

Contribution à l'analyse des séquences de protéines : Similarité, Clustering et Alignement

Par

Abdellali Kelil

Thèse présentée au Département d'informatique
En vue de l'obtention du grade de philosophiæ docteur (PhD.)

FACULTÉ DES SCIENCES
UNIVERSITÉ DE SHERBROOKE

Sherbrooke, Québec, Canada
Février 2011



Library and Archives
Canada

Published Heritage
Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et
Archives Canada

Direction du
Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 978-0-494-75066-7
Our file Notre référence
ISBN: 978-0-494-75066-7

NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

■*■
Canada

Le 9 février 2011,

*le jury a accepté la thèse de Monsieur Abdellali Kelil
dans sa version finale.*

Membres du jury

Professeur Shengrui Wang
Directeur de recherche
Département d'informatique

Professeur Ryszard Brzezinski
Codirecteur de recherche
Département de biologie

Professeur Rafaël Josef Najmanovich
Membre
Faculté de médecine
Département de biochimie

Professeur Alioune Ngom
Membre externe
Windsor University
School of Computer Science

Professeur Gabriel Girard
Président rapporteur
Département d'informatique

*À ma courageuse mère Kheira, à mon dévoué père Abdelouahab,
Les êtres les plus chères à mon cœur*

SOMMAIRE

La prédiction des fonctions biologiques des protéines est primordiale en biologie cellulaire. On peut comprendre facilement tout l'enjeu de pouvoir différencier efficacement les protéines par leurs fonctions, quand on sait que ceci peut rendre possible la réparation des protéines anormales causants des maladies, ou du moins corriger ou améliorer leurs fonctions.

Les méthodes expérimentales, basées sur la structure tridimensionnelle des protéines sont les plus fiables pour la prédiction des fonctions biologiques des protéines. Néanmoins, elles sont souvent coûteuses en temps et en ressources, et ne permettent pas de traiter de grands nombres de protéines. Il existe toutefois des algorithmes qui permettent aux biologistes d'arriver à de bons résultats de prédictions en utilisant des moyens beaucoup moins coûteux. Le plus souvent, ces algorithmes sont basés sur la similarité, le clustering, et l'alignement. Cependant, les algorithmes qui sont basés sur la similarité et le clustering utilisent souvent l'alignement des séquences et ne sont donc pas efficaces sur les protéines non alignables. Et lorsqu'ils ne sont pas basés sur l'alignement, ces algorithmes utilisent souvent des approches qui ne tiennent pas compte de l'aspect biologique des séquences de protéines. D'autre part, l'efficacité des algorithmes d'alignements dépend souvent de la nature structurelle des protéines, ce qui rend difficile le choix de l'algorithme à utiliser quand la structure est inconnue. Par ailleurs, les algorithmes d'alignement ignorent les divergences entre les séquences à aligner, ce qui contraint souvent les biologistes à traiter manuellement les séquences à aligner, une tâche qui n'est pas toujours possible en pratique.

Dans cette thèse nous présentons un ensemble de nouveaux algorithmes que nous avons conçus pour l'analyse des séquences de protéines. Dans le premier chapitre, nous présentons CLUSS, le premier algorithme de clustering capable de traiter des séquences de protéines non-alignables. Dans le deuxième chapitre, nous présentons CLUSS2 une version améliorée

de CLUSS, capable de traiter de plus grands ensembles de protéines avec plus de fonctions biologiques. Dans le troisième chapitre, nous présentons SCS, une nouvelle mesure de similarité capable de traiter efficacement non seulement les séquences de protéines mais aussi plusieurs types de séquences catégoriques. Dans le dernier chapitre, nous présentons ALIGNER, un algorithme d'alignement, efficace sur les séquences de protéines indépendamment de leurs types de structures. De plus, ALIGNER est capable de détecter automatiquement, parmi les protéines à aligner, les groupes de protéines dont l'alignement peut révéler d'importantes propriétés biochimiques structurelles et fonctionnelles, et cela sans faire appel à l'utilisateur.

REMERCIEMENTS

Je tiens à remercier trois personnes, sans qui tout cela n'aurait jamais pu arriver; ma bienaimée et dévouée épouse Dalel, et mes deux Professeurs, Dr. Shengrui Wang et Dr. Ryszard Brzezinski.

Je remercie aussi les membres du jury pour avoir accepté d'évaluer ma thèse;

Je tiens aussi à remercier les membres du laboratoire prospectus avec qui j'ai appris beaucoup de choses et aussi avec qui j'ai eu l'opportunité de collaborer pendant mon doctorat.

Je remercie aussi les techniciens et les professionnels en informatique du département d'informatique, qui m'ont aidé pour la conception et la maintenance des deux serveurs web CLUSS et ALIGNER.

Je remercie tous les professeurs du département d'informatique ainsi que tous les membres du staff administratif et académique.

