Learning a Markov logic network for supervised gene regulation inference

application to the ID2 regulatory network in human keratinocytes

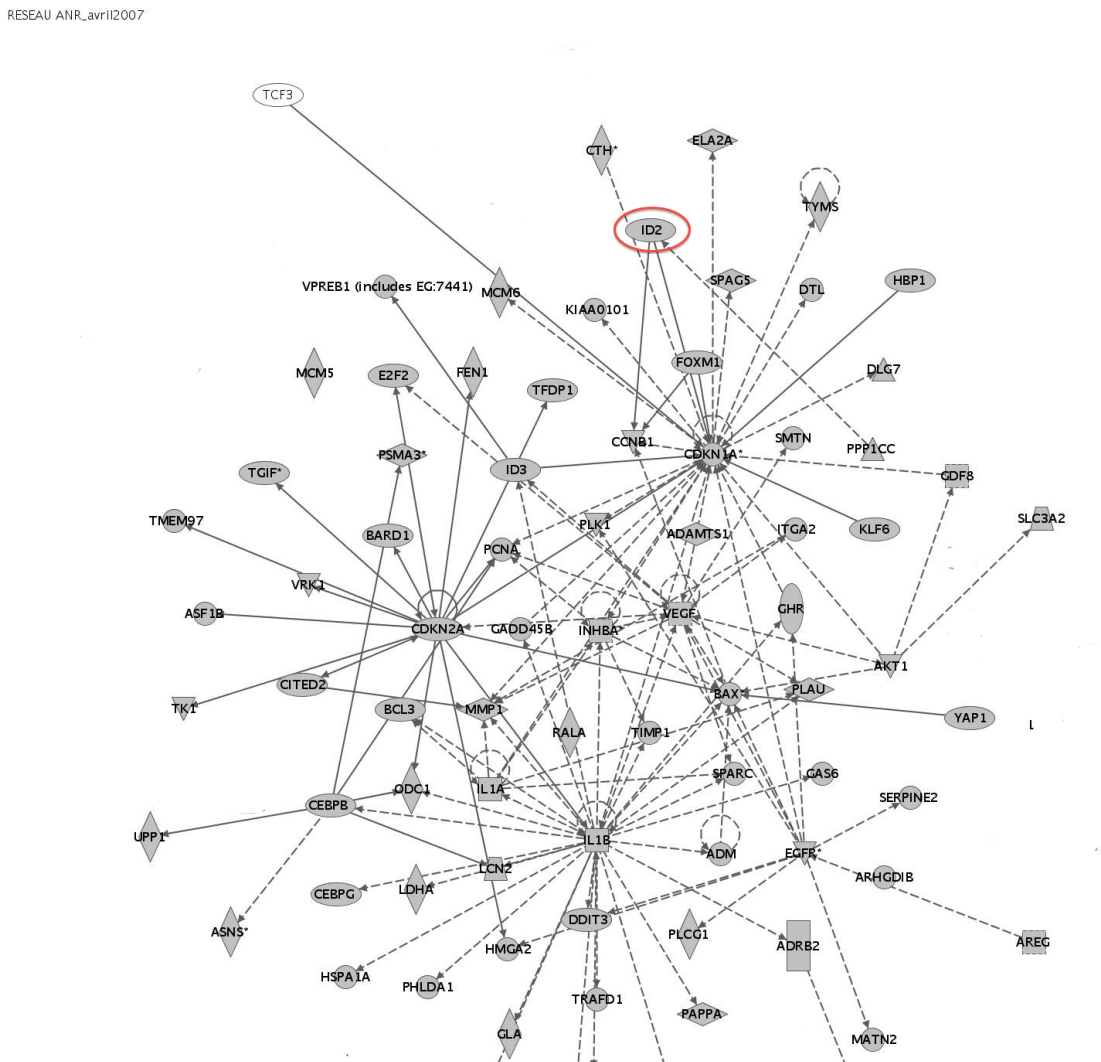
C. Brouard[1], C. Vrain[2], J. Dubois[1,2], D. Castel[3], M-A. Debily[3], F. d'Alché-Buc[1,4]

[1] IBISC, Université d'Evry Val d'Essonne
[2] LIFO, Université d'Orleans
[3] CEA, Evry
[4] INRIA, LRI umr CNRS 8623, Université Paris Sud, Orsay
France

