A Protocol to Detect Local Affinities Involved in Proteins Distant Interactions

Christophe N. Magnan Cécile Capponi
François Denis
Laboratoire d'Informatique Fondamentale de Marseille, CNRS, Aix-Marseille I,
39, rue F. Joliot Curie, 13013 Marseille, France
{Firstname.Lastname}@lif.univ-mrs.fr

Abstract

The tridimensional structure of a protein is constrained or stabilized by some local interactions between distant residues of the protein, such as disulfide bonds, electrostatic interactions or hydrogen links. The in silico *prediction of the disulfide connectivity has been widely studied: most results were based on few amino-acids around bonded cysteines, which we call* local environments of cysteines*. In order to evaluate the impact of such local information onto residue pairing, we propose a machine learning based protocol, independent from the type of contact, to detect affinities between local environments which would contribute to residues pairing. Finally, we experiment our protocol on proteins that feature disulfide or salt bridges. The results show that local environments contribute to the formation of salt bridges. However, results on disulfide bridges are not significantly positive with the class of linear functions used by the perceptron-type algorithm we propose.*

1. Introduction

The prediction of proteins tridimensional (3D) structure starting from primary amino acids sequence is a current challenge for both biologists and computer-scientists. About 5 millions of proteins sequences are available in different databases, whereas approximatively only 40.000 3D structures are known. Moreover, determining 3D structures experimentally is a long, expensive and unreliable task. It is the reason why many researchers work at developing automatic learning methods for predicting the structure of new proteins from experimentally designed structures.

The most widespread bioinformatic method to determine proteins structures consists in predicting different structural elements and long-range contacts, then to propose the set of 3D structures matching these predictions. The prediction of the secondary (2D) structure have received considerable attention from researchers. β-sheets or α-helix are examples of 2D structure elements.

Some punctual interactions between distant residues of the primary sequence of a protein are also of interest even if they are sometimes considered as a consequence of the 3D conformation rather than a cause. Among them, *disulfide bridges* have been widely studied. The prediction of such covalent bonds from primary sequences is a two-stages process: (i) prediction of the oxidization state of cysteines; (ii) prediction of the disulfide connectivity: which cysteine is bonded with such other given oxidized cysteine?

The first stage has been widely studied [6, 3], leading to fairly good performances. Some methods have been proposed for the prediction of the correct connectivity [9, 4]. However, none have reached 63% of correct connectivity despite their strong biological, theoretical, and algorithmic foundations. Actually, most of these methods use the local environments around cysteines to predict their pairing (*i.e.* the 5 ou 6 amino-acids that range on each side of each cysteine in the primary sequence). Hence, in order to improve automatic methods, it is worthwhile to wonder which information is actually useful for predicting the disulfide connectivity. Basically, are local environments informatory?

In order to evaluate the impact of local environments of cysteines onto disulfide connectivity, we study an experimental protocol aiming at revealing a potential *affinity* between oxidized cysteines for further bonding. Our hypothesis is that, before the protein folds, some couples of cysteines are more likely to bond than others, given the amino-acids of their primary sequence neighborhood. Such an affinity should then be involved in the observed bridges. Assuming that such an affinity exists, we suppose that we can detect, extract and evaluate it with machine learning methods. More generally, we realize this study among several types of contacts which may be driven by local information.

We thus propose a formalization of the affinity between residues: we focus on a protocol for learning a function representing this affinity from labeled examples available in databases. Starting from machine learning considerations, the main idea of our proposal is to assume that actual bonded residues (positive examples) are not the only ex-

amples of high propensity residue pairs: some non-bonded cysteines might also be propitious to form a disulfide bridge according to their neighborhood while some other information does not allow them to actually bond. In previous works, observed bonded residues are considered as positive examples, while non-bonded residues are definitely negative examples: pairs that cannot contact. We argue that our hypothesis may be used for improving usual machine learning methods for predicting residue connectivity. Indeed in a previous work [5], we considered non-bonded residues as unlabeled examples: we then obtained better predictive performances, using a naive bayesian classifier, than when non-bonded residues where labeled as negative examples.