Je remercie aussi ma famille, mes amis, mes collègues, ainsi que toutes les personnes de l'Université de Sherbrooke, qui m'ont aidé et soutenue pour la préparation de cette thèse;

TABLE DES MATIÈRES

SOMMAIRE	i
REMERCIEMENTS	iii
TABLE DES MATIÈRES	iv
INTRODUCTION	1
1. Préambule	1
2. Les protéines	4
3. Prédiction des fonctions biologiques des protéines	6
4. Similarité	10
4.1. État de l'art	10
4.2. SMS	13
4.3. tSMS	14
4.4. SAF	16
4.5. SCS	16
4.6. Méthode d'utilisation de la théorie de Karlin	18
5. Clustering	20
5.1. État de l'art	20
5.1.1. Algorithmes de clustering hiérarchiques	22
5.1.2. Algorithmes de clustering non-hiéarchiques	26
5.2. CLUSS	27
5.3. CLUSS2	29
6. Alignement	29
6.1. État de l'art	29
6.1.1. Alignement progressif	30
6.1.2. Alignement itératif	31
6.1.3. Model de Markov caché	32
6.1.4. Algorithmes génétique	33
6.1.5. Recuit simulé	33
6.1.6. Recherche de motifs	34
6.1.7. Alignement local et global	35
6.2. ALIGNER	36
7. Conclusion	40

Chapitre 1 CLUSTERING DES FAMILLES DE PROTÉINES	41
Chapitre 2 CLUSTERING DES GRANDES FAMILLES DE PROTÉINES.....	79
Chapitre 3 SIMILARITÉ DES SÉQUENCES CATÉGORIQUES	102
Chapitre 4 ALIGNEMENT DES PROTÉINES APPARENTÉES	128
CONCLUSION.....	157
ANNEXE 1 : ÉVALUATION DU CLUSTERING.....	161
ANNEXE 2 : COMPLEXITÉ.....	164
ANNEXE 3 : LE SERVEUR WEB CLUSS	168
ANNEXE 4 : LE SERVEUR WEB ALIGNER.....	169
ANNEXE 5 : Liste des publications	170
ANNEXE 6 : Aperçu des travaux publiés en lien avec nos recherches	172
REFERENCES.....	175

INTRODUCTION

1. Préambule

Tout organisme vivant porte au fin fond de lui son patrimoine génétique communément appelé gé nome, une véritable base de données contenant des gènes. Les gènes sont en réalité des recettes permettant, entre autres, de reproduire des protéines à volonté, un des matériaux essentiels pour la survie de tout organisme vivant. Chaque recette contient les instructions pour fabriquer au moins une protéine, dans un langage très particulier utilisant un alphabet de quatre lettres (A, C, G et T), d'où l'importance d'identifier les gènes, ce qui veut dire là où ils commencent et là où ils finissent, et l'ordre exact des lettres qui les composent. Également, un gène n'est pas forcément associé à une seule protéine mais bien souvent à plusieurs protéines différentes. Ce phénomène est appelé l'épissage alternatif. Quant aux protéines, elles peuvent être comparées à de longues chaînes de caractères, où chaque caractère représente un acide aminé unique, qui est en réalité une molécule biologique plus ou moins complexe. Dans la nature, il existe 20 acides aminés différents quel que soit l'organisme vivant, voir Tableau 1. La séquence d'une protéine et la succession des acides aminés qui la composent déterminent exactement sa forme et sa conformation et donc sa fonction dans l'organisme parmi toutes les autres protéines. Pour une lecture détaillée veuillez consulter Lodish *et al.* [86].

Identifier la séquence linéaire des acides aminés composant une protéine, appelé aussi séquençage, est donc primordial pour comprendre sa structure et sa fonction. Tous les mécanismes dans tous les organismes vivants font appel aux protéines. Par exemple, le transport de l'oxygène et des nutriments vers les cellules, l'élasticité de la peau, la rigidité des os, la digestion des aliments ou l'absorption des nutriments sont tous assurés par autant de protéines différentes et spécialisées. Si l'on veut espérer pouvoir un jour comprendre

Tableau 1. Acide aminés

Acides aminés	Code à une lettre	Code à trois lettres
Alanine	A	Ala
Arginine	R	Arg
Asparagine	N	Asn
Aspartate	D	Asp
Cystéine	C	Cys
Glutamate	E	Glu
Glutamine	Q	Gln
Glycine	G	Gly
Histidine	H	His
Isoleucine	I	Ile
Leucine	L	Leu
Lysine	K	Lys
Méthionine	M	Met
Phénylalanine	F	Phe
Proline	P	Pro
Sérine	S	Ser
Thréonine	T	Thr
Tryptophane	W	Trp
Tyrosine	Y	Tyr
Valine	V	Val

comment les protéines fonctionnent et interagissent entre elles - ce qui se révèle d'ores-et-déjà d'une difficulté titanique - il est nécessaire d'identifier et de comprendre leurs fonctions individuelles. D'autre part, on sait déjà que la modification d'un seul acide aminé dans une protéine dite « *normale* » peut rendre cette protéine « *anormale* », ce qui peut mener à l'apparition de certaines maladies. On peut comprendre facilement tout l'enjeu de pouvoir différencier les protéines normales de celles qui sont anormales. Ceci pourra rendre possible la réparation des protéines anormales, ou du moins corriger ou améliorer leurs fonctions biologiques. Les livres publiés par Friedman [35] et Oliver [106] offrent une lecture riche et détaillée à propos de ce sujet.