September 8, 2012

Switch proliferation/differentiation of skin primary cells (human keratinocytes)

Source: Ingenuity



Switch proliferation/differentiation of skin primary cells (human keratinocytes)

- Collaboration with two biologists: Marie-Anne Debily and David Castel (CEA, Evry, France)
- This laboratory (Xavier Gidrol) has identified protein ID2 as a major component in this switch
- **Experimental data**: Transcriptomic analysis by microarray experiments of HaCaT cells presenting stable overexpression or transient knock-down achieved by RNA interference of ID2 expression.
- **Existing network**: provided by Ingenuity (text-mining) on a subset of 63 differentially expressed genes $\approx$ 157 known regulations
- **Background knowledge**: cellular localization of proteins, biological processes, protein-protein interactions, position of genes on chromosomes

Goal of the study

Given a gene regulatory network provided by Ingenuity (text-mining), confront it to experimental data and background knowledge, build a method able to complete the network with new candidate genes

Machine Learning for biological network inference

Two main families of methods

- Modeling the behavior of the network as a (dynamical) system
- Modeling/predicting edges in the graph: given an ordered pair of genes (A,B) ,predict if A regulates B

Modeling/predicting edges in the graph

- **Supervised network inference**: (PPI) pairwise SVM [Ben-Hur and Noble, 2005, Hue and Vert, 2010], mixture of feature experts [Qi, 2008], KCCA [Yamanishi et al., 2004], metric learning [Yamanishi and Vert, 2005], output kernel regression tree [Geurts et al., 2006;2007]; (GR) local classifiers [Bleakley et al. 2007], [Mordelet at al. 2008]
- **Semi-supervised network inference**: PPI: Kernel Matrix completion using EM [Tsuda et al., 2003], [Kato et al., 2005], Link Propagation [Kashima et al., 2009], Training set expansion [Yip and Gerstein, 2009] , Operator-valued kernel [Brouard et al. 2011]
- **Unsupervised**: (GR) Gaussian graphical models [Shafer and Strimmer et al. 2005], [Wille and Buehlman et al.2006]

Our approach: learning a Markov Logic Network

Motivation

- Supervised link prediction
- Combine the efficiency of statistical learning with the interpretability of first order logic

Proposed solution

- Build a classifier based on a set of weighted first order logic rules that conclude on the target predicate "Regulates": if Propr1(A,C) and Propr2(B,D) and Prop3(A,B) then Regulates (A,B).
- Markov Logic network recently introduced by Domingos et al. 2006

Outline

Using first order logic to encode data

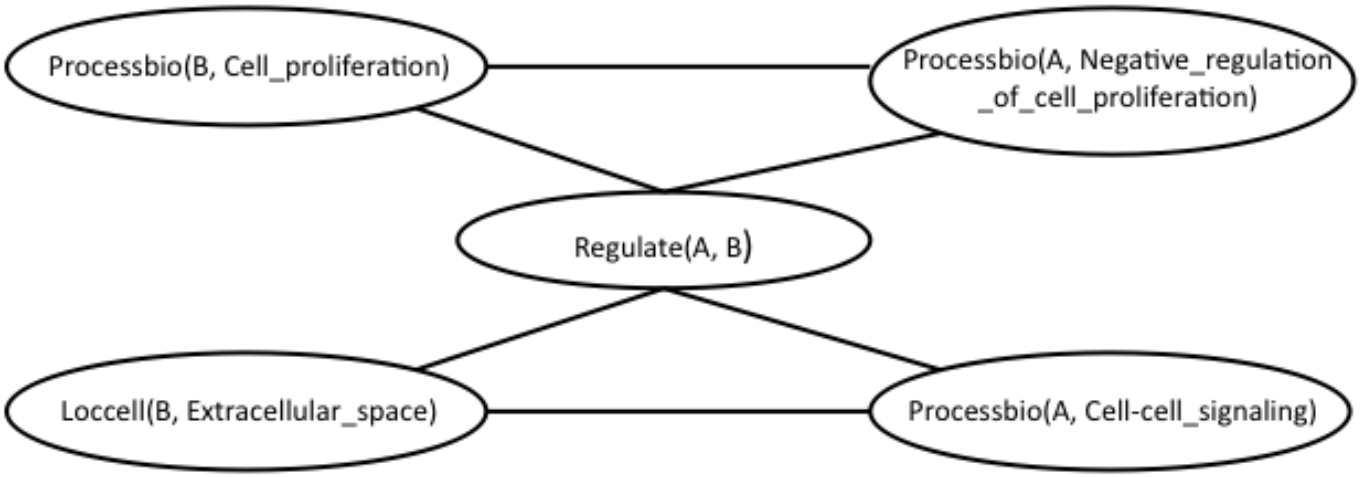
- **Variables** : gene, (protein), level, loc, process
- **Constants** : Id2, Cdkn2a,Cytoplasm,...
- **Atoms** : $P(t_1, \ldots, t_n)$,
  where P is a predicate ans $t_1, \ldots, t_n$ are variables or constants
  - $Loccell(Akt1, Cytoplasm)$
  - $Regulates(x, y)$
- A ground atom is an atom with no variable, only constants; It can be true or false
- A *possible world*: an assignment of truth values to all possible groundings of predicates