In these preliminary works, we considered that there exists an affinity function g, defined on pairs of local environments, which can only take two values: 1 means a high affinity between both environments, while 0 reveals a low affinity. We postulate that pairs with high affinity are more likely to bound than pairs with low affinity. Thus, bounded and unbounded pairs available in proteins databases can be considered as *examples of pairs labeled with* g. Furthermore, these pairs might have been corrupted with classification noise: not all unbonded pairs (resp. bonded pairs) have low (resp. high) affinity. Such a model of noise have already been studied in the machine learning litterature. It is refered to as *class-conditional classification noise* (CCCN). If our base hypothesis is correct, a learning algorithm that is capable of learning from such noisy data, should be able to learn the affinity function g from pairs of cysteines issued from the proteins databases. Then we should be able to prove that local environments carry some information on the connectivity by checking that pairs with high affinity are more likely to be bonded than pairs with low affinity.

Section 2 is concerned with the presentation and the formal modelling of the biological problem in terms of machine learning methods. Section 3 concerns the algorithms that we propose to learn the affinity function, which are proven to be efficient in some noisy contexts. Section 4 reports some of the numerous experiments we performed: a discussion on the presented results is worthwhile and we hope it will give rise to advices from the whole community of structural biologists and computer scientists.

2 Affinity of protein distant interactions

2.1 Disulfide bridges and salt bridges

A protein may be represented by its primary structure – a sequence of amino-acids– from which a 3D structure is gathered. Nowadays, some interactions are known that contribute to protein stability, such as hydrogen links, electrostatic interactions, covalent bonds. As a matter of fact, the prediction of such interactions should be of great help for the prediction of the structure from the sequence. We are interested by the prediction of affinity between cysteines to bond, making up disulfide bridges. However, the protocol we study can be applied on other punctual contacts.

Disulfide bridges are involved in the 3D structure of a protein as covalent bonds between two oxidized cysteines (amino-acid C). Such a physical interaction is a strong, well-conserved link, thus a strong constraint for the stability of the protein structure. Experimental ways of determining them, through RMN, X-ray crystallography or site-directed mutagenesis, is a long and expensive process.

Salt bridges are relatively weak ionic hydrogen bonds made up of the interaction between two charged residues. As disulfide bonds, they contribute to the stability of the structure.

2.2 A model of affinity between residues

We present a model and a protocol to detect neighborhood affinity implied in the formation of interaction between two distant residues of a protein. We present the protocol through disulfide bonds, but other bonds are also directly concerned as long as the distant contacts involve few residues.

2.2.1 Modeling the data

The primary structure of a protein p can be considered as a word w of Σ^* where Σ represents the set of twenty amino acids or any other similar alphabet. Let $\mathcal{P} \subset \Sigma^*$ be the set of proteins containing an even number of cysteines involved in disulfide bridges (oxidized cysteines). Let $\mathcal{P}_l \subset \mathcal{P}$ be the proteins with $2l$ cysteines involved in disulfide bridges.

Let $\mathcal{G}$ be the set of non-oriented graphs where nodes have degree 1. For a protein $p \in \mathcal{P}$, nodes of the associate graph in $\mathcal{G}$ represent oxidized cysteines of p, and an edge represent a disulfide bond between two cysteines of p. Let $\phi : \mathcal{P} \rightarrow \mathcal{G}$ be a function which associates a graph in $\mathcal{G}$ (the disulfide connectivity) to a protein in $\mathcal{P}$. Then, our aim is to approximate the function ϕ with the highest precision, using examples issued from experiments.

To do so, many authors use local environment of cysteines, *i.e.* amino acids located around the cysteines. Usually, segments centered on cysteines of size $2r+1$ are considered. Let P be a probability distribution over $\mathcal{P}$ and let $\Omega_r = \Sigma^{2r+1}$ be the set of protein segments of size $2r+1$. The elements of Ω_r are local environments of cysteines, also called *windows*: a sequence of residues whose center is an oxidized cysteine. For $w, w' \in \Omega_r$, let $P(w)$ be the probability that w is a local environment of a cysteine into a protein $p \in \mathcal{P}$, $P(w, w')$ be the probability that w and w' are distinct local environments of a cysteine into a protein $p \in \mathcal{P}$, $P(w, w'|l)$ be the probability that w and w' are distinct local environments of a cysteine into a protein $p \in \mathcal{P}_l$ and

$P(B(w,w')|w,w',l)$ be the probability that w and w' are bonded knowing that there are distinct local environments of cysteines into a protein $p \in \mathcal{P}_l$.

2.2.2 Modeling the role of local environment on bonds

Past results of automatic methods for the prediction of disulfide bridges based on the proteins sequence are not satisfactory. The error rate remains high (about 40%) while the results are not stable. Most of these methods relies upon the local environments of cysteines, namely the environments w modeled above. Our aim is to answer the question: *is there local information involved in the formation of disulfide bonds*? Is there any information in the neighborhood of the cysteines that would help to predict their bonding?

In order to answer that question, an affinity measure among cysteines based on their local environment must be highlighted through a functional represesention. The affinity between cysteines must be considered as a necessary, but not sufficient, condition for their actual physical distant interactions. We assume that if such a function exists, then there is a way to learn it from examples. In this section, we draw a model of affinity as well as a protocol to learn it from known disulfide bridges.

Let p be a protein with l bridges ($2l$ involved cysteines). If there is no local information for pairing cysteines into bridges, then there is $2l-1$ pairing possibilities for each cysteine, so $P(B(w,w')|w,w',l)=\frac{1}{2l-1}$. Reciprocally if $P(B(w,w')|w,w',l)=\frac{1}{2l-1}$, there is no local information since the actual pairing does not depend on w nor on w'.

Such an equivalence provides us a probabilistic way to determine if the local context of oxidized cysteines is involved into the formation of the bridges, which requires the estimation of $P(B(w,w')|w,w',l)$. However, estimating $P(B(w,w')|w,w',l)$ without additional hypotheses is impossible. Indeed, with $r = 3$ (which means that we only consider 3 amino-acids on each side of the cysteine in the sequence), the solution space is of size $|\{(w,w'),w,w' \in \Omega_r\}| = 20^{12} \simeq 4.10^{15}$, while only few hundreds examples are available in databases!

Our solution is to assume the existence of an affinity function $g : \Omega_r \text{ x } \Omega_r \rightarrow Y$ such as: $g(w_1,w_2) = g(w_1',w_2') \Rightarrow P(B(w_1,w_2)|w_1,w_2,l)=P(B(w_1',w_2')|w_1',w_2',l)$ and $y<y' \Rightarrow P(B(w_1,w_2)|g(w_1,w_2)=y) < P(B(w_1',w_2')|g(w_1',w_2')=y'),\ y,y' \in Y$. The simplest case is $Y=\{0,1\}$ (Figure 1: 0 means low affinity between local environments, whereas 1 means a high affinity). In such a case, pairs of windows are partitioned into two classes, corresponding to two affinity levels and $P(B(w,w')|w,w',l) =$

$$P(B(w,w')|g(w,w'),l) = \begin{cases} \alpha_1^l \ if \ g(w,w') = 1 \\ \alpha_0^l \ if \ g(w,w') = 0 \end{cases}$$

Assuming that g exists and plays a role in the interaction, then we must have α_1^l significantly higher than α_0^l. In other words, there are more bonded cysteines when there is a high affinity between their local environments ($g(w,w')=1$) than when there is a low affinity ($g(w,w')=0$).

The observed bonded and unbonded pairs of local environments centered on oxidized cysteines are then indirect information on g since our model does not exclude that pairs with a high affinity level could be unbonded, neither that a bridge can hold among a pair of cysteines for which $g=0$.

2.2.3 Observed cysteines pairs as noisy exemples of g

The observed classes (bonded or non-bonded) of observed examples (local environments of cysteines) do not carry direct information about the affinity function g. Some pairs of environment are labeled "bonded" while their affinity is 0, and vice-versa. Such a phenomenon is quite usual in machine learning: we interpret these mislabeled pairs as *noisy* labels with regards to the function g. More precisely, using the previous expression of $P(B(w,w')|g(w,w'),l)$, one can observe that pairs such that $g = 1$ correspond to the observation of a bridge with noise $\eta^+ = 1 - \alpha_1^l$, and, symetrically, pairs such that $g = 0$ correspond to a non-bonded pairs with noise $\eta^- = \alpha_0^l$. One could observe that the noise is somehow a measure of mislabelling rates.

On the one hand, this kind of noise is a generalization of the *uniform classification noise* (CN) where it is supposed that positive and negative examples are corrupted according to the same noise rate ($\eta^+ = \eta^-$). On the other hand, it is a particular case of *constant-partition classification noise* (CPCN) where it is supposed that the description space is partitioned into a finite number of regions in each of which the noise rate is constant. Such a noise has been studied in [7], it is refered as *class-conditional classification noise* (CCCN). Most of learning algorithms that are tolerant to noise, such as soft-margins SVM, cannot handle data corrupted by CCCN except if the noise rates are very small, which is not the case in our application. New methods have to be created.

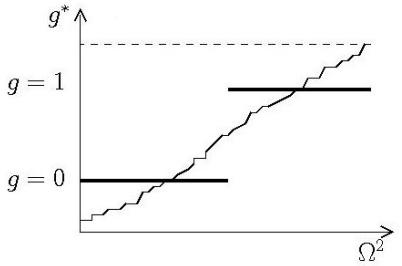

Figure 1. A two-levels affinity function g in function of the values of $g^* = P(B(w,w')|w,w',l)$. The pairs (w,w') of local environments are ordered by the value of g^*.

2.2.4 Setting up the protocol to learn g

If a local information exists (*i.e.* if local environments of cysteines contribute to their pairing), and if it can be represented by a function learnable under CCCN, then we should be able to detect, extract and evaluate it. Our study thus concerns the setting up, and the test, of a protocol for learning the hypothetic affinity function from proteins where disulfide bridges are known.

In section 3, we propose an efficient algorithm to learn *the linear threshold functions* under CCCN. This algorithm is a generalization of the perceptron algorithm. [2, 1] sketched two adaptations of it in the CN learning context that we generalize in the CCCN context. We thus propose a perceptron in order to apply our protocol on real datasets (section 4).

3 A CCCN-learning perceptron algorithm

In order to experiment the protocol proposed in section 2, this section briefly explain how the perceptron algorithm can be adapted for data corrupted by a CCCN noise.

Let $S = \{(x_1, l(x_1)), ..., (x_n, l(x_n))\}$ be a set of labeled examples[1] (for instance, the set of all pairs of local environments of cysteines), such that $x_i \in \mathbb{R}^m$ (m the size of the descriptive attributes space) and $l(x_i) \in \{-1, 1\}$. S is linearly separable by a hyperplane H if the positive and negative examples of S are separated by H. We refer to a *linearly separable set* S if S is separable by a hyperplane which passes by the origin[2]. Such a hyperplane H is identified by a vector $w^* \in \mathbb{R}^m$ such that $||w^*|| = 1$ and $\forall x \in \mathbb{R}^m$, $x \in H$ if and only if $x \cdot w^* = 0$. We say that w^* separates examples in S with margin σ if $min_{x \in S} |cos(w^*, x)| = min_{x \in S} w^* \cdot l(x)\bar{x} = \sigma$ ($\bar{x}=\frac{x}{||x||}$).

In this context, our first purpose is to infer, from a linearly separable set S, a hypothesis w such that w separates positive and negative examples of S: $\forall (x, 1) \in S$, $w \cdot x > 0$ and $\forall (x, -1) \in S, w \cdot x < 0$, or $\forall (x, l(x)) \in S, w \cdot l(x)x > 0$.

3.1 Perceptron Algorithm

The perceptron algorithm [8] is an iterative method for infering a hyperplane w passing by origin, that separates a linearly separable dataset S. A sketch of this algorithm is given on algorithm 1. In the usual form, $x_{upd} = l(x_B)\bar{x}_B$, where x_B is an example wrongly classified by the current hyperplane w, and $\bar{x}_B = \frac{x_B}{||x_B||}$. This algorithm requires at most $\frac{1}{\sigma^2}$ iterations, where σ is the maximal margin among hyperplanes separating S [1].

[1] We use annotations 1 or $+$ for bonded pairs – positive examples –, and either -1 or $-$ for unbonded pairs – negative examples –.

[2] One may transform any set S, linearly separable, by a hyperplane H that does not pass the origin, into another set S' that is linearly separable by a hyperplane H' passing the origin.

Algorithm 1 Sketch of the perceptron algorithm

Require: $S = \{(x_1, l(x_1)), (x_2, l(x_2)), ..., (x_n, l(x_n))\}$
  $w = \overrightarrow{0}$
  while $\exists (x, l(x)) \in S$ such that $w \cdot l(x)x < 0$ **do**
    let x_{upd} be such that $w^* \cdot x_{upd} > 0$ and $w \cdot x_{upd} < 0$
    $w = w + x_{upd}$
  end while
Ensure: w such that $\forall (x, l(x)) \in S, w \cdot l(x)x > 0$

Others possibilities exist for choosing x_{upd}, for instance $\sum l(x_B)x_B$, or any other colinear vector such as the average of the misclassified examples, or its normalized sum.

3.2 Classification noise

Observed classes of the examples may be corrupted by a noise process: the assigned class for some examples may be wrong, for any reason. The most studied noise process is the uniform classification noise (CN), where the labels of examples are supposed to be independently flipped with a constant noise rate $\eta < 0.5$. Two adaptations of the perceptron algorithm in a CN context have been proposed in [2, 1]. The second one presents a direct analysis which computes, when η is known, an estimated value of $\sum l(x_B)x_B$ where x_B are misclassified examples[3]. In order to select a good hypothesis when the noise rate is unknown, it is usual to scan the rate within $[0, 0.5[$ for selecting the hypothesis that leads to the smallest error.

3.3 CCCN-Perceptron Algorithm

We generalize this noise process by assuming that the noise rate over positive examples is not the same than the noise rate over negative examples, *i.e.* η^+ and $\eta^- \in [0, 1]$, $\eta^+ + \eta^- < 1$ to avoid any ambiguity. Introduced in [7], this new kind of noise was refered to as *class-conditional classification noise* (CCCN).

As in a CN context [1], an estimate of the value of $\sum l(x_B)x_B$ can be obtained from data corrupted by CCCN noise when η^+ and η^- are known. Proof and formulas are not reported in order to stay in the scope of this paper. For more details, the reader is adviced to refer to [1, 7] or to a complete version of this article[4].

When the noise rates are unknown, it is necessary to scan the interval $[0, 1]$ for the values of η^+ and η^- with a step $s \geq \frac{1}{n}$ where $n = |S|$. The algorithm is then launched for each pair for computing a hypothesis. However, the empirical risk minimization principle does not necessarily hold in CCCN context. We thus propose a consistent criterion to se-

[3] $l(x_B)$ is the correct label of x_B.

[4] http://hal.archives-ouvertes.fr/hal-00167520/fr/

lect a hyperplane when data is corrupted by a noise CCCN (proof is not given here[5]).

4 Experimentations

We tested the protocol presented in section 2 on two datasets featuring disulfide and salt bridges. We applied the algorithm proposed in section 3 in order to learn an affinity function supposed to be involved in the pairing of residues.

4.1 Protocol

We ran the protocol over two proteins datasets featuring proteins which contains from two to five bonds. The first dataset contains experimentally observed disulfide bridges in proteins; it is known as SPX [4], featuring 1676 disulfide bridges within 567 proteins. The homology rate of proteins of SPX is smaller than 30%.

The second dataset compiles 1836 intern salt bridges in 570 proteins; we call it G3D, for it was created from PDB by the ACI GENOTO3D consortium, a french group working on the prediction of the 3D structure of proteins. The homology rate of proteins is smaller than 25%.

For both kinds of bonds, we distinguish the study according to the number of bonds, since the noise rates induced by proteins containing k bridges are different from the noise rates induced by proteins containing $l \neq k$ bridges (section 2.2.3). Indeed, $l(2l-1)$ pairs of cysteines could be formed within a protein containing $2l$ oxidized cysteines, but only l pairs are actually bonded while $2l(l-1)$ remain unbonded.

4.1.1 Coding of local environments pairs

From a protein p_l containing l bonds ($2l$ oxidized cysteines), we extract $l(2l-1)$ pairs of local environments centered on a cysteine, with radius r (*i.e.* windows of size $2r+1$). We set r to 6 (*i.e.* local environments of size 13, including the central cysteine, because it corresponds to the best results we and other authors have obtained so far). For any pair (w_i, w_j) of p_l local environments, we extract 169 residue pairs (A_i, A_j) $(i,j \in \{1,...,13\})$ where $A_i \in w_i$ and $A_j \in w_j$. Each pair (w_i, w_j) is modeled with a vector of $\mathbb{R}^m$, where m is the number of ordered pairs of residues within the alphabet Σ, and where each coordinate is the number of times a pair is observed in (w_i, w_j). The alphabet Σ contains a symbol for each amino-acid, and a symbol X which denotes any unknown amino-acid ($|\Sigma| = 21$ and $m = 231$) . The coding of salt bridges is quite the same, except the central amino acid since salt bridges occur between two charged residues (Aspartic Acid (D) or Glutamic Acid (E) with Lysine (K), Arginine (R) or Histidine (H)).

[5]the reader is adviced to refer to the complete version given previously

Table 1. Characteristics of the affinity functions g learned on SPX.

Bonds - Prot.	P(g=1)	P(B|g=1)	P(B|g=0)	P(B)
2 - 211 prot.	0.622 *± 0.088*	**0.338** *± 0.009*	**0.325** *± 0.018*	0.333
3 - 219 prot.	0.436 *± 0.031*	**0.228** *± 0.005*	**0.179** *± 0.002*	0.200
4 - 88 prot.	0.608 *± 0.087*	**0.154** *± 0.003*	**0.124** *± 0.008*	0.143
5 - 49 prot.	0.528 *± 0.051*	**0.116** *± 0.005*	**0.105** *± 0.005*	0.111

Table 2. Characteristics of the affinity functions g learned on G3D.

Bonds - Prot.	P(g=1)	P(B|g=1)	P(B|g=0)	P(B)
2 - 182 prot.	0.649 *± 0.037*	**0.381** *± 0.009*	**0.246** *± 0.012*	0.333
3 - 166 prot.	0.485 *± 0.068*	**0.243** *± 0.005*	**0.160** *± 0.007*	0.200
4 - 136 prot.	0.482 *± 0.061*	**0.174** *± 0.005*	**0.114** *± 0.003*	0.143
5 - 86 prot.	0.593 *± 0.047*	**0.129** *± 0.003*	**0.084** *± 0.005*	0.111

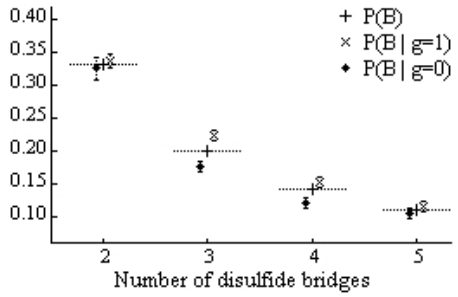

Figure 2. Graphical view of table 1.

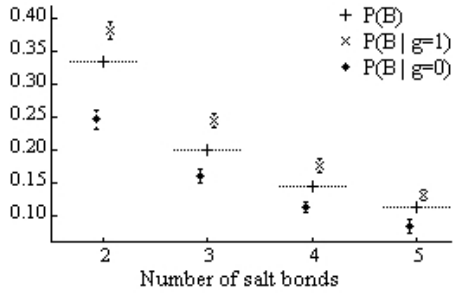

Figure 3. Graphical view of table 2.

4.2 Results

We launched two experiments: one for learning the affinity function involved in disulfide bridges (table 1 and figure 2), and the other for learning the affinity function involved in the salt bridges formation (table 2 and figure 3).

Three criteria are reported with standard square deviations: $P(g{=}1)$, the probabilities that pairs of local environments have a high level of affinity (computed with linear functions infered with the perceptron algorithm), $P(B|g{=}1)$, the probability to observe a bond knowing that the pair is predicted to have a high level of affinity and $P(B|g{=}0)$ knowing that the pair is predicted to have a low level of affinity. Each reported result is an average of five 10-fold cross-validations on the subset of proteins containing $2{\leq}n{\leq}5$ bridges. Standard deviations are italized.

It is worthwhile to notice that, in order to ensure that the detected local information is not correlated with beta sheets or alpha helix in proteins, we launched experiments to look after (i) the ratio of residues involved both in bridges and beta sheets, (ii) the ratio of residues involved both in bridges and alpha helix, and (iii) those only involved in either disulfide or salt bridges. These experiments show that no correlation exists between the level of affinity predicted for local environments pairs and these 2D structural elements.

4.3 Discussion

On one hand, results on salt bridges reveal that a clear signal is detected: there exists local information that is involved in the formation of salt bridges. That signal is quite constant along the experiments. Figure 2 shows that whatever the number of bonds is, our algorithm learns an affinity function g that always classify more observed bonds as having high affinity than having low affinity. In other words, we pointed out an affinity function between local environments of salt bridges. The detected affinities might be explained either by the ionic nature of salt bridges, which often involves the charge of their local environments, or/and by the hydrophilic property of many residues around salt bridges.

On the other hand, the results on disulfide bridges pictured on figure 1, are not as clear as expected: probabilities $P(B|g = 1)$ and $P(B|g = 0)$ are really close to the baseline probabilities $P(B)$. These results may be explained by several independent reasons.

Biology reality. The first insight of our results is that there might be no local information that would guide the formation of disulfide bridges during the 3D conformation of proteins. Such an explanation would be shared with many biologists and biochemists: disulfide bonds are so strong links that the propensity between their environments is not enough determining for guiding actual bonds.

Data sparsity. In order to estimate the impact of sparsity on our experiments, we used hyperplanes infered by this algorithm, for relabelling the learning data. A soft-margin algorithm has then been launched on these re-labelled dataset in order to optimize the margin. However, no significant improvement has been observed on test data.

Learning a function in an unsuitable concept classes. The affinity function g that we try to learn might be not representable by a linear threshold function such as learnt by any perceptron. Obviously, we still have to design other algorithms adapted to CCCN noise in other concept classes. It would be very interesting to adapt soft-margins method to CCCN context.

However, this work does not allow us to know which assumption is the most probable. The case of the disulfide bridges remains an open question.

5 Conclusions and future works

We presented a machine-learning based protocol to answer the question of the presence of local affinities that would be involved in the pairing of distant residues in proteins. We validated this protocol since we were able to learn an affinity between local environments of salt bridges. However, the same protocol has not yet indicated any impact of local environments on the formation of disulfide bonds. More generally, the protocol can be used to detect any affinity between pairs of local environments residues.

In a machine learning point of view, this work is a success for it proves that it is crucial to theoretically study other algorithms fitting the CCCN context.

The presented protocol initiates the state of the art for the question of the existence of local affinities involved in local interactions. Yet many studies have to be done using this protocol for surrounding this question.

References

[1] A. Blum, A. M. Frieze, R. Kannan, and S. Vempala. A polynomial-time algorithm for learning noisy linear threshold functions. In *IEEE Symposium on Foundations of Computer Science*, pages 330–338, 1996.

[2] Bylander. Learning linear threshold functions in the presence of classification noise. In *COLT*, 1994.

[3] A. Ceroni, P. Frasconi, A. Passerini, and A. Vullo. Predicting the disulfide bonding state of cysteines with combinations of kernel machines. *J. of VLSI Signal Pr. Syst.*, 35(3), 2003.

[4] J. Cheng, H. Saigo, and P. Baldi. Large-scale prediction of disulphide bridges using kernel methods, two-dimensional recursive neural networks, and weighted graph matching. *Proteins*, 62(3):617–629, 2006.

[5] C. N. Magnan. Asymmetrical semi-supervised learning and prediction of disulfide connectivity in proteins. In *R.I.A.*, volume 20(6), pages 673–695, 2006.

[6] P. L. Martelli, P. Fariselli, L. Malaguti, and R. Casadio. Prediction of the disulfide-bonding state of cysteines in proteins at 88% accuracy. *Protein Science*, 11(11):2735–9, 2002.

[7] L. Ralaivola, F. Denis, and C. Magnan. CN = CPCN. In *ICML '06*, pages 721–728, 2006.

[8] F. Rosenblatt. Principles of neurodynamics. 1962.

[9] A. Vullo and P. Frasconi. Disulfide connectivity prediction using recursive neural networks and evolutionary information. *Bioinformatics*, 20(5):653–659, 2004.