En 1953, la première séquence complète d'une protéine, l'insuline, a été obtenue par le célèbre biochimiste Frederick Sanger. Depuis lors, le nombre de séquences de protéines répertoriées n'a jamais cessé de croître, voir Figure 1 et Figure 2. Tous organismes confondus, ce nombre avoisine aujourd'hui les 12 000 000 de protéines (source :

UniProtKB/TrEMBL), et cela augmente toujours. Et quand on sait que la caractérisation expérimentale d'une seule protéine peut nécessiter des années de recherche, alors on peut imaginer facilement l'ampleur de la tâche qui attend les biologistes. C'est pourquoi parmi cet afflux de données, il y a juste une infime minorité de protéines qui ont été étudiées expérimentalement afin de déterminer leurs structures et fonctions. À ce jour, on a pu identifier expérimentalement la structure de seulement 61 860 protéines (source : PDB), voir Figure 3, ce qui correspond à moins de 1% des toutes les protéines connues. Face à ce déluge de données, la bioinformatique s'imposa comme étant un outil incontournable pour l'étude et l'analyse des protéines.

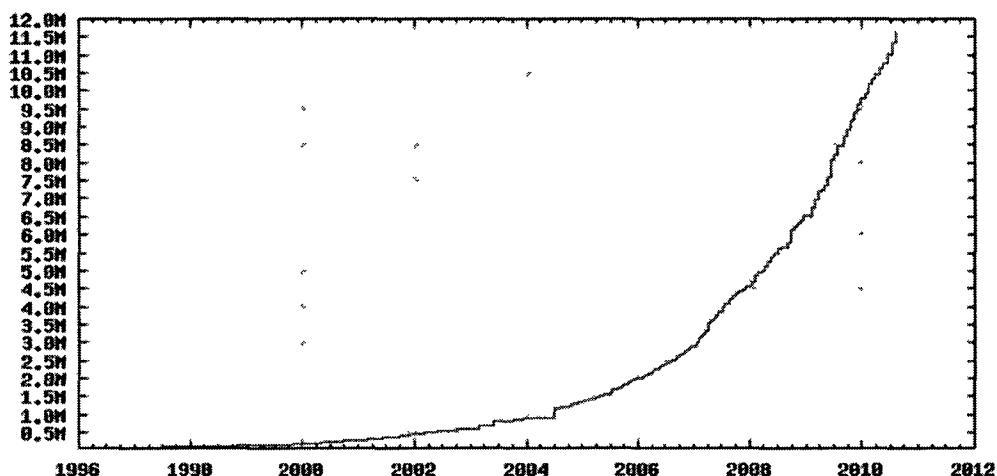


Figure 1. Croissance par année du nombre de protéines dans la banque de donnée UniProtKB/TrEMBL

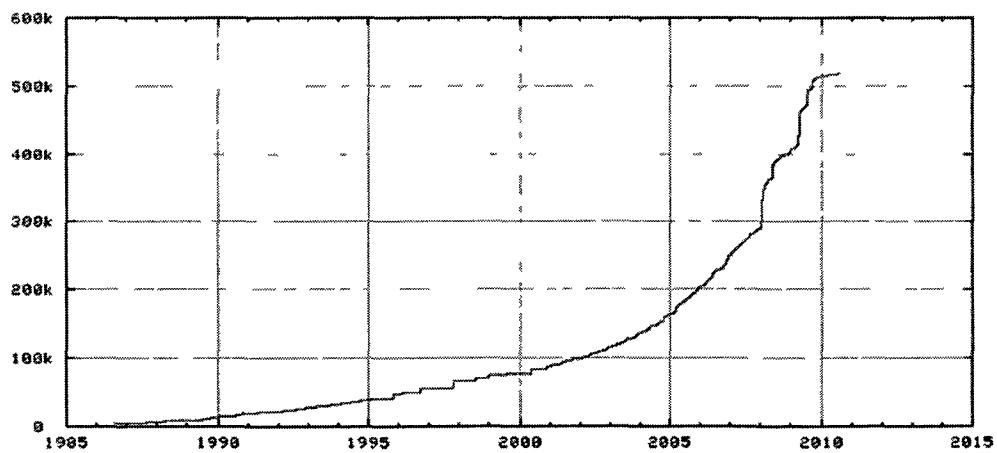


Figure 2. Croissance par année du nombre de protéines dans la banque de données UniProtKB/Swiss-Prot

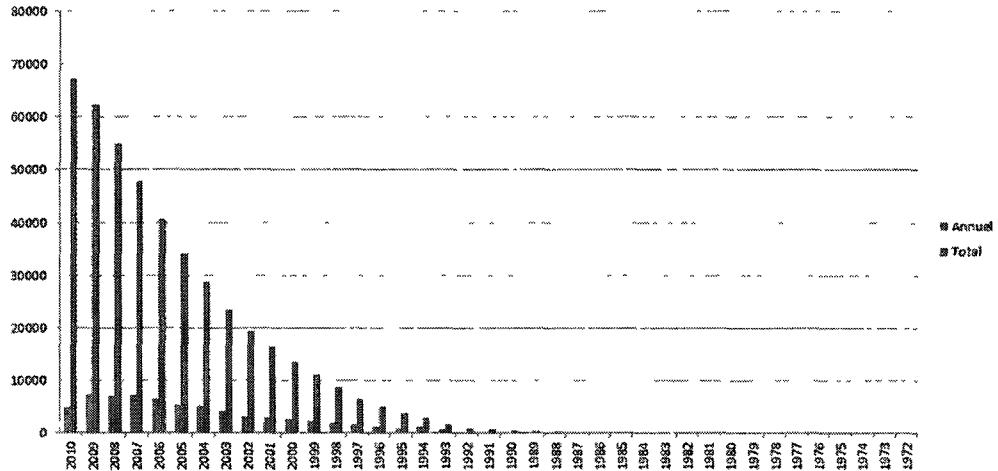


Figure 3. Croissance par année du nombre de structures de protéines établi expérimentalement (source : PDB)

2. Les protéines

Les protéines sont des macromolécules organiques constituées de chaînes linéaires, plus ou moins longues, d'acides aminés amarrés les uns aux autres séquentiellement par des liaisons chimiques. Cette structure est appelée « *séquence* » ou « *structure primaire* » de la protéine, et est définie par le gène qui code chaque protéine. La structure primaire de la protéine se replie sur elle-même pour former une « *structure secondaire* ». Plusieurs structures secondaires s'agencent les unes par rapport aux autres pour former la « *structure tertiaire* », voir Figure 4 pour une illustration des quatre structures. Les forces qui gouvernent ces repliements et agencements sont les forces d'attraction et de répulsion classiques de la physique [4]. Le rôle que joue chaque protéine au sein de la cellule vivante est conféré par la manière dont les acides aminés qui la constituent sont agencés les uns par rapport aux autres dans l'espace. Il devient vital alors pour la cellule vivante que chaque protéine puisse se replier correctement afin qu'elle puisse assurer son rôle. Pour plus de détails sur la fonction et la structure des protéines veuillez consulter le livre référence Lodish *et al.* [87].

Tous organismes confondus, il existe dans la nature 20 acides aminés différents. Avec ces 20 acides aminés, on peut théoriquement assembler un nombre astronomique de séquences de protéines, par exemple, on peut construire pas moins de 20^{100} séquences de protéiques