Predicates encoding experimental data and prior knowledge

- Regulation: Regulates (gene1,gene2)
- Expression data :
  - Expwt(gene, level), Expsiid2(gene,level), Expprcid2(gene,level)
  - For instance, Expsiid2(G,L) states that the level of expression of gene G is L when the level of expression of ID2 has been increased
- Position on chromosomes :
  - Samechro(gene1, gene2), Sameband(gene1, gene2)
- Biological processes to which genes are contributing :
  - Processbio(gene, process)
- Cellular localization of proteins
  - Loccell(protein, loc)
- Physical interaction between proteins ...
- Links between a gene and a protein ...

Structure of a small Markov Logic Network (example)

- A MLN is a set $\mathcal{F}$ of formula (clauses) and a weight vector (each formula is weighted)
- Together with a finite set $\mathcal{C}$ of constants, among which the variables can take their values, a MLN defines a Markov Network.
- node: a ground atom
- edge: each time two ground atoms appear in the same ground formula



Markov Logic Network (MLN)

- Let $\mathcal{X}$ be the set of all propositions describing a world
- w_i is the weight (positive or negative) associated with the clause $f_i \in \mathcal{F}$, and $\mathcal{Z}$, the normalizing constant
- Then, the probability of a particular truth assignment x of variables in $\mathcal{X}$ is given by the formula:

$$P(\mathcal{X} = x) = \frac{1}{\mathcal{Z}} \exp(\sum_{f_i \in \mathcal{F}} w_i n_i(x))$$

$n_i(x)$ is the number of true groundings of f_i in x and $\mathcal{Z}$ known as the partition function is the normalization coefficient

MLN for a supervised prediction of a regulation

Notations

- Let $\mathcal{Y}$ the set of query atoms (regulate predicate)
- $y = (y_{11}, \ldots, y_{nn})$ where y_{ij} correspond to the instantiated predicates **Regulate(**G_i**,** G_j**)**
  and thus to the labeled data.
- x correspond to all the other instanciated predicates

Modeling the posterior probability of a regulation between i and j

$$P(\mathcal{Y} = y_{ij}|x, w) = \frac{\exp(\sum_{k \in \mathcal{F}_{y_{ij}}} w_k n_k(x, y_{ij}))}{\sum_{t=0,1} \exp(\sum_{k \in \mathcal{F}_{y_{ij}}} w_k n_k(x, y_{|Y_{ij}=t}))}$$

Discriminative learning of weights given the structure

Maximization of the penalized conditional log-likelihood

$$\mathcal{L}(w) = \log P(\mathcal{Y} = y | \mathcal{X} = x, w) + \log P(\mathcal{X} = x, w) \quad (1)$$

$$\approx \sum_{i,j=1}^{n} \log P(\mathcal{Y}_{ij} = y_{ij} | \mathcal{X} = x, w) + \log P(w) \quad (2)$$

$$(3)$$

- ℓ_2 norm: P(w) $\propto \exp(-\lambda \parallel w \parallel^2)$
- Implementation with Alchemy (Kok et al. 2007)
- N.B. Sparse models with ℓ_1 constraint also possible not implemented here

Discriminative learning of the structure

- Used tool: **Aleph** (Srinivasan, 2001), Inductive Logic Programming
  - Selection of a positive example
  - Construction of the most specific rule satisfied by this example
  - Generalization of this rule by a top-down search
  - The process is iterated until all the positive examples be covered

Description of the experimental studies

Gene regulatory network associated with *ID*2 in human cells:

- $\mathcal{G}_A$: set of the 63 genes of interest
- Regulations between these genes were obtained using Ingenuity

We conducted three numerical studies to assess the performance of our method:

1. Cross-validation measurements on a well-balanced classification task
2. Updating the network using asymmetric bagging
3. Inference of regulations with a new set of genes using asymmetric bagging

The two last studies were defined with the biologist Marie-Anne Debily and considered by her as necessary *in silico* assessment before processing to new wet experiments.

Comparison using a baseline pairwise SVM

- Pairwise SVM [Ben-Hur and Noble, 2005, Hue and Vert, 2010]
- A kernel between ordered pairs of genes is built using a kernel k between single data :

$$K((G_1, G_2), (G_3, G_4)) = k(G_1, G_3)k(G_2, G_4).$$

- Definition of six gaussian kernels k_i for each feature previously described
- Two ways of combining kernels:

$$K_{pairwisesum}((G_1, G_2), (G_3, G_4)) = \frac{1}{6}\sum_{i=1}^{6} k_i(G_1, G_3)k_i(G_2, G_4)$$

$$K_{sum}((G_1, G_2), (G_3, G_4)) = \bar{k}(G_1, G_3)\bar{k}(G_2, G_4)),$$

where $\bar{k}(G_j, G_k) = \frac{1}{6}\sum_{i=1}^{6} k_i(G_j, G_k)$.

Averaged cross-validation measurements on balanced samples (1)

- Genes of $\mathcal{G}_A$
- R_1: dataset labeled using Ingeniuity in 2007
- R_1^+: set of 106 positive examples of regulations between genes of $\mathcal{G}_A$
- R_1^-: set of all the "negative" examples (no regulation proven)
- 30 samples of negative examples $R_{1,i}^-$, $i = 1, \ldots, 30$ randomly sampled from R_1^-
- With each set $R_1^+ \cup R_{1,i}^-$: 10-fold cross-validation for each set
- Evaluation metric: averaged AUC-ROC and AUC-PR values obtained within a large range of values of the regularization parameter λ (resp. C) of the MLN (resp. SVM).

Averaged cross-validation measurements on balanced samples (2)

λ	MLN	
	AUC-ROC	AUC-PR
20	80.8 ± 6.1	82.7 ± 5.4
50	84.3 ± 3.5	85.5 ± 4.0
100	$\mathbf{84.4 \pm 2.8}$	$\mathbf{86.2 \pm 3.2}$
500	83.4 ± 2.7	86.0 ± 2.7
750	83.3 ± 2.8	85.8 ± 2.8

Pairwise SVM				
C	Pairwise sum		Sum	
	AUC-ROC	AUC-PR	AUC-ROC	AUC-PR
10^{-3}	70.9 ± 3.5	73.1 ± 3.4	82.5 ± 2.3	84.3 ± 2.1
10^{-2}	70.9 ± 3.5	73.1 ± 3.4	82.5 ± 2.3	84.3 ± 2.1
10^{-1}	70.9 ± 3.5	73.1 ± 3.4	82.5 ± 2.3	84.3 ± 2.1
1	76.4 ± 3.1	78.7 ± 3.0	$\mathbf{85.2 \pm 2.8}$	$\mathbf{87.3 \pm 2.5}$
10^{1}	77.5 ± 3.2	79.4 ± 3.5	84.3 ± 3.4	86.3 ± 3.1
10^{2}	77.5 ± 3.2	79.4 ± 3.5	84.3 ± 3.4	86.3 ± 3.1
10^{3}	77.5 ± 3.2	79.4 ± 3.5	84.3 ± 3.4	86.3 ± 3.1

Network completion with a new set of genes (1)

- TRAINING SET: R_2 contains all the ordered labeled pairs between genes of $\mathcal{G}_A$ (updated data 2009)
- TEST set R_3: containing all the ordered pairs between genes of $\mathcal{G}_A$ and $\mathcal{G}_B$
- Goal: test the ability of the classifier to label correctly the regulations between the genes of $\mathcal{G}_A$ and $\mathcal{G}_B$
- The test was made under real conditions: the whole set of positive (55) and negative examples (2969) of R_3 was considered to assess the performance in prediction.

Asymmetric bagging

- Bootstrap sampling only on the over-represented class
- Each generated predictor is trained on a balanced dataset
- Average of their predictions on the test set to provide a single prediction

Network completion with a new set of genes (2)

Bagged MLNs		
λ	Auc-roc	Auc-pr
50	72.8	6.7
100	73.1	7.7
500	73.2	9.2
750	**73.4**	9.5
1000	73.1	9.5
5000	73.0	**9.8**
10000	72.8	9.5

Bagged pairwise SVMs				
C	Pairwise Sum		Sum	
	Auc-roc	Auc-pr	Auc-roc	Auc-pr
0.001	62.8	4.0	66.2	7.8
0.01	62.8	4.0	66.2	7.8
0.1	62.8	4.0	66.2	7.8
1	65.3	7.7	67.4	**8.6**
10	65.4	6.1	**67.5**	8.3
100	65.4	6.1	67.5	8.3
1000	65.4	6.1	67.5	8.3

And what about the rules ?

- Aleph did not find rules involving the positions of genes on chromosomes
- Examples of rules with a high positive weight:
  - 0.20 $ProtLoccell(g_2, Plasma\_membrane) \wedge Expsiid2(g_2, Level3) \wedge Expsiid2(g_1, Level3) \Rightarrow Regulates(g_1, g_2)$
  - 0.30 $Processbio(g_2, Cell\_proliferation) \wedge Processbio(g_1, Negative\_regulation\_of\_cell\_proliferation) \Rightarrow Regulates(g_1, g_2)$
- Promising results but it should be possible to find more relevant rules given some constraints on the rule learner
- More relevant rules if data are richer (for instance kinetics during the switch)

Conclusion and perspectives

- First-order logic as a framework to encode heterogeneous data, readable by biologists: not a black box
- Consistency of the built classifier with the experimental data and available knowledge
- In this example, MLN performs as well as SVM in artificial tasks and better in the realistic completion task
- **Future work on rules extraction**: (1) rules can be improved, constraints on the kind of rules to be built by Aleph must be imposed, (2) rules less numerous (sparse modeling)

- 2 Postdoc positions open at IBISC, Genopole Evry and INRIA, LRI University of Paris Sud, France
  - 1-year postdoc position on protein-protein interaction network inference (CFTR network) with Alexander Edelman (Necker Hospital) and Christine Froidevaux (Paris Sud))
  - 2-year postdoc position on Dynamical modeling for understanding of endothelium dysfunctions in normal tissues following ionizing radiation exposure with Olivier Guipaud (IRSN, Paris)
  - CONTACT : florence.dalche AT ibisc.fr

2. Updating a graph (1)

- Still genes of $\mathcal{G}_A$
- R_2^+: set of regulations between the 63 genes of interest obtained with Ingenuity two years after the construction of the first dataset.
- 51 new regulations were discovered by Ingenuity between these two dates
- Prediction of the updated graph : use $R_1 - R_2^+$ to see if we could retrieve the new regulations in $R_2^+ \setminus R_1^+$ using **asymmetric bagging**

Asymmetric bagging

- Bootstrap sampling only on the over-represented class
- Each generated predictor is trained on a balanced dataset
- Average of their predictions on the test set to provide a single prediction

2. Updating a graph (2)

- Positive training set: dataset R_1^+
- 30 subsamples $R_{1,i}^-$ from $R_1^- \setminus R_2^+$, such that $|R_{1,i}^-| = |R_1^+|$
- For each sampling, the predictor learned was applied to the 51 new regulations and the predictions obtained were averaged.

Selection of a threshold θ:

- For each sampling,
  - $\frac{2}{3}$ of R_1^+ and $R_{1,i}^-$ considered for the training set and $\frac{1}{3}$ for a validation set.
- The F_1-measure was computed on each validation set for different thresholds:

$$F_1 = 2.\frac{Precision.Recall}{Precision + Recall}$$

- Selection of the threshold maximizing the averaged F_1-measure, that is maximizing precision and recall at the same time.

2. Updating a graph(3)

- Prediction on pairs of genes in R_2^+

Bagged MLNs	
λ	TPR
20	64.7
50	64.7
100	72.6
500	80.4
750	84.3
1000	**90.2**
2000	88.2
5000	84.3

Bagged pairwise SVMs		
C	Pairwise sum	Sum
	TPR	TPR
0.001	**90.2**	58.8
0.01	88.3	58.8
0.1	88.3	58.8
1	74.5	52.9
10	64.7	43.1
100	64.7	43.1
1000	64.7	43.1