regions of interest.

vi. Data analysis: preprocess data (optional), analyze the treatment effects by means of the ANOVA and determine the locus of any significant effect using a post hoc test.

Chapter 3

Computational Biology and Drug Discovery

3.1 Abstract

The drug discovery process is complex, time consuming and expensive, and includes preclinical and clinical phases. The pharmaceutical industry is moving from a symptomatic relief focus towards a more pathology-based approach where a better understanding of the pathophysiology should help deliver drugs whose targets are involved in the causative processes underlying the disease. Computational biology and bioinformatics have the potential not only to speed up

the drug discovery process, thus reducing the costs, but also to change the way drugs are designed. In this review we focus on the different computational and bioinformatics approaches that have been proposed and applied to the different steps involved in the drug development process. The development of 'network-reconstruction' methods is now making it possible to infer a detailed map of the regulatory circuit among genes, proteins and metabolites. It is likely that the development of these technologies will radically change, in the next decades, the drug discovery process, as we know it today. This chapter is based on my recently published review on *Current Bioinformatics* (Ambesi-Impiombato and Di Bernardo, 2006).

3.2 Introduction

The drug discovery process is very similar across different pharmaceutical companies. It consists of preclinical and clinical phases. In the *target identification and validation* step, "druggable" biological targets are identified. In the *hit identification* step, library of compounds ranging from tens to hundreds of thousands of compounds are screened against the "druggable" targets to identify those compounds that "hit" the targets using high throughput screening (HTS). HTS methods based on experimental assays are reviewed extensively elsewhere (Hart, 2005). The number of compounds selected after this step is in the order of hundreds. By analyzing the structure of the selected compounds and identifying common active substructures, novel compounds containing those substructures are synthesized to significantly lower the number of lead compounds. This step is

called *lead identification.* Structural bioinformatics and chemical informatics approaches to drug discovery are particularly useful in this step, however, widely used methods like structure-activity relationship (SAR) are outside the scope of this review. We refer the interested reader to Bredel *et al.* (Bredel and Jacoby, 2004) and Fagan *et al.* (Fagan and Swindells, 2000).

The leads identified are further refined to comply with pharmacokinetic constraints such as absorption and bioavailability, and to increase their potency and efficacy, while decreasing side effects and toxicity. This step is called *lead optimization*. Knowledge of the mode of action (MOA), that is, the identification of the therapeutic molecular target of the drug, can simplify the task of optimizing the drug candidate. Understanding the MOA can help predicting the effect of drug interactions and allow structure-activity relationships (SAR) to guide medicinal chemistry efforts toward optimization (Hart, 2005). However, for many drugs, the targets are unknown and difficult to find among the thousands of gene products in a typical genome.

Many new compounds fail when they are tested in humans due to lack of efficacy. Testing for efficacy early during the drug discovery process (*i.e.* before the clinical phases) is essential for reducing costs and time required. Therefore, the development of experimental and computational approaches to test for efficacy in vitro is critical.

After the preclinical phases, a candidate compound is then selected and the clinical phase on the process can begin. This consists of clinical phase I, phase II, and phase III and possibly the launch into the market. Many compounds fail in the

clinical phases of the process thus leading to consistent waste of time and money. A good review of the evolution of the drug discovery process can be found in Ratti *et al.* (Ratti and Trist, 2001).

Computational biology and bioinformatics have the potential not only of speeding up the drug discovery process thus reducing the costs, but also of changing the way drugs are designed. In this review we will focus on the different computational and bioinformatics approaches that have been proposed and applied to the different steps involved in drug development as shown in Table 3-1. Our aim is to describe the different computational methods that have been used so far to tackle these problems by giving examples of applications. Since we cannot be comprehensive in our review, we tried to compensate for this by referring the interested readers to other reviews with a different focus that have been written on this subject. The organization of this paper is based on classifying drug discovery approaches into two major categories. Section 3.3 reviews Classifier-based algorithms which try to determine drug specific patterns as biomarkers of a compound activity, while section 3.4 assesses more complex methods that attempt to infer the network of gene-gene interactions that are perturbed by a drug. We further subdivided those sections in subsections, each focusing on specific steps of the drug discovery process.

3.3 Classifier-based algorithms

A classifier is an algorithm that uses a set of input or predictor variables $x = (x_1, x_2, ..., x_n)$ to predict one or more response variables $y = (y_1, y_2, ..., y_m)$

(Figure 3-1). For example x can be a set of measurements of the expression of n genes in response to a drug treatment in a tumor cell type and y can represent the efficacy of the drug for that tumor cell type. Classifiers can be further subdivided in supervised-learning methods and unsupervised-learning methods. In supervised-learning a training set of 'solved cases' is used to train a model to recognize what will be the response y given the input variables x. Supervised-learning methods may be thought of as a "learning with a teacher model" in which a student gives an answer $\hat{y}$ to each question x in the training set, and the teacher provides the correct answer y. After the training, the student should be able to give the correct answer to a new question that was not in the training set. If y and $\hat{y}$ are coded as numerical values, we can define a loss function $L(y,\hat{y})$, for example, $L(y,\hat{y}) = (y - \hat{y}(\theta))^2$, where θ are the parameters of the model to be learned. By minimizing this function over θ on the training set, one finds the values of the model parameters θ. For example, Linear Discriminant Analysis (LDA) is a supervised learning where $\hat{y} = \theta x$.

In unsupervised-learning, or "learning without a teacher", one has a set of n observations $(x_1, x_2, ..., x_n)$ without the correct response variables. Cluster analysis is an example of unsupervised-learning method whose goal is to group a collection of objects into subsets or "clusters", such that the objects within each cluster are more closely related to one another than those assigned to different clusters. In addition the goal can also be to arrange the clusters in a natural hierarchy. A commonly used hierarchical clustering is the one described by Eisen (Eisen et al., 1998). Unsupervised methods have the advantage that they are 'data

driven' and do not rely on *a priori* knowledge. A comprehensive and detailed description of these methods can be found in the excellent book by Hastie *et al.* (Hastie et al., 2001).

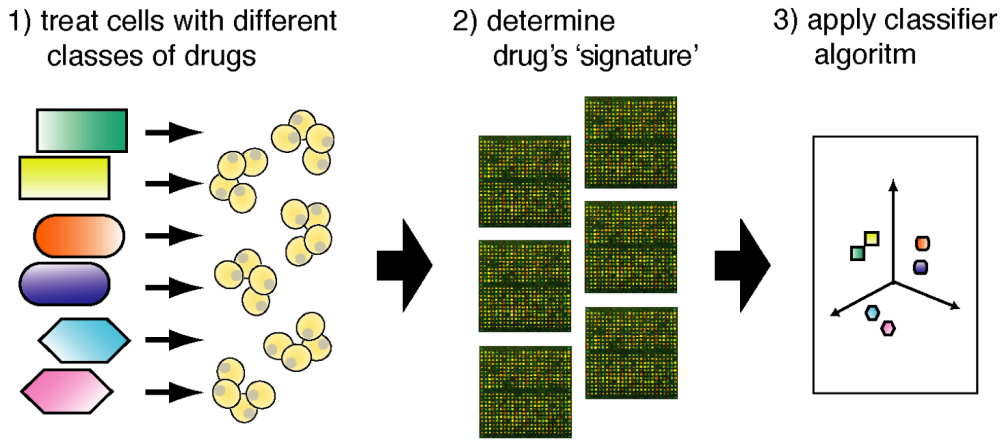


Figure 3-1: Schematic diagram of the classifier-based algorithms.

3.4 Target identification and validation

Whole-genome gene expression data, proteomic data or metabolomic data, also named "molecular profiling" in a recent review (Stoughton and Friend, 2005), can be used to build classifier algorithms able to help in the process of identifying 'druggable' gene/protein/metabolites targets.

An example of an unsupervised-learning method can be found in Hughes *et al.* (Hughes et al., 2000). These authors constructed a reference database of whole-genome expression profiles referred to as a gene expression "compendium" generated by 300 diverse mutations and chemical treatments in *Saccharomyces cerevisiae*. A 2D hierarchical clustering (Eisen et al., 1998; Hartigan, 1975) was

used to cluster genes and experiments using as the similarity measure the correlation coefficient. Genes and experiments were reordered according to the resulting clustering similarity trees. By examining the clusters the authors were able to find an unknown ORFs that clustered among genes involved in the ergosterol biosynthesis and experiments that were perturbing this pathway, thus deducing these ORFs to belong to this pathway. They then experimentally confirmed that 8 of these ORFs were indeed required for sterol metabolism. Since sterol metabolism is a 'druggable' pathway in yeast for antimycotic drugs, this work shows how novel targets can be identified via bioinformatics approaches. A similar method has been applied by Gasch *et al*. (Gasch et al., 2000) that performed a hierarchical clustering of 142 whole-genome arrays in *S. cerevisiae* in response to environmental changes and were able to clarify the regulation mechanisms in which three transcriptions factors were involved.

An example of supervised learning for understanding the function of gene from gene expression data is given in Brown *et al.* 2000 (Brown et al., 2000), in which Support Vector Machines (SVMs) (Hastie et al., 2001) are used. When applied to gene expression data, an SVM begins with a set of genes that have a common function: for example, genes coding for ribosomal proteins or genes coding for components of the proteasome. In addition, a separate set of genes that are known not to be members of the functional class is specified. These two sets of genes are combined to form a set of training examples in which the genes are labeled positively if they are in the functional class and are labeled negatively if they are known not to be in the functional class. By analyzing expression data from 2,467 genes from the budding yeast *S. cerevisiae* measured in 79 different DNA

microarray hybridization experiments the authors were able to correctly assign genes to five functional classes from the Munich Information Center for Protein Sequences Yeast Genome Database (MYGD). The method is compared with hierarchical clustering and shown to be marginally better, but this could have been expected since supervised learning methods have access to additional information as provided by the training set.

An original high throughput drug screening strategy based on unsupervised-learning is used by Segmaier *et al.* (Stegmaier et al., 2004), which, unlike most commonly used methods, does not simply screen for compounds that interact with specific molecular targets. The authors preliminarily define a gene expression signature for the target post-treatment phenotype, or 'cellular state' of interest. Specifically, in this study the target cellular state was differentiated neutrophils and monocytes from control individuals vs. pretreatment bone marrow samples derived from Acute Myelogenous Leukemia (AML) patients. A "handful" of marker genes were selected, unfortunately not in a generalized manner but rather arbitrarily, from the differentiation-correlated genes. Those markers were then used to develop a detection assay called Gene Expression-based High-Throughput Screening (GE-HTS) based on multiplexed RT-PCR and Single Base Extension (SBE) reaction followed by MALDI-TOF mass spectrometry. Eight target compounds identified by GE-HTS in this study were validated in several ways including morphological observations and functional measures. Interestingly, the broader cellular genetic program of differentiation beyond the selected handful of marker genes was also investigated, again through a correlation-based statistical test. The authors analyzed triplicate microarray expression data from HL-60 cell

lines treated with eight different compounds. Six out of the eight expression profiles were found statistically significantly similar to the gene expression differences characterizing the original AML-*vs.*-controls primary cells, as determined by the Mantel test (Stegmaier et al., 2004). This test is an unbiased, global measure of similarity, and indicates that the six compounds induced a nonrandom pattern of gene expression consistent with differentiation. The advantage of GE-HTS is that the development of the assay does not require any specialized assays such as traditional methods based on antibodies or reporter constructs or cellular phenotypes, and, once the gene expression signature pattern is defined, the procedure is rather straightforward.

Although they may result in outstanding accuracy performance, correlation-based methods do not easily provide insight into the mechanisms of action common to the therapeutic category, but rather capture silent features of drug efficacy by their correlation to biomarker signatures based on gene expression patterns.

3.5 Hit identification, Lead identification and optimization: Mode of Action (MOA)

One of the first bioinformatics approaches to determine mode of action of a compound was based on a simple supervised-learning approach (Paull et al., 1989). In 1985 the National Cancer Institute (USA) established a primary screen in which compounds were tested in vitro for their ability to inhibit growth of 60 different human cancer cell lines (Weinstein et al., 1997). To each compound

tested it is possible to associate a value quantifying the differential growth inhibition (GI) for each cell line (treated *vs.* untreated). The algorithm developed by Allen and coworkers, named COMPARE, measures the similarity of the GI "signature" of a novel compound against a database of "signatures" of compounds with known MOA. The similarity is obtained simply by computing the average differences between the signatures of the test compound and each of the signatures in the database. Ranking according to this measure of similarity, one can infer the MOA of the novel compound as the one of the most similar compound in the database. An extension of this approach based on hierarchical clustering and integration of different data set from the NCI 60 cell lines has been proposed by Weinstein in 1997 (Weinstein et al., 1997). A more sophisticated approach using SVM to classify drugs into 5 mechanistic classes using drug activity profiles and the gene expression profiles of each of the untreated NCI 60 cell lines, has been proposed by Bao *et al.* (Bao et al., 2002).

Unsupervised approaches have been applied extensively in this area. Marton *et al.* (Marton et al., 1998) were pioneers of the "signature approach" based on gene expression profile following drug treatment. In this approach the drug signature is compared to a mutant strain signature using a correlation coefficient as a measure of similarity, $p = \frac{\sum x_k y_k}{\sqrt{\sum x_k^2 \sum y_k^2}}$. They also proposed a further 'decoder' step where the mutant strains whose expression profiles were most similar to the drug-treated cells are treated with the drug, generating an expression signature in the mutant strain. If the mutated gene encodes a protein involved in the pathway affected by the drug, then the signature in mutant cell should be different or,

ideally, absent. Marton *et al.* did a proof of principle study on FK506 and the calcineurin signaling pathway as a model system.

The previously described work by Hughes *et al.* (Hughes et al., 2000), is another good example of how hierarchical clustering and correlation can be used for understanding the MOA of a drug. The authors used the gene expression 'compendium' to identify the target of the commonly used topical anesthetic dyclonine. In order to find the target of the compound, the authors treated the yeast cells with the compound and compared the gene expression profile to the most similar expression profiles in the compendium using the correlation coefficient as the similarity measure. The erg2Δ strain (knock-out of the erg2 gene) was most similar to the dyclonine treatment thus suggesting, correctly as verified experimentally, that this gene is the molecular target of the drug. Since this gene is conserved in human but codes for the sigma receptor, a neurosteroid-interacting protein, the MOA of the drug in human has also been explained.

Hierarchical clustering methods have been applied not only to gene expression data, but also to chemical-genetic and genetic interaction data. Parsons *et al.* (Parsons et al., 2004) screened ~4700 yeast deletion mutants for hypersensitivity to 12 diverse inhibitory compounds. Hypersensitivity was measured from digital images of plates by quantifying colony area growing in drug-medium versus no-drug control medium. Hypersensitive strains for a given drug were coded as 1, and with a 0 otherwise. These data (a vector of ~4700 0s and 1s for each drug) were used for 2D hierarchical clustering. Both genes and compounds are clustered together upon the similarity of their chemical-genetic interactions. By analyzing

the clusters they were able to detect genes whose deletion was associated with sensitivity to multiple compounds, thus enabling them to identify a multidrug-resistant gene set. To identify the mode of action of a compound, they performed synthetic lethal screens between ERG11 mutants and the ~4700 deletion strains. The overlap between the genes that were synthetic lethal with ERG11 mutants, with the genes whose deletions were lethal after treatment with flucanozole, was used to infer the MOA of this drug.

Related to these methods are drug-induced haploinsufficiency screens first proposed by Giaever *et al.* (Giaever et al., 1999). Drug-induced haploinsufficiency occurs when lowering the dosage of a single gene from two copies to one copy in diploid cells results in a heterozygote that displays increased sensitivity to the drug as compared to the wild-type strain. These screens make use of a fitness defect score (Giaever et al., 2004) that is computed using different methods (Baetz et al., 2004; Lum et al., 2004).

Hierarchical clustering has been applied also to data derived from automated microscopy in order to identify drug MOA. Perlman *et al.* (Perlman et al., 2004) chose 200 compounds, 90 of which were drugs with known MOA. They cultured HeLa (human cancer) cells in 384-well plates to near confluence, and treated them with 13 threefold dilutions of each drug for 20 hour, covering a final concentration range from micromolar to picomolar. They chose 11 distinct fluorescent probes covering a range of biological processes. Using automated fluorescence microscopy they measured for each cell, region and probe, a set of descriptors including size, shape, intensity, as well as ratios of intensities between

regions for a total of 93 descriptors. For each descriptor they developed a titration-invariant similarity score (TISS) to allow comparison between dose-response profiles independent of starting dose. TISS scores for 61 compounds were computed and used for hierarchical clustering; the data matrix used for clustering consisted of 61 compounds by 93 TISS scores. Once again they found that drug with similar mechanism of action clustered together, thus allowing inference of drug MOA for drugs with unknown molecular targets.

Signature Expression profiles were used by Betts (Baetz et al., 2004) to determine the differential mode of action of three active drugs against *Mycobacterium tuberculosis*, and as a means of identifying novel and efficacy-optimized active drugs. In this study the authors show that although global response profiles of isoniazide and thiolactomycine are more closely related to each other than to that of triclosan, there are differences that distinguish the mode of action of these two drugs. A mathematical model is proposed to discriminate between the three compounds and also the vehicle control treatment. The main sources of variance of the data were obtained by Principal Component Analysis (PCA). The principal components are a linear combination of all the gene intensities. Partial least squares discriminant analysis was performed on a subset of data selecting the dose and the time point that maximized separation of experimental groups. The 500-top ranking genes thus identified, were further processed by stepwise linear discriminant analysis in order to generate a mathematical model for the probability $P_i(x)$ of a gene expression signature x belonging to classification group i based on the following discriminant function:

$$P_i(x) = \frac{e^{D_i^2(x)}}{\sum_{j=1}^{n} e^{D_j^2(x)}} \qquad i = 1,2,...,n \tag{1}$$

where $D_i^2(x)$ is the discriminant score of the signature x for group i.

Methods that rely on a dataset for the construction of a classifier model, without implementing more robust statistical analyses, such as running a series of training and testing data in a 'leave-one-out' manner, although accurately performing on the training dataset may lead to the construction of a model that 'overfits' the data, and thus may not perform well on new data obtained using different treatments.

3.6 Hit identification, Lead identification and optimization: Efficacy and Toxicity

A large part of the efforts based on computational and bioinformatics approaches have been directed to predict sensitivity of cancer cell lines to different compounds. Scherf *et al.* (Scherf et al., 2000) aimed at relating sensitivity to therapy with gene expression using an unsupervised approach. They used the database of drug activity profiles (Growth Inhibition after 48h of drug treatment) of more than 70,000 compounds on NCI 60 cell lines, together with gene expression profiles of 9,703 genes measured using cDNA microarrays for each of the 60 untreated cell lines. They then performed a hierarchical clustering of 118 compounds with known mechanism of action. In order to integrate drug activity profile with gene expression data, they chose Pearson correlation coefficient as a

measure of similarity. This coefficient was calculated for each combination of a gene (expression profile across 60 cell lines) and a drug (GI activity profile across 60 cell lines). This yielded 1376 correlation coefficients for each of the 118 drugs. Using this technique they were able to associate sensitivity of leukemic lines to L-asparagine to the amount of asparagine synthetase. A similar technique has been proposed by Dan *et al.* (Dan et al., 2002). They were able to identify gene markers for chemosensitivity for 55 anticancer drugs using gene expression data across 39 cell lines and drug activity profiles (GI). Similarly, Szakacs (Szakacs et al., 2004) and co-workers correlated expression profiles of all 48 human ABC transporters with patterns of drug activity in the NCI 60 cell lines. They were able to identify candidate substrates for several ABC transporters and compounds whose toxicities are potentiated by ABCB1-MDR1.

One potential application of microarrays in toxicology is their use in predicting toxicity of undefined chemicals by comparing their gene expression patterns in a biological model with databases of microarray-generated gene expression data corresponding to known toxicants. Feasibility of compound classification based on gene expression profiles is proven by several experiments. Hammadeh et al. (Hamadeh et al., 2002a; Hamadeh et al., 2002b), for example, analyzed rat liver gene expression patterns elicited by peroxisome proliferators, and enzyme inducers. These authors used several computational analyses including hierachical cusltering (Eisen et al., 1998), PCA, pairwise Pearson correlation of gene expression profiles, and finally a combination of a genetic algorithm and K-nearest neighbor (GA/KNN) (Li et al., 2001). Their results confirm that compound classification based on gene expression is feasible, and showed both

strong within-class correlation of expression profiles and between-class highly distinguishable patterns.

The work of Staunton *et al.* (Staunton et al., 2001) is an example of a supervised-learning approach. Specifically, the authors investigated whether patterns of gene expression were sufficient to predict sensitivity or resistance of the NCI 60 cell lines to 232 chemical compounds whose GI activity profile had been previously measured. They measure gene expression of 6817 genes in each of the 60 untreated cell lines using Affymetrix chips. Chemosensitivity prediction was modeled as a binary classification problem, and thus for each compound two classes of cell lines were defined: sensitive (class 1) and resistant (class 2), according to the GI profiles. They then divided the data set into a training set and a test set. The classifier was implemented using a weighted voting algorithm, in which correlated genes "vote" on whether a cell is predicted to be sensitive or resistant. Correlation in the training set between a compound c and a gene g is defined as:

$$P(g,c) = \frac{\mu_1(g) - \mu_2(g)}{\sigma_1(d) + \sigma_2(d)} \tag{2}$$

Large values of the correlation $P(g,c)$ indicate that the gene expression is a good indicator of class distinction. A weighted sum of the gene expression level of strongly correlated genes is then used to classify. Classifiers with up to 200 genes were tested, with the median accuracy of the classifiers reaching 75%. From this work one can conclude that indeed gene expression profiles in untreated cells can be used to predict whether a cell line is sensitive or resistant to a particular drug.

Other interesting examples of supervised classification methods applied to drug-treated human neural cell cultures come from two studies of Gunther and colleagues. The aim of first study (Gunther et al., 2003) was to investigate whether high content statistical categorization of drug-induced gene expression profiles can be used to predict the drug's therapeutical class among different classes of psychoactive compounds. Primary cultures of human neuronal precursor cells were treated with multiple members of antidepressants (AD), antipsychotics (AP), and opioid receptor agonists (OP). Arguably, however, one of the most used class of psychoactive drugs, the class of antianxiety compounds, would have been an interesting choice. Gene expression was measured using DNA microarrays containing about 11k oligonucleotide probes. Data was analyzed by supervised statistical classification including Classification Tree (CT) and Random Forest (RF) methods. Both methods are based on a "leave-one-out" training and testing series, so that the class of the naive test sample can be predicted after training over all other samples. The former method resulted in 88.9% of correct predictions, and relied on few strong markers. Notably, accuracy did not decline significantly when the classification was repeated after withholding the predominant classifier genes from the analysis. The latter method, is based on stochastic feature evaluation, and resulted in a correct prediction rate of 83.3% based on a much larger set (326) of week marker genes. Interestingly, two examples are given in which one subclass of AD (SSRIs, or tricyclic) could be successfully predicted as belonging to the antidepressants class after being excluded from the training using the RF. Although the accuracy of prediction of novel subclass unrepresented in the training was surprisingly high (100%), it is

unclear why a similar analysis after withholding the third subclass of AD adopted in this study, the MAOIs, is missing. The authors of this work recently published a new study (Gunther et al., 2005) in which they propose a novel algorithm for drug efficacy-profiling, called Sampling Over Gene Space (SOGS), and applied it to drug-treated human cortical neuron 1A cell line. While less appealing from a physiological point of view, cell line monocultures provide a simpler system more suitable for reproducible chemical genomics screening. This procedure is based on supervised classification methods such as Linear Discriminant Analysis (LDA) and Support Vector Machines (SVM), expected to yield stronger predictions than stochastic feature evaluation such as RF on one hand, but on the other they are more prone to 'overfitting' the training data. SOGS however builds multiple classifier methods iteratively sampling random sets of features using LDA or SVM, and the final classification is based on the most frequent classification over the multiple iterations. The authors claim that such a combination of stochastic feature evaluation with the stable LDA and SVM modeling methods minimizes overfit, while increasing prediction strength.

3.7 Network/Pathway reconstruction

Perturbations to the state of the cell have been used extensively in molecular biology to infer the function of a single gene or protein. With the advent of high throughput quantitative methods it has become possible to move from a qualitative biology to a quantitative biology, thus enabling the use of methodologies typical of engineering and physics to the study of the biological

processes and the emergence of "systems biology", *i.e.* the integrated study of biological processes [for a good review of systems biology refer to Brent (Brent, 2004) and for its application in drug discovery refer to Butcher *et al.* (Butcher et al., 2004) and Apic *et al.* (Apic et al., 2005)]. Biological processes are the result of complex interaction among thousands of components. Network, or, graph theory, is a mathematical formalism that is very well suited for describing such interactions. Hence the renewed interest in network theory and its potential impact on molecular biology and medicine.

In the area of drug development, particular relevance assumes "reverse engineering" whose goal is to map gene, protein and metabolite interactions in the cell, thus elucidating the regulatory circuits used by the cell for its functioning, and their malfunctioning during diseases. A very good review was recently published on this topic (Gardner and Faith, 2005).

We can distinguish two different reverse engineering strategies (Gardner and Faith, 2005): the "physical approach" and the "influence approach". In the former, the aim is to use RNA expression data to identify the transcription factors (TFs) and the DNA binding sites to which the factor binds. The interactions thus inferred are true physical interactions between TFs and the promoters of the regulated genes. In the latter, the aim is to find regulatory influences between RNA transcripts that do not necessarily have to be of the TF-DNA binding site kind. The general model, as shown in Figure 3-2, requires that some RNA transcripts act as regulatory "inputs" whose concentration variations drive the expression of an "output" transcript. Such a model therefore does not describe

physical interactions, since an mRNA does not control directly the level of other mRNAs, but rather aims at inferring the regulatory influence between two or more transcripts that may as well be indirect through the action of proteins, metabolites and other molecules.

Reverse engineering algorithms make use of measurements of transcript concentrations in response to perturbations to the state of the cell in order to infer regulatory interactions.

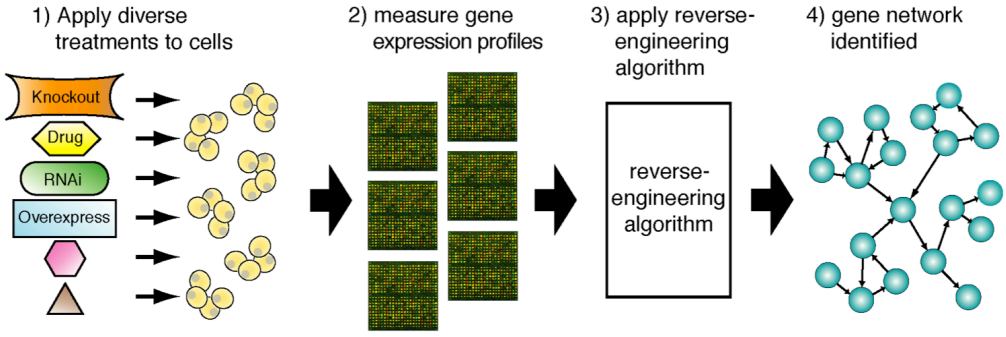


Figure 3-2: Schematic diagram of reverse-engineering approaches to drug discovery. Gene expression profiles following a variety of perturbations to the cells are used to reconstruct the network of interactions of gene, proteins and metabolites.

3.8 Target identification and Validation

For a detailed description on computational and bioinformatics methods to infer interactions among genes and proteins we refer the interested reader to an excellent review on this topic by Gardner et al (Gardner and Faith, 2005). Here we

will briefly discuss two recent examples based on two different methodologies that use the "influence strategy" as defined above.

The first methodology describes a gene network as a system of ordinary differential equations (De Jong et al., 2004). The rate of change in concentration of a particular transcript, x_i, is given by a nonlinear influence function, f_i, of the concentrations of other RNAs:

$$\frac{dx_i}{dt} = f_i(x_i,...,x_n) \tag{3}$$

Where n is the number of genes or transcripts in the network. The function f_i can have different forms. The easiest form that this function can assume is the linear form where Equation (3) becomes:

$$\frac{dx_i}{dt} = \sum_j w_{ij} x_i + p_i \tag{4}$$

where w_{ij} represents the influence of gene j on gene i, and p_i an externally applied perturbation to the level of transcript i. We developed an inference algorithm named Network Identification by Regression (NIR) (Gardner et al., 2003) that uses the differential equation model of a gene network in Equation (4) to infer the regulatory interactions among 9 genes part of the *Escherichia coli* SOS pathway. The strategy we adopted was to overexpress each of the 9 genes in the network using an exogenous plasmid carrying a copy of the gene under the control of an inducible promoter. After transfection and induction of the vector, the gene expression change of the 9 genes in the network was measured at steady-state, *i.e.* when the cell has reached a new equilibrium and all the transient effects

are over. Under these conditions, the term on the left hand-side of Equation (4) becomes $\frac{dx_i}{dt} = 0$, so that the equation can be rewritten as:

$$-p_i = \sum_j w_{ij} x_j \tag{5}$$

where both p_i and x_j for all the 9 different perturbation experiments are experimentally measured, whereas the weights w_{ij} are the unknown parameters that we would like to learn from the data. Using multiple linear ridge regression, we were able to recover a network model, shown in Figure 3-3, that correctly identified 25 of the previously known regulatory interactions between the 9 transcripts, as well as 14 interactions that could be novel, or possibly false positives. These results were obtained with a noise-to-signal ration of 68%. From a drug discovery point of view, this approach would be powerful for finding new targets for antibiotics, since the 9 genes are part of the SOS pathway involved in response to DNA damage. The genes that are the 'hubs' of the network, *i.e.* those genes that are the main regulators of the system, are ideal targets for new antibiotics because they would block the response of the bacteria to damage, thus preventing their survival.

As a second example of successful network inference applied to a mammalian system, we will illustrate the work of Basso *et al.* (Basso et al., 2005). The approach used by these authors is based on information theory. Their approach named ARACNE is based on the computation of mutual information among pair of genes. For a pair of discrete random variables, x and y, the mutual information is defined as

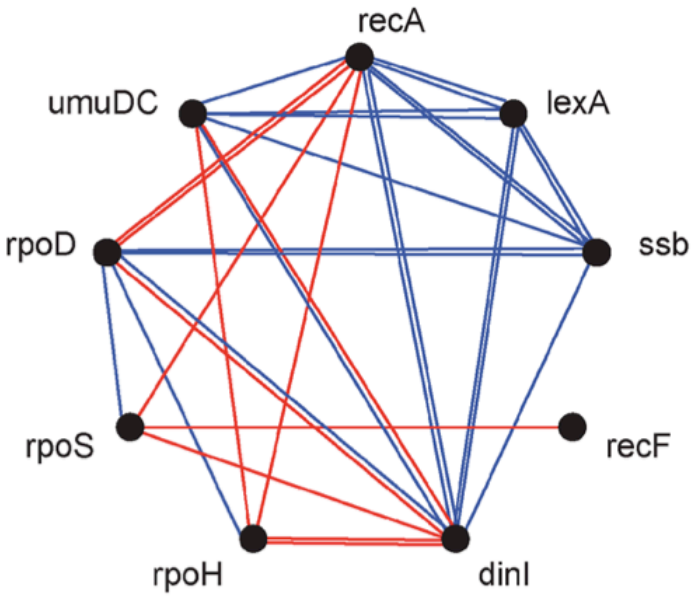
$$I(x,y) = S(x) + S(y) - S(x,y) \tag{7}$$

where $S(.)$ defines the entropy. For a given discrete stochastic variable t the entropy is defined as:

$$S(t) = \sum_i \Pr(t = t_i)\log(\Pr(t = t_i)) \tag{8}$$

As it can be intuitively appreciated from the above definition entropy is maximal for a uniformly distributed variable. The probability is estimated using Montecarlo simulations. To each value of the mutual information $I(x,y)$ is associated a *p*-value computed again using Montecarlo simulations. The null hypothesis associated to the *p*-value corresponds to pair of nodes that are disconnected from the network and from each other. The final step of their algorithm is a pruning step that tries to reduce the number of false positives (*i.e.* inferred interactions among two genes that are not direct interaction in the real biological pathway). They use Data Processing Inequality principle that asserts that if both (x,y) and (y,z) are directly interacting, and (x,z) are indirectly interacting through y, then $I(x,z) <= I(x,y)$ and $I(y,z) <= I(x,y)$. This condition is necessary but not sufficient, *i.e.* the inequality can be satisfied even if (x,z) are directly interacting, therefore the authors acknowledge that by applying this pruning step using DPI they may be discarding some direct interactions as well. The authors applied their algorithm on a data set consisting of 336 whole-genome expression profiles representative of perturbations of B cell lines and are able to find novel direct targets of the Transcription Factor MYC.

(a) Network inferred by NIR method



(b) Network model: connection weight matrix

	recA	lexA	ssb	recF	dinI	umuD	rpoD	rpoH	rpoS
recA	0.40	-0.18	-0.01	0	0.10	0	-0.01	0	0
lexA	0.39	-0.67	-0.01	0	0.09	-0.07	0	0	0
ssb	0.04	-1.19	-0.28	0	0.05	0	0.03	0	0
recF	0	0	0	0	0	0	0	0	0
dinI	0.28	0	0	0	-1.09	0.16	-0.04	0.01	0
umuDC	0.11	-0.40	-0.02	0	0.20	-0.15	0	0	0
rpoD	-0.17	0	-0.02	0	0.03	0	-0.51	0.02	0
rpoH	0.10	0	0	0	0.01	-0.03	0	0.52	0
rpoS	0.22	0	0	-1.68	0.67	0	0.08	0	-2.92

Figure 3-3: Inference of a nine-transcript subnetwork of the SOS pathway in E. coli using the NIR algorithm. (a) Graph depiction of the network model identified by the NIR algorithm. Previously known regulatory influences are marked in blue, novel influences (or false positives) are marked in red. The strengths and directions of the identified connections are not labeled in the graph. (b) The network model is also depicted as a matrix of interaction strengths. The colors are the same as in panel (a).

3.9 Hit identification, Lead identification and optimization

Network identification can be used to infer the direct gene and protein targets of a compound with unknown mode of action. One of the earliest approaches of this kind has been proposed by Imoto *et al.* (Imoto et al., 2003). Although the approach described in the paper is somewhat confusing, we decided to include it in our review since to our knowledge this is one of the first papers to propose that network inference can be used for lead optimization. The authors termed their approach the "virtual gene technique". Briefly, using an algorithm by Maki *et al.* (Maki et al., 2001) they reconstruct a directed acyclic graph (DAG) describing gene regulatory interactions considering the drug as a "virtual gene". Let $V = \{g_1, g_2, ..., g_n\}$ the set of all genes and $D = \{d_1, d_2, ..., d_m\} \subseteq V$ the set of genes to be knocked out in order to perturb the system. D is assumed to contain also the virtual gene and the perturbation experiment associated to this virtual gene is treatment with the drug. By observing how the genes change in response to the gene disruption they are able to find a DAG by drawing an edge between two nodes of the graph if a certain equivalence relationship is satisfied. By considering the DAG whose root is the virtual gene, the children of this virtual gene would be the candidate genes directly affected by the drug. From their paper is not clear how well their method performs since the experimental results on deletion strains of *S. cerevisiae* are poorly described. However, their method is an illustrative example of how network inference can be applied to drug discovery.

Another example of network inference to drug discovery is the work of Haggarty et al (Haggarty et al., 2003). Their approach is based on wildtype and nine

different gene deletion strains in *S. cerevisiae*. Each of the strains is treated with all the possible combinations of 2 molecules drawn from a set of 24 small molecules. The authors propose a method that can be used to understand which of the molecules have similar mode of action by measuring the similarity of chemo-genomic networks. For each strain the data were represented as an adjacency matrix, A, with one row and one column for each of the 24 molecules tested. The element a_{ij} of matrix A is 0 when no observable effect on growth after treatment with compound i and j is found, 1 if there is a measurable growth defect. For each compound in each strain, information in A can be used for clustering the compounds on the basis of similarity in their pattern of biological activity. However the authors do not test thoroughly this prediction.

The NIR algorithm we developed and briefly described in section 3.8, can also be used for compound mode of action discovery. The network model can be used as a predictive tool for analyzing new RNA expression data obtained by measuring transcript responses to a drug treatment. As a proof-of-principle, we applied the antibiotic mitomycin C to *E. coli*, we observed the changes in all nine measured SOS transcripts. However the known mediator of mytomicin C is only the gene recA. The network model obtained by the NIR algorithm enables us to separate secondary changes from primary changes due to direct interaction with the drug. In this case Equation (5) can be solved to find the p_i value for each $i = 1...9$, since the network model w_{ij} is known while x_j are the measured response of the cell to the drug treatment. If p_i is close to 0, then gene i is not a direct target of the drug, otherwise gene i is directly interacting with the compound.

The network model correctly filters the RNA expression response to the drug treatment to reveal the recA gene as the direct target. The same target was identified for treatment with UV irradiation and the antibiotic pefloxacin, both of which stimulate recA transcript, but not for novobiocin, a drug that should not directly interact via the recA gene.

We recently proposed an extension of the NIR algorithm called Mode of action by Network Identification (MNI), that computes the likelihood that gene products and associated pathways are targets of a compound (Di Bernardo et al., 2005). Our approach is described in Figure 3-4. We first reverse-engineer a network model of regulatory interactions in the organism of interest using a training data set of whole-genome expression profiles. The network model is based on ordinary linear differential equations under steady-state conditions described by Equation (5). We then use the model to analyze the expression profile of the compound-treated cells to determine the pathways and genes targeted by the compound. The algorithm assumes that the expression profile training data set are obtained at steady-state following a variety of treatment, including compounds, RNAi, and gene-specific mutations (Figure 3-4).

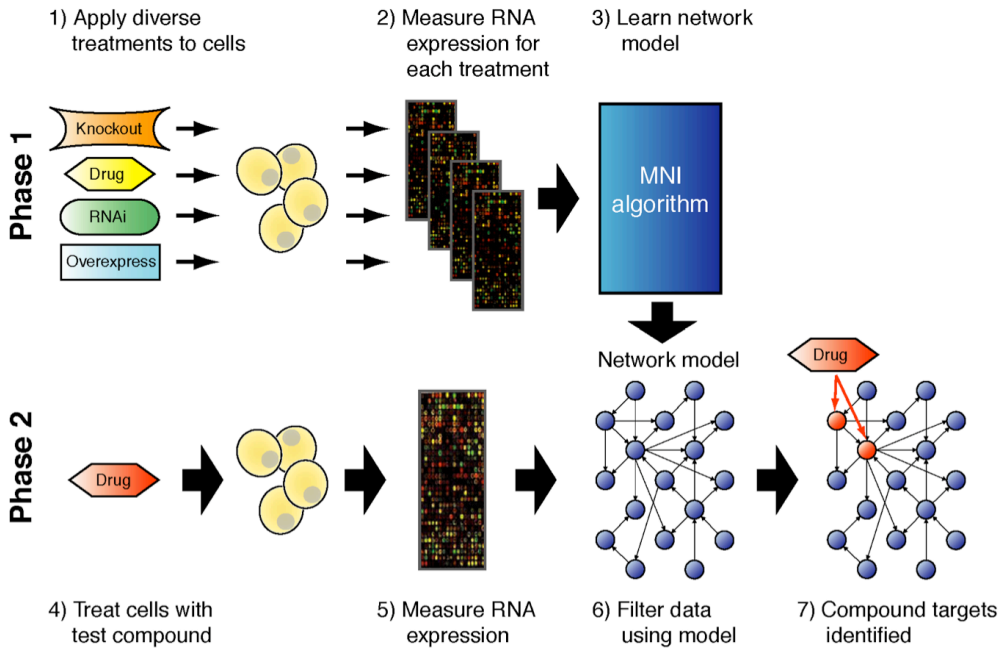


Figure 3-4: Overview of the NMI method. In phase 1, a set of treatments is applied to cells. Changes in mRNA species are measured. The data are then used by the MNI algorithm of infer a model of the regulatory network among the genes. In phase 2, cells are treated with the test compounds and the expression changes of all the mRNA species is measured. The expression data are then filtered using the network model to distinguish the targets of the test compound from secondary responders.

The ability to use different treatment types is an important advance over earlier model estimation techniques that require knowledge of the targets of the perturbations. To infer a network model without requiring gene-specific perturbations the algorithm employs an iterative procedure. It first predicts the targets of treatment using an assumed network model, and then uses those predicted targets to estimate a better model. The procedure stops once convergence criteria are met. Once the regulatory model has been learned, we

applied it to the expression profile of a test compound to predict its targets. We applied this method to the *S. cerevisiae* using as a training data set 515 whole-genome yeast expression profiles resulting from a variety of treatment (Hughes et al., 2000; Mnaimneh et al., 2004). We then used MNI algorithm to identify the probable targets of 15 compounds, 13 of which were drawn from the Hughes compendium (Hughes et al., 2000) and from other studies (Ueda et al., 2003). Of these 15 compounds, 9 had previously known targets, while the targets of other six were previously unknown. MNI ranks the ~6000 genes in yeast according to their probability of being direct targets of the compound. By selected the top 50 genes predicted by MNI for a compound, it is possible to infer the pathways directly affected by the drug looking for significantly overrepresented Gene Ontology (GO) processes among the highly ranked genes.

For 7 out of 9 compounds with known mode of action, MNI correctly identified the known target pathway and for 6 out of this 9 it was able also to identify the correct target gene. We then demonstrated the use of MNI on a tetrazole-containing compound, 1-phenyl-1H-tetrazole-5-ylsulfonyl-butanenitrile (PTSB) found to inhibit both wt *S. cerevisiae* and human small lung carcinoma cells. We applied MNI to the expression profile after treatment with PTSB and found two genes thioredoxin reductase (TRR1, MNI_rank=32) and thioredoxin (TRX2, MNI_rank=36) while the overrepresented GO process among the top 50 genes was the 'cell redox homeostasis'. We validated the prediction made by MNI with appropriate biochemical assays and confirmed that PTSB acts by inhibiting these two targets.

3.10 Conclusions

Computational biology and bioinformatics approaches have the potential to completely change the way drugs are discovered and designed. Already these methods are having an impact on the different stages of the drug discovery process. We have shown in this review how computational methods like classification and network-based algorithms can be used to understand the mode of action and the efficacy of a given compound and to help elucidating the pathophysiology of a disease. But these computational tools, in our opinion, may also be used in a different and innovative way to promote a change of paradigm in how drugs are designed. In the pharmacological industry there has already been a shift from symptomatic oriented drugs, that can relieve the symptoms but not the cause of the disease, to pathology-based drugs whose targets are the genes and proteins involved in the etiology of the disease. Drugs targeting the affected pathway have thus the potential to become therapeutic. An example of this is enzyme replacement therapy in genetic sulfatase deficiency syndromes (Cosma et al., 2003). The sulfatase enzyme is the missing protein that when reintroduced in the organism is able to restore the pathway that had been altered by the disease process. The passage from symptomatic-centered drug discovery to disease-centered drug-discovery has been forced upon the industry by the availability of the full sequence of the human genome, with its implicit promise of novel potential targets. However, as reported in a recent review by Csermely *et al.* (Csermely et al., 2005), the number of successful drugs did not increase appreciably in the recent years. With the current paradigm, an ideal drug is both potent and specific, *i.e.* it targets specifically a single protein. In our opinion a

second shift is now necessary and will be driven by the availability of sophisticated computational biology and bioinformatics tools: a shift from single-target drugs to "network drugs". By network drug we define a compound or a set of compounds that is able to alter a biological pathway disregulated by a disease in a predefined way so as to restore its normal physiological function. A similar concept has been put forward by Csermely *et al.* (Csermely et al., 2005) in their review, where they propose the partial inactivation of multiple targets as a novel paradigm for drug design. They argue that such kind of multi-target drug could be much more efficient than a drug directed at a single target. They proposed that a network approach to drug design would examine the effect of drugs in the contest of a network of relevant protein-protein, regulatory and metabolic interactions. The end result would be the development of a drug that would hit multiple targets selected in such a way as to decrease network integrity and so completely disrupt the functioning of the network. Our idea is to take this approach one step further and aim not at disrupting the network, but into developing compounds and delivery techniques able to change the behavior of the network in a controllable and predictable manner.

Thanks to network-inference approaches, some of which were described in this review, it is now becoming possible to have a detailed map of the regulatory circuit among genes, proteins and metabolites. This in turn allows a better understanding of how biological pathways are regulated and how they accomplish their function. The approaches presented in this review also allow the screening of a compound to quickly identify the proteins it interacts with. This gives us all the necessary tools to identify and repair the disregulated biological pathway causing

the disease, much as an engineer would do to restore a malfunctioning electronic circuit. If she/he finds that a specific component of the circuit is malfunctioning, it would be bypassed using extra wires that would bridge different parts of the circuit. Sometimes this would not be sufficient since those parts of the circuit should be in contact only under precisely defined conditions. In this case, she/he would also need to add a microchip that would take care of activating those connections only when necessary.

Similarly one could think of delivering multiple compounds, each directed to a specific biological target, in a coordinated way controlled by a computer chip that would release the drugs in the organism only when needed to restore physiological behavior of the pathway disregulated by the disease. The key step in this approach is to have a detailed knowledge of the network of protein, gene and metabolite interactions in the different biological pathways.

Although this picture may seem farfetched all the tools to accomplish this feat have already been developed and are here to stay, and hopefully in the next decades the way we think of drugs will be completely different.

Drug Discovery		Classifiers	Network/Pathway reconstruction
Target identification & validation		Stoughton *et al.* 2005 (Review) Walker 2001 (Review); Hughes *et al.* 2000; Gasch 2000; Stegmeir 2004; Brown 2000	Gardner at al 2003; Basso *et al.* 2005; Gardner *et al.* 2005 (Review); Apic 2005 (Review)
Hit identification, Lead identification & optimization	Mode of action (MOA)	Perlman *et al.* 2004; Parsons *et al.* 2003; Parsons *et al.* 2004; Marton 1998; Giaever 2004; Giaever 1999; Lum 2005; Hughes *et al.* 2000; Betts 2003; Paull 1989; Weinstein 1997; Bao *et al.* 2002	di Bernardo *et al.* 2005; Imoto 2003 ; Haggarty 2003
	Efficacy & Toxicity	Bugrim *et al.* 2004 (Review), Szakacs 2004; Staunton *et al.* 2001; Scherf 2000, Gunther 2003; Gunther 2005; Hamadeh 2002; Dan *et al.* 2002	Not known

Table 3-1: Classification of the reviewed manuscripts according to the computational methods used and their application to the drug discovery process.

Chapter 4

Quetiapine Experiments

4.1 Abstract

Neuronal expression of immediate-early genes in response to a drug is a powerful screening tool for dissecting anatomical and functional brain circuitry affected by psychoactive compounds. We examined the effect of dopaminergic perturbation on two *Homer 1* gene splice variants, *Homer 1a* and *ania-3* in rat forebrain. Rats were treated with the 'typical' antipsychotic haloperidol, the 'atypical' quetiapine, or the selective dopamine transporter (DAT) inhibitor GBR 12909 in acute and chronic paradigms. Our results show that the high affinity dopamine D_2 receptor

antagonist haloperidol strongly induces *Homer 1* gene expression in the caudate-putamen whereas quetiapine, a fast D2R dissociating antagonist, does not. This confirms that *Homer 1* may be considered a predictor of 'atypicality' of antipsychotic compounds in acute and also chronic regimens. Chronic treatment with GBR 12909 showed a strong induction in the parietal cortex resembling the activation of 'sensitization' circuitry by stimulants. Finally, we describe a differential spatial induction pattern of *Homer 1* gene within the caudate-putamen by typical antipsychotics and DAT blockers, and propose a novel method to quantitate it.

4.2 Introduction

In the present study we investigated the differential effects of several agents that modulate the dopaminergic neurotransmission on *Homer 1a* and *ania-3* gene expression by means of quantitative *in situ* hybridization on rat forebrain slices. The treatments included the typical antipsychotic haloperidol and the atypical quetiapine, as well as the DAT inhibitor GBR 12909, all in acute and chronic regimens. Stimulants such as cocaine and amphetamine are also known to induce immediate early genes, and specifically *Homer 1* (Berke et al., 1998; Yano and Steiner, 2005). Moreover, the expression patterns of stimulant induced IEGs can be directly affected by pretreatment schedules (Curran et al., 1996). Since both antipsychotics and DAT inhibitors, with opposite effects on dopamine transmission, are known to upregulate *Homer 1* gene expression, we attempted to detect any subtle differential response to stimulating *vs.* blocking dopamine

transmission. With the chronic treatment schedule we tried to assess whether there is tolerance phenomena to *Homer 1a* induction as found for IEG induction after repeated stimulant administration (Persico et al., 1993).

4.3 Materials and methods

The procedure for *in situ* hybridization histochemistry was taken from several standard published protocols (Ambesi-Impiombato et al., 2003; Austin et al., 1992; Young et al., 1986). Refer to Chapter 2 for detailed procedure.

4.3.1 Drug treatment and tissue preparation

Quetiapine was chosen based on the observation that this compound, as opposed to haloperidol, binds to the dopamine D_2 receptor (D2R) with fast dissociation dynamics, which is correlated to low propensity of this drug to induce EPSEs (Kapur et al., 2000b). In preclinical studies quetiapine has been shown to have a clozapine-like activity in a wide range of behavioral and biochemical tests, while showing no significant liability for hematological side effects, such as neutropenia (Nemeroff et al., 2002). The selective dopamine transporter blocker GBR 12909, also known as vanoxerine (1-{2-[bis-(4-fluorophenyl)methoxy]ethyl]}-4-(3-phenylpropyl)piperazine), shares the same mechanism of action as cocaine, as it blocks dopamine reuptake, by selectively binding to the Dopamine Transporter (DAT). A DAT inhibitor was included in our experiments because dopamine re-uptake inhibitors have been shown to regulate *Homer 1* gene products in a regionally selective manner (Swanson et al., 2001), and because *Homer 1a* is strongly induced in the striatum by cocaine (Brakeman et al., 1997).

4.3.2 Acute experiment.

On the day of the experiment rats were randomly assigned to one of the following treatment groups: A) 0.9% NaCl (SAL); B) 15 mg/kg quetiapine (QUE15); C) 30 mg/kg quetiapine (QUE30); D) haloperidol 0.8 mg/Kg (HAL); E) GBR 12909 30 mg/kg (GBR). The animals were sacrificed by decapitation 90 minutes after the treatment.

4.3.3 Chronic experiment.

Rats were treated daily for 21 days after their assignment to the following experimental groups: A) 0.9% NaCl (SAL); B) 15 mg/kg quetiapine (QUE) (the daily dose was divided in two administration 12 hours a part); C) haloperidol 0.8 mg/Kg (HAL); D) GBR 12909 15mg/kg (GBR).

The drug dosages of the antipsychotics were chosen based on previous animal studies in which behavioral effects are elicited that are predictive of antipsychotic activity (Pira et al., 2004), or within the range most commonly used in rat brain gene expression studies (Cochran et al., 2002; Merchant and Dorsa, 1993; Robertson and Fibiger, 1992). The dose of GBR 12909 is consistent with a previous study in which submaximal behavioral responses are obtained (Lane et al., 2005). All treatments were performed intraperitoneally (*i.p.*). The animals were sacrificed by decapitation 90 minutes after the last injection. The brains were rapidly removed, quickly frozen on powdered dry ice and stored at -70^{O}C prior to sectioning.

4.3.4 Radiolabeling and purification of oligonucleotide probes

The *Homer 1a* probe was a 48-base oligodeoxyribonucleotide complementary to bases 2527-2574 of the rat *Homer* mRNA (GenBank # U92079) (MWG Biotech; Firenze, Italy). The *ania-3* probe was a 48-base oligodeoxyribonucleotide complementary to bases 1847-1894 of the rat *ania-3* mRNA (GenBank # AF030088) (MWG Biotech; Firenze, Italy). For each probe a 50µl labeling reaction mix was prepared on ice using DEPC treated water, 1X tailing buffer, 7.5pmol/µl of oligo, 125 Units of TdT and 100mCi ^{35}S-dATP. The mix was incubated 20 min at 37°C. The unincorporated nucleotides were separated from radiolabeled DNA using ProbeQuant G-50 Micro Columns (Amersham Biosciences; Milano, Italy). As an assessment of the probe specificity, the autoradiographic signal distribution was compared and found to be consistent with previous *in situ* hybridization studies (Brakeman et al., 1997; Polese et al., 2002). The specificity of each probe was also tested by a control experiment using the corresponding sense oligo.

4.3.5 Image analysis

Signal intensity analysis was carried out on digitized autoradiograms measuring mean optical density within outlined Regions of Interest (ROIs) in correspondence of subregions of the cortex, caudate-putamen, and nucleus accumbens (oval templates Figure 4-1a). Sections were quantitated blind to the treatment conditions. In order to test for inter-observer reliability an independent quantitation was performed by a second investigator. Only quantitatively

comparable results, in terms of consistency of statistically significant effects obtained by the two investigators, were considered reliable.

Quantitative measurements of the spatial distribution of the signal within the caudate-putamen were carried out using ImageJ as follows. The digital image of autoradiogram of interest is rotated 45 degrees so that a rectangle can be selected along a direction form DL to VM subregions of caudate-putamen (rectangular template in Figure 4-1b). The ImageJ command `plot profile` is used to output the average signal intensity profile along the horizontal axis of the tilted selection. This command computes the average intensity of pixels in each vertical line on the tilted rectangle template.

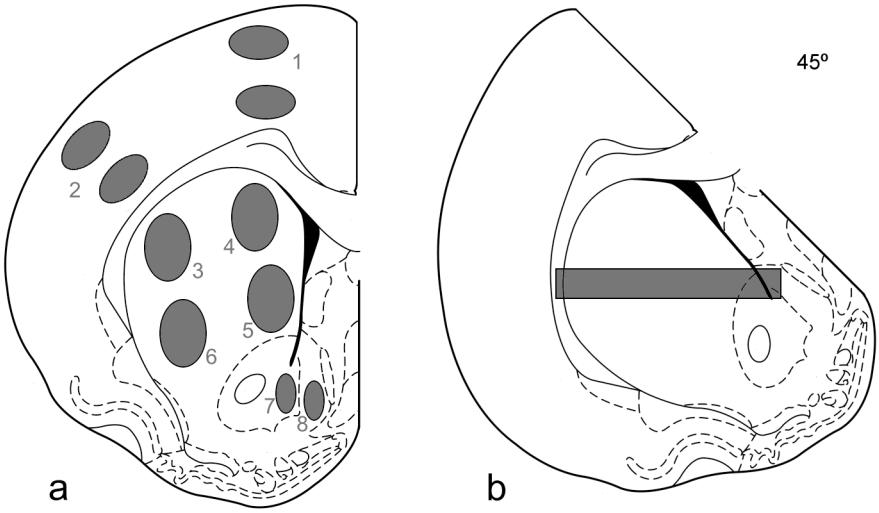


Figure 4-1: *Panel a*: diagram of regions of interest (ROIs) quantitated on autoradiographic film images in rat forebrain. 1 = Frontal cortex (FC); 2 = Parietal cortex (PC); 3 = dorsolateral caudate-putamen (DL); 4 = dorsomedial caudate-putamen (DM); 5 = ventromedial caudate-putamen (VM); 6 = ventrolateral caudate-putamen (VL); 7 = core of accumbens (AcbCo); 8 = shell of accumbens

(AcbSh). *Panel b*: rectangle selection used for the quatitation of spatial distribution profiles on a 45º rotated image. Modified from Paxinos and Watson Rat Brain Atlas (Paxinos and Watson, 1997)

4.3.6 Data processing

Measurements of mean optical density (OD) within ROIs were converted using a calibration curve based on the standard scale co-exposed to the sections. Standard values from 4 through 12 have been previously cross-calibrated to ^{35}S brain paste standards, in order to assign a dpm/mg tissue wet weight value to each OD measurement through a calibration curve. For this purpose a "best fit" 3rd degree polynomial was used. For each animal measurements from the 3-7 adjacent sections were averaged. A One Way Analysis of Variance (ANOVA) was used to analyze treatment effects, and to determine the locus of effects in any significant ANOVA the Student-Neuman-Keuls *post hoc* test was used.

The spatial distribution profiles acquired (as described in the 'Image analysis' section) from previously published *in situ* hybridization autoradiograms (Berke et al., 1998), were analyzed by a correlation based clustering (`clusterdata` command in Matlab) in order to verify if an unsupervised classification could separate between antipsychotic- and stimulant-induced spatial profiles in an independent dataset.

4.4 Results

Homer 1 gene expression was detected in several regions of the forebrain of control animals with higher intensity of the autoradiographic signal in caudate-putamen, in the nucleus accumbens, cortex and islands of Calleja, with a similar pattern for the two splice variants *Homer 1a* and *ania-3* (Figure 4-2). Statistically significant effects of treatments *vs.* saline are summarized in Table 4-1.

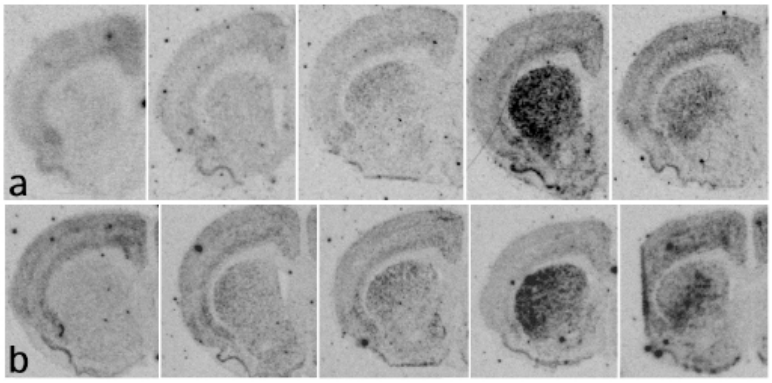


Figure 4-2: Autoradiographic film images of Homer 1a (panel a) and ania-3 (panel b) mRNA detected by means of in situ hybridization histochemistry (ISHH) in coronal brain sections after acute treatment with saline (SAL), quetiapine 15mg (QUE15), quetiapine 30mg (QUE30), haloperidol (HAL), or GBR 12909 (GBR).

		Acute Treatment				Chronic Treatment		
		HAL	QUE15	QUE30	GBR	HAL	QUE	GBR
cortex	FC outer							
	FC inner				*ania-3*			
	PC outer							*Homer1a, ania-3*
	PC inner							*Homer1a, ania-3*
caudate-putamen	DM	*Homer1a, ania-3*			*ania-3*	*ania-3*		
	DL	*Homer1a ania-3*				*Homer1a, ania-3*		
	VL	*Homer1a, ania-3*			*ania-3*	*Homer1a, ania-3*		
	VM	*Homer1a, ania-3*			*ania-3*			
nucleus accumbens	core							
	shell	*ania-3*		*ania-3*	*ania-3*			

Table 4-1: summary of statistically significant changes compared to SAL at the post hoc test. FC = frontal cortex; PC = parietal cortex; DM = dorso-medial; DL = dorso-lateral; VL = ventro-lateral; VM = ventro-medial.

4.4.1 Acute Treatment.

As reported previously (de Bartolomeis and Iasevoli, 2003; Polese et al., 2002), acute haloperidol treatment robustly induced the expression of *Homer 1a* in all subregions of the caudate-putamen. Statistically significant differences were detected among experimental groups in all subregions of caudate-putamen (DM: ANOVA p=0.0024; DL: ANOVA p=0.0131; VL: ANOVA p=0.0041; VM: ANOVA p=0.0067), but not in cortex and nucleus accumbens (Figure 4-2 and Figure 4-3). The *post-hoc* test performed on each caudate-putamen subregion revealed a significant signal increase in the caudate-putamen in haloperidol group compared to control, and to any other experimental group. The other treatments showed no statistically different difference of *Homer 1a* gene expression compared to SAL. GBR induced the expression of *ania-3* in the inner layer of the

frontal cortex (ANOVA, p=.0027), with a statistically significant increase compared to all other experimental groups. This layer of the cortex roughly corresponds to the peak distribution of D_1 and D_2 dopamine receptors within the cortex (Boyson et al., 1986), and resembles the distribution of dopamine-containing axon terminals (Dawson et al., 1986). Within the caudate-putamen *ania-3* splice variant shows a haloperidol-induced expression pattern similar to that of *Homer 1a*. In all the caudate-putamen subregions statistically significant changes (ANOVA p<.0001) were detected, where the *post hoc* test showed that HAL increases the expression of *ania-3* in all the subregions and GBR in all of them except DL. *Ania-3* was upregulated also in the shell of nucleus accumbens (ANOVA p= .0024), where at the *post hoc* test, the expression levels induced by QUE30, HAL, and GBR, were significantly higher compared to the saline group. Compared to *ania-3*, the variant *Homer 1a* had a similar trend in the nucleus accumbens, but not statistically significant (p=.0593).

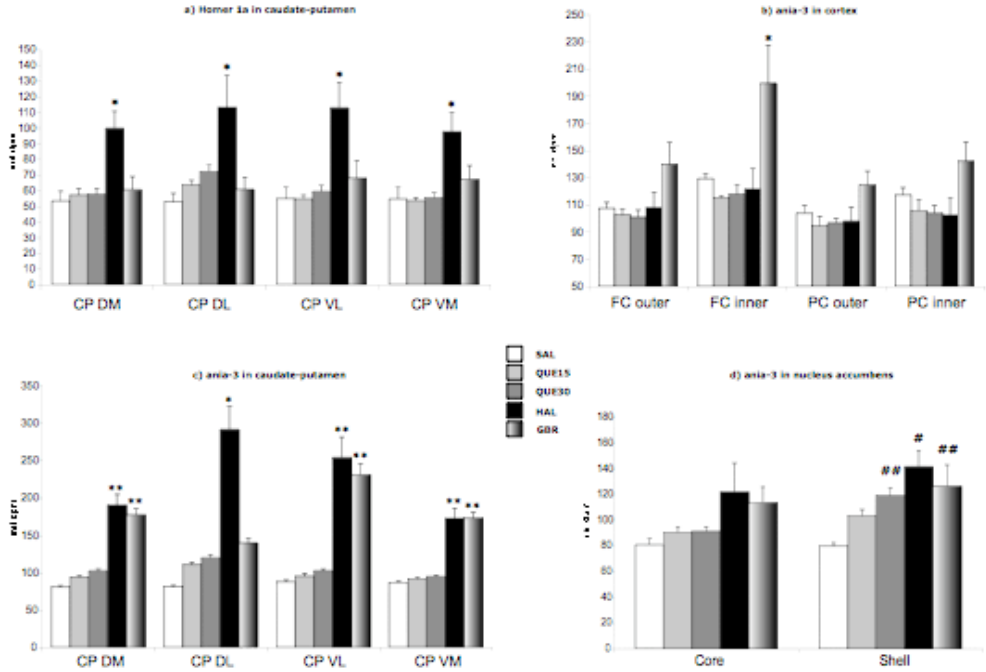


Figure 4-3: Homer 1a and ania-3 mRNA levels after acute treatment. Panel a: Homer 1a mRNA levels in caudate-putamen. Panels b, c, and d: ania-3 mRNA levels in cortex, caudate-putamen and nucleus accumbens. Data are reported in relative dpm as mean ± S.E.M.

* vs. all other treatments (ANOVA, p<0.05);

** vs. SAL, QUE15, QUE30 (ANOVA, p<0.05);

vs. SAL, QUE15 (ANOVA, p<0.05);

vs. SAL (ANOVA, p<0.05).

4.4.2 Chronic Treatment.

Statistical analysis showed a statistically significant increase of both *Homer 1a* (ANOVA, outer layer p=.0040; inner layer p<.0203) and *ania-3* (ANOVA, outer layer p=.0003; inner layer p<.0001) gene expression in the parietal cortex of rats treated with GBR (Figure 4-4 and Figure 4-5). This induction was not detected in the acute treatment. Apart from this observation, the expression patterns of the

two *Homer 1* splice variants in the chronic treatment were similar to the acute treatment. Statistically significant changes of *Homer* gene expression were found in lateral caudate-putamen regions (ANOVA: DL p= .0014; VL p= .0038) by chronic haloperidol treatment. A similar upward trend was found in DM, that is not significant at the *post hoc* test compared to SAL, but it was compared to GBR. *Ania-3* splice variant showed a statistically significant induction in most caudate-putamen subregions (ANOVA: DM p=.0086, DL p=.0023, VL p=.0019) by HAL compared to SAL and to all other treatments. No statistically significant changes were found in nucleus accumbens.

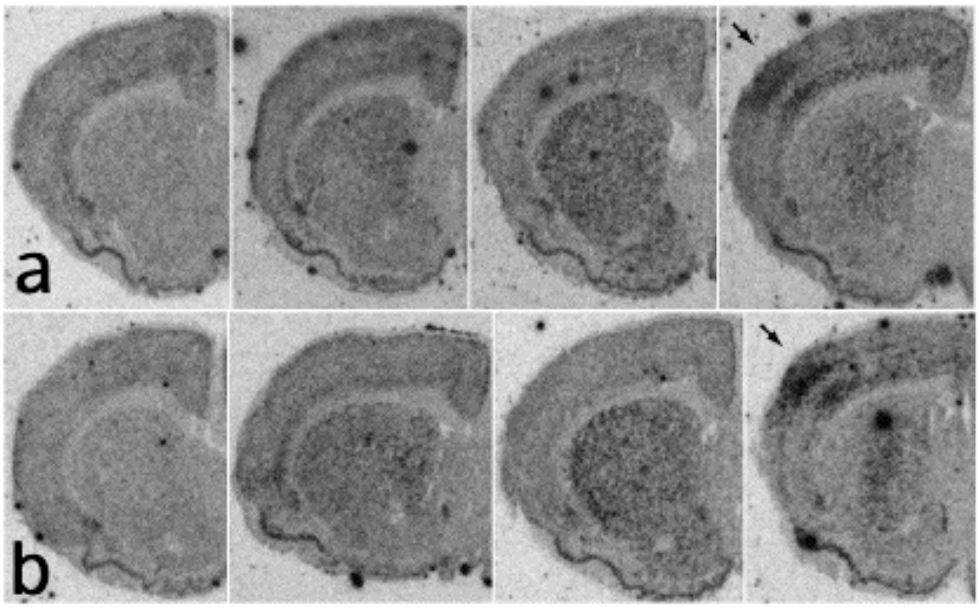


Figure 4-4: Autoradiographic film images of Homer 1a (panel a) and ania-3 (panel b) mRNA detected by means of in situ hybridization histochemistry (ISHH) in coronal brain sections after chronic treatment with saline (SAL), quetiapine (QUE), haloperidol (HAL), or GBR 12909 (GBR).

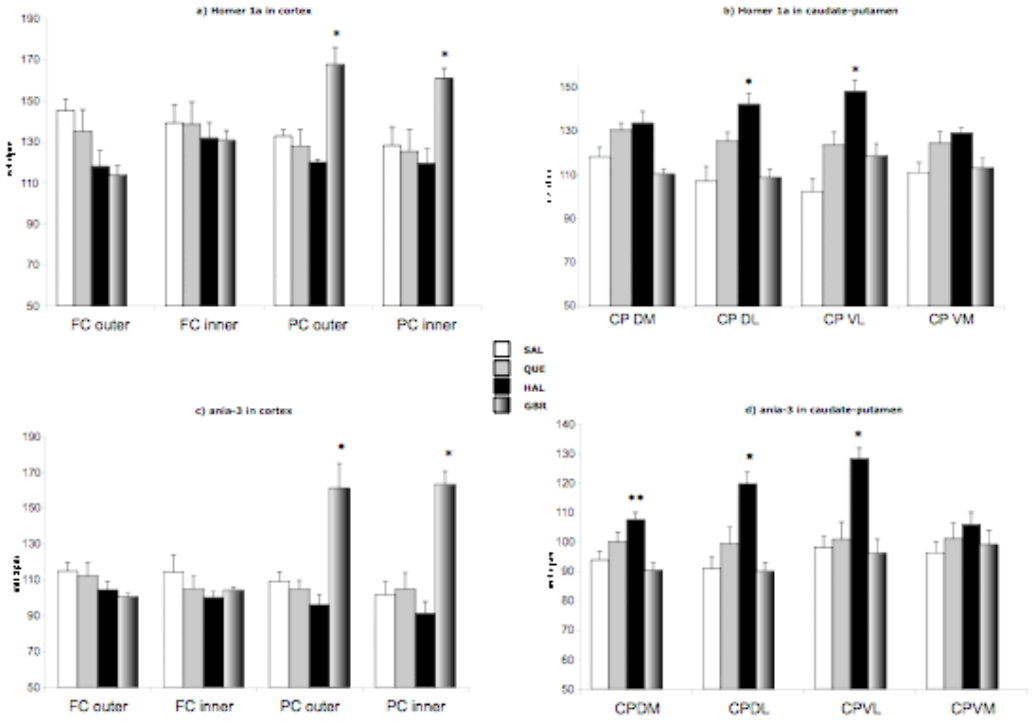


Figure 4-5 : Homer 1a and ania-3 mRNA levels after chronic treatment. Panels a and b: Homer 1a mRNA levels in cortex and caudate-putamen. Panels c and d: ania-3 mRNA levels in cortex and caudate-putamen. Data are reported in relative dpm as mean ± S.E.M.

* vs. all other treatments (ANOVA, $p<0.05$);

** vs. SAL, GBR (ANOVA, $p<0.05$);

4.4.3 Spatial distribution analysis.

Representative spatial profiles of *Homer* 1a signal from acute GBR and HAL are compared in Figure 4-6. This plot shows the distinct profiles of the signal distribution for the two experimental groups within the caudate putamen. We applied the spatial quantitation method to the autoradiographic images of 9 distinct IEG probes (c-fos, MKP-1, ania-3, ania-1, ania-4, ania-6, ania-7, ania-8, and ania-9) from an independently published set of in situ hybridization

experiments (Berke et al., 1998). The clustering analysis correctly distinguished the two classes with only 3 misclassifications out of 18 occurred (Figure 4-7), which corresponds to a cumulative p value of p=.0038 assuming a binomial background distribution ($\pi = .5$).

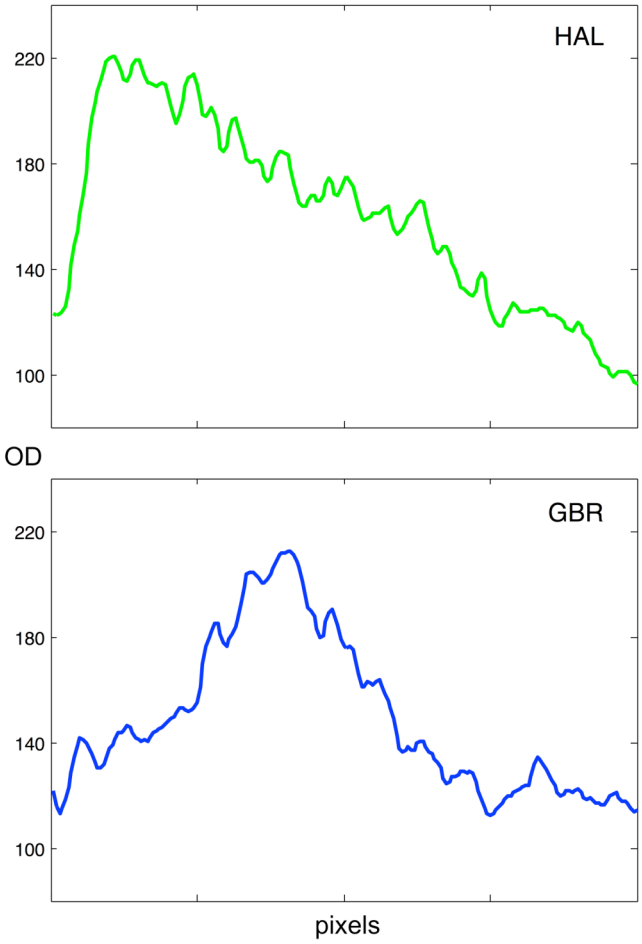


Figure 4-6: Spatial profiles representing the average signal intensity gradient of *Homer 1a* expression measured at an angle of 45º on representative autoradiograms of the acute treatment.

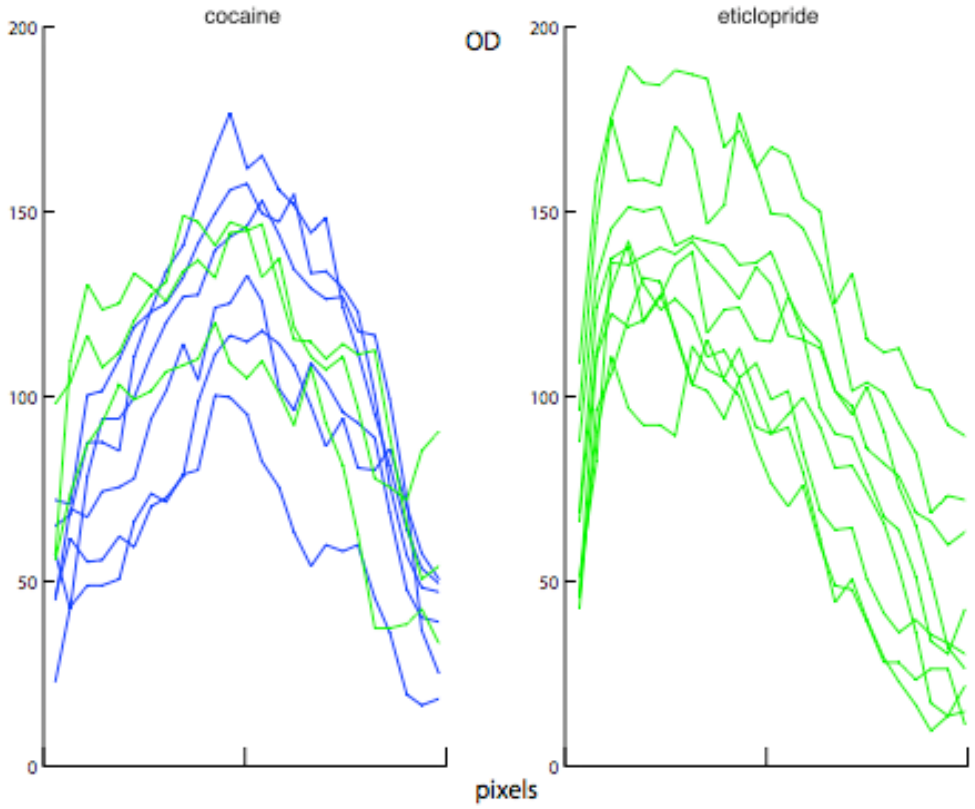


Figure 4-7: Correlation based clustering of spatial profiles measured within caudate-putamen sections hybridized with 9 IEG different probes (autoradiograms from Berke *et al.*). The classifier correctly classified cocaine *vs.* eticlopride treatments, with a misclassification rate of 3 out of 18.

4.5 Discussion

Our previous findings on the differential regulation of *Homer 1a* by typical and atypical antipsychotics (de Bartolomeis et al., 2002) have led us to propose that measuring the increase of *Homer 1a* in caudate-putamen in rodents can be used to discriminate typical from atypical antipsychotics. Specifically, our previous results on the differential expression of haloperidol *vs.* atypical antipsychotics olanzapine and clozapine (de Bartolomeis et al., 2002; Polese et al., 2002)

suggested that typical antipsychotics with high D2R affinity strongly induce an overexpression of *Homer 1a* in subcortical regions of the rat brain after acute treatment. To further confirm this hypothesis, it needs to be tested on several other antipsychotics and/or other agents that target D2Rs. The response of *Homer 1* gene to antipsychotics was also assessed with a chronic administration, resembling the clinical treatment regimens required for antipsychotic effects to be observed in humans. In the present paper we confirm the strong induction of *Homer 1a* gene expression in rat caudate-putamen after acute administration of the typical antipsychotic haloperidol and show a lack of induction by the atypical quetiapine (Figure 4-3a and Figure 4-3c), consistent with the previously reported *Homer 1a* induction patterns in the rat striatum. Quetiapine only induced the gene expression of *ania-3* in the shell of the nucleus accumbens, showing selectivity to limbic structures. The induction of *ania-3* by GBR after acute treatment occurred in brain regions involved in rewarding effects of stimulant drugs (Wise et al., 1996), including, the ventral striatum regions, the accumbens shell, and inner layer of frontal cortex which receives mesocortical dopamine projections (Dawson et al., 1986).

In most subcortical regions, chronic treatments showed a similar effect on *Homer 1* gene, as haloperidol but not quetiapine induced its expression (Figure 4-5b and Figure 4-5d). As opposed to the acute treatment, chronic GBR did not affect *ania-3* in caudate-putamen (Figure 5d). Most remarkably, in the cortical regions both *Homer 1a* (figure 5a) and *ania-3* (Figure 5c) showed a strong induction by chronic GBR limited to the parietal cortex, both statistically significant ($p<.0001$). This induction in the parietal cortex was absent or not statistically significant after

the acute treatment and could be related to the recruitment of 'sensitization' specific neuronal networks by stimulants after a pretreatment schedule, as previously suggested by Curran and coworkers (Curran et al., 1996). Their results show an increase of *c-fos* expression in the somatosensory cortex by cocaine after a sensitization pretreatment schedule with amphetamine, cocaine or morphine. The neuroanatomical specificity of *c-fos* induction by cocaine after a pretreatment schedule found by Curran strikingly resembles the response of *Homer 1* gene expression to chronic GBR (Figure 4-5a and Figure 4-5c), suggesting a common activation mechanism of the two IEGs, even though our chronic treatment did not include a withdrawal period as in the experiment by Curran. As opposed to *c-fos*, *Homer 1* is an effector gene and is directly involved in synaptic plasticity through a dominant negative effect on mGluRs signal transduction. The induction of *Homer 1* in somatosensory cortex after repeated GBR administration might be involved in a compensatory blunting of cortical activity as was shown to occur after *Homer 1a* overexpression by viral vector infusion in the frontal cortex of rats (Lominac et al., 2005).

Overall, the two splice variants of *Homer 1* gene showed a similar expression pattern. The major difference between them is the response to GBR in the frontal cortex after the acute treatment where *ania-3* is upregulated (Figure 4-4b) and *Homer 1a* is not. Acute GBR also induced *ania-3* in most caudate-putamen subregions (Figure 4-4c). Other significant changes specific to the *ania-3* probe were detected in the nucleus accumbens after acute treatments (Figure 4d), where QUE30, HAL, and GBR increased *ania-3* expression compared to SAL in the shell of accumbens. The increase of *ania-3* in nucleus accumbens but not in

caudate-putamen, is consistent with a prominent effect on limbic regions by quetiapine, while sparing the nigrostriatal pathway that is implicated in EPSEs (Tada et al., 2004; Westerink, 2002). The differences in expression patterns found between the two splice variants of *Homer 1* gene may suggest a differential regulation with a neuroanatomical specificity. However a simpler explanation for those differences could be that the signal-to-noise ratio for *ania-3* is higher than for *Homer 1a*. Thus, further investigation is needed to conclusively determine whether the two variants are indeed differentially expressed. Should this be confirmed it would be interesting and challenging to pin down the mechanisms involved in such differential regulation.

Apart from the strong induction of *Homer 1* gene by chronic GBR in the parietal cortex, we also report a novel finding about the GBR-induced expression pattern in the striatum. The anatomical distribution of the signal induced by acute (Figure 4-2) and chronic (Figure 4) GBR in caudate-putamen shows a distinctive spatial gradient of expression for both *Homer 1* probes. In detail, following a direction from VM to DL (oblique box in Figure 4-1b), the signal is more intense at the center and decreases at the extremities, such that higher levels are found in the caudate-putamen regions VL-center-DM, and lower levels in VM and DL (Figure 4-6). To our knowledge, this distinctive regional distribution of induced expression of IEGs within the caudate-putamen has never been described before. Interestingly, the same distribution within the caudate-putamen can be appreciated by carefully observing the autoradiograms of cocaine-treated rat forebrains from previously published paper by Berke and coworkers (Berke et al., 1998). Most of the IEGs studied in their work display this distinctive signal distribution within

the caudate-putamen in the cocaine group, whereas the induction by the antipsychotic eticlopride, resembles the classical anatomical distribution induced by HAL in our experiments. Our clustering analysis performed on this independent set of autoradiograms detected a statistically significant difference between spatial gene expression profiles between cocaine and eticlopride treated animals. The fact that both haloperidol and a DAT inhibitor, with opposite effects on dopaminergic neurotransmission, induce *Homer 1* in caudate-putamen could argue against a relevant role of *Homer 1* gene in the action of antipsychotics. However this conclusion is not so straightforward, as the neuroanatomical distribution of the gene expression induced by GBR and haloperidol indeed showed differential effects in both acute and chronic regimens. Specifically, the spatial distribution was different within the caudate-putamen, and in the chronic treatment a differential induction was observed in the parietal cortex.

In conclusion, our results confirm that *Homer 1* gene induction pattern within striatal structures may be considered as a predictor of 'atypicality' of antipsychotic compounds in both acute and chronic regimens, with possibly a slightly different pattern observed for the two splice variants *Homer 1a* and *ania-3*. Chronic treatment with GBR 12909 showed a strong induction in the parietal cortex resembling the activation of 'sensitization' circuitry by stimulants as shown for *c-fos*. Finally, our results provide strong evidence, compatible with independently published imaging data, of a differential anatomical induction pattern by agents that directly affect dopaminergic neurotransmission, namely (typical) antipsychotics and DAT blockers.

Chapter 5

Ziprasidone Experiments

5.1 Abstract

The essential difference between typical and atypical antipsychotics is the lower incidence of extrapyramidal side effects of the latter. Several animal studies have shown a differential effect on neuronal gene expression of immediate early genes, including the effector gene homer 1. The D2 dopamine receptor, blocked by all antipsychotics, plays a crucial role in the mechanisms of antipsychotic effects as well as extrapyramidal side effects, and appears to mediate homer 1a differential expression. Acute induction of homer 1a gene expression in rat striatum has been

recently proposed as a novel preclinical characterization of antipsychotics. However the effect on homer 1 gene after prolonged antipsychotics administration has not been assessed so far. In order to further characterize the differential effects of antipsychotics on post synaptic density genes, the novel atypical antipsychotic Ziprasidone was used in both acute and chronic paradigms. Rats were treated (i.p.) with clozapine, haloperidol, ziprasidone, or vehicle, and sacrificed 90 min after the injection in the acute paradigm. In the chronic schedule haloperidol ziprasidone or veichle treated animals were sacrificed at either 90 min or 24h after the last injection. Quantitative in situ hybridization was carried out for of postsynaptic density genes homer1a, 1b, and shank in rat striatum. Our results show a dose dependent induction of homer 1a gene expression in lateral caudate putamen of the rat after acute administration of ziprasidone, as well as after 90 mins after the chronic treatment, but not at 24 hrs from the last injection of the chronic treatment. This suggests that the effect on homer 1a is transient and also that it does not saturate over prolonged administration. The dose dependency of this effect may correlate to a higher D2 receptor occupancy obtained by the higher concentrations of the drug. Finally, these findings also show that homer 1a induction pattern in rat striatum can separate ziprasidone 4mg/kg and clozapine 15mg/kg from haloperidol 0.8mg/kg, as shown for other atypical antipsychotics.

5.2 Introduction

To investigate the dynamics of homer 1a response to antipsychotics, and because the effect on homer 1 gene after prolonged antipsychotics administration has not

been assessed so far, we expanded our paradigm to include a chronic administration. The advantage of a prolonged treatment administration is that it more closely resembles the clinical administration schedule of these compounds. We used an experimental design that allowed us to explore the effects on homer gene expression after short (90 min) and long (24h) duration of withdrawal. In our chronic paradigm we also investigated the effects on other relevant genes, not described as IEGs, involved in postsynaptic glutamatergic system, including homer 1b, Shank1, PSD-95, and IP_3R, all known to modulate postsynaptic signaling through its direct or indirect interaction with homer (Naisbitt et al., 1999). These genes encode major components of postsynaptic density and interact directly or indirectly with each other (de Bartolomeis and Iasevoli, 2003). Moreover, they cooperate in clustering glutamate receptors and in directing the intracellular second messenger-mediated response to extracellular signals, including those evoked by antipsychotics (Yang et al., 2004), and thus can be considered as candidate genes for psychosis.

In the present study we investigated the effects of acute administration of two different dosages of the atypical antipsychotic ziprasidone compared with the typical antipsychotic haloperidol and with clozapine, the prototype atypical antipsychotic, on gene expression of homer1a in rat forebrain. Ziprasidone is an atypical antipsychotic, whose efficacy in schizophrenia has clearly been demonstrated on both short and long period with an excellent tolerability profile both regarding movement disorders and metabolic side effects (Nasrallah and Newcomer, 2004). The choice of ziprasidone in the present study was based on its receptor affinity profile and clinical properties, since it shows a low liability to

EPSEs despite its relatively high affinity for D_2 receptors, making this compound particularly interesting in terms of how it might affect homer gene expression. Indeed, homer gene expression is sensitive to dopamiergic manipulation, and has been shown to be involved in subtle tuning of motor function (Tappe and Kuner, 2006).

5.3 Materials and methods

The procedure for *in situ* hybridization histochemistry was taken from several standard published protocols (Ambesi-Impiombato et al., 2003; Austin et al., 1992; Young et al., 1986). Refer to Chapter 2 for detailed procedure.

5.3.1 Drugs

Haloperidol (Serenase® 2mg/ml. Lusofarmaco; Milan, Italy) was used as commercially available ampoules and diluted in saline. Clozapine was dissolved in saline. Ziprasidone was provided by Pfizer as a powder and was dissolved by few drops of DMSO and 0.9% NaCl. All injections were performed intraperitoneally using an equal injection volume.

5.3.2 Drug treatment: Acute paradigm

On the day of the experiment rats were randomly assigned to each of the following treatment groups: A) 0.9% NaCl (SAL); B) 15mg/kg clozapine (CLO) C) haloperidol 0.8mg/Kg (HAL); D) ziprasidone 4mg/kg (ZIP4); E) ziprasidone

10mg/kg (ZIP10). The animals were sacrificed by decapitation 90 min. after the treatment.

5.3.3 Drug treatment: Chronic Paradigm

Animals were randomly assigned to each of the following treatment groups: A) 0.9% NaCl (SAL); B) haloperidol 0.8mg/Kg (HAL); C) ziprasidone 4mg/kg (ZIP). On the last day of injection, the animals were divided in two subgroups: one was sacrificed at 90 minutes after the last injection whereas the other after 24 hours, resulting in the following experimental groups: SAL90', HAL90', ZIP90' (sacrificed at 90 min.); and SAL24h, HAL24h, ZIP24h (sacrificed at 24 hrs.).

5.3.4 Radiolabeling and purification of oligonucleotide probes

The homer probe was a 48-base oligodeoxyribonucleotide complementary to bases 2527-2574 of the rat homer 1a mRNA (GenBank # U92079) (MWG Biotech; Firenze, Italy). The Shank probe was a 48-base oligodeoxyribonuce complementary to bases 2757-2804 of the rat Shank 1 mRNA (GenBank # NM_0317751) (MWG Biotech; Firenze, Italy). The PSD95 probe was a 45-base oligodeoxyribonucleotide complementary to bases 225–269 of the rat PSD95 mRNA (GenBank # M96853) (MWG Biotech; Firenze, Italy). The homer1b was a 48-base oligodeoxyribonucleotide complementary to bases 1306-1354 of the rat homer1b mRNA (GenBank Accession AF093267).The IP_3R probe was a 48-base oligodeoxyribonuce complementary to bases 7938-7985 of the rat IP_3R mRNA (GenBank # NM_001007235) (MWG Biotech; Firenze, Italy). These sequences were checked with blastn algorithm against GenBank, to avoid cross-hybridization. For each probe a 50µl labelling reaction mix was prepared on ice

using depc treated water, 1X tailing buffer, 1.5mM $CoCl_2$, 7.5pmol/µl of oligo, 125 Units of TdT and 100mCi ^{35}S-dATP. The mix was incubated 20 min at 37°C. The unincorporated nucleotides were separated from radiolabeled DNA using ProbeQuant G-50 Micro Columns (Amersham Biosciences; Milano, Italy). The autoradiographic signal distribution of homer matched that of previous ISSH studies (Berke et al., 1998; Brakeman et al., 1997; de Bartolomeis et al., 2002; Polese et al., 2002). Also, the specificity of each probe was tested by a control experiment using the corresponding sense oligo.

5.4 Results

5.4.1 Anatomical distribution of gene expression

Gene expression for homer 1a and shank was detected in control animals with a signal distribution that was comparable between acute and chronic paradigms. Low levels of homer 1a gene expression were detected in the forebrain of control animals in both cortical and subcortical regions (Figure 5-1), consistently with previous experiments (de Bartolomeis et al., 2002; Polese et al., 2002). Shank gene expression was detected in several regions of the forebrain of control animals with higher intensity of the autoradiographic signal in cortical layers and the in the islands of Calleja, and low levels in subcortical regions.

Qualitatively, homer 1b and PSD-95 were both abundantly expressed throughout cortical and subcortical regions, with a homogeneously spatial distribution of the signal. High levels of the signal was also present in the islands of Calleja. Finally,

IP3R signal was detected throughout the caudate-putamen and accumbens, and with a slightly lower intensity in the cortex.

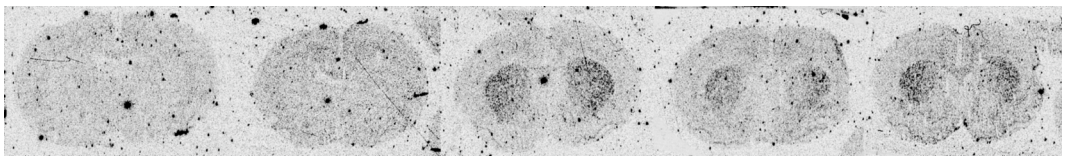


Figure 5-1: Autoradiographic film images of homer 1a mRNA detected by means of in situ hybridization histochemistry (ISHH) in coronal forebrain sections from rats assigned to the following treatment groups (from left to right): SAL, CLO, HAL, ZIP4, ZIP10.

5.4.2 Acute paradigm

Statistically significant differences were detected for homer 1a among experimental groups in all subregions of the caudate-putamen (DL: ANOVA $p<0.0001$; VL: ANOVA $p<0.0001$; VM: ANOVA $p=0.0003$; DM: ANOVA $p=0.0001$). Post hoc tests revealed a statistically significant signal increase in all subregions of the caudate-putamen for HAL, ZIP4 and ZIP10 groups compared to SAL and CLO (Figure 5-2). On the other hand, post hoc tests revealed no statistically significant differences of homer1a gene expression between CLO and SAL in the caudate-putamen. Remarkably, in the lateral subregions (DL and VL), a further statistically significant difference at the post hoc test was detected between the two different dosages of ziprasidone, with the level of ZIP4 being not only higher than SAL and CLO but also lower than ZIP10 and HAL. No statistically significant differences were detected in measurements of nucleus accumbens. Consistent results were obtained by an independent in situ hybridization experiment using different forebrain sections from the same animals

(data not shown). Statistical analysis for shank signal showed no significant change across experimental groups in any of the striatal sub-regions.

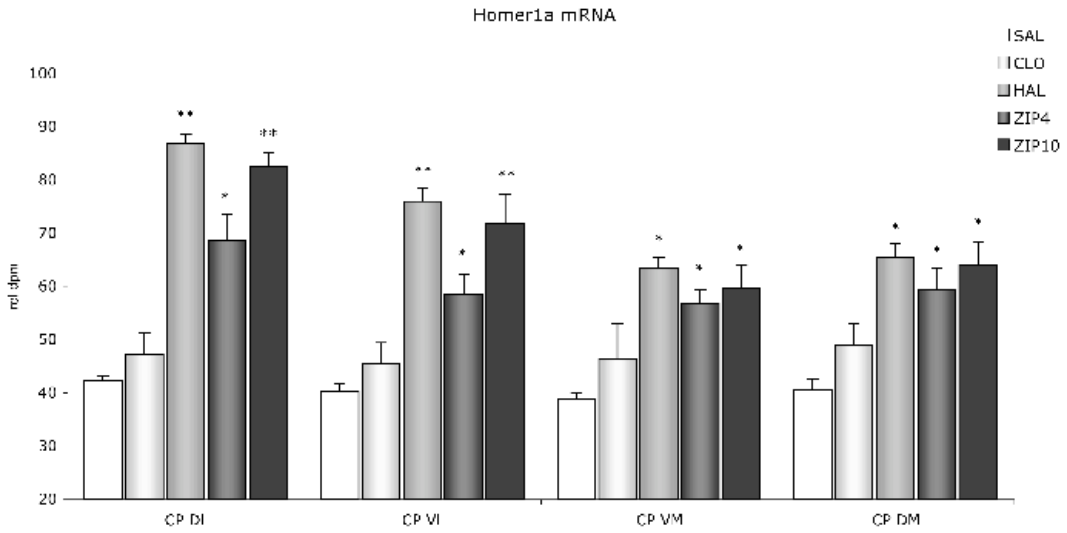


Figure 5-2: Homer 1a mRNA levels measured after acute treatment in caudate-putamen subregions (CP DL = dorsolateral; CP VL = ventrolateral, ANOVA $p<0.0001$; CP VM = ventromedial, ANOVA $p=0.0003$; CP DM = dorsomedial, ANOVA $p=0.0001$), quantitated by densitometry of in situ hybridization histochemistry autoradiograms. Post-hoc test levels of significance: *treatment vs. SAL, CLO; **treatment vs, SAL, CLO, ZIP4. Data are expressed as relative dpm ± S.E.M.

5.5 Chronic Paradigm

Quantitative analysis of homer 1a signal in the chronic experiment (Figure 5-3) revealed a statistically significant signal increase in all subregions of the caudate-putamen for HAL90' group compared to SAL90' (ANOVA, $p<0.05$, Figure 5-4). HAL90' also shows a significant induction of homer1a gene compared to ZIP90'

group in all caudate-putamen subregions (but only a positive trend in DL) and in both core and shell of accumbens (Figure 5-4). Post hoc tests also revealed a significant increase of homer1a gene expression of ZIP90' over SAL90' in: DL, VL, and VM caudate-putamen, and a positive trend in DM (Figure 5-4). No significant change was found between ZIP90' and SAL90' in the two subregions of the nucleus accumbens. No change was detected between groups sacrificed at 24 hours from last injection in any of the subregions assessed. The quantitative analysis of shank, homer1b, PSD-95, and IP_3R signal levels in the chronic experiment showed no statistically significant change in any of the regions analyzed in both 90 minutes and 24 hours groups.

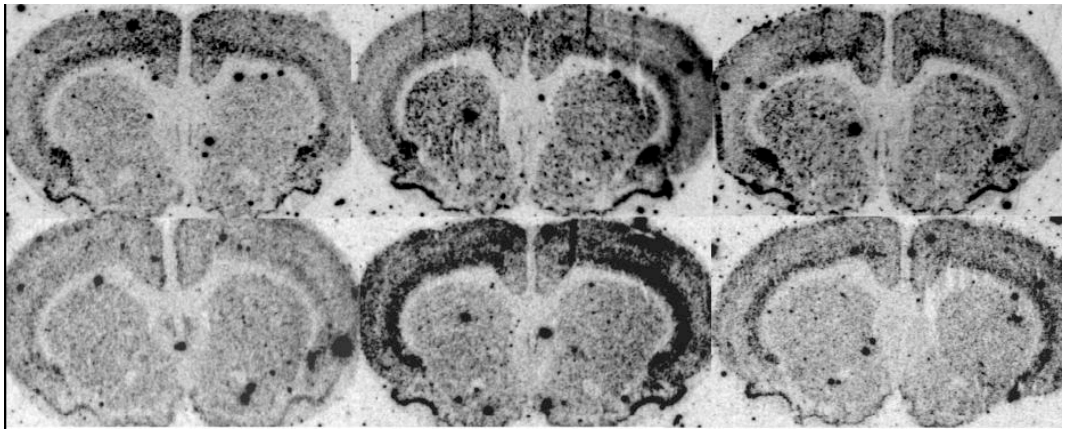


Figure 5-3: Autoradiographic film images of homer 1a mRNA detected by means of in situ hybridization histochemistry (ISHH) in coronal forebrain sections from rats assigned to the following treatment groups: SAL90', HAL90', ZIP90' (from left to right, upper row); SAL24h, HAL24h, ZIP24h (from left to right, lower row).

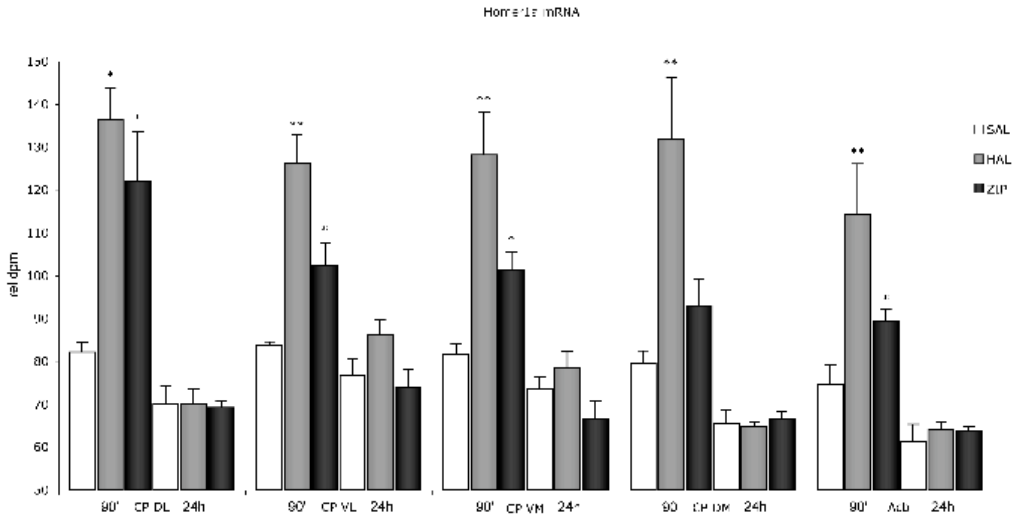


Figure 5-4: Homer 1a mRNA levels measured after chronic treatment in caudate-putamen subregions and nucleus accumbens (90 minutes paradigm: CP DL = dorsolateral, ANOVA p= 0.0141; CP VL = ventrolateral, ANOVA p=0.0021; CP VM = ventromedial, ANOVA p=0.0040; CP DM = dorsomedial, ANOVA p=0.0113; Acb = nucleus Accumbens p=0.0024; 24 hours paradigm: ANOVA p>0.05 in all subregions), quantitated by densitometry of in situ hybridization histochemistry autoradiograms. Post-hoc test levels of significance: *treatment vs. SAL90'; **treatment vs, SAL90', ZIP90'. Data are expressed as relative dpm ± S.E.M.

5.6 Discussion

Previous studies have shown that acute treatment with typical or atypical APS may modulate the expression of the IEG form of homer in the rat forebrain differently (de Bartolomeis et al., 2002; Polese et al., 2002). In this study, we investigated the effects of the novel atypical antipsychotic agent ziprasidone compared to other antipsychotics on the neural expression of genes known to

respond to dopaminergic stimulation. This is especially appealing due to the fact that ziprasidone has a relatively high affinity to the D_2 receptor compared to other atypical antipsychotics (Seeman and Tallerico, 1998), and induces a striatal D_2 occupancy, as assessed by PET in humans at clinically effective doses, which is lower than that induced by haloperidol, similarly to the other atypical compounds risperidone and olanzapine (Mamo et al., 2004). Moreover, for the first time to our knowledge we investigated whether homer 1a acute response is affected by previous repeated administrations of antipsychotics, by assessing whether the IEG-like response of homer 1a is conserved after a 21-day chronic antipsychotic administration and whether the induction of the gene is sustained also after 24 hours from last injection, as may be expected due to persisting levels of antipsychotics in target tissues following a chronic administration.

The dosages of antipsychotics used in this study were all chosen as to fit within the dosage range that produce effects in animal behavioral models that are predictive of antipsychotic efficacy in humans, such as the conditioned avoidance suppression test (Seeger et al., 1995). Moreover the higher dose of ziprasidone used in our paradigm (10mg/kg) is also compatible with the dosage that produces minimal catalepsy, 12.1 mg/kg (9.7-15.1, 95% C.I.) (Seeger et al., 1995).

Our results show a dose dependent induction of homer 1a gene expression in rat caudate putamen after acute administration of ziprasidone, with a significantly greater effect for the higher dose, which is shown to be correlated to EPSE liability in animal models. Homer 1a expression by the higher ziprasidone dose overlaps that obtained by haloperidol. At the lower ziprasidone dosage, shown to

exert responses in animal models predictive of antipsychotic efficacy, but not EPS liability, the level of change observed in Homer 1a expression was significantly lower than with the higher dosage. This provides further evidence that the expression of genes involved in postsynaptic glutamatergic function is differentially modulated by typical and atypical antipsychotics and that this modulation may positively correlate with their degree of dopamine D_2 receptor occupancy.

The dose dependency of the observed effect on homer gene expression by ziprasidone is thus particularly interesting. Specifically, homer induction pattern by ziprasidone in the lateral striatum separates the lower dose (ZIP4) from the higher dose (ZIP10), which shows a consistently similar behavior to HAL. In medial caudate-putamen subregions (VM, DM) ZIP4 shares the same level of significance with ZIP10 and HAL, whereas in lateral subregions (DL and VL) ZIP4 has an intermediate level of significance between CLO and ZIP10. The dose dependency of homer induction by ziprasidone may correlate to an increasing D_2 dopamine receptor occupancy, and possibly to animal responses that are predictive of an increased risk of EPSE. The differential effects on homer gene expression seem to be neuroanatomically specific to lateral caudate-putamen, as no statistically significant differences were detected in the other regions analyzed, including nucleus accumbens. The specific pattern of homer differential activation observed in the lateral caudate-putamen could be explained by a more pronounced action of haloperidol and very high doses of ziprasidone on the striato-nigral pathway compared to clozapine and lower doses of ziprasidone. Consistently, clozapine and ziprasidone are known to have a lower incidence of EPSEs in

clinical practice (Stimmel et al., 2002). Since the lower dose of ziprasidone 4mg/kg and clozapine 15mg/kg are predicted to have antipsychotic effects in animal behavioral models, with a negligible potential for EPSE, our results may capture the distinctive therapeutic properties of these drugs. Specifically, a distinct functional effect of these drugs on the glutamatergic postsynaptic density may contribute to some of their biological effects such as low EPSE liability.

It can be expected that a chronic treatment with psychotropic compounds may cause gene expression to be up- or down-regulated, as seen for a number of target genes (Chen and Chen, 2005; Feher et al., 2005; Semba et al., 1996). To assess this prediction we sacrificed the animals after a chronic schedule and at two time points, acutely (90 min), and delayed (24 h) from last injection. After a chronic antipsychotic treatment homer1a induction is observed in rats sacrificed at 90 minutes but not in the littermates sacrificed at 24 hours. Conversely, c-fos, a different antipsychotic responding IEG (Feher et al., 2005; MacGibbon et al., 1994; Semba et al., 1996), is reported not to be induced at 45 minutes from last administration in a chronic antipsychotic treatment paradigm (Semba et al., 1996). While c-fos induction undergoes a tolerance phenomenon after repeated antipsychotic administration, this seems not the case for homer1a. Our data show that it is induced at 90 minutes from administration in both acute and chronic paradigms. In contrast to the 90 minutes challenge, no change from basal expression has been detected for any of the compounds used in the 24 hours paradigm. Thus, homer1a gene did not show any sensitization or tolerance phenomena after a chronic antipsychotic treatment and it appears to retain the IEG response liability even after a long-term treatment by antipsychotics. Moreover,

homer1a induction after long-term treatment remains specific as demonstrated by the clear separation between haloperidol- and ziprasidone-induced effects in our study. Hence evaluation of homer 1a pattern of expression may provide a compelling tool to estimate the molecular outcome of sustained antipsychotic treatment in preclinical models.

At the molecular level, considering that homer 1a functions as a "dominant negative" on the signal transmission efficiency of group 1 mGluRs (Xiao et al., 1998), it can be predicted that the increase of homer 1a will displace constitutively expressed CC-homers (1b/c, 2, 3), thus disrupting the clustering of mGluRs and uncoupling them from their intracellular effectors. The overall effect would therefore be a reduction of glutamate activation of the striatal neurons. Although a direct correlation of this effect to any biological effect induced by antipsychotics is speculative, a specific pattern of homer 1a induction might represent a first step that triggers complex molecular and neuronal events that ultimately lead to biological effects of the drugs. Moreover, the differential pattern of homer expression may provide a powerful molecular tool for further investigation into the differential molecular mechanisms of typical and atypical antipsychotics, as well as a potential predictor of 'atypicality' for putative novel antipsychotic agents. Finally, our results confirm that antipsychotic compounds acting prevalently at the dopamine receptors can perturb homer 1a, a relevant effector of glutamatergic signaling, which, differently from other early genes such as c-fos, has a direct role in synaptic plasticity.

Chapter 6

Computational Prediction of Antipsychotics Gene Targets

6.1 Abstract

Control of gene expression is essential for the establishment and maintenance of all cell types, and is involved in pathogenesis of several diseases. Accurate computational predictions of transcription factor regulation may thus help in understanding complex diseases, including mental disorders in which dysregulation of neural gene expression is thought to play a key role. However, predictions via bioinformatics tools are typically poorly specific. We have

developed and tested a computational workflow to computationally predict Transcription Factor Binding Sites on proximal promoters of vertebrate genes. The computational framework was applied to groups of genes found to respond to antipsychotic drugs. Our approach for the prediction of regulatory elements is based on a search for known regulatory motifs retrieved from TRANSFAC, on DNA sequences of genes' promoters. Predictions are thus weighted by conservation. These predictions are further refined using a logistic regression to integrate data from co-regulated genes. Consistent results were obtained on a large simulated dataset consisting of 5460 simulated promoter sequences, and on a set of 377 vertebrate gene promoters for which binding sites are known (TRANSFAC gene set). Our results show that integrating information from multiple data sources, such as genomic sequence of genes' promoters, conservation over multiple species, and gene expression data, can improve the accuracy of computational predictions. The validation of our method allowed us to apply the computational framework to Homer1 promoter as a means to infer direct targets of antipsychotics.

6.2 Computational Framework

Our Computational Framework for transcription factor Binding site Identification (CFBI) supplies a set of novel tools to fetch and integrate data from multiple sources and analyze it to make predictions, all in an automated and flexible bioinformatics workflow (Figure 6-1). Differently from previous approaches, CFBI does not require alignment of ortholog gene promoters, nor a linearity

assumption, as in the case of linear regression based algorithms. Our framework can also be applied to qualitative expression data, such as developmental and/or neuroanatomical expression data such as that obtained by in situ hybridization histochemistry.

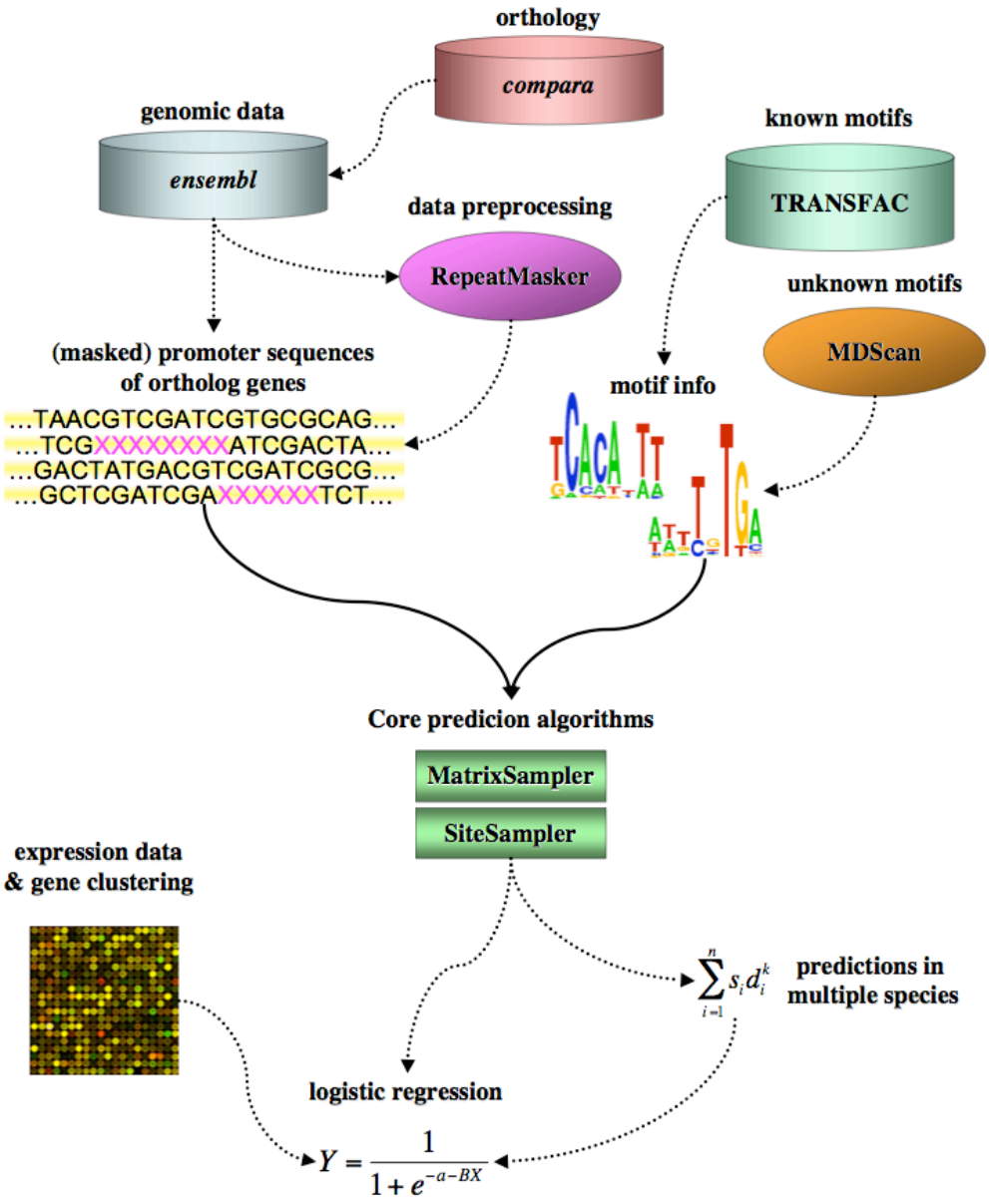


Figure 6-1: Diagram illustrating the structure of the framework for the computational prediction of transcription factor binding sites. The diagram shows the multiple sources of input data, including *ensembl*, *compara*, TRANSFAC (or

alternatively, novel motifs obtained by a motif-finding algorithm, such as MDScan), the optional data preprocessing RepeatMasker step, and the post-processing steps including data integration from multiple species and logistic regression of gene expression defined classes of genes. Dotted lines indicate optional or alternative steps

6.3 Methods

6.3.1 Sequence and motif data retrieval

Promoter sequences were retrieved from the latest build of *ensembl* database (build 32), and ortholog gene IDs were obtained by querying the *compara* database (Hubbard et al., 2005). Finally, all of the 145 vertebrate motif data were fetched from TRANSFAC 9.2. Each transcription binding site motif was modeled as a position weight matrix (PWM).

6.3.2 Position weight matrix score

We computed PWM scores using a statistical formula proposed by Stormo *et al.* (Stormo, 2000; Stormo and Fields, 1998). This score is based on the ratio between the probability of a subsequence being generated from the PWM over that of being generated by the background Markov model. The score of a motif of length w over a promoter sequence of length l is given by:

$$s = \log_2 \sum_{i=1}^{l-w+1} \frac{\prod_{j=1}^{w} p_{ij}}{\prod_{j=1}^{w} p'_{ij}} \quad (1)$$

were p_{ij} is the probability of a base at position $i+j$ based on the PWM and and p'_{ij} is the probability of it being generated by the background Markov model. For this purpose a species-specific 3rd Order Markov model was trained on large (10kb) intergenic regions upstream of a set of human neural genes, including Dopamine D_2 receptor, 5-HT2A, Tryptophan hydroxylase 1, Homer 1, Neuronal acetylcholine receptor alpha-10, c-Myc and c-Fos. Alternatively, a different set of background sequences may be specified each time.

6.3.3 Phylogenetic data integration

For each motif, the PWM score in the promoter of ortholog genes in k different species was integrated by the following mathematical formula that is based on the assumption that some of the regulatory machinery of gene expression is conserved in evolutionary related species:

$$relSum = \sum_{i=1}^{k} s_i(1 - d_i) \tag{2}$$

were s_i is the PWM score of the motif in the promoter of the gene in species i, and d_i is a weight proportional to evolutionary distance from the main species (human), ranging from 0 (same species) to 1 (farthest species). The distance weight d_i (Table 6-1) was calculated using the multi-species alignment of coding sequences of the *myc* gene using the program DNADIST (Felsenstein, 1989). We named this score 'relatedness sum' (*relSum*, for short) since it takes into account how related promoters of different species are.

6.3.4 Qualitative data integration: Logistic regression

If *a priori* information is available indicating that a gene of interest is part of a set of genes that may share common regulatory motifs in their promoters, then this information can be used to increase the specificity of *in silico* predictions. This *a priori* information can be obtained, for example, by selecting a cluster of co-expressed genes from microarray experiments. A value y=1 is assigned to the cluster of co-expressed genes to which the gene of interest belongs, while a value y=0 is assigned to a background set of genes that are thought not to share any common regulatory motifs. Logistic regression is then used to identify the shared regulatory motifs in the co-expressed dataset. The general model for a logistic regression is:

$$y_i = \frac{1}{1 + e^{-\mathbf{a}-\mathbf{b}^T\mathbf{x}}} \qquad \text{for } i = 1...n \tag{3}$$

where n is the total number of target genes in the two sets, the response variable $y_i \in \{0,1\}$ is equal to the class of the i^{th} gene, and **x** is a vector of scores (*relSum*s) for m 'candidate' motifs (regressors). The vectors **a** and **b** are the parameters of the model. Parameter **b** is a vector of size m of fitted weights. The greater the weight, the more likely the corresponding motif is functional. The variance σ^2 of **b** may be computed as:

$$\hat{\sigma}^2 = diag(X'wX)^{-1}$$

where X is the $n \times m$ matrix of *relSum* scores and the diagonal matrix w is equal to:

$$w = \frac{e^{-Xb}}{(1 + e^{-Xb})^2}$$

6.3.5 Generation of the in silico promoter dataset

The sequence generator *Seq-gen* algorithm (Rambaut and Grassly, 1997) was used to build simulated datasets of ortholog sequences. *Seq-gen* is able to generate simulated DNA sequences of a given length and the corresponding 'ortholog' sequences at different evolutionary distances, starting from the 9-species phylogenetic distance matrix previously described (Table 6-1). *Seq-gen* was run to generate 90 simulated DNA sequences with the corresponding 'ortholog' in 9 species (human, chimp, dog, cow, mouse, rat, chicken, fugu, and zebrafish). This program implemented the Hasegawa, Kishino and Yano (HKY) model (Hasegawa et al., 1985) for the generation of simulated data. Motif sequences were randomly selected from the list of known binding sites in TRANSFAC, and inserted in random non-overlapping positions within the simulated promoter sequences. In order to account for the evolutionary distance, we decreased the frequencies of inserted motifs with the evolutionary distance. Thus, human and chimp promoters received two inserts, cow and dog received 1.5 inserts on average, mouse and rat 1 insert, chicken 0.5 inserts and finally fugu and zebrafish 0.2 inserts. Only the high quality subset of 145 TRANSFAC matrices, *i.e.* compiled from 20 or more binding sites, was considered for the generation of simulated datasets. Thus a total of 13050 promoters were analyzed (145 different datasets of 90 genes)

Species	distance
Human	0
Chimp	0.0041
Dog	0.0954
Cow	0.1071
Mouse	0.2595
Rat	0.1352
Chicken	0.3705
Fugu	0.4805
Zebrafish	0.7485

Table 6-1: Phylogenetic distance weights used to compute the 'relatedness sum' score (variable *d* in equation 2).

6.4 Results

The CFBI approach we developed proceeds as follows (Figure 6-1): the gene of interest is selected and its promoter sequence, together with promoter sequences of ortholog genes in other species are retrieved from *ensembl* database (www.ensembl.org) and *compara* for orthology information (Hubbard et al., 2005). A list of motifs of all known vertebrate transcription factors (TFs) is obtained by the TRANSFAC database, or a list of novel motifs may be predicted by MDScan (Liu et al., 2002). Motifs are then modeled as Position Weight Matrices (PWMs). A PWM score for each motif is computed in each promoter of the ortholog gene set. The PWM scores in the ortholog gene set are integrated using a weighted sum calibrated on the phylogenetic distances between the species. This final score can then be used to rank the motifs and select the ones with the highest probability of being functional transcription factor binding sites.

These predictions can be refined using logistic regression to integrate data from potentially co-regulated genes. The logistic regression makes use of two sets: a set of promoters of potentially co-regulated genes, and a background set of gene promoters that do share any regulatory motifs. For further details please refer to the Methods section.

In order to establish the performance of CFBI, we counted the number of true positives (TP), true negatives (TN), false positives (FP), false negatives (FN), and presented the results as Positive Predictive Value (PPV) = $\frac{TP}{TP + FP}$, and Sensitivity = $\frac{TP}{TP + FN}$.

6.4.1 Simulated data

Performance and usability of the CFBI was tested on an *in silico* dataset consisting of 1450 genes with ortholog sequences in 9 different species (see Methods).

The predictive performance of CFBI on this dataset is shown in Figure 6-2. Robustness of the logistic regression step was tested by progressively introducing 'noise' in the set of co-regulated genes and in the background set of genes (see Methods). Noise was added to simulate a more realistic scenario, in which only some of the genes in the co-regulated set, do share a common regulatory motif in their promoters. The noise free case (black continuous line in Figure 6-2) consisted of the 10 motif-positive promoters assigned to the co-regulated set of genes, and the null promoters (with no insertions) assigned to the background set.

Promoters in the background set were progressively misassigned to the co-regulated set, and the corresponding performances are shown in Figure 6-2.

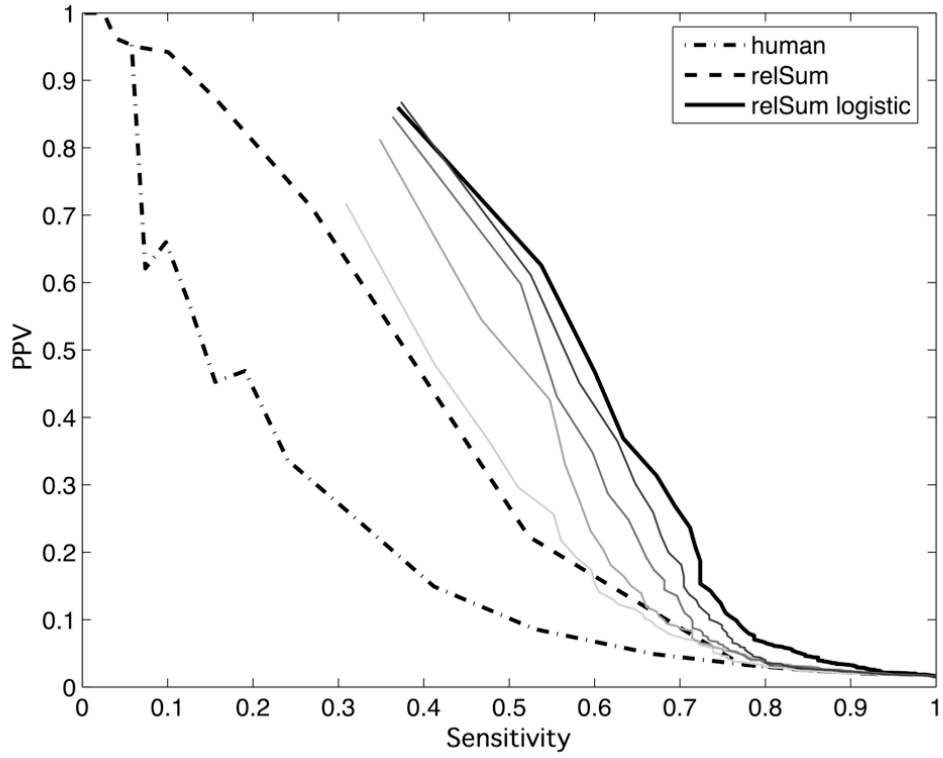


Figure 6-2: Positive Predictive Value (PPV) *vs.* Sensitivity plot showing the results in the simulated dataset. Continuous lines: performance profile obtained using the logistic regression step (black thick line shows performance with zero noise, and thin gray scale lines show performance when miss-assignmets are progressively introduced).

6.4.2 TRANSFAC genes dataset

The TRANSFAC dataset consists of promoters of 407 human genes from TRANSFAC gene table, for which transcription factors are known and experimentally validated with an annotated 5'-UTR. Ortholog gene sequences were fetched via the automated workflow, for each of 9 species where available.

The analysis was limited to the subset of 161 groups of ortholog genes for which all 9 orthologs were available, for a total of 1449 promoter sequences. All promoters were 1kb long, with 300bp downstream of the transcript start site.

Results on the human TRANSFAC genes dataset confirm the results obtained on the simulated dataset. Single species performance appears to resemble the evolutionary distance of the species (Figure 6-3). The PPV reached a maximum of approximately 30% when the ortholog gene promoter sequences are used, as compared to an average peak of <20% for the human species alone. We also compared the performance of CFBI with one of the most commonly used algorithms for TFBS prediction, MATCH (Kel et al., 2003) using both the 'minimize FP' and 'minimize FN' options (Figure 6-3).

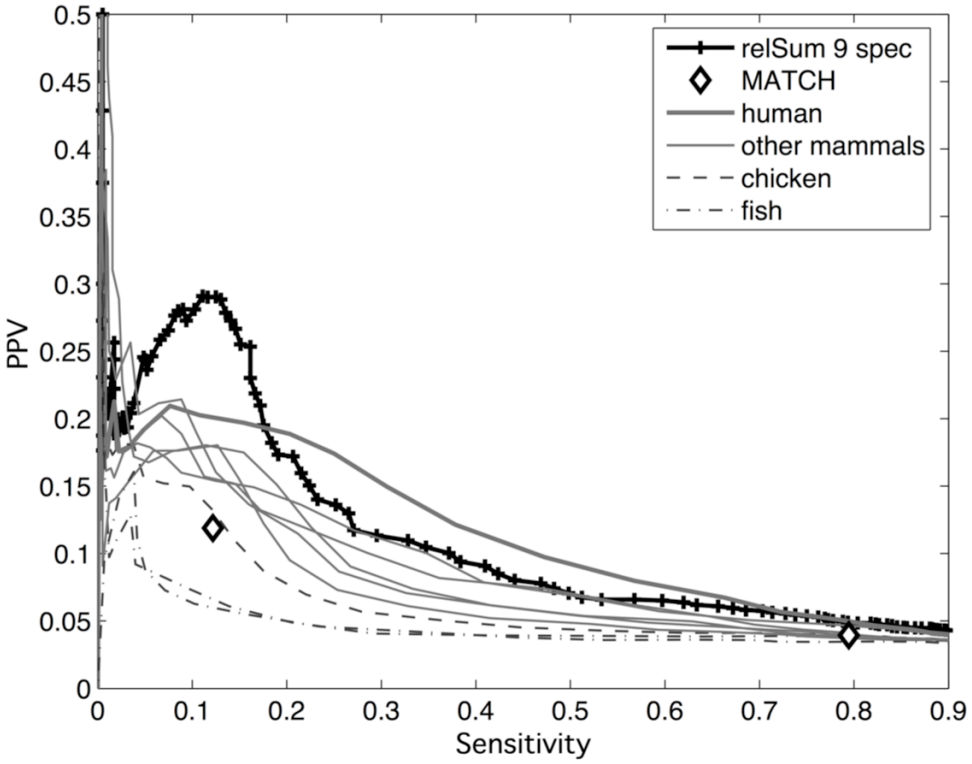


Figure 6-3: PPV *vs.* Sensitivity on the TRANSFAC genes dataset. Plain gray lines: scores obtained on the individual species; continuous lines: mammals (the thicker line is the human); dashed lines: chicken; dot dashed: fugu and zebrafish. Performance obtained using MATCH: two bordered white diamonds correspond to 'minimize false positives' and 'minimize false negatives'

6.4.3 Myc targets dataset

In order to confirm our results on an independent dataset, we selected a subset of Myc target genes from the Myc database (Basso et al., 2005). The Myc gene a transcription factor vastly implicated in neuroscience (Knoepfler et al., 2002; Pession and Tonelli, 2005; West et al., 2004), whose primary targets have been extensively validated. Only the top 17 high quality targets were included in the analysis, *i.e.* those validated as primary targets by both Chromatin

ImmunoPrecipitation (ChIP) and biochemical assays, in order to have a small but highly reliable dataset (Basso et al., 2005).

Performance on this dataset confirms the advantage of integrating phylogenetic sequence information over using a single species, and a boost (> two fold) in performance when integrating information on co-regulated genes via logistic regression (Figure 6-4).

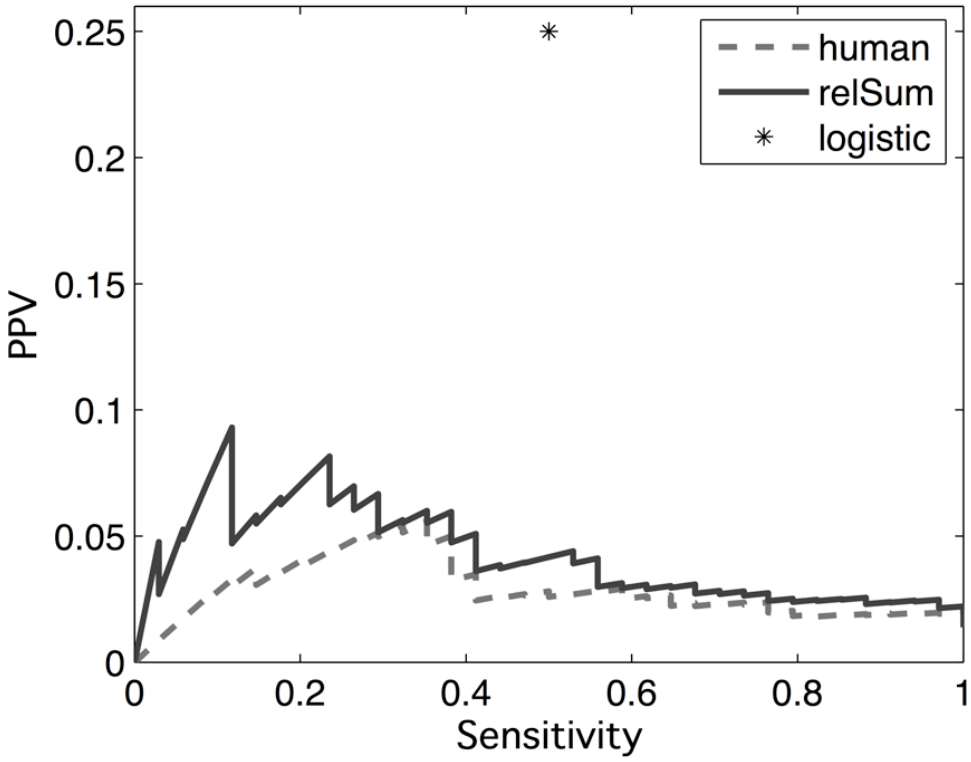


Figure 6-4: PPV *vs.* Sensitivity on the small set of 'high quality' Myc target genes dataset. Continuous line: performance of the weighted sum over 9 species; dashed line: human alone. The asterisk shows the peak performance obtained by the logistic of the 17 *relSum* scores against 100 promoters of random genes not included in the Myc database

6.5 Homer 1 promoter analysis

The promoter sequence 2kbp + 500bp within 5'-UTR upstream of the rat Homer 1 gene (chr2:23580396-23582896, NCBI build 36) was retrieved from UCSC Genome Browser and analyzed through our computational framework. Because the annotation of Homer 1 is missing in the current build of the rat genome, the precise transcription start site was identify by the alignment of human Homer 1 gene (NM_004272.3). The Top predictions based on a *z* score greater than 3 are shown in Table 6-2. Remarkably, the top hit is the CREB binding site, which is indeed expected because of the involvement of the MAP kinase in Homer 1a induction (Sato et al., 2001), and confirms a previous bioinformatics analysis by Bottai et al (Bottai et al., 2002).

Factor	postition	*z score*
CREB	-627	5.622147
TAXCREB	-629	4.995546
CREB	-626	4.931094
AP2	-266	4.809325
CMAF	-631	4.6919475
TAXCREB	-655	4.4109797
CREBATF	-650	4.227764
CREB	-667	4.1864557
FOXP1	-1976	4.1154675
SP1	-316	4.0611258
SP1	-311	4.0611258
SP1	-306	4.0611258
KROX	-81	4.017264
KROX	-304	3.9865954
CREBATF	-624	3.9597793
SP1	-312	3.953564
CREBATF	-666	3.9340138
E4F1	-650	3.9250305
ATF	-626	3.9170463
UF1H3BETA	-409	3.9129333
SP1	-307	3.9102688
ATF1	-665	3.901047
CREB	-627	3.7579746
CDPCR1	257	3.6762116
ELK1	-702	3.619494
KROX	-365	3.6090372
CREB	-689	3.5926433
R	-756	3.5275745
VJUN	-691	3.4715352
ATF	-652	3.4485972
CDPCR3HD	257	3.350472
EVI1	-1841	3.3475976
SP1	-315	3.2414412
SP1	-310	3.2414412
SP1	-305	3.163656
SP1	-351	3.037524

Table 6-2: Top predicted Transcription factor binding sites predicted by King Algorithm. Positions are shown relative to the transcription start site and the ranking based on *z* scores.

Chapter 7

Conclusions

7.1 Homer 1 as 'atypicality' predictor

Findings from our previous experiments on the differential regulation of *Homer 1a* by typical and atypical antipsychotics (de Bartolomeis et al., 2002) have led us to propose that measuring the increase of *Homer 1a* in caudate-putamen in rodents can be used to discriminate typical from atypical antipsychotics. Specifically, our previous results on the differential expression of haloperidol *vs.* atypical antipsychotics olanzapine and clozapine (de Bartolomeis et al., 2002; Polese et al., 2002) suggested that typical antipsychotics with high D2R affinity strongly induce an overexpression of *Homer 1a* in subcortical regions of the rat

brain after acute treatment. We investigated the response of *Homer 1* gene to antipsychotics with a acute and chronic administration of other antipsychotics, including Quetiapine and Ziprasidone. The chronic administration of antipsychotics to animals was used to resemble the clinical treatment regimens required for antipsychotic effects to be observed in humans. The results presented in this manuscript confirm the strong induction of *Homer 1a* gene expression in rat caudate-putamen after acute administration of the typical antipsychotic haloperidol and show a lack of induction by the atypical quetiapine (Figure 4-3a and Figure 4-3c) and ziprasidone 4mg/kg, consistent with the previously reported *Homer 1a* induction patterns in the rat striatum (de Bartolomeis et al., 2002). The atypical agents Quetiapine and Ziprasidone only induced showed an induction for Homer 1 in nucleus accumbens suggesting selectivitiy for limbic structures.

The increase of *ania-3* in nucleus accumbens but not in caudate-putamen, is consistent with a prominent effect on limbic regions by quetiapine, while sparing the nigrostriatal pathway that is implicated in EPSEs (Tada et al., 2004; Westerink, 2002). The differences in expression patterns found between the two splice variants of *Homer 1* gene may suggest a differential regulation with a neuroanatomical specificity. However a simpler explanation for those differences could be that the signal-to-noise ratio for *ania-3* is higher than for *Homer 1a*. Thus, further investigation is needed to conclusively determine whether the two variants are indeed differentially expressed. Should this be confirmed it would be interesting and challenging to pin down the mechanisms involved in such differential regulation.

Moreover, the differential pattern of homer expression may provide a powerful molecular tool for further investigation into the differential molecular mechanisms of typical and atypical antipsychotics, as well as a potential predictor of 'atypicality' for putative novel antipsychotic agents. Finally, our results confirm that antipsychotic compounds acting prevalently at the dopamine receptors can perturb homer 1a, a relevant effector of glutamatergic signaling, which, differently from other early genes such as c-fos, has a direct role in synaptic plasticity.

The other dopaminergic agent used in our studies GBR, which shares the mechanism of action with cocaine, induced *ania-3* by GBR after acute treatment in brain regions involved in rewarding effects of stimulant drugs. Most remarkably, in the cortical regions both *Homer 1a* (figure 5a) and *ania-3* (Figure 5c) showed a strong induction by chronic GBR limited to the parietal cortex, both statistically significant ($p<.0001$). This induction in the parietal cortex was absent or not statistically significant after the acute treatment and could be related to the recruitment of 'sensitization' specific neuronal networks by stimulants after a pretreatment schedule, as previously suggested by Curran and coworkers (Curran et al., 1996).. The induction of *Homer 1* in somatosensory cortex after repeated GBR administration might be involved in a compensatory blunting of cortical activity as was shown to occur after *Homer 1a* overexpression by viral vector infusion in the frontal cortex of rats (Lominac et al., 2005).

7.2 Computational predictions

We have developed a novel strategy for increasing the accuracy of computational predictions of TFBSs on genomic DNA sequences. Key factors of our computational framework include the integration of phylogenetic information from multiple species, and the possibility to include a priori information such as that available from quantitative or qualitative gene expression data.

Regulation of gene expression is a key factor determining complexity of biological systems. There is an increasing interest in understanding regulation of gene expression in the brain, where the dynamics of gene expression may play a role in drug response and in brain disorders. There are examples in which neural gene expression profiles could accurately discriminate among classes of psychoactive compounds (Gunther et al., 2003; Gunther et al., 2005) or even between complex social behaviors within honeybees (Whitfield et al., 2003).

Here, we developed a novel strategy for increasing the accuracy of computational predictions of TFBSs on genomic DNA sequences. Key factors of our computational framework include the integration of phylogenetic information from multiple species, and the possibility to include a priori information such as that available from quantitative or qualitative gene expression data.

One novelty of our computational approach, compared to others that make use of phylogenetic information, is that it does not require aligning promoter sequences from different species, thus overcoming the problem of aligning promoter sequences that have diverged with evolution.

A second novelty is the use of non-linear logistic regression to integrate additional *a priori* information on gene regulation. The source of *a priori* information could be microarray gene expression profiles. Clusters of genes that share a common expression profile with a gene of interest can be identified, and considered against a set of genes that do not change. The hypothesis is that genes that are co-expressed should be co-regulated and therefore share common regulatory motifs in their promoters, while the second set of non-changing genes is used as a background set to reduce false positives. Alternatively, contrasting sets of genes could be identified from biological knowledge or from different experimental data such as a specific pattern of expression by *in situ* hybridization. For example, a pattern of expression in specific neuroanatomical regions in response to a drug may be used to select one group of genes, whereas a (larger) set of genes not responding, or responding with a different pattern may be used as the background set. Logistic regression is different from the linear regression method by Conlon *et al*. (Tadesse et al., 2004), in that the linear regression model relies on the assumption that the gene expression levels are linearly related to the sequence matching scores of the motifs. Such an effect could be true in lower animals but is not easy to detect in mammals. In addition, the use a background set makes logistic regression less prone to false positive predictions.

Our results on *Homer 1* gene regulation by antipsychotics have stimulated us to the investigation of the molecular mechanisms involved in such a regulation. Sequence analysis of *Homer 1* promoter was perfomed implementing our computational framework for the prediction of TFBSs after it had been validated on simulated datasets as well as on 'real' datasets. Accurate predictions of direct

drug targets through combined analysis of genomic expression data are promising in the drug discovery process, and the investigation of mechanisms of action of compounds.

7.3 Future directions

Our investigation on the identification of drug targets antipsychotics and gene networks directly perturbed by such drugs is still an ongoing process that will require experimental validation of predicted targets and analysis of *Homer 1a* co-regulated genes. Our future goals are to implement the computational framework on promoters of all genes that are expressed in the brain and find correlations based on the patterns top scoring predicted TFBSs, in order to recover a functional network of genes that are expressed in the brain, and that are the direct targets of antipsychotic drug treatments. Predictions will be integrated with gene expression data obtained by *in situ* hybridization or microarray technology and will be validated experimentally.

Bibliography

Ambesi-Impiombato A, Bansal M, Lio P, di Bernardo D. 2006. Computational framework for the prediction of transcription factor binding sites by multiple data integration. BMC Neurosci 7 Suppl 1:S8.

Ambesi-Impiombato A, D'Urso G, Muscettola G, de Bartolomeis A. 2003. Method for quantitative in situ hybridization histochemistry and image analysis applied for Homer1a gene expression in rat brain. Brain Res Brain Res Protoc 11(3):189-196.

Ambesi-Impiombato A, Di Bernardo D. 2006. Computational Biology and Drug Discovery: From Single-Target to Network Drugs. Current Bioinformatics 1(1):3-13.

Angulo JA, Cadet JL, McEwen BS. 1990. Effect of typical and atypical neuroleptic treatment on protachykinin mRNA levels in the striatum of the rat. Neurosci Lett 113(2):217-221.

Apic G, Ignjatovic T, Boyer S, Russell RB. 2005. Illuminating drug discovery with biological pathways. FEBS Lett 579(8):1872-1877.

Arnt J. 1998. Pharmacological differentiation of classical and novel antipsychotics. Int Clin Psychopharmacol 13 Suppl 3:S7-14.

Austin MC, Schultzberg M, Abbott LC, Montpied P, Evers JR, Paul SM, Crawley JN. 1992. Expression of tyrosine hydroxylase in cerebellar Purkinje neurons of the mutant tottering and leaner mouse. Brain Res Mol Brain Res 15(3-4):227-240.

Baetz K, McHardy L, Gable K, Tarling T, Reberioux D, Bryan J, Andersen RJ, Dunn T, Hieter P, Roberge M. 2004. Yeast genome-wide drug-induced haploinsufficiency screen to determine drug mode of action. Proc Natl Acad Sci U S A 101(13):4525-4530.

Bailey TL, Elkan C. 1994. Fitting a mixture model by expectation maximization to discover motifs in biopolymers. Proc Int Conf Intell Syst Mol Biol 2:28-36.

Bao L, Guo T, Sun Z. 2002. Mining functional relationships in feature subspaces from gene expression profiles and drug activity profiles. FEBS Lett 516(1-3):113-118.

Basso K, Margolin AA, Stolovitzky G, Klein U, Dalla-Favera R, Califano A. 2005. Reverse engineering of regulatory networks in human B cells. Nat Genet 37(4):382-390.

Beasley CM, Jr., Tollefson G, Tran P, Satterlee W, Sanger T, Hamilton S. 1996. Olanzapine versus placebo and haloperidol: acute phase results of the

North American double-blind olanzapine trial. Neuropsychopharmacology 14(2):111-123.
Berke JD, Paletzki RF, Aronson GJ, Hyman SE, Gerfen CR. 1998. A complex program of striatal gene expression induced by dopaminergic stimulation. J Neurosci 18(14):5301-5310.
Borison RL, Arvanitis LA, Miller BG. 1996. ICI 204,636, an atypical antipsychotic: efficacy and safety in a multicenter, placebo-controlled trial in patients with schizophrenia. U.S. SEROQUEL Study Group. J Clin Psychopharmacol 16(2):158-169.
Boscia F, Gala R, Pignataro G, de Bartolomeis A, Cicale M, Ambesi-Impiombato A, Di Renzo G, Annunziato L. 2006. Permanent focal brain ischemia induces isoform-dependent changes in the pattern of Na+/Ca2+ exchanger gene expression in the ischemic core, periinfarct area, and intact brain regions. J Cereb Blood Flow Metab 26(4):502-517.
Bottai D, Guzowski JF, Schwarz MK, Kang SH, Xiao B, Lanahan A, Worley PF, Seeburg PH. 2002. Synaptic activity-induced conversion of intronic to exonic sequence in Homer 1 immediate early gene expression. J Neurosci 22(1):167-175.
Boyson SJ, McGonigle P, Molinoff PB. 1986. Quantitative autoradiographic localization of the D1 and D2 subtypes of dopamine receptors in rat brain. J Neurosci 6(11):3177-3188.
Brakeman PR, Lanahan AA, O'Brien R, Roche K, Barnes CA, Huganir RL, Worley PF. 1997. Homer: a protein that selectively binds metabotropic glutamate receptors. Nature 386(6622):284-288.
Bredel M, Jacoby E. 2004. Chemogenomics: an emerging strategy for rapid target and drug discovery. Nat Rev Genet 5(4):262-275.
Brent R. 2004. A partnership between biology and engineering. Nat Biotechnol 22(10):1211-1214.
Brown MP, Grundy WN, Lin D, Cristianini N, Sugnet CW, Furey TS, Ares M, Jr., Haussler D. 2000. Knowledge-based analysis of microarray gene expression data by using support vector machines. Proc Natl Acad Sci U S A 97(1):262-267.
Bulyk ML, McGuire AM, Masuda N, Church GM. 2004. A motif co-occurrence approach for genome-wide prediction of transcription-factor-binding sites in Escherichia coli. Genome Res 14(2):201-208.
Bussemaker HJ, Li H, Siggia ED. 2001. Regulatory element detection using correlation with expression. Nat Genet 27(2):167-171.
Butcher EC, Berg EL, Kunkel EJ. 2004. Systems biology in drug discovery. Nat Biotechnol 22(10):1253-1259.
Bymaster FP, Felder CC, Tzavara E, Nomikos GG, Calligaro DO, McKinzie DL. 2003. Muscarinic mechanisms of antipsychotic atypicality. Prog Neuropsychopharmacol Biol Psychiatry 27(7):1125-1143.
Chen ML, Chen CH. 2005. Microarray analysis of differentially expressed genes in rat frontal cortex under chronic risperidone treatment. Neuropsychopharmacology 30(2):268-277.
Cochran SM, McKerchar CE, Morris BJ, Pratt JA. 2002. Induction of differential patterns of local cerebral glucose metabolism and immediate-early genes by acute clozapine and haloperidol. Neuropharmacology 43(3):394-407.

Conlon EM, Liu XS, Lieb JD, Liu JS. 2003. Integrating regulatory motif discovery and genome-wide expression analysis. Proc Natl Acad Sci U S A 100(6):3339-3344.
Cosma MP, Pepe S, Annunziata I, Newbold RF, Grompe M, Parenti G, Ballabio A. 2003. The multiple sulfatase deficiency gene encodes an essential and limiting factor for the activity of sulfatases. Cell 113(4):445-456.
Csermely P, Agoston V, Pongor S. 2005. The efficiency of multi-target drugs: the network approach might help drug design. Trends Pharmacol Sci 26(4):178-182.
Curran EJ, Akil H, Watson SJ. 1996. Psychomotor stimulant- and opiate-induced c-fos mRNA expression patterns in the rat forebrain: comparisons between acute drug treatment and a drug challenge in sensitized animals. Neurochem Res 21(11):1425-1435.
Dahl JP, Kampman KM, Oslin DW, Weller AE, Lohoff FW, Ferraro TN, O'Brien CP, Berrettini WH. 2005. Association of a polymorphism in the Homer1 gene with cocaine dependence in an African American population. Psychiatr Genet 15(4):277-283.
Dan S, Tsunoda T, Kitahara O, Yanagawa R, Zembutsu H, Katagiri T, Yamazaki K, Nakamura Y, Yamori T. 2002. An integrated database of chemosensitivity to 55 anticancer drugs and gene expression profiles of 39 human cancer cell lines. Cancer Res 62(4):1139-1147.
Dawson TM, Gehlert DR, McCabe RT, Barnett A, Wamsley JK. 1986. D-1 dopamine receptors in the rat brain: a quantitative autoradiographic analysis. J Neurosci 6(8):2352-2365.
de Bartolomeis A, Aloj L, Ambesi-Impiombato A, Bravi D, Caraco C, Muscettola G, Barone P. 2002. Acute administration of antipsychotics modulates Homer striatal gene expression differentially. Brain Res Mol Brain Res 98(1-2):124-129.
de Bartolomeis A, Iasevoli F. 2003. The Homer family and the signal transduction system at glutamatergic postsynaptic density: potential role in behavior and pharmacotherapy. Psychopharmacol Bull 37(3):51-83.
De Jong H, Gouze JL, Hernandez C, Page M, Sari T, Geiselmann J. 2004. Qualitative simulation of genetic regulatory networks using piecewise-linear models. Bull Math Biol 66(2):301-340.
Di Bernardo D, Thompson MJ, Gardner TS, Chobot SE, Eastwood EL, Wojtovich AP, Elliott SJ, Schaus SE, Collins JJ. 2005. Chemogenomic profiling on a genome-wide scale using reverse-engineered gene networks. Nat Biotechnol 23(3):377-383.
DiMasi JA, Hansen RW, Grabowski HG. 2003. The price of innovation: new estimates of drug development costs. J Health Econ 22(2):151-185.
Eisen MB, Spellman PT, Brown PO, Botstein D. 1998. Cluster analysis and display of genome-wide expression patterns. Proc Natl Acad Sci U S A 95(25):14863-14868.
Eskin E, Pevzner PA. 2002. Finding composite regulatory patterns in DNA sequences. Bioinformatics 18 Suppl 1:S354-363.
Fagan R, Swindells M. 2000. Bioinformatics, target discovery and the pharmaceutical/biotechnology industry. Curr Opin Mol Ther 2(6):655-661.

Feher LZ, Kalman J, Puskas LG, Gyulveszi G, Kitajka K, Penke B, Palotas M, Samarova EI, Molnar J, Zvara A, Matin K, Bodi N, Hugyecz M, Pakaski M, Bjelik A, Juhasz A, Bogats G, Janka Z, Palotas A. 2005. Impact of haloperidol and risperidone on gene expression profile in the rat cortex. Neurochem Int 47(4):271-280.

Felsenstein J. 1989. PHYLIP - Phylogeny Inference Package (Version 3.2). Cladistics 5:164-166.

Fujibuchi W, Anderson JS, Landsman D. 2001. PROSPECT improves cis-acting regulatory element prediction by integrating expression profile data with consensus pattern searches. Nucleic Acids Res 29(19):3988-3996.

Gardner TS, di Bernardo D, Lorenz D, Collins JJ. 2003. Inferring genetic networks and identifying compound mode of action via expression profiling. Science 301(5629):102-105.

Gardner TS, Faith JJ. 2005. Reverse-engineering transcription control networks. Physics of Life Reviews 2:65-88.

Gasch AP, Spellman PT, Kao CM, Carmel-Harel O, Eisen MB, Storz G, Botstein D, Brown PO. 2000. Genomic expression programs in the response of yeast cells to environmental changes. Mol Biol Cell 11(12):4241-4257.

Giaever G, Flaherty P, Kumm J, Proctor M, Nislow C, Jaramillo DF, Chu AM, Jordan MI, Arkin AP, Davis RW. 2004. Chemogenomic profiling: identifying the functional interactions of small molecules in yeast. Proc Natl Acad Sci U S A 101(3):793-798.

Giaever G, Shoemaker DD, Jones TW, Liang H, Winzeler EA, Astromoff A, Davis RW. 1999. Genomic profiling of drug sensitivities via induced haploinsufficiency. Nat Genet 21(3):278-283.

Giuffrida R, Musumeci S, D'Antoni S, Bonaccorso CM, Giuffrida-Stella AM, Oostra BA, Catania MV. 2005. A reduced number of metabotropic glutamate subtype 5 receptors are associated with constitutive homer proteins in a mouse model of fragile X syndrome. J Neurosci 25(39):8908-8916.

Gunther EC, Stone DJ, Gerwien RW, Bento P, Heyes MP. 2003. Prediction of clinical drug efficacy by classification of drug-induced genomic expression profiles in vitro. Proc Natl Acad Sci U S A 100(16):9608-9613.

Gunther EC, Stone DJ, Rothberg JM, Gerwien RW. 2005. A quantitative genomic expression analysis platform for multiplexed in vitro prediction of drug action. Pharmacogenomics J 5(2):126-134.

Haggarty SJ, Clemons PA, Schreiber SL. 2003. Chemical genomic profiling of biological networks using graph theory and combinations of small molecule perturbations. J Am Chem Soc 125(35):10543-10545.

Hallikas O, Palin K, Sinjushina N, Rautiainen R, Partanen J, Ukkonen E, Taipale J. 2006. Genome-wide prediction of mammalian enhancers based on analysis of transcription-factor binding affinity. Cell 124(1):47-59.

Hamadeh HK, Bushel PR, Jayadev S, DiSorbo O, Bennett L, Li L, Tennant R, Stoll R, Barrett JC, Paules RS, Blanchard K, Afshari CA. 2002a. Prediction of compound signature using high density gene expression profiling. Toxicol Sci 67(2):232-240.

Hamadeh HK, Bushel PR, Jayadev S, Martin K, DiSorbo O, Sieber S, Bennett L, Tennant R, Stoll R, Barrett JC, Blanchard K, Paules RS, Afshari CA.

2002b. Gene expression analysis reveals chemical-specific profiles. Toxicol Sci 67(2):219-231.
Hart CP. 2005. Finding the target after screening the phenotype. Drug Discov Today 10(7):513-519.
Hartigan JA. 1975. Clustering Algorithms. New York: John Wiley & Sons.
Hasegawa M, Kishino H, Yano T. 1985. Dating of the human-ape splitting by a molecular clock of mitochondrial DNA. J Mol Evol 22(2):160-174.
Hastie T, Tibshirani R, Friedman JH. 2001. The Elements of Statistical Learning. New York, NY: Springer.
Hennou S, Kato A, Schneider EM, Lundstrom K, Gahwiler BH, Inokuchi K, Gerber U, Ehrengruber MU. 2003. Homer-1a/Vesl-1S enhances hippocampal synaptic transmission. Eur J Neurosci 18(4):811-819.
Hong EJ, West AE, Greenberg ME. 2005. Transcriptional control of cognitive development. Curr Opin Neurobiol 15(1):21-28.
Hubbard T, Andrews D, Caccamo M, Cameron G, Chen Y, Clamp M, Clarke L, Coates G, Cox T, Cunningham F, Curwen V, Cutts T, Down T, Durbin R, Fernandez-Suarez XM, Gilbert J, Hammond M, Herrero J, Hotz H, Howe K, Iyer V, Jekosch K, Kahari A, Kasprzyk A, Keefe D, Keenan S, Kokocinsci F, London D, Longden I, McVicker G, Melsopp C, Meidl P, Potter S, Proctor G, Rae M, Rios D, Schuster M, Searle S, Severin J, Slater G, Smedley D, Smith J, Spooner W, Stabenau A, Stalker J, Storey R, Trevanion S, Ureta-Vidal A, Vogel J, White S, Woodwark C, Birney E. 2005. Ensembl 2005. Nucleic Acids Res 33(Database issue):D447-453.
Hughes JD, Estep PW, Tavazoie S, Church GM. 2000. Computational identification of cis-regulatory elements associated with groups of functionally related genes in Saccharomyces cerevisiae. J Mol Biol 296(5):1205-1214.
Imoto S, Savoie CJ, Aburatani S, Kim S, Tashiro K, Kuhara S, Miyano S. 2003. Use of gene networks for identifying and validating drug targets. J Bioinform Comput Biol 1(3):459-474.
Inoue Y, Honkura N, Kato A, Ogawa S, Udo H, Inokuchi K, Sugiyama H. 2004. Activity-inducible protein Homer1a/Vesl-1S promotes redistribution of postsynaptic protein Homer1c/Vesl-1L in cultured rat hippocampal neurons. Neurosci Lett 354(2):143-147.
Kalivas PW, Szumlinski KK, Worley P. 2004. Homer2 gene deletion in mice produces a phenotype similar to chronic cocaine treated rats. Neurotox Res 6(5):385-387.
Kapur S, Seeman P. 2001. Does fast dissociation from the dopamine d(2) receptor explain the action of atypical antipsychotics? A new hypothesis. Am J Psychiatry 158(3):360-369.
Kapur S, Zipursky R, Jones C, Remington G, Houle S. 2000a. Relationship between dopamine D(2) occupancy, clinical response, and side effects: a double-blind PET study of first-episode schizophrenia. Am J Psychiatry 157(4):514-520.
Kapur S, Zipursky R, Jones C, Shammi CS, Remington G, Seeman P. 2000b. A positron emission tomography study of quetiapine in schizophrenia: a preliminary finding of an antipsychotic effect with only transiently high dopamine D2 receptor occupancy. Arch Gen Psychiatry 57(6):553-559.

Kel AE, Gossling E, Reuter I, Cheremushkin E, Kel-Margoulis OV, Wingender E. 2003. MATCH: A tool for searching transcription factor binding sites in DNA sequences. Nucleic Acids Res 31(13):3576-3579.

Knoepfler PS, Cheng PF, Eisenman RN. 2002. N-myc is essential during neurogenesis for the rapid expansion of progenitor cell populations and the inhibition of neuronal differentiation. Genes Dev 16(20):2699-2712.

Lane EL, Cheetham S, Jenner P. 2005. Dopamine uptake inhibitor-induced rotation in 6-hydroxydopamine-lesioned rats involves both D1 and D2 receptors but is modulated through 5-hydroxytryptamine and noradrenaline receptors. J Pharmacol Exp Ther 312(3):1124-1131.

Li L, Darden TA, Weinberg CR, Levine AJ, Pedersen LG. 2001. Gene assessment and sample classification for gene expression data using a genetic algorithm/k-nearest neighbor method. Comb Chem High Throughput Screen 4(8):727-739.

Lieberman JA, Stroup TS, McEvoy JP, Swartz MS, Rosenheck RA, Perkins DO, Keefe RS, Davis SM, Davis CE, Lebowitz BD, Severe J, Hsiao JK. 2005. Effectiveness of antipsychotic drugs in patients with chronic schizophrenia. N Engl J Med 353(12):1209-1223.

Liu XS, Brutlag DL, Liu JS. 2002. An algorithm for finding protein-DNA binding sites with applications to chromatin-immunoprecipitation microarray experiments. Nat Biotechnol 20(8):835-839.

Lominac KD, Oleson EB, Pava M, Klugmann M, Schwarz MK, Seeburg PH, During MJ, Worley PF, Kalivas PW, Szumlinski KK. 2005. Distinct roles for different Homer1 isoforms in behaviors and associated prefrontal cortex function. J Neurosci 25(50):11586-11594.

Ludwig MZ. 2002. Functional evolution of noncoding DNA. Curr Opin Genet Dev 12(6):634-639.

Lum PY, Armour CD, Stepaniants SB, Cavet G, Wolf MK, Butler JS, Hinshaw JC, Garnier P, Prestwich GD, Leonardson A, Garrett-Engele P, Rush CM, Bard M, Schimmack G, Phillips JW, Roberts CJ, Shoemaker DD. 2004. Discovering modes of action for therapeutic compounds using a genome-wide screen of yeast heterozygotes. Cell 116(1):121-137.

MacGibbon GA, Lawlor PA, Bravo R, Dragunow M. 1994. Clozapine and haloperidol produce a differential pattern of immediate early gene expression in rat caudate-putamen, nucleus accumbens, lateral septum and islands of Calleja. Brain Res Mol Brain Res 23(1-2):21-32.

Maki Y, Tominaga D, Okamoto M, Watanabe S, Eguchi Y. 2001. Development of a system for the inference of large scale genetic networks. Pac Symp Biocomput:446-458.

Mamo D, Kapur S, Shammi CM, Papatheodorou G, Mann S, Therrien F, Remington G. 2004. A PET study of dopamine D2 and serotonin 5-HT2 receptor occupancy in patients with schizophrenia treated with therapeutic doses of ziprasidone. Am J Psychiatry 161(5):818-825.

Marton MJ, DeRisi JL, Bennett HA, Iyer VR, Meyer MR, Roberts CJ, Stoughton R, Burchard J, Slade D, Dai H, Bassett DE, Jr., Hartwell LH, Brown PO, Friend SH. 1998. Drug target validation and identification of secondary drug target effects using DNA microarrays. Nat Med 4(11):1293-1301.

Meltzer HY, Zhang Y, Stockmeier CA. 1992. Effect of amperozide on rat cortical 5-HT2 and striatal and limbic dopamine D2 receptor occupancy: implications for antipsychotic action. Eur J Pharmacol 216(1):67-71.
Merchant KM, Dorsa DM. 1993. Differential induction of neurotensin and c-fos gene expression by typical versus atypical antipsychotics. Proc Natl Acad Sci U S A 90(8):3447-3451.
Minami I, Kengaku M, Smitt PS, Shigemoto R, Hirano T. 2003. Long-term potentiation of mGluR1 activity by depolarization-induced Homer1a in mouse cerebellar Purkinje neurons. Eur J Neurosci 17(5):1023-1032.
Mnaimneh S, Davierwala AP, Haynes J, Moffat J, Peng WT, Zhang W, Yang X, Pootoolal J, Chua G, Lopez A, Trochesset M, Morse D, Krogan NJ, Hiley SL, Li Z, Morris Q, Grigull J, Mitsakakis N, Roberts CJ, Greenblatt JF, Boone C, Kaiser CA, Andrews BJ, Hughes TR. 2004. Exploration of essential gene functions via titratable promoter alleles. Cell 118(1):31-44.
Morgan JI, Curran T. 1991. Stimulus-transcription coupling in the nervous system: involvement of the inducible proto-oncogenes fos and jun. Annu Rev Neurosci 14:421-451.
Naisbitt S, Kim E, Tu JC, Xiao B, Sala C, Valtschanoff J, Weinberg RJ, Worley PF, Sheng M. 1999. Shank, a novel family of postsynaptic density proteins that binds to the NMDA receptor/PSD-95/GKAP complex and cortactin. Neuron 23(3):569-582.
Nasrallah HA, Newcomer JW. 2004. Atypical antipsychotics and metabolic dysregulation: evaluating the risk/benefit equation and improving the standard of care. J Clin Psychopharmacol 24(5 Suppl 1):S7-14.
Nemeroff CB, Kinkead B, Goldstein J. 2002. Quetiapine: preclinical studies, pharmacokinetics, drug interactions, and dosing. J Clin Psychiatry 63 Suppl 13:5-11.
Norton N, Williams HJ, Williams NM, Spurlock G, Zammit S, Jones G, Jones S, Owen R, O'Donovan MC, Owen MJ. 2003. Mutation screening of the Homer gene family and association analysis in schizophrenia. Am J Med Genet B Neuropsychiatr Genet 120(1):18-21.
Palin K, Ukkonen E, Brazma A, Vilo J. 2002. Correlating gene promoters and expression in gene disruption experiments. Bioinformatics 18 Suppl 2:S172-180.
Parsons AB, Brost RL, Ding H, Li Z, Zhang C, Sheikh B, Brown GW, Kane PM, Hughes TR, Boone C. 2004. Integration of chemical-genetic and genetic interaction data links bioactive compounds to cellular target pathways. Nat Biotechnol 22(1):62-69.
Paull KD, Shoemaker RH, Hodes L, Monks A, Scudiero DA, Rubinstein L, Plowman J, Boyd MR. 1989. Display and analysis of patterns of differential activity of drugs against human tumor cell lines: development of mean graph and COMPARE algorithm. J Natl Cancer Inst 81(14):1088-1092.
Paxinos G, Watson C. 1997. The Rat Brain Sereotaxic Coordiantes. New York, USA: AcademicPress.
Pennacchio LA, Rubin EM. 2001. Genomic strategies to identify mammalian regulatory sequences. Nat Rev Genet 2(2):100-109.

Perlman ZE, Slack MD, Feng Y, Mitchison TJ, Wu LF, Altschuler SJ. 2004. Multidimensional drug profiling by automated microscopy. Science 306(5699):1194-1198.
Persico AM, Schindler CW, O'Hara BF, Brannock MT, Uhl GR. 1993. Brain transcription factor expression: effects of acute and chronic amphetamine and injection stress. Brain Res Mol Brain Res 20(1-2):91-100.
Pession A, Tonelli R. 2005. The MYCN oncogene as a specific and selective drug target for peripheral and central nervous system tumors. Curr Cancer Drug Targets 5(4):273-283.
Pira L, Mongeau R, Pani L. 2004. The atypical antipsychotic quetiapine increases both noradrenaline and dopamine release in the rat prefrontal cortex. Eur J Pharmacol 504(1-2):61-64.
Polese D, de Serpis AA, Ambesi-Impiombato A, Muscettola G, de Bartolomeis A. 2002. Homer 1a gene expression modulation by antipsychotic drugs: involvement of the glutamate metabotropic system and effects of D-cycloserine. Neuropsychopharmacology 27(6):906-913.
Rambaut A, Grassly NC. 1997. Seq-Gen: an application for the Monte Carlo simulation of DNA sequence evolution along phylogenetic trees. Comput Appl Biosci 13(3):235-238.
Ratti E, Trist D. 2001. The continuing evolution of the drug discovery process in the pharmaceutical industry. Farmaco 56(1-2):13-19.
Robertson GS, Fibiger HC. 1992. Neuroleptics increase c-fos expression in the forebrain: contrasting effects of haloperidol and clozapine. Neuroscience 46(2):315-328.
Sala C, Futai K, Yamamoto K, Worley PF, Hayashi Y, Sheng M. 2003. Inhibition of dendritic spine morphogenesis and synaptic transmission by activity-inducible protein Homer1a. J Neurosci 23(15):6327-6337.
Sato M, Suzuki K, Nakanishi S. 2001. NMDA receptor stimulation and brain-derived neurotrophic factor upregulate homer 1a mRNA via the mitogen-activated protein kinase cascade in cultured cerebellar granule cells. J Neurosci 21(11):3797-3805.
Scatton B, Sanger DJ. 2000. Pharmacological and molecular targets in the search for novel antipsychotics. Behav Pharmacol 11(3-4):243-256.
Scherf U, Ross DT, Waltham M, Smith LH, Lee JK, Tanabe L, Kohn KW, Reinhold WC, Myers TG, Andrews DT, Scudiero DA, Eisen MB, Sausville EA, Pommier Y, Botstein D, Brown PO, Weinstein JN. 2000. A gene expression database for the molecular pharmacology of cancer. Nat Genet 24(3):236-244.
Seeger TF, Seymour PA, Schmidt AW, Zorn SH, Schulz DW, Lebel LA, McLean S, Guanowsky V, Howard HR, Lowe JA, 3rd, et al. 1995. Ziprasidone (CP-88,059): a new antipsychotic with combined dopamine and serotonin receptor antagonist activity. J Pharmacol Exp Ther 275(1):101-113.
Seeman P, Tallerico T. 1998. Antipsychotic drugs which elicit little or no parkinsonism bind more loosely than dopamine to brain D2 receptors, yet occupy high levels of these receptors. Mol Psychiatry 3(2):123-134.
Semba J, Sakai M, Miyoshi R, Mataga N, Fukamauchi F, Kito S. 1996. Differential expression of c-fos mRNA in rat prefrontal cortex, striatum,

N. accumbens and lateral septum after typical and atypical antipsychotics: an in situ hybridization study. Neurochem Int 29(4):435-442.
Staunton JE, Slonim DK, Coller HA, Tamayo P, Angelo MJ, Park J, Scherf U, Lee JK, Reinhold WO, Weinstein JN, Mesirov JP, Lander ES, Golub TR. 2001. Chemosensitivity prediction by transcriptional profiling. Proc Natl Acad Sci U S A 98(19):10787-10792.
Stegmaier K, Ross KN, Colavito SA, O'Malley S, Stockwell BR, Golub TR. 2004. Gene expression-based high-throughput screening(GE-HTS) and application to leukemia differentiation. Nat Genet 36(3):257-263.
Stimmel GL, Gutierrez MA, Lee V. 2002. Ziprasidone: an atypical antipsychotic drug for the treatment of schizophrenia. Clin Ther 24(1):21-37.
Stormo GD. 2000. DNA binding sites: representation and discovery. Bioinformatics 16(1):16-23.
Stormo GD, Fields DS. 1998. Specificity, free energy and information content in protein-DNA interactions. Trends Biochem Sci 23(3):109-113.
Stoughton RB, Friend SH. 2005. How molecular profiling could revolutionize drug discovery. Nat Rev Drug Discov 4(4):345-350.
Sudarsanam P, Pilpel Y, Church GM. 2002. Genome-wide co-occurrence of promoter elements reveals a cis-regulatory cassette of rRNA transcription motifs in Saccharomyces cerevisiae. Genome Res 12(11):1723-1731.
Swanson CJ, Baker DA, Carson D, Worley PF, Kalivas PW. 2001. Repeated cocaine administration attenuates group I metabotropic glutamate receptor-mediated glutamate release and behavioral activation: a potential role for Homer. J Neurosci 21(22):9043-9052.
Szakacs G, Annereau JP, Lababidi S, Shankavaram U, Arciello A, Bussey KJ, Reinhold W, Guo Y, Kruh GD, Reimers M, Weinstein JN, Gottesman MM. 2004. Predicting drug sensitivity and resistance: profiling ABC transporter genes in cancer cells. Cancer Cell 6(2):129-137.
Szumlinski KK, Abernathy KE, Oleson EB, Klugmann M, Lominac KD, He DY, Ron D, During M, Kalivas PW. 2006a. Homer isoforms differentially regulate cocaine-induced neuroplasticity. Neuropsychopharmacology 31(4):768-777.
Szumlinski KK, Kalivas PW, Worley PF. 2006b. Homer proteins: implications for neuropsychiatric disorders. Curr Opin Neurobiol 16(3):251-257.
Szumlinski KK, Lominac KD, Kleschen MJ, Oleson EB, Dehoff MH, Schwarz MK, Seeburg PH, Worley PF, Kalivas PW. 2005a. Behavioral and neurochemical phenotyping of Homer1 mutant mice: possible relevance to schizophrenia. Genes Brain Behav 4(5):273-288.
Szumlinski KK, Lominac KD, Oleson EB, Walker JK, Mason A, Dehoff MH, Klugmann M, Cagle S, Welt K, During M, Worley PF, Middaugh LD, Kalivas PW. 2005b. Homer2 is necessary for EtOH-induced neuroplasticity. J Neurosci 25(30):7054-7061.
Tada M, Shirakawa K, Matsuoka N, Mutoh S. 2004. Combined treatment of quetiapine with haloperidol in animal models of antipsychotic effect and extrapyramidal side effects: comparison with risperidone and chlorpromazine. Psychopharmacology (Berl) 176(1):94-100.
Tadesse MG, Vannucci M, Lio P. 2004. Identification of DNA regulatory motifs using Bayesian variable selection. Bioinformatics 20(16):2553-2561.

Tappe A, Kuner R. 2006. Regulation of motor performance and striatal function by synaptic scaffolding proteins of the Homer1 family. Proc Natl Acad Sci U S A 103(3):774-779.
Tauscher J, Hussain T, Agid O, Verhoeff NP, Wilson AA, Houle S, Remington G, Zipursky RB, Kapur S. 2004. Equivalent occupancy of dopamine D1 and D2 receptors with clozapine: differentiation from other atypical antipsychotics. Am J Psychiatry 161(9):1620-1625.
Tu JC, Xiao B, Yuan JP, Lanahan AA, Leoffert K, Li M, Linden DJ, Worley PF. 1998. Homer binds a novel proline-rich motif and links group 1 metabotropic glutamate receptors with IP3 receptors. Neuron 21(4):717-726.
Ueda M, Kinoshita H, Yoshida T, Kamasawa N, Osumi M, Tanaka A. 2003. Effect of catalase-specific inhibitor 3-amino-1,2,4-triazole on yeast peroxisomal catalase in vivo. FEMS Microbiol Lett 219(1):93-98.
Weinstein JN, Myers TG, O'Connor PM, Friend SH, Fornace AJ, Jr., Kohn KW, Fojo T, Bates SE, Rubinstein LV, Anderson NL, Buolamwini JK, van Osdol WW, Monks AP, Scudiero DA, Sausville EA, Zaharevitz DW, Bunow B, Viswanadhan VN, Johnson GS, Wittes RE, Paull KD. 1997. An information-intensive approach to the molecular pharmacology of cancer. Science 275(5298):343-349.
West AB, Kapatos G, O'Farrell C, Gonzalez-de-Chavez F, Chiu K, Farrer MJ, Maidment NT. 2004. N-myc regulates parkin expression. J Biol Chem 279(28):28896-28902.
Westerink BH. 2002. Can antipsychotic drugs be classified by their effects on a particular group of dopamine neurons in the brain? Eur J Pharmacol 455(1):1-18.
Whitfield CW, Cziko AM, Robinson GE. 2003. Gene expression profiles in the brain predict behavior in individual honey bees. Science 302(5643):296-299.
Wise SP, Murray EA, Gerfen CR. 1996. The frontal cortex-basal ganglia system in primates. Crit Rev Neurobiol 10(3-4):317-356.
Xiao B, Tu JC, Petralia RS, Yuan JP, Doan A, Breder CD, Ruggiero A, Lanahan AA, Wenthold RJ, Worley PF. 1998. Homer regulates the association of group 1 metabotropic glutamate receptors with multivalent complexes of homer-related, synaptic proteins. Neuron 21(4):707-716.
Yang L, Mao L, Tang Q, Samdani S, Liu Z, Wang JQ. 2004. A novel Ca2+-independent signaling pathway to extracellular signal-regulated protein kinase by coactivation of NMDA receptors and metabotropic glutamate receptor 5 in neurons. J Neurosci 24(48):10846-10857.
Yano M, Steiner H. 2005. Methylphenidate (Ritalin) induces Homer 1a and zif 268 expression in specific corticostriatal circuits. Neuroscience 132(3):855-865.
Young WS, 3rd, Bonner TI, Brann MR. 1986. Mesencephalic dopamine neurons regulate the expression of neuropeptide mRNAs in the rat forebrain. Proc Natl Acad Sci U S A 83(24):9827-9831.
Zhu Z, Shendure J, Church GM. 2005. Discovering functional transcription-factor combinations in the human cell cycle. Genome Res 15(6):848-855.

Appendix A

Java Code Examples

1.1 Package bioinfo

1.1.1 Bioinfo.MarkovModel Class

```
//
//  MarcovModel.java
//
//
//  Created by Alberto Ambesi on Tue Sep 02 2003.
//  Copyright (c) 2004 __Tigem__. All rights reserved.
//
package bioinfo;

import java.io.*;
import bioinfo.bio.*;
import java.util.*;
```

```
/*
 * Markov Model object
 */

public class MarkovModel implements Serializable {
     private int order; //order of Markov Model
     protected boolean useBothStrands;
     private boolean freqsComputed;
     protected LinkedHashMap tupleCountMap_f; // using simple base freq
model
     protected LinkedHashMap tupleCountMap_r;
     private int[] counter;
     private int[] voidCounter; //counts tuples with no occurrances in
training seqs.
     private static HashMap reverseMap = makeReverseMap();
     private boolean consolidated;
         private Vector speciesV; //not used: should be a single species,
whith get method

     /**
      * constructor using Markov Models
      * @param order order of Markov Model
      */
     public MarkovModel(boolean useBothStrands, int order) throws
Exception {
          this.order = order;
          //this.promoter = promoter;
          //To do: check order
          this.useBothStrands = useBothStrands;

          initialize();
     }
         /**
      * constructor using Markov Models
      * @param order order of Markov Model
      */
     public MarkovModel(boolean useBothStrands, int order, Vector
speciesV) throws Exception {
          this.order = order;
          this.speciesV = speciesV;
          //To do: check order
          this.useBothStrands = useBothStrands;

          initialize();
     }

     public static HashMap makeReverseMap() {
          reverseMap= new HashMap();
          reverseMap.put("a", "t");
          reverseMap.put("c", "g");
          reverseMap.put("g", "c");
          reverseMap.put("t", "a");
          return reverseMap;
     }
```

```
    /**
     * this test the code of markov model generation: it calcs the prob
of @param tuple
     */
    public void testMarkovModel(String tuple, int order) throws
Exception {
        long tStart = System.currentTimeMillis();
        System.out.println("probability of " + tuple + " using order " +
order + " = " + calcProbabilityOfTupleVerbose(tuple, true, order));
        try {
            System.out.println("frequency of   " + tuple + " = " +
calcMMFreq(tuple, true));//forwardStrand
        } catch(NullPointerException e) { //prints the exact frequency
only if available
        }
    }

    /**
     * @param string must be lowercase only a, c, g, t are allowed
     */
    private String reverseComplement(String s) { //throws Exception {
        String revComp = "";
        for (int i = s.length(); i>0; i--)
            revComp += (String)reverseMap.get(s.substring(i-1, i));
        return revComp;
    }

    private void initialize() throws Exception {
        //initializing map and counter[]
        freqsComputed = false;
        String[] base = new String[]{"a", "c", "g", "t"};
        tupleCountMap_f = new LinkedHashMap();
        Vector lastTempV = new Vector();
        lastTempV.addElement("");
        int w = order + 1;
        counter= new int[w];
        for (int j =0; j<w; j++) {
            Vector tempV = new Vector();
            for (Iterator k=lastTempV.iterator(); k.hasNext();) {
                String tuple = k.next().toString();
                for (int b=0; b<base.length; b++) {
                    tempV.addElement(tuple + base[b]);
                }
            }
            for (Iterator k=tempV.iterator(); k.hasNext();)
                tupleCountMap_f.put(k.next(), new Integer(0)); //adding
back to main map
            lastTempV = tempV;
            counter[j]=0;
        }
        tupleCountMap_r = (LinkedHashMap)tupleCountMap_f.clone();
    }


    /**
```

```
     * computes relative frequencies of all n-tuples with n {1:order}
     */
     public void addMMCounts(String seqStr) throws Exception {
        System.out.println("adding a sequence counts for this Markov
Model of order " + this.order + "...");
        int w = order + 1;
        int L = seqStr.length();
        seqStr = seqStr.toLowerCase();
        int illegalSymbolCounter =0;
        for (int i = 0; i < L; i++) { // iterating base by base a window
of w
             String s = "";
             try {
                           try {

                                  s = seqStr.substring(i, i + w);
                                  //System.out.println(s);
                           } catch (IndexOutOfBoundsException e2) {
//reaching end of sequence
                                  s = seqStr.substring((L - --w), L);
                                  //System.out.println("cought (index "
+ i + "): " + s);
                           }
                           //System.out.print(s.charAt(0));
                           for (int j = 1; j<=w; j++) {
                                  String jTupleSeqStrF = s.substring(0,
w-j+1);
                                  try {
                                         int jTupleCount_f =
((Integer)tupleCountMap_f.get(jTupleSeqStrF)).intValue();

tupleCountMap_f.put(jTupleSeqStrF, new Integer(++jTupleCount_f));
                                         //System.out.println("\t" +
jTupleSeqStrF + " count updated: " + jTupleCount_f);
                                         String jTupleSeqStrRC =
reverseComplement(jTupleSeqStrF);
                                         int jTupleCount_r =
((Integer)tupleCountMap_r.get(jTupleSeqStrRC)).intValue();

tupleCountMap_r.put(jTupleSeqStrRC, new Integer(++jTupleCount_r));
                                         counter[j-1]++;
                                  }catch(NullPointerException e2){
                                         //exception if sequence
contains symbols different than 'a','c','g', or 't' (eg.'n')
                                         illegalSymbolCounter++;
                                  }
                           }
                     } catch(StringIndexOutOfBoundsException e) {
                            System.out.println("WARNING sequence too
short: " + s);
                     }
        }
        if (illegalSymbolCounter>0)
             System.out.println("illegal symbols: " +
illegalSymbolCounter);
```

```
    }

    /**
     * consolidates Markov Model adding correction to those frequencies
that are equal to 0
     */
    public void consolidate() throws Exception {
        if (consolidated)
            throw new Exception("cant consolidate an already
consolidated MarkovModel");

        consolidate(tupleCountMap_f);
        if (useBothStrands)
            consolidate(tupleCountMap_r);
        consolidated = true;
    }
    /**
     * consolidates Markov Model adding counts the zero frequency
tuples, needed for the correction factor in method calcMMFreq()
     */
     private void consolidate(LinkedHashMap tupleCountMap) throws
Exception {
        //System.out.println("adding a sequence counts for this Markov
Model of order " + this.order + "...");
        voidCounter= new int[order+1];
        for (int i=0; i<order+1; i++)
            voidCounter[i] = 0;
        for (Iterator i = tupleCountMap.keySet().iterator();
i.hasNext();) {
            String tuple = i.next().toString();
            if (((Integer)tupleCountMap.get(tuple)).intValue() ==0) {
                voidCounter[tuple.length()-1]++;
            }
        }
    }

    /**
     * calcs Probability of a single tuple from this tupleCountMap
adding the correctionfactor;
     */
    public double calcMMFreq(String tuple, boolean forwardStrand) throws
Exception {
        if (!consolidated)
            throw new Exception("MarkovModel is not consolidated");
        tuple = tuple.toLowerCase();
        LinkedHashMap tupleCountMap;
        if (forwardStrand)
            tupleCountMap = tupleCountMap_f;
        else tupleCountMap = tupleCountMap_r;
        int j = tuple.length();
        //System.out.println(tuple);
        int tupleCount;
        try {
            tupleCount= ((Integer)tupleCountMap.get(tuple)).intValue();
        } catch(NullPointerException e) {
```

```
                throw new Exception("can't score tuple: " + tuple);
            }
            double correctionfactor = (double)voidCounter[j-
1]/(2*Math.pow(4, j));
            double freq= (double)tupleCount/counter[j-1];
            return freq + correctionfactor;
        }

        public LinkedHashMap getTupleCountMap_f() {
            return tupleCountMap_f;
        }

        public LinkedHashMap getTupleCountMap_r() {
            return tupleCountMap_r;
        }

        /*
         * use method in MarkovModelTools.java
         * returns the probability of the 4 symbols in a given position
based on the previous jTuple from this markovModel
         */
        public double[] fetchProbabilitiesOfNextSymol(String jTuple, boolean
plusStrand) throws Exception {
            double[] probabilities = new double[4];
            int totalCounts=0;
            int x=0;
            LinkedHashMap tupleCountMap;
            if (plusStrand)
                tupleCountMap= getTupleCountMap_f();
            else tupleCountMap= getTupleCountMap_r();
            while (totalCounts==0) { // will rely on a lesser order if the
totalCount is zero
                try {
                    jTuple = jTuple.substring(x++);
                } catch(StringIndexOutOfBoundsException e) {
                    jTuple="";
                }
                probabilities[0] = calcMMFreq(jTuple + "a", plusStrand); //
forward strand
                probabilities[1] = calcMMFreq(jTuple + "c", plusStrand);
                probabilities[2] = calcMMFreq(jTuple + "g", plusStrand);
                probabilities[3] = calcMMFreq(jTuple + "t", plusStrand);
                totalCounts += ((Integer)tupleCountMap.get(jTuple +
"a")).intValue();
                totalCounts += ((Integer)tupleCountMap.get(jTuple +
"c")).intValue();
                totalCounts += ((Integer)tupleCountMap.get(jTuple +
"g")).intValue();
                totalCounts += ((Integer)tupleCountMap.get(jTuple +
"t")).intValue();
            }
            double sum = 0;
            for (int i=0; i<4; i++) {
                sum+=probabilities[i];
            }
```

```
        if (sum==0)
            throw new Exception("probability cannot be zero");
        for (int i=0; i<4; i++) {
            probabilities[i]/=sum;
        }
        return probabilities;
    }

    public String generateRandomSeqStr(int order, int length) throws
Exception {
        if (!consolidated)
            throw new Exception("MarkovModel is not consolidated");
        String seqStr = "";
        boolean testMode =false;
        while (order > this.order) // max order is this.order
            order--;
        System.out.println("generating random seq using order "  + order
+ " (" + this.order + ") and length " + length);
        for (int i=0; i<order; i++) { //priming...
            String sym;
            double rand = Math.random();
            double[] probs = fetchProbabilitiesOfNextSymol(seqStr,
true);
            if (rand < probs[0])
                sym = "a";
            else if (rand < probs[0]+probs[1])
                sym = "c";
            else if (rand < probs[0]+probs[1]+probs[2])
                sym = "g";
            else
                sym = "t";
            seqStr += sym;
        }       int factor = 1; //number of times the length for
training the random seq.
        for (int i=order; i<length *factor; i++) {
            String sym;
            double rand = Math.random();
            double[] probs;
            probs=
fetchProbabilitiesOfNextSymol(seqStr.substring(seqStr.length() - order),
true);
            if (rand < probs[0])
                sym = "a"; // a
            else if (rand < probs[0]+probs[1])
                sym = "c"; //c
            else if (rand < probs[0]+probs[1]+probs[2])
                sym = "g"; //g
            else
                sym = "t"; //t
            seqStr += sym;
        }
        String[] seqStrArr = new String[factor];
        for (int s=0; s<factor; s++)
            seqStrArr[s] = seqStr.substring(s*length, ((s+1)*length));
        return seqStrArr[factor-1];
```

```
    }


    /**
     * Computes probability of a word using this Markov Model
     * @param wTuple word for which to compute the probability.
     */
    public double calcProbabilityOfTuple(String wTuple, boolean
forwardStrand, int desiredOrder) throws Exception {
        int w = wTuple.length();
        int order = desiredOrder;
        if (desiredOrder>this.order)
            order = this.order;
        while (order > w - 1) // decreases order for short tuples
shorter than order + 1
            order--;
        if (order!=desiredOrder)
            System.out.println("warning using order "  + order);
        LinkedHashMap tupleCountMap;
        if (forwardStrand)
            tupleCountMap = tupleCountMap_f;
        else tupleCountMap = tupleCountMap_r;
        double numerator = 1;
        for (int i=0; i<w-order; i++) {
            numerator *= calcMMFreq(wTuple.substring(i, i+order+1),
forwardStrand); //freq
        }
        double denominator = 1;
        try {
            for (int i = 0; i<w-order-1; i++) {
                denominator *=  calcMMFreq(wTuple.substring(i+1,
i+order+1), forwardStrand);
            }
        } catch(NullPointerException e) {
            //allows for order == 0
        }
        return numerator/denominator;
    }

    /**
     * Exactly the same as calcProbabilityOfTuple() only with output use
just for test
     */
    public double calcProbabilityOfTupleVerbose(String wTuple, boolean
forwardStrand, int desiredOrder) throws Exception {
        int w = wTuple.length();
        int order = desiredOrder;
        if (desiredOrder>this.order)
            order = this.order;
        while (order > w - 1) // decreases order for short tuples
shorter than order + 1
            order--;
        if (order!=desiredOrder)
            System.out.println("warning using order "  + order);
        LinkedHashMap tupleCountMap;
```

```
        if (forwardStrand)
             tupleCountMap = tupleCountMap_f;
        else tupleCountMap = tupleCountMap_r;
        double numerator = 1;
        System.out.print("p(" + wTuple + ") = \n   ");
        for (int i=0; i<w-order; i++) {
             numerator *= calcMMFreq(wTuple.substring(i, i+order+1),
forwardStrand); //freq
             System.out.print("p(" + wTuple.substring(i, i+order+1) + ")
");
        }
        System.out.print("\n   ----------------------\n        ");
        double denominator = 1;
        try {
             for (int i = 0; i<w-order-1; i++) {
                  denominator *=  calcMMFreq(wTuple.substring(i+1,
i+order+1), forwardStrand);
                  System.out.print("p(" + wTuple.substring(i+1,
i+order+1) + ") ");
             }
             System.out.println();
        } catch(NullPointerException e) {
             //allows for order == 0
        }
        return numerator/denominator;
    }

    public static void writeDataSetSpeciesModels(int order,
LinkedHashMap map, String outputFile) throws Exception {

        LinkedHashMap spec_modelMap = new LinkedHashMap();
        int j=0;
        int counter=0;
        for (Iterator i = map.keySet().iterator(); i.hasNext();) {
             String seqName = i.next().toString();
             String[] split = seqName.split("\\|");
             String species = "";
             try {
                  species = new Species(split[0]).getName();
             } catch(Exception e) {
                  try {
                       species = new Species(split[1]).getName();
                  } catch(Exception e1) {
                       try {
                            species = new Species(split[2]).getName();
                       } catch(Exception e2) {
                            throw new Exception("species not in fasta
heading: >" + seqName);
                       }
                  }
             }
             MarkovModel markovM;
             if (!spec_modelMap.keySet().contains(species)) {
                  System.out.println("species: " + species);
                  markovM = new MarkovModel(true, order); //use
```

```
bothStrands
                    spec_modelMap.put(species, markovM);
               } else markovM = (MarkovModel)spec_modelMap.get(species);
               String seq = map.get(seqName).toString();
               markovM.addMMCounts(seq);
               counter++;
          }
          for (Iterator i = spec_modelMap.values().iterator();
i.hasNext();)
               ((MarkovModel)i.next()).consolidate();
          ObjectOutputStream out = new ObjectOutputStream(new
FileOutputStream(outputFile));
          out.writeObject(spec_modelMap); // species > MarkovModel
          out.close();
          System.out.println("added seqs= " + counter);
     }

         /*public static void trainMarkovModels() {

         }*/

     public static void main(String[] args) throws Exception {
          System.out.println("\nUsage: java MarkovModel order dataset.fa
\n");
          String fileName= "input/dataset.txt"; //default
          int order = 6;                        //default
          System.out.println("testing markov modeld order 3");
                 MarkovModel testmm = new MarkovModel(true, order);
                 testmm.addMMCounts("catcatg");
     }
}
```

1.2 Package bioinfo.bio

1.2.1 bioinfo.bio.DNASequence Class

```
/*
 * code for DNA sequences
 * bioinfo.bio.DNASequence.java
 * created by @author Alberto Ambesi
 */

package bioinfo.bio;

import java.util.*;
import java.io.*;

/**
```

```
 *
 * @author alberto
 */
public class DNASequence implements Serializable {
    private String name;
    private String sequence;
    private HashMap featMap; // maps track name to a Features
    private HashMap annoMap;    //annotations, with no ref to locations.


    /**
     * @param name name of the sequence, @param seqeunce sequence
     */
    public DNASequence(String name, String sequence) throws
IllegalDNASymbolException {
        this.name = name;
        this.sequence = sequence;
        annoMap = new HashMap();
        featMap = new HashMap();
        //check sequence:
        Vector allowedSymbols = new Vector();
        if (!sequence.matches("[aAcCgGtTxXnN\\-]*"))
            throw new
IllegalDNASymbolException(this);//sequence.substring(i-1, i));
    }
    public String getName() {
        return name;
    }
    public String getSequence() {
        return sequence;
    }
    /**
     * returns a Vector of String features ... must be changed with
overlapping
     */
    public Vector getOverlappingFeatures(Location loc) {
        Vector v = (Vector)featMap.get(loc);
        return v;
    }

    /**
     * contains needs to be tested....
     */
    public void addFeature(Location loc, String feat) {
        Vector v;
        if (featMap.keySet().contains(loc))
            v=(Vector)featMap.get(loc);
        else {
            v = new Vector();
            featMap.put(loc, v);
        }
        v.addElement(feat);
    }

    public void addAnnotation(String annType, String ann) {
```

```
        annoMap.put(annType, ann);
    }

    public HashMap getFeatMap() {
        return featMap;
    }
    public HashMap getAnnoMap() {
        return annoMap;
    }
    public static void main(String[] args) throws Exception {
        //System.out.println("A".matches("[[aAcCgGtTxXnN\\-]"));
        DNASequence seq = new DNASequence("name", "AcTgNxXn-aCtG");
        System.out.println(seq.getName() + " " + seq.getSequence());
        try {
            DNASequence seq2 = new DNASequence("name2", "AcTgNxfXn-
aCtG");
            System.out.println(seq2.getName() + " " +
seq2.getSequence());
        } catch(IllegalDNASymbolException e) {
            System.out.println(e.toString());
        } try {
            DNASequence seq3 = new DNASequence("name3", "wAcqTgNxfXn-
aCtGs");
            System.out.println(seq3.getName() + " " +
seq3.getSequence());
        } catch(IllegalDNASymbolException e) {
            System.out.println(e.toString());
        }
    }
}

/** */
class IllegalDNASymbolException extends Exception {
    public IllegalDNASymbolException(DNASequence seq) {
        String sequence = seq.getSequence();
        String name = seq.getName();
        int min =3;
        int counter = 0;
        boolean printAll = true;
        for (int i=0; i<sequence.length(); i++) {
            String base = Character.toString(sequence.charAt(i));
            if (!base.matches("[aAcCgGtTxXnN\\-]")) {
                printAll = counter++<min;
                if (counter<2)
                    System.out.println("Exception in parsing DNASequence:
'" + name + "'");
                if (printAll) {
                    System.out.println("\tIllegal symbol: " + base + " at
position: " + (i+1));
                }
            }
        }
        if (!printAll)
            System.out.println("... and " + (counter-min) + " others.");
    }
```

1.2.2 bioinfo.bio.Species Class

```
/*
 * Species.java
 */

package bioinfo.bio;

import java.util.*;
import java.io.*;
import bioinfo.IOTools;

/**
 *
 * @author alberto
 */
public class Species implements Serializable {
    private String name;
    private String shortName;

    /**
     * @param fullName name of a species as "homo sapiens"
     */
    public Species(String fullName) {
        name = fullName.toLowerCase();
        shortName = new StringTokenizer(name," ").nextToken();
    }
    public String getName() {
        return name;
    }
    public String getShortName() {
        return shortName;
    }

    public boolean equals(Species sp) {
        return this.name.equalsIgnoreCase(sp.getName());
    }
    public static void main(String[] args) throws Exception {
        LinkedHashMap specMap = new LinkedHashMap();
        String homoName = "homo sapiens";
        String musName = "mus musculus";
        String ratName = "rattus norvegicus";

        Species human = new Species(homoName);
        Species mouse = new Species("mus musculus");
        Species rat = new Species("rattus norvegicus");

        System.out.println(mouse.getShortName());
        System.out.println(rat.getShortName());
        System.out.println(human.getShortName());
```

```
        specMap.put(musName, mouse);
        specMap.put(ratName, rat);
        specMap.put(homoName,human);

        //boolean write = false; //
        //if (write)
        //    IOTools.writeObject(specMap, fileName);
    }

}
```

1.3 Package bioinfo.ensj

1.3.1 bioinfo.ensj.LocationFetcher Class

```
package bioinfo.ensj;
import bioinfo.*;

import java.util.*;
import java.io.*;
import bioinfo.bio.EnsemblSpecies;
import org.ensembl.driver.CoreDriver;
import org.ensembl.datamodel.Sequence;
import org.ensembl.datamodel.Location;
import org.ensembl.driver.SequenceAdaptor;
/*
 */

public class LocationFetcher {


        /**
         * fetch location from a list of location w/ format:
chrX\t10000\t10100
         **/
    public static void main(String[] args) throws Exception {
            String usage = "Usage: java bioinfo.ensj.LocationFectcher
file spec";
            System.out.println(usage);
            String file = args[0];
            String spec = args[1];
            fetch(file, spec);
            //fetchPromoterWLoc(file,spec,4000,1000);
        }

        /**
         *
         **/
        public static void fetch(String file, String spec) throws
Exception {
```

```
            EnsemblSpecies species =
(EnsemblSpecies)EnsemblSpecies.getSpeciesMultiAccessMap().get(spec);
            Vector v = IOTools.parseStringsFromFile(file);
            PromoterFetcher fetcher = new PromoterFetcher(species);
            CoreDriver driver = fetcher.getCoreDriver();
            SequenceAdaptor seqAdaptor = driver.getSequenceAdaptor();
            for (int i =0; i< v.size();i++) {
                String line = v.elementAt(i).toString();
                if (line.startsWith("chr")) {
                    String[] spl = line.split("\t");
                    String chr = spl[0].substring(3);
                    int start = Integer.parseInt(spl[1]);
                    int end = Integer.parseInt(spl[2]);

                    Sequence seq = seqAdaptor.fetch(new
Location("chromosome",chr,start,end,1));
                    System.out.println(">" + species.getShortName() + "_"
+ chr + ":" + start + "-"+ end);
    //CoordinateSystem,sequenceRegion,start,end,strand
                    System.out.println(seq.getString());
                    //seq.printFasta(outputFileName);
                } //else System.out.println(line);
            }
        }
}
```

1.4 Package bioinfo.program

1.4.1 bioinfo.program.MatrixSampler Class

```
//
//  MatrixSampler3.java
//
//
//  Created by Alberto Ambesi on Fri Apr 15 2005.
//  Copyright (c) 2003 __MyCompanyName__. All rights reserved.
//
package bioinfo.program;

import java.io.*;
import java.io.Serializable;
import java.util.regex.*;
import java.util.*;
//import org.biojava.bio.*;
//import org.biojava.bio.seq.*;
//import org.biojava.bio.seq.impl.ViewSequence;
//import org.biojava.bio.symbol.*;
```

```
//import org.biojava.utils.ChangeVetoException;
import bioinfo.*;
import bioinfo.transfac.*;
import bioinfo.bio.EnsemblSpecies;

/**
 * This is the Gibbs sampling algorithm implementation as of Conlon 2002
 * scores must be taken the log (base 2): Math.log(score)/Math.log(2)
 */
public class MatrixSampler {
    private double threshold;
     private boolean useRevComp;
    private MarkovModel mm;
     private int order;
     private String sequenceStr;
     private String sequenceStr_rc;
     private String sequenceName;
     private double[] scoreLogSum; //[strand]
     private double[][] scoreProfile;
     //private double altScore_f; //alternative score based on
infoVector[position:1->L-w+1]
     //private double altScore_r; //alternative score based on
infoVector[position:1->L-w+1]

     /**
      * constructor using matrices
      * sampler2 : avoid using Sequence instatnce variable
      */
     public MatrixSampler(String[] sequence, MarkovModel mm, int order)
throws Exception {
         sequenceStr = sequence[0].toLowerCase();
         sequenceName = sequence[1];
          this.mm = mm;
          this.order = order;
         this.threshold = .85f; //default
     }

     public MatrixSampler(String[] sequence, MarkovModel mm, int order,
double threshold) throws Exception {
          this(sequence, mm, order);
         this.threshold = threshold;
     }

     /*
      * Core gibbs sampling algorithm
      */
    private double scoreTulpe(Matrix mat, int offset, boolean
forwardStrand) throws Exception {
         double matrixP =1; //motif sampler score Thijs '01
         double backgroundP =1;
          //double backgroundMax =0;
          //dists = mat.getDistributionsArray();
          String seqStr;
          if (forwardStrand)
               seqStr = sequenceStr;
```

```
          else seqStr = sequenceStr_rc;

          for (int i=0; i<mat.getInformationV().length; i++) {
             String base = "";
              String tuple = "";
              int j = -1; //base 0 1 2 3
              int end = offset+i; // position of base
              if (end>0) {
                   int start =end-order;
                   if (start <0)
                        start=0;
                   tuple = seqStr.substring(start, end); // one position
uptream of base
               }
               base = seqStr.substring(end, end+1);
               if (base.matches("[NnXx-]")) //.equalsIgnoreCase("n") ||
base.equalsIgnoreCase("x"))
                   throw new IllegalDNASymbolException(base); ///  N's are
masked
               else if (base.equalsIgnoreCase("a")) j=0;
               else if (base.equalsIgnoreCase("c")) j=1;
               else if (base.equalsIgnoreCase("g")) j=2;
               else if (base.equalsIgnoreCase("t")) j=3;
               try {//check that no n's are in previous tuple
                   while (tuple.substring(0,1).matches("[NnXx-]"))
//.equalsIgnoreCase("n")
                        //|| tuple.substring(0,1).equalsIgnoreCase("x"))
                            tuple=tuple.substring(1);
               } catch(StringIndexOutOfBoundsException e) {
                   tuple="";
               }
               matrixP *= mat.getDistributionsArray()[i][j];
               //System.out.println("tuple: " + tuple);
               try {
                   backgroundP *=
MarkovModelTools.fetchProbabilitiesOfNextSymol(tuple, mm,
forwardStrand)[j]; //(!)TESTMODE boolean forwardStrand
               } catch(Exception e) {
                   if (tuple.split("n|N|x|X|-", -2).length>1)
                        tuple="";             // correttion for input seq
of kind: // GGGCGGNGGA
                   backgroundP *=
MarkovModelTools.fetchProbabilitiesOfNextSymol(tuple, mm,
forwardStrand)[j]; //(!)TESTMODE boolean forwardStrand
               }
               //System.out.println(forwardStrand + ", consensus (" +
mat.getConsensus() + "); mat.freqs["+i+"]["+j+"]= "
               // + mat.freqs[i][j] + "; backgroundP(" + base + "|"+ tuple
+")="
               // + MarkovModelTools.fetchProbabilitiesOfNextSymol(tuple,
mm, forwardStrand)[j]);
          }
           double score = matrixP/backgroundP;
           if (forwardStrand) {
               scoreLogSum[0] += score;
```

```
                scoreProfile[0][offset] = score;
            } else {
                scoreLogSum[1] += score;
                scoreProfile[1][sequenceStr.length() -
mat.getInformationV().length -offset] = score;
            }
            return score;
        }


        /*
         * this implementation scores the sequence
         *
         */
        public void scoreSequence(Matrix mat, boolean bothStrands) throws
Exception {
            int positions = sequenceStr.length() -
mat.getInformationV().length + 1;
            scoreLogSum = new double[2]; //is initialized to zeros
            //int counter=0;
            try {
                scoreProfile = new double[2][positions]; //{fwd,rev}{scores}
                for(int offset = 0; offset < positions; offset++) {
                    try {
                        scoreTulpe(mat, offset, true);
                        //counter++;
                    }catch(IllegalDNASymbolException e) {
                    }
                }
                scoreLogSum[0] = Math.log(scoreLogSum[0])/Math.log(2);
                if (bothStrands) {
                    sequenceStr_rc =
BioinfoTools.reverseComplement(sequenceStr);
                    scoreLogSum[1] = 0;
                    for(int offset = 0; offset < positions; offset++) {
                        try {
                            scoreTulpe(mat, offset, false);
                        }catch(IllegalDNASymbolException e) {
                        }
                    }
                    scoreLogSum[1] = Math.log(scoreLogSum[1])/Math.log(2);
                }
            } catch(NegativeArraySizeException e) { // motif is longer than
sequence
                scoreProfile = new double[2][0];
            }
        }


        public String getSequenceStr() {
            return sequenceStr;
        }

        public String getSequenceName() {
            return sequenceName;
```

```
    }

    public double[] getScoreLogSum() {
        return scoreLogSum;
    }

    public double[][] getScoreProfile() {
        return scoreProfile;
    }

    public static void main(String[] args) throws Exception {
        //System.out.println(args[0].split("n|N|x|X", -2).length);
        if (args.length<3)
            throw new Exception("Usage: java
bioinfo.transfac.MatrixSampler dataset.fa motifMap markovModelMap [
noOfMatrices ]");
        String seqFile = args[0];
        String matrixFile = args[1];
        String markovModelFile = args[2];//"objects/homo_orth26_tf8-
3_5k-1keMrM.markovM5";
        LinkedHashMap matrixMap =
(LinkedHashMap)IOTools.loadObject(matrixFile);
        LinkedHashMap seqMap = BioinfoTools.parseFastaFile(seqFile);
        HashMap specMap = EnsemblSpecies.getSpeciesMultiAccessMap();
        int noOfMatrices = matrixMap.values().size();
        if (args.length>3)
            noOfMatrices = Integer.parseInt(args[3]);

        int order =3;
        LinkedHashMap markovModelMap =
(LinkedHashMap)IOTools.loadObject(markovModelFile);
        PrintStream ps = IOTools.createPrintStream(seqFile,
"_MatrixSampler.xls");
        //int counter=0;
        for (Iterator seqIt = seqMap.keySet().iterator();
seqIt.hasNext();) {
            String seqName =
seqIt.next().toString();//sequence.getName();
            String seqStr = seqMap.get(seqName).toString();
            String[] sequence = new String[]{seqStr, seqName};
            int index = seqName.indexOf("|");
            String spec = seqName.substring(index+1,
seqName.indexOf("|", index+1));

            MarkovModel model =
(MarkovModel)markovModelMap.get((EnsemblSpecies)specMap.get(spec));
            String geneID = seqName.substring(0, index);
            MatrixSampler sampler = new MatrixSampler(sequence, model,
order);
            /*ps.print("geneID\tmatrixID");
            for (Iterator j =matrixMap.values().iterator();
j.hasNext();)
                ps.print("\t" + ((Matrix)j.next()).getMatrixID());
            ps.println();*/
            for (Iterator j =matrixMap.values().iterator();
```

```
j.hasNext();) {
                    Matrix m= (Matrix)j.next();
                    sampler.scoreSequence(m, true);
                    System.out.println("\t" + geneID + "(" + spec + ")_"
                         + m.getMatrixID()
                         + "= f" + sampler.getScoreLogSum()[0]
                         + " r" + sampler.getScoreLogSum()[1]
                    );
                    ps.print(geneID + "\t" + m.getMatrixID());
                    double[][] profile = sampler.getScoreProfile();
                    for (int k =0; k<profile[0].length; k++) {
                         double maxScore = Math.max(profile[0][k],
profile[1][k]);
                         ps.print("\t" + maxScore);
                    }
                    ps.println();
               }
          }
          ps.close();
     }
}
```

1.5 Package bioinfo.transfac

1.5.1 bioinfo.transfac.Matrix Class

```
//
//  Matrix.java
//
//
//  Created by Administrator on Mon Nov 03 2003.
//  Copyright (c) 2003 __MyCompanyName__. All rights reserved.
//
package bioinfo.transfac;
import java.lang.*;
import java.io.*;
import java.util.*;
//import org.biojava.bio.dist.*;
//import org.biojava.bio.dp.*;
//import org.biojava.bio.symbol.*;
//import org.biojava.bio.seq.*;
//import org.biojava.bio.seq.io.*;
import bioinfo.*;

/**
 * a class for matrix records in Transfac Matrix table
 */
```

```
public class Matrix implements Cloneable, Serializable {
     private String matrixID;
    private String name;
    private String description;
    private String bindingFactors;
    private String consensus;
    private Vector countsV;
    protected double[][] freqs; //position i, base j
    protected double max, min;
    protected double[] info;
    protected int length;
    protected double pseudoCount;
    public static final double pseudoCount_default=0.0001; //default
value
     private static String transfacVersion;

    public Matrix(String id, String name, String consensus, String
description, String bindingFactors, Vector countsV) throws Exception {
         this(id, name, consensus, description, bindingFactors, countsV,
pseudoCount_default);
     }

    public Matrix(String id, String name, String consensus, String
description, String bindingFactors, Vector countsV, double pseudoCount)
throws Exception {
        matrixID = id;
        transfacVersion = TransfacTools.currentVersion();
         this.name = name;
        this.description = description;
        this.consensus = consensus;
        this.bindingFactors = bindingFactors;
        this.countsV = countsV;
        this.pseudoCount = pseudoCount;
         length = countsV.size();
        createDistributionsArray(pseudoCount);
        createInformationVector();
        calcMaxAndMin();
   }
   /*
    * as in nature review gen wasserman '04
    */
   private void createInformationVector() throws Exception {
       info = new double[length];
       for (int i=0; i<length; i++) {
           info[i] = 2;
           for (int j =0; j<4; j++) {
               if (freqs[i][j]!=0)
                   info[i] += freqs[i][j] *
Math.log(freqs[i][j])/Math.log(2); // Math.log() is base e
         //else log(4*0) ~ 0
           }
           //System.out.println("\tinfo[" + i + "]= " + info[i]);
       }
    }
```

```
    private void calcMaxAndMin() throws Exception {
        //System.out.println("calculating max and min...");
        min = 0;
        max = 0;
        for (int i=0; i<length; i++) {
            double minFreq = freqs[i][0];
            double maxFreq = freqs[i][0];
            for (int j=1; j<4; j++) {
                if (minFreq>freqs[i][j]) minFreq = freqs[i][j];
                if (maxFreq<freqs[i][j]) maxFreq = freqs[i][j];
            }
            //System.out.println("\tminFreq: " + minFreq + ", maxFreq: "
+ maxFreq + ", info[" + i + "]:" + info[i]);
             min += info[i]*minFreq;//Math.log(minFreq)/Math.log(2);
             max += info[i]*maxFreq;//Math.log(maxFreq)/Math.log(2);
        }
        //System.out.println("...calculated max and min: " + max + ", " +
min);
    }

    /*
     * returns an array of weights with pseudoCount is used by
MatchAnnotator
     */
    private void createDistributionsArray(double pseudoCount) throws
Exception {
        //System.out.println("creating distributions array using " +
pseudoCount + " pseudocounts.");
        freqs = new double[countsV.size()][4];
        //FiniteAlphabet dna = DNATools.getDNA();
        for (int i=0; i<countsV.size(); i++) {
            double[] counts= (double[])(countsV.elementAt(i));
             double sum =0;
            for (int j=0; j<4; j++) {
                 freqs[i][j] = counts[j] + pseudoCount;
                 //System.out.print(freqs[i][j] + "\t" );
                sum += freqs[i][j];
            }
            for (int j=0; j<4; j++) {
                freqs[i][j] /= sum;
                 //System.out.print(freqs[i][j] + "\t" );
            }
            //System.out.println();
        }
    }

    /*
     * Attention: shuffles only matrix not consensus
     */
    public Matrix shuffle() throws Exception {
        Matrix shuffledMatrix = (Matrix)(this.clone());
        List weightsL = (List) countsV;
        Collections.shuffle(weightsL);
```

```
    setCountsV(new Vector(weightsL));
    return shuffledMatrix;
}

public Matrix reverseComplement() throws Exception {
    Matrix revCompMatrix = (Matrix)(this.clone());
    List weightsL = (List)countsV;
    Collections.reverse(weightsL);
    for (int i =0; i<weightsL.size(); i++) {
        double[] weightsAtCurrentPos = (double[])weightsL.get(i);
        double weightA = weightsAtCurrentPos[0];
        double weightC = weightsAtCurrentPos[1];
        weightsAtCurrentPos[0] = weightsAtCurrentPos[3]; // a <- t
        weightsAtCurrentPos[1] = weightsAtCurrentPos[2]; // c <- g
        weightsAtCurrentPos[2] = weightC; // g <- c
        weightsAtCurrentPos[3] = weightA; // t <- a
        weightsL.set(i, weightsAtCurrentPos);
    }
    setCountsV(new Vector(weightsL));
    return revCompMatrix;
}
public String getMatrixID() throws Exception {
    return matrixID;
}

public String getName() throws Exception {
    return name;
}
public String getDescription() throws Exception {
    return description;
}
public String getConsensus() throws Exception {
    return consensus;
}
public Vector getCountsV() throws Exception {
    return countsV;
}
public String getBindingFactors() throws Exception {
    return bindingFactors;
}
public static String getTransfacVersion() throws Exception {
     return transfacVersion;
 }
public double[][] getDistributionsArray() throws Exception {
    return freqs;
}

public double getMax() throws Exception {
    return max;
}
public double getMin() throws Exception {
    return min;
}
public double[] getInformationV() throws Exception {
    return info;
```

```
    }

     public void changePseudoCountAndUpdateMatrix(double i) throws
Exception {
          this.pseudoCount = i;
          createDistributionsArray(pseudoCount);
         createInformationVector();
         calcMaxAndMin();
     }

    public void setMatrixID(String s) throws Exception {
        matrixID= s;
    }
    public void setName(String s) throws Exception {
        name= s;
    }
    public void setDescription(String s) throws Exception {
        description= s;
    }
    public void setConsensus(String s) throws Exception {
        consensus= s;
    }
    public void setCountsV(Vector v) throws Exception {
        countsV= v;
    }
    public void setBindingFactors(String s) throws Exception {
        bindingFactors= s;
    }
    /*
     public static LinkedHashMap writeAllMatrices(boolean
vertebratesOnly, boolean removeFlankingNNN) throws Exception {
          return writeAllMatrices(vertebratesOnly, removeFlankingNNN,
pseudoCount_default);
     }*/

     public static LinkedHashMap writeAllMatrices(boolean
vertebratesOnly, boolean removeFlankingNNN, double pseudoCount) throws
Exception {
          Properties props = new Properties();
          InputStream inputStream = new FileInputStream(new
File("config/match.props"));
          props.load(inputStream);
          String version = props.getProperty("transfacVersion");
          //String home = props.getProperty("home");
          String option = "";
          if (vertebratesOnly)
               option = "Vert";
          if (removeFlankingNNN)
               option += "Nr";
          //option += ("" + pseudoCount); //.replaceAll("\\.", "_");

          String outputFile = "objects/all" + option + "_matrices_" +
version + ".map";
          System.out.println("writing all " + option + " matrices to " +
outputFile + "...");
```

```
        LinkedHashMap map =
TransfacTools.fetchAllMatrices(vertebratesOnly, removeFlankingNNN);
        if (pseudoCount!=pseudoCount_default)
            for (Iterator i = map.values().iterator(); i.hasNext();)

    ((Matrix)i.next()).changePseudoCountAndUpdateMatrix(pseudoCount);
        //ObjectOutputStream out = new ObjectOutputStream(new
FileOutputStream(outputFile));
        //out.writeObject(map); // ensemblID > TransfacGene
        //out.close();
        System.out.println("... done. (pseudoCount=" + pseudoCount + ");
removeFlankingNNN: " + removeFlankingNNN);
        return map;
    }

    /**
     * this main writes the matrix map to file
     */
    public static void main(String[] args) throws Exception {
        if (args.length <2) throw new Exception ("Usage: java
bioinfo.transfac.Matrix vertebratesOnly(t|*) removeFlankingNNN(t|*) [
pseudoCount ]");
        long timePoint = System.currentTimeMillis();

        boolean vertebratesOnly = args[0].equalsIgnoreCase("t");
        boolean removeFlankingNNN = args[1].equalsIgnoreCase("t");
        double pseudoCount;
        if (args.length==3) {
            pseudoCount = Double.parseDouble(args[2]);
        } else pseudoCount = Matrix.pseudoCount_default;

        System.out.println("stored " + writeAllMatrices(vertebratesOnly,
removeFlankingNNN, pseudoCount).values().size() + " matrices.");

        System.out.println("time: " + (System.currentTimeMillis() -
timePoint));
        timePoint = System.currentTimeMillis();
    }
}
```

1.5.2 bioinfo.transfac.Motif Class

```
//
//  Matrix.java
//
//
//  Created by Administrator on Mon Nov 03 2003.
//  Copyright (c) 2003 __MyCompanyName__. All rights reserved.
//
package bioinfo.transfac;
import java.lang.*;
import java.io.*;
```

```
import java.util.*;
//import org.biojava.bio.dist.*;
//import org.biojava.bio.dp.*;
//import org.biojava.bio.symbol.*;
//import org.biojava.bio.seq.*;
//import org.biojava.bio.seq.io.*;
import bioinfo.*;

/**
 * a class for matrix records in Transfac Matrix table
 */


public class Matrix implements Cloneable, Serializable {
     private String matrixID;
    private String name;
    private String description;
    private String bindingFactors;
    private String consensus;
    private Vector countsV;
    protected double[][] freqs; //position i, base j
    protected double max, min;
    protected double[] info;
    protected int length;
    protected double pseudoCount;
    public static final double pseudoCount_default=0.0001; //default
value
     private static String transfacVersion;

    public Matrix(String id, String name, String consensus, String
description, String bindingFactors, Vector countsV) throws Exception {
         this(id, name, consensus, description, bindingFactors, countsV,
pseudoCount_default);
     }

    public Matrix(String id, String name, String consensus, String
description, String bindingFactors, Vector countsV, double pseudoCount)
throws Exception {
        matrixID = id;
        transfacVersion = TransfacTools.currentVersion();
         this.name = name;
        this.description = description;
        this.consensus = consensus;
        this.bindingFactors = bindingFactors;
        this.countsV = countsV;
        this.pseudoCount = pseudoCount;
         length = countsV.size();
        createDistributionsArray(pseudoCount);
        createInformationVector();
        calcMaxAndMin();
   }
   /*
    * as in nature review gen wasserman '04
    */
   private void createInformationVector() throws Exception {
```

```
        info = new double[length];
        for (int i=0; i<length; i++) {
            info[i] = 2;
            for (int j =0; j<4; j++) {
                if (freqs[i][j]!=0)
                    info[i] += freqs[i][j] *
Math.log(freqs[i][j])/Math.log(2); // Math.log() is base e
                                   // else log(4*0) ~ 0
            }
            //System.out.println("\tinfo[" + i + "]= " + info[i]);
        }
    }


    private void calcMaxAndMin() throws Exception {
        //System.out.println("calculating max and min...");
        min = 0;
        max = 0;
        for (int i=0; i<length; i++) {
            double minFreq = freqs[i][0];
            double maxFreq = freqs[i][0];
            for (int j=1; j<4; j++) {
                if (minFreq>freqs[i][j]) minFreq = freqs[i][j];
                if (maxFreq<freqs[i][j]) maxFreq = freqs[i][j];
            }
            //System.out.println("\tminFreq: " + minFreq + ", maxFreq: "
+ maxFreq + ", info[" + i + "]:" + info[i]);
             min += info[i]*minFreq;//Math.log(minFreq)/Math.log(2);
             max += info[i]*maxFreq;//Math.log(maxFreq)/Math.log(2);
        }
        //System.out.println("...calculated max and min: " + max + ", " +
min);
    }

    /*
     * returns an array of weights with pseudoCount is used by
MatchAnnotator
     */
    private void createDistributionsArray(double pseudoCount) throws
Exception {
        //System.out.println("creating distributions array using " +
pseudoCount + " pseudocounts.");
        freqs = new double[countsV.size()][4];
        //FiniteAlphabet dna = DNATools.getDNA();
        for (int i=0; i<countsV.size(); i++) {
            double[] counts= (double[])(countsV.elementAt(i));
             double sum =0;
            for (int j=0; j<4; j++) {
                 freqs[i][j] = counts[j] + pseudoCount;
                 //System.out.print(freqs[i][j] + "\t" );
                sum += freqs[i][j];
            }
            for (int j=0; j<4; j++) {
                freqs[i][j] /= sum;
                 //System.out.print(freqs[i][j] + "\t" );
```

```
        }
        //System.out.println();
    }
}

/*
 * Attention: shuffles only matrix not consensus
 */
public Matrix shuffle() throws Exception {
    Matrix shuffledMatrix = (Matrix)(this.clone());
    List weightsL = (List) countsV;
    Collections.shuffle(weightsL);
    setCountsV(new Vector(weightsL));
    return shuffledMatrix;
}

public Matrix reverseComplement() throws Exception {
    Matrix revCompMatrix = (Matrix)(this.clone());
    List weightsL = (List)countsV;
    Collections.reverse(weightsL);
    for (int i =0; i<weightsL.size(); i++) {
        double[] weightsAtCurrentPos = (double[])weightsL.get(i);
        double weightA = weightsAtCurrentPos[0];
        double weightC = weightsAtCurrentPos[1];
        weightsAtCurrentPos[0] = weightsAtCurrentPos[3]; // a <- t
        weightsAtCurrentPos[1] = weightsAtCurrentPos[2]; // c <- g
        weightsAtCurrentPos[2] = weightC; // g <- c
        weightsAtCurrentPos[3] = weightA; // t <- a
        weightsL.set(i, weightsAtCurrentPos);
    }
    setCountsV(new Vector(weightsL));
    return revCompMatrix;
}
public String getMatrixID() throws Exception {
    return matrixID;
}

public String getName() throws Exception {
    return name;
}
public String getDescription() throws Exception {
    return description;
}
public String getConsensus() throws Exception {
    return consensus;
}
public Vector getCountsV() throws Exception {
    return countsV;
}
public String getBindingFactors() throws Exception {
    return bindingFactors;
}
public static String getTransfacVersion() throws Exception {
     return transfacVersion;
 }
```

```
    public double[][] getDistributionsArray() throws Exception {
        return freqs;
    }

    public double getMax() throws Exception {
        return max;
    }
    public double getMin() throws Exception {
        return min;
    }
    public double[] getInformationV() throws Exception {
        return info;
    }

     public void changePseudoCountAndUpdateMatrix(double i) throws
Exception {
          this.pseudoCount = i;
          createDistributionsArray(pseudoCount);
         createInformationVector();
         calcMaxAndMin();
     }

    public void setMatrixID(String s) throws Exception {
        matrixID= s;
    }
    public void setName(String s) throws Exception {
        name= s;
    }
    public void setDescription(String s) throws Exception {
        description= s;
    }
    public void setConsensus(String s) throws Exception {
        consensus= s;
    }
    public void setCountsV(Vector v) throws Exception {
        countsV= v;
    }
    public void setBindingFactors(String s) throws Exception {
        bindingFactors= s;
    }
    /*
     public static LinkedHashMap writeAllMatrices(boolean
vertebratesOnly, boolean removeFlankingNNN) throws Exception {
          return writeAllMatrices(vertebratesOnly, removeFlankingNNN,
pseudoCount_default);
     }*/

     public static LinkedHashMap writeAllMatrices(boolean
vertebratesOnly, boolean removeFlankingNNN, double pseudoCount) throws
Exception {
          Properties props = new Properties();
          InputStream inputStream = new FileInputStream(new
File("config/match.props"));
          props.load(inputStream);
          String version = props.getProperty("transfacVersion");
```

```
        //String home = props.getProperty("home");
        String option = "";
        if (vertebratesOnly)
            option = "Vert";
        if (removeFlankingNNN)
            option += "Nr";
        //option += ("" + pseudoCount); //.replaceAll("\\.", "_");

        String outputFile = "objects/all" + option + "_matrices_" +
version + ".map";
        System.out.println("writing all " + option + " matrices to " +
outputFile + "...");
        LinkedHashMap map =
TransfacTools.fetchAllMatrices(vertebratesOnly, removeFlankingNNN);
        if (pseudoCount!=pseudoCount_default)
            for (Iterator i = map.values().iterator(); i.hasNext();)

    ((Matrix)i.next()).changePseudoCountAndUpdateMatrix(pseudoCount);
        //ObjectOutputStream out = new ObjectOutputStream(new
FileOutputStream(outputFile));
        //out.writeObject(map); // ensemblID > TransfacGene
        //out.close();
        System.out.println("... done. (pseudoCount=" + pseudoCount + ");
removeFlankingNNN: " + removeFlankingNNN);
        return map;
    }

    /**
     * this main writes the matrix map to file
     */
    public static void main(String[] args) throws Exception {
        if (args.length <2) throw new Exception ("Usage: java
bioinfo.transfac.Matrix vertebratesOnly(t|*) removeFlankingNNN(t|*) [
pseudoCount ]");
        long timePoint = System.currentTimeMillis();

        boolean vertebratesOnly = args[0].equalsIgnoreCase("t");
        boolean removeFlankingNNN = args[1].equalsIgnoreCase("t");
        double pseudoCount;
        if (args.length==3) {
            pseudoCount = Double.parseDouble(args[2]);
        } else pseudoCount = Matrix.pseudoCount_default;

        System.out.println("stored " + writeAllMatrices(vertebratesOnly,
removeFlankingNNN, pseudoCount).values().size() + " matrices.");

        System.out.println("time: " + (System.currentTimeMillis() -
timePoint));
        timePoint = System.currentTimeMillis();
    }
}
```

1.6 Package bioinfo.ucsc

1.6.1 bioinfo.ucsc.RefSeqGene Class

```
/*
 * RefseqGene.java
 *
 * Created on August 23, 2006, 7:41 PM
 *
 * To change this template, choose Tools | Template Manager
 * and open the template in the editor.
 */

package bioinfo.ucsc;
import bioinfo.bio.EnsemblSpecies;
import java.util.*;
import java.io.*;
import java.util.zip.GZIPInputStream;
import org.ensembl.datamodel.Location;


/**
 *
 * @author alberto
 */
public class RefseqGene {
    private String id;
    private Location geneLoc;
    private Location cdsLoc;
    private Vector exonV;
    private Vector intronV;
    private static String build;
    private static String species;
    private static String file;
    private Vector altOverlapLocs;
    private Vector altNonOverlapLocs;
    private static Vector ambiguousRefseqsV;

    /** Creates a new instance of RefseqGene */
    public RefseqGene(String id, Location geneLoc, Location cdsLoc,
Vector exonV, Vector intronV) {
        this.id =id;
        this.geneLoc=geneLoc;
        this.cdsLoc=cdsLoc;
        this.exonV = exonV;
        this.intronV = intronV;
        altOverlapLocs = new Vector();
        altNonOverlapLocs = new Vector();
    }


    public static void main(String[] args) throws Exception {
        String aSpecies = "mus";
```

```
        String aBuild = "35";
        if (args.length>1) {
            aSpecies = args[0];
            aBuild = args[1];
        }
        aSpecies =
((EnsemblSpecies)EnsemblSpecies.getSpeciesMultiAccessMap().get(aSpecies))
.getShortName();
        String usage = "Usage: java bioinfo.ucsc.RefseqGene [ species
build ]";
        System.out.println(usage);
        String[] chromosomes = new
String[]{"1","2","3","4","5","6","7","8","9","10","11","12","13","14","15
","16","17","18","19","X","Y","M"};
        if (!aSpecies.equals("mus")) {
            System.out.println("currently works only for mus. Valid
chromosomes are: 1-19, X, Y, M");
            return;
        }
        LinkedHashMap map = parse(aSpecies, aBuild);
        System.out.println("total number of refseqs: " +
map.entrySet().size());
        LinkedHashMap<String,Vector> chrMap = new
LinkedHashMap<String,Vector>();
        for (Iterator i = map.values().iterator(); i.hasNext();) {
            RefseqGene gene = (RefseqGene)i.next(); //test first gene
            String refseq = gene.getId();
            int count = 0;
            Location geneLoc = gene.getGeneLoc();
            String chr = geneLoc.getSeqRegionName();
            Vector refseqV;
            if (chrMap.containsKey(chr))
                refseqV=chrMap.get(chr);
            else {
                refseqV=new Vector();
                chrMap.put(chr,refseqV);
            }
            if (!refseqV.contains(refseq))
                refseqV.addElement(refseq);

            /*for (Iterator i=gene.getExonV().iterator(); i.hasNext();)
                System.out.println("exon  " + ++count + ": " +
((Location)i.next()).toString());
            count=0;
            for (Iterator i=gene.getIntronV().iterator(); i.hasNext();)
                System.out.println("intron " + ++count + ": " +
((Location)i.next()).toString());
            */
        }
        /*for (int i = 0; i<chromosomes.length; i++) {
            try {
                System.out.println(chromosomes[i] + "\t" +
chrMap.get(chromosomes[i]).size());
            } catch(NullPointerException e) {
                System.out.println(chromosomes[i] + "\t0");
```

```
            }
        }*/
        for (Iterator i = chrMap.entrySet().iterator(); i.hasNext();) {
            Map.Entry ent = (Map.Entry)i.next();

            System.out.println(ent.getKey() + "\t" +
((Vector)ent.getValue()).size());

        }
        //System.out.println("test gene " + gene.getId() + ", 5 prime
UTR: " + gene.get5PrimeUTR().toString());
    }
    /**
     * @param args the command line arguments
     */
    public static LinkedHashMap parse(String aSpecies, String aBuild)
throws Exception {
        species = aSpecies;
        build = aBuild;
        String home = "/Users/alberto";
        //file = "data/ucsc_" + aSpecies + "_" + aBuild +
"/refGene.txt.gz";
        file = home + "/Public/" +
UcscTools.convertNCBIBuild(aSpecies,Integer.parseInt(aBuild)) +
"/database/refGene.txt.gz";

        //String refFile = "data/ucsc_" + spec + "_" + build +
"/refGene.xls";
        File testFile = new File(file);
        if (!testFile.exists())
            System.out.println("refseq file not found: " + file
                    + "\ndownload refGene.txt.gz from ucsc
http://hgdownload.cse.ucsc.edu/downloads.html, annotation database
section");
        BufferedReader br= new BufferedReader(
                            new InputStreamReader(
                                new GZIPInputStream(
                                    new FileInputStream(
                                        file
                            ))));
        System.out.println("reading refseq file: " + file );
        String line = br.readLine();
        LinkedHashMap map = new LinkedHashMap();
        ambiguousRefseqsV = new Vector();
        while (line!=null) {
            String[] parsedL = line.split("\t");
            int k =0;

            String id =  parsedL[k];
            try { //bugfix: new files start with a number.
                Integer.parseInt(id);
                id =  parsedL[++k];
            } catch(NumberFormatException ex) {

            }
```

```
            String chr = parsedL[++k].substring(3);
            int str = 0;
            if (parsedL[++k].equals("+"))
                str = 1;
            else if (parsedL[k].equals("-"))
                str =-1;
            else throw new Exception("strand is illegal: " + parsedL[k] +
"\nline:" + line);

            int start = Integer.parseInt(parsedL[++k]);
            int end   = Integer.parseInt(parsedL[++k]);
            Location geneLoc = new
Location("chromosome",chr,start,end,str);

            start = Integer.parseInt(parsedL[++k]);
            end   = Integer.parseInt(parsedL[++k]);
            Location cdsLoc  = new
Location("chromosome",chr,start,end,str);

            Vector exonV = new Vector();
            Vector intronV = new Vector();
            int exonsNo = Integer.parseInt(parsedL[++k]);
            String[] exonSt = parsedL[++k].split(",");
            String[] exonEn = parsedL[++k].split(",");
            if (exonSt.length!=exonsNo || exonEn.length!=exonsNo)
                throw new Exception();
            for (int i=0; i<exonsNo;i++) {
                int st = Integer.parseInt(exonSt[i]);
                int en = Integer.parseInt(exonEn[i]);
                Location exon = new Location("chromosome",chr,st,en,str);
                exonV.addElement(exon);
                try {
                    int st2 = Integer.parseInt(exonSt[i+1]);
                    Location intron = new
Location("chromosome",chr,en+1,st2-1,str);
                    intronV.addElement(intron);

                } catch(IndexOutOfBoundsException e) {

                }
            }
            if (str==-1) { //invert order of exon and intron Vectors
                Vector tmpV = new Vector();
                for (int i=0; i<exonV.size(); i++)
                    tmpV.add(0, exonV.elementAt(i));
                exonV = tmpV;
                tmpV = new Vector();
                for (int i=0; i<intronV.size(); i++)
                    tmpV.add(0, intronV.elementAt(i));
                intronV = tmpV;
            }
//            //TEST
//            if (id.equals("NM_207668"))
//                System.out.println("testing: " + id);
            RefseqGene gene = new RefseqGene(id, geneLoc, cdsLoc, exonV,
```

```
intronV);
            if (!geneLoc.getSeqRegionName().endsWith("_random")) {
                RefseqGene prev = (RefseqGene)map.put(id, gene);
                if (prev!=null) {
                    //boolean sameTSS = false;
                    Location prevLoc = prev.getGeneLoc();

                    if (overlaps(prevLoc,geneLoc)) {//overlaps, choose
most upstream
                        if (str>=0) {
                            if (prevLoc.getStart()<geneLoc.getStart()) {
                                map.put(id, prev); //put back
                                prev.altOverlapLocs.addElement(geneLoc);
                                //System.out.println(id + ": " +
geneLoc.getStart() + " <- " + prevLoc.getStart());
                                //if (!ambiguousRefseqsV.contains(id))
                                //    ambiguousRefseqsV.addElement(id);
                            } else {
                                gene.altOverlapLocs.addElement(prevLoc);
                            }

                        } else {
                            if (prevLoc.getEnd()>geneLoc.getEnd()) {
                                map.put(id, prev);
                                prev.altOverlapLocs.addElement(geneLoc);
                                //if (!ambiguousRefseqsV.contains(id))
                                  //  ambiguousRefseqsV.addElement(id);
                                //System.out.println(id + ": " +
geneLoc.getEnd() + " <- " + prevLoc.getEnd());
                            } else {
                                gene.altOverlapLocs.addElement(prevLoc);
                            }
                        }
                    } else {
                        map.put(id, prev); //put back
                        prev.altNonOverlapLocs.addElement(geneLoc);
                        if (!ambiguousRefseqsV.contains(id))
                             ambiguousRefseqsV.addElement(id);
                        System.err.println("WARNING: duplicate entry not
overlapping for refseq " + id);
                    }
                }
            }

            //System.out.println("map contains ID: " + id + " -> " +
map.containsValue(id));
            /*if (id.equals("NM_001001999"))
                System.out.println("TEST NM_001001999: " + id + " "
                        + geneLoc.getSeqRegionName() + " "
                        + geneLoc.getStart() + " "
                        + geneLoc.getEnd() + " "
                        + gene.getId() + ". "
                        +" Map contains ID: " + map.containsKey(id)
                        );
            */
```

```
            line = br.readLine();
        }
        System.out.println("no of ambiguous refseqs: " +
ambiguousRefseqsV.size());
        //for (Iterator i = getAmbiguousRefseqsV().iterator();
i.hasNext();)
          //  System.out.println(i.next().toString());
//        System.out.println("map contains NM_207668: " +
map.containsKey("NM_207668"));

        return map;
    }

    public String getId() {
        return id;
    }

    public Location getGeneLoc() {
        return geneLoc;
    }

    public Location getCdsLoc() {
        return cdsLoc;
    }

    public Vector getExonV() {
        return exonV;
    }

    public Vector getIntronV() {
        return intronV;
    }
    public Location get5PrimeUTR() {
        Location utrLoc = null;
        int start = geneLoc.getStart();
        int end = geneLoc.getEnd();
        int strand = geneLoc.getStrand();

        if (strand>0) {
            if (start<cdsLoc.getStart()) // hasUTR
                utrLoc = new Location("chromosome",
geneLoc.getSeqRegionName(), start,cdsLoc.getStart()-1,strand);
        } else {
            if (cdsLoc.getEnd()<end) // hasUTR
                utrLoc = new Location("chromosome",
geneLoc.getSeqRegionName(), cdsLoc.getEnd()+1, end, strand);
        }

        return utrLoc;
    }
    public Location get3PrimeUTR() {
        Location utrLoc = null;
        int start = geneLoc.getStart();
        int end = geneLoc.getEnd();
        int strand = geneLoc.getStrand();
```

```
        if (strand>0) {
            if (cdsLoc.getEnd()<end) // hasUTR
                utrLoc = new Location("chromosome",
geneLoc.getSeqRegionName(), cdsLoc.getEnd()+1, end, strand);
        } else {
            if (start<cdsLoc.getStart()) // hasUTR
                utrLoc = new Location("chromosome",
geneLoc.getSeqRegionName(), start,cdsLoc.getStart()-1,strand);
        }

        return utrLoc;
    }

    public static Vector getAmbiguousRefseqsV() {
        return ambiguousRefseqsV;
    }
        public static boolean overlaps(Location a, Location b) throws
Exception {
            boolean bool = false;
            //System.out.println("test " + a.getSeqRegionName());
            //System.out.println("test " + a.getSeqRegionName());
            if (a.getSeqRegionName().equals(b.getSeqRegionName())) {
                int startA = a.getStart();
                int endA = a.getEnd();
                int startB = b.getStart();
                int endB = b.getEnd();

                if (endA<startA || endB<startB)
                    throw new Exception("end < start:");
                bool =((endA>=startB && endA<=endB) || (endB>=startA &&
endB<=endA));
            }
            return bool;
        }

    public Vector getAltNonOverlapLocs() {
        return altNonOverlapLocs;
    }

    public Vector getAltOverlapLocs() {
        return altOverlapLocs;
    }
}
```

Appendix B

Matlab Code Examples

2.1 Logistic regression

```
% function x = logistic(a, y, w)
%
% Logistic regression.  Design matrix X (nxm), targets y (1xn)
% n data elements, m regressors,
% optional instance weights W.
% Model is E(y) = 1 ./ (1+exp(-a-X*b)).
% Outputs are: regression coefficients b (1xm),
%              last iteration iter,
%              variance var

function [b, iter, covar]= logistic(X, y, w)


epsilon = 1e-10; %-10
ridge = 1e-1; % default 5, diego choose 1
maxiter = 100;
```

```
[n, m] = size(X);

a = ones(n,1);
%X = [a X];
%m = m+1;

if nargin < 3
    w = ones(n,1);
end

b = zeros(m,1);
oldexpy = -ones(size(y));

for iter = 1:maxiter
    adjy = X * b;
    expy = 1 ./ (1 + exp(-adjy));
    deriv = max(epsilon*0.001, expy .* (1-expy));
    adjy = adjy + (y-expy) ./ deriv;
    weights = spdiags(deriv .* w, 0, n, n);

    xwx = X' * weights * X + ridge*speye(m);

    %   if(cond(xwx)>100000000000)
    %       cond(xwx)
    %   end

    covar = inv(xwx);
    b = covar * X' * weights * adjy;

    if (sum(abs(expy-oldexpy)) < n*epsilon)
        break;
    end
    oldexpy = expy;
end
```

2.2 logistic regression implementation

```
%Cross-validation loss function using k=n; n observations (patterns);
% m features; two classes y [1, 0]
% X must include a first column of ones: size(x) == n x m+1

function [loss_ni, err, pval, loss, b] = logRegLoss(X, y);

[r,c] = size(y);
[n,m1] = size(X); % n x 1+m
m = m1-1;
if (r~=n || c~=1)
    y=y';
    if (c~=n || r~=1)
```

```
        'error: wrong y input'
        return;
    end
end
if X(:,1)~=ones(n,1)
    'error: fist column of x must be ones'
    return;
end

%B = [];
ind1= find(y==1);
ind0= find(y==0);

n1 = length(ind1);
n2 = length(ind0);
loss_ni =[];
loss  =[];
[b maxiter covar] = logistic(X,y);
for i=1:n
    Xni= X([1:i-1 i+1:n],:);
    yni= y([1:i-1 i+1:n]);
    [bni maxiter covar] = logistic(Xni,yni);
    %var =diag(covar);
    %z =(bni./sqrt(var));
    %B(:,i) = bni(2:m1);

    xi = X(i,:); % 1xm left-one-out
    yi = y(i);   % 1x1
    y_hat = 1./( 1+exp(-xi*bni) );

    loss_ni(i) = -2 * (yi*bni'*xi' - log(1+exp(bni'*xi')));
    loss(i)  = -2 * (yi*b'  *xi'   - log(1+exp(b'  *xi')));

    if y_hat<.5
        err(i) = yi~=0;
    else
        err(i) = yi==0;
    end
end
%binomial
p = .6;
%var = n*p*(1-p);

pval = binocdf(sum(err),n,p);
```

Appendix C

List of publications

3.1 Peer reviewed journal articles

2006 Sep:M. Bansal, G. Della Gatta *A. Ambesi-Impiombato*, C. Missero, and D. di Bernardo. "Integrated Computational and Experimental approach to identify the targets of p63 on Genome Scale". To be submitted.

2006 Sep:M. Bansal, *A. Ambesi-Impiombato*, V. Belcastro, and D. di Bernardo. "Systems Biology in practice: how to infer gene networks from gene expression profiles". Under review: **Molecular Systems Biology**.

2006 Sep: *A. Ambesi-Impiombato*, M. Bansal, P. Lio' and D. di Bernardo. "Transcription Factor Binding Sites prediction by integration of genomic, evolutionary, and gene expression data". **BMC Neuroscience 7(Suppl 1):S8**.

2005 Sep: *A. Ambesi-Impiombato*, D. di Bernardo; Computational Biology and Drug Discovery: from single-target to network drugs. **Current Bioinformatics, 1:3-13**.

2005 Sep: P. Carninci, T. Kasukawa, S. Katayama, J. Gough, M. Frith, N. Maeda, R. Oyama, T. Ravasi, B. Lenhard, C. Wells, R. Kodzius, K. Shimokawa, V. B. Bajic, S. E. Brenner, S. Batalov, A. R. R. Forrest, M. Zavolan, M. J. Davis, L. G. Wilming, V. Aidinis, J. Allen, *A. Ambesi-Impiombato*, ... (and 172 others); The Transcriptional Landscape of the Mammalian Genome. **Science 309 (5740): 1559-63**.

2005 Aug: F. Boscia, R. Gala, G. Pignataro, A. de Bartolomeis, M. Cicale, *A Ambesi-Impiombato*, G Di Renzo, L Annunziato; Permanent focal brain ischemia induces isoform-dependent changes in the pattern of Na(+)/Ca(2+) exchanger gene expression in the ischemic core, periinfarct area, and intact brain regions. **J Cereb Blood Flow Metab 26(4):502-17**.

2003 Jan: *A. Ambesi-Impiombato*, G. D'Urso, G.Muscettola, and A. de Bartolomeis; A method for quantitative in situ hybridization histochemistry and image analysis applied for Homer1a gene expression in the rat brain. **Brain Research Protocols 11(3):189-96**.

2002 Dec: D. Polese, A. Amato de Serpis, *A. Ambesi-Impiombato*, G. Muscettola and A. de Bartolomeis; Homer 1a Gene Expression Modulation by Antipsychotic Drugs: Involvement of the Glutamate Metabotropic System and Effects of D-Cycloserine. **Neuropsychopharmacology 27(6) 906-13.**

2002 Oct: M. Cicale, *A. Ambesi-Impiombato*, V. Cimini, G. Fiore, G.Muscettola, L. C. Abbott and A. de Bartolomeis; Decreased gene expression of calretinin and ryanodine receptor type 1 in tottering mice. **Brain Research Bulletin 59 (1) 53-8**.

2002 Jan: A. de Bartolomeis, L. Aloj, *A. Ambesi-Impiombato*, D. Bravi, C. Caraco, G. Muscettola, P. Barone. Acute administration of antipsychotics modulates Homer striatal gene expression differentially. **Brain Research Molecular Brain Research 98 (1-2): 124-9**.

3.2 Book Chapters

2007 Jan: *A. Ambesi-Impiombato*, M. Bansal, G. Della Gatta and Diego di Bernardo. Chapter 2: "Gene Networks and Application to Drug Discovery". In: Emanuele de Rinaldis and Armin Lahm **DNA Microarrays: Current Applications; Horizon Bioscience, Norwich U.K**.

2003 Mar: A. de Bartolomeis, *A. Ambesi-Impiombato*. "Le basi molecolari e biologiche della psicofarmacoterapia: implicazioni per la pratica clinica". In: G. B. Cassano, P. Panchehri, A. Pazzagli, L. Ravizza, R. Rossi, E. Smeraldi, V. Volterra; Trattato Italiano Psichiatria, 3rd edition; Masson, Milano. Italy. Volume I: Parmacoterapia psichiatrica, p. 41-85.

Index