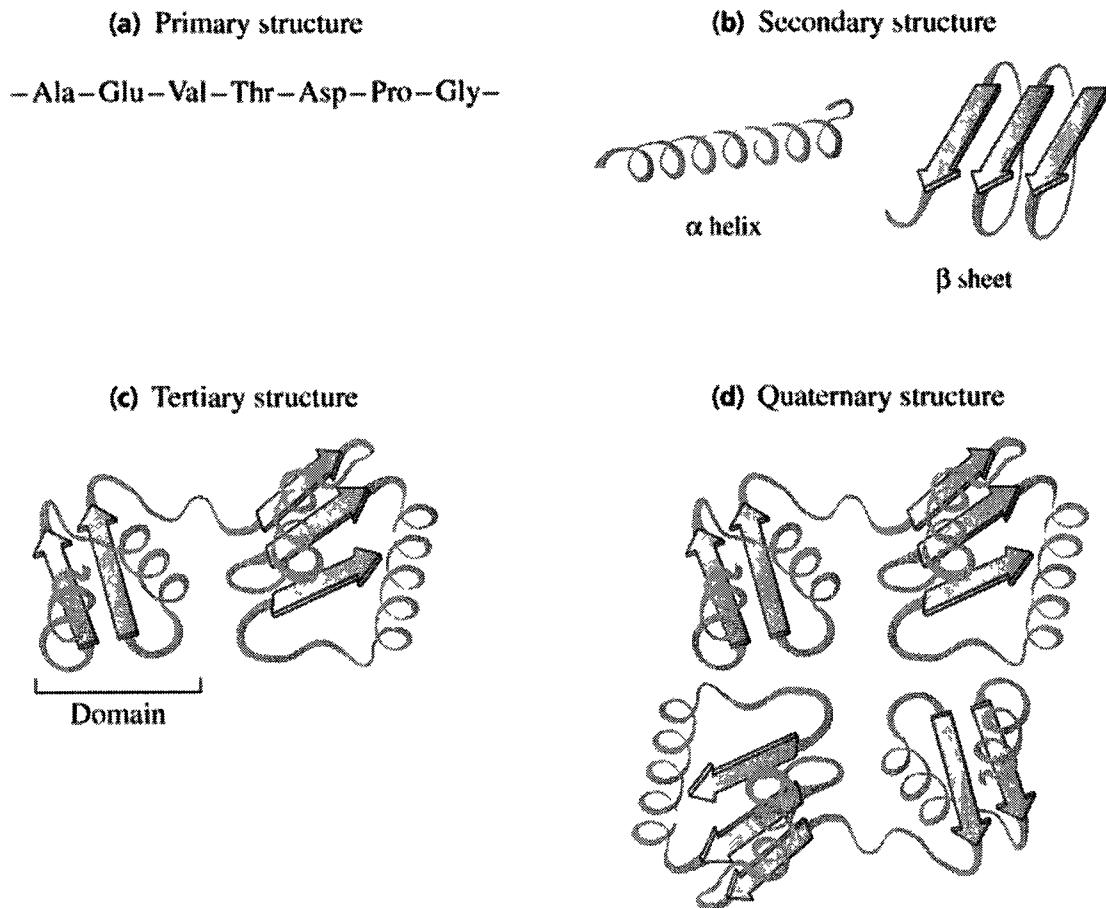


Figure 4. Exemple des quatre niveaux de la structure des protéines. (a) la séquence linéaire d'acides aminés définit la structure primaire. (b) la structure secondaire est composée de régions contenant des conformations répétées, comme les hélices alpha et feuillets bêta, (c) la structure tertiaire décrit la forme de la chaîne polypeptidique repliée, ici un exemple de deux domaines, (d) la structure quaternaire se réfère à la disposition des chaînes de deux ou plusieurs polypeptides dans une molécule de plusieurs sous-unités. (Source: Principles of biochemistry, H. Robert Horton et al.)

différentes de longueur de 100 acides aminés seulement. Cependant, seule une infime partie de toutes les combinaisons possibles existent réellement dans les organismes vivants, voir Figure 1 et Figure 2. Spécialement lorsque on sait que la longueur d'une séquences de protéines peut atteindre plusieurs milliers d'acides aminés, comme celle de la souris « *titin* » qui contient 35 213 acides aminés (source : UniProtKB/SwissProt).

Les protéines adoptent en effet une multitude de formes qui assurent de multiples fonctions qui sont indispensables à la survie de la cellule et de l'organisme. En réalité, la quasi-totalité

des fonctions biologiques dans les organismes vivants est assurée par des protéines [87]. Par exemple :

- Elles ont un rôle structurel comme l'actine, une protéine du cytosquelette qui donne aux cellules leurs formes particulières.
- Elles ont un rôle dans la mobilité, comme la myosine qui a un rôle fondamental dans les mécanismes de la contraction musculaire.
- Elles ont un rôle dans le transport, telle l'hémoglobine qui transporte l'oxygène vers les cellules.
- Elles ont un rôle dans la communication, telle l'insuline qui peut véhiculer un message vers les organes les informant de l'état nutritionnel et de l'activité physique de l'organisme.
- Elles ont un rôle dans le système immunitaire, telles les immunoglobulines qui permettent la reconnaissance des corps étranges dans l'organisme.
- Elles ont un rôle catalytique, telles les enzymes qui permettent d'accélérer les réactions chimiques à l'intérieur ou à l'extérieur des cellules.
- Elles ont un rôle de régulation de la compaction et l'enroulement de l'ADN, telles les histones, ou l'expression des gènes tels les facteurs de transcriptions, etc.

3. Prédiction des fonctions biologiques des protéines

Il existe dans la nature un bon nombre de familles de protéines structurellement et fonctionnellement apparentées. Ces familles s'organisent autour de sous-familles comportant un nombre limité de caractéristiques structurelles révélant certaines fonctions biologiques. Cela sous-entend que la détermination des caractéristiques structurelles d'une protéine peut nous informer sur sa fonction biologique [87]. Toutefois, ces caractéristiques sont souvent rendues méconnaissables du fait qu'elles sont immergées dans un océan d'informations structurelles généralement non-pertinentes. En conséquence, la majorité des protéines

séquencées à ce jour ne présentent pas de similitudes structurelles significatives apparentes avec d'autres protéines, ce qui rend leurs caractérisations particulièrement ardues. Il est clair à présent que l'un des défis les plus importants en biologie cellulaire est la détermination, ou en termes plus précis, la prédiction des propriétés structurelles et fonctionnelles des protéines. Les découvertes les plus récentes concernant les propriétés des protéines ont été revues dans le livre référence de Lodish *et al.* [87].

La littérature fait état d'un bon nombre d'approches développées pour la prédiction des fonctions des protéines. En général, les approches expérimentales, basées sur la détermination de la structure tridimensionnelle des protéines, sont celles qui donnent les résultats les plus probants biologiquement. A titre d'exemple, on peut citer les travaux reconnus historiquement comme « *pionniers* » dans le domaine, tels que la première cristallisation du lysozyme du blanc d'œuf (HEWL) réalisée par Blake *et al.* [13], la première détermination de la structure de la ribonucléase réalisée par Kartha [62], et aussi la première détermination de la structure d'une protéase, la papaïne, réalisée par Drenth *et al.* [26]. On peut citer aussi l'exemple de la famille 46 des Glycoside Hydrolase (GH46), qui appartient à la grande famille des Carbohydrate-Active Enzymes (CAZy) [18], et pour laquelle plusieurs travaux basés sur la structure ont été réalisés afin de mieux comprendre les mécanismes d'actions des Chitosanases, dont ceux réalisés par Boucher *et al.* [16], Marcotte *et al.* [92], Fukamizo *et al.* [37], Côté *et al.* [21], et aussi Saito *et al.* [120]. Néanmoins, ces approches sont souvent très complexes, et elles sont coûteuses en temps et en ressources et nécessitent de la part des biologistes du temps et des moyens drastiques. Plus encore, elles ne permettent pas de traiter un grand nombre de protéines dans des délais et avec des ressources raisonnables.

Toutefois, il existe un grand nombre d'approches algorithmiques pour la prédiction des fonctions des protéines [107]. Ces approches permettent aux biologistes d'arriver à leurs fins en utilisant des moyens souvent très efficaces et beaucoup moins complexes à mettre en œuvre en pratique. Pour prédire les fonctions des protéines, les biologistes utilisent le plus souvent trois catégories différentes d'approches algorithmiques, en raison de leurs simplicités et de leurs efficacités à déchiffrer les propriétés structurelles et fonctionnelles cachées des

protéines. Ces catégories d'approches sont basées respectivement sur : la similarité, le clustering, et l'alignement. Nous les résumons comme suit (une discussion détaillée suivra) :

- **La similarité**, est utilisée pour comparer les séquences de protéines. Une similarité marquée entre deux séquences de protéines peut refléter un lien structurel ou fonctionnel. La similarité s'avère très efficace pour identifier de manière exhaustive les groupes de protéines partageant certaines caractéristiques importantes. En pratique, on peut souvent identifier des protéines non caractérisées en cherchant des séquences similaires dans les banques de données en utilisant des programmes informatiques de recherche de similarités comme BLAST [2] ou FASTA [108]. Voir Figure 5 pour un exemple.
- **Le clustering**, est utilisé pour subdiviser des ensembles de protéines en des groupes de protéines avec des propriétés biochimiques similaires. Ainsi, une fonction biologique peut être attribuée à une protéine non-caractérisée avec une bonne certitude, si dans le même groupe il existe au moins une protéine dont la fonction a été déterminée auparavant. Inversement, une fonction biologique nouvellement découverte pour une protéine peut être étendue sur tous les membres du groupe. Voir Figure 6 pour un exemple.
- **L'alignement**, est utilisé pour trouver le meilleur appariement entre des séquences de protéines, de sorte que les positions des acides aminés identiques ou similaires dans les différentes séquences soient alignées. L'alignement a pour objectif d'identifier les régions conservées dans les séquences qui peuvent révéler des motifs significatifs d'une importance structurelle ou fonctionnelle dans un ensemble de séquences de protéines. Voir Figure 7 pour un exemple.

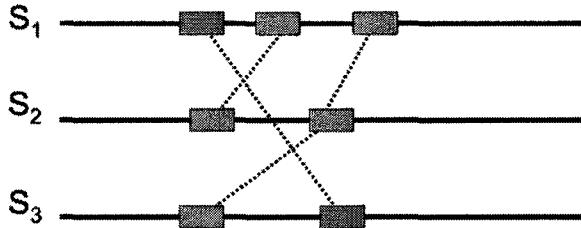


Figure 5. Un motif important partagé entre des séquences de protéines peut refléter un lien structurel ou fonctionnel important

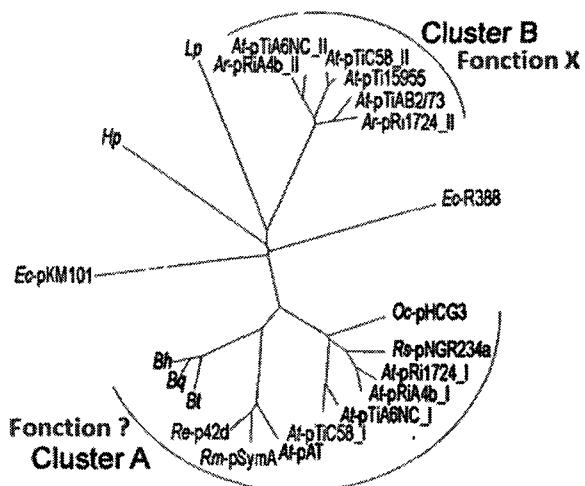


Figure 6. Clustering d'un groupe de protéines. La fonction « X » est attribuée à une protéine non-caractérisée si elle est classée dans le cluster B. La fonction d'une protéine caractérisée peut être étendue à tous les membres du cluster A si elle est classée dans ce cluster.

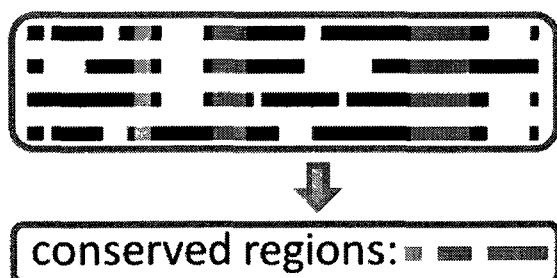


Figure 7. L'alignement d'un ensemble de séquences de protéines peut révéler des régions conservées qui ont un rôle structurel ou fonctionnel dans ces protéines

4. Similarité

4.1. État de l'art

De nombreuses approches pour mesurer la similarité entre les séquences de protéines ont été développées. Parmi les approches les plus fréquemment utilisées, il y a celles conçues spécifiquement pour comparer une séquence de protéines à une banque de données contenant un très grand nombre de séquences de protéines. On peut citer les deux algorithmes les plus connues : BLAST [2] (source : NCBI¹) qui détecte les régions isolées similaires par l'utilisation de paires de segments à haut scores basés sur l'alignement, et aussi FASTA [108] (source : EBI²) qui utilise des *K-mots* pour la construction d'alignements locaux afin de capturer les similitudes locales les plus importantes. Deux versions différentes de BLAST ont été aussi développées, PSI-BLAST [3] qui est un BLAST itérative, et GAPPED-BLAST [3] qui est un BLAST qui autorise les gaps. Tous les deux ont été développés pour détecter des similarités plus faibles. Ces algorithmes sont capables de traiter de grands ensembles de séquences de protéines, dans des délais très raisonnables, en utilisant diverses techniques pour accélérer l'examen des relations entre les séquences de protéines. Malheureusement, ces algorithmes ne sont pas très sensibles aux subtiles différences qui peuvent exister entre les séquences de protéines quand elles sont très similaires, et c'est pourquoi ils ne sont pas très efficaces pour traiter les protéines de mêmes familles. Toutefois, il existe d'autres approches capables de mesurer la similarité entre les séquences de protéines avec plus de précision.

La méthode communément appelée « *distance de Levenshtein* » ou « *distance d'édition* » [82] est à l'origine une mesure de distance entre des chaînes de caractères. Elle est basée sur le calcul du coût minimum nécessaire pour transformer une séquence en une autre en utilisant les opérations « *insertion* », « *suppression* », et « *remplacement* », où chaque opération lui est assignée un coût. Le coût total ainsi calculé est d'autant plus grand que le nombre de différences entre les deux séquences. Cette méthode, malgré qu'elle ne soit pas spécialement

¹ <http://blast.ncbi.nlm.nih.gov>

² <http://www.ebi.ac.uk>

développée pour la bioinformatique, est couramment utilisée pour comparer des séquences de protéines [76, 123, 148, 153].

L’alignement des séquences [100, 126] est l’une des approches les plus fréquemment utilisées pour mesurer la similarité entre les séquences de protéines. L’alignement a pour objectif d’apparier une paire de séquences de protéines en insérant des « *gaps* » dans les positions appropriées afin que les acides aminés identiques ou similaires dans les deux séquences soient alignés, ainsi les régions similaires dans les deux séquences sont mises en évidences. L’alignement ainsi obtenu est ensuite utilisé pour calculer la similarité entre les deux séquences de protéines.

Cependant, vu que la distance d’édition et l’alignement des séquences sont basés sur l’appariement des acides aminés dans des positions équivalentes (i.e., même ordre chronologique), elles sont inefficaces pour traiter des séquences de protéines qui contiennent des régions similaires dans des positions non équivalentes. En plus, ces deux méthodes dépendent beaucoup des coûts qu’assigne l’utilisateur aux opérations « *insertion* », « *suppression* », et « *remplacement* » dans le cas de la « *distance de Levenshtein* », ou aux pénalités d’ouverture et d’extension de « *gaps* » dans le cas de l’alignement des séquences. Cela crée des ambiguïtés et complique la tâche de mesure de similarité, en particulier pour les séquences de longueurs très différentes. Car il est difficile d’apparier les régions similaires qui sont dans des positions éloignées dans les séquences de protéines à cause du coût de la pénalité engendré [49].

Il existe aussi dans la littérature différentes approches pour mesurer la similarité entre les séquences de protéines qui ne sont pas basées sur l’appariement des acides aminés. La plupart de ces approches transforment les séquences de protéines en des vecteurs dans des espaces multidimensionnels, pour lesquels on peut utiliser les outils classiques de l’algèbre linéaire et la théorie des statistiques [14, 25, 73]. Ces vecteurs sont définis par les fréquences des *K-mots* dans les séquences de protéines. Les *K-mots* sont l’ensemble de tous les motifs possibles d’une longueur fixe *K*. On peut citer par exemple l’approche développée par Wu *et al.* [151] qui utilise les motifs de courte longueur en tant qu’indices comme dans le domaine de la

recherche d'information dans le traitement du texte en langage naturel, ou l'approche développée par Bogan-Marta *et al.* [14] qui calcule l'entropie croisée appliquée sur les *K-mots* recueillis avec une longueur fixe.

Il est bien connu que les méthodes basées sur les *K-mots* ont un inconvénient majeur lié au choix de *K*, qui est fixé manuellement par l'utilisateur, et ne reflète pas nécessairement les propriétés structurelles et fonctionnelles des protéines [95]. Ceci provoque souvent la collecte de *K-mots* qui constituent juste du bruit ou alors l'exclusion d'importants *K-mots* qui représentent dans les séquences de protéines des régions conservées. Ces approches ont été examinées en détail par plusieurs auteurs, dont Reinert *et al.* [116], Rocha *et al.* [117], Edgar [28], Vinga *et al.* [143], et Dai *et al.* [22].

Il faut mentionner aussi l'existence d'une approche purement mathématique, communément appelée « *complexité de Kolmogorov* », qui n'est pas basée sur l'appariement des acides aminés. La complexité de Kolmogorov définit la complexité d'un objet, telle une séquence de protéine, par la taille des ressources de calcul minimums nécessaires qui permettent de reproduire cet objet [83]. Cette approche a été appliquée la première fois en bioinformatique par Li *et al.* [83] pour comparer des séquences mitochondriales, et aussi par Kocsor *et al.* [72] pour comparer des séquences de protéines. Bien que la complexité de Kolmogorov soit une méthode théoriquement bien formulée, la notion est cependant non-calculable [83, 117]. En pratiques, elle est approchée par la longueur des séquences compressées calculées par un algorithme de compression. Cette approximation a pour conséquence de rendre cette méthode moins précise que les méthodes standards. Pour cette raison, la complexité de Kolmogorov n'est généralement utilisée que sur de petits ensembles de séquences de protéines ou plus simplement sur des domaines de protéines, ce qui la rend moins compétitive que les approches standards. En effet, dans les travaux réalisés par Rocha *et al.* et [117], il a été montré que l'application de la complexité de Kolmogorov sur de grands ensembles de séquences de protéines produisait souvent des résultats non satisfaisants.

Dans plusieurs cas, les méthodes qui ne dépendent pas de l'appariement des acides aminés peuvent grandement améliorer la comparaison des séquences de protéines, en particulier

celles qui ne sont pas alignables. Cependant, ces méthodes considèrent que les protéines sont des séquences purement catégoriques, et considèrent que les acides aminés sont seulement des caractères d'un alphabet de 20 lettres, et ne tiennent pas compte des relations biologiques importantes qui existent entre les acides aminés, telles que les relations physico-chimiques, structurelles, génétiques, ou de substitutions (source : PROWL³) . Pour résoudre ce problème, certains chercheurs, comme Edgar [28], ont suggéré l'utilisation des méthodes de correction de la similarité, comme celle introduite par Kimura [71] ou Felsenstein [32]. Toutefois, pour obtenir des résultats acceptables, l'approche décrite par Edgar [28] effectue un raffinement itératif du résultat, incluant un alignement profile-profile à chaque itération, ce qui augmente considérablement la complexité.

Pour faire face à ces différents problèmes, nous avons développé une série de méthodes, motivées par des considérations biologiques et des observations connues liés à la nature structurelle des protéines et de leur évolution, pour calculer la similarité entre les séquences de protéines. Nous les résumons ci-dessous.

4.2. SMS

L'objectif principal de cette nouvelle mesure de similarité est de faire un usage plus efficace de l'information biologique contenue dans les acides aminés constituant les séquences de protéines, ce qui conduirait à une meilleure mesure de similarité. L'idée principale consiste à détecter des motifs de longueurs suffisamment importantes qui soient partagés par les séquences de protéines, et d'utiliser ensuite les propriétés physico-chimiques connues des acides aminés par le biais des relations de substitutions pour pondérer l'importance de chaque motif dans les séquences de protéines. Et pour ce faire, nous utilisons l'une des matrices de substitutions telle que BLOSUM62 [46], ou PAM250 [23], qui sont habituellement utilisées par les algorithmes d'alignement des séquences de protéines.

³ <http://prowl.rockefeller.edu/aainfo/contents.htm>

SMS, que nous avons publié dans Kelil *et al.* [69], est une nouvelle mesure de similarité qui ne dépend pas de l'alignement, même si elle utilise des matrices de substitutions conçues pour l'alignement. SMS est capable de détecter les motifs les plus significatifs, ceux qui reflètent le mieux les caractéristiques structurelles et fonctionnelles des protéines. SMS utilise une nouvelle approche d'appariement de pairs de motifs identiques que nous avons développée en nous basant sur une nouvelle méthode d'utilisation de la théorie statistique introduite par Karlin *et al.* [61]. SMS est efficace tant avec les séquences de protéines alignables que non alignables. Nous discuterons plus en détails de la nouvelle méthode d'utilisation de la théorie de Karlin plus loin dans ce chapitre.

Nos tests expérimentaux ont montré que, par rapport aux approches existantes, SMS est plus efficace pour détecter les motifs significatifs, qui représentent le mieux les propriétés intrinsèques des protéines, comme les dépendances chronologiques et les caractéristiques structurelles et fonctionnelles. Toutefois, SMS a tendance à être moins efficace lorsqu'elle est appliquée à de grands ensembles de protéines contenant de grands nombres de fonctions biologiques. En plus de cela, malgré l'utilisation de techniques d'optimisation pour accélérer la mise en correspondance des motifs, il ne nous a pas été possible de réduire considérablement sa complexité. Tous ces facteurs empêchent SMS d'être très efficace sur les grands ensembles de séquences de protéines.

4.3. tSMS

Selon l'évolution, dans les séquences de protéines, il y a des régions qui mutent plus que d'autres. On appelle ce phénomène la « *mutabilité* ». Il y a aussi des régions qui sont conservées plus que d'autres. On appelle ce phénomène la « *conservation* ». Les régions dans les protéines qui mutent le moins ou qui sont les plus conservées sont celles qui jouent un rôle important dans la fonction et la structure des protéines. Dans notre mesure de similarité SMS, seul l'information de conservation est utilisée, vu que seuls les acides aminés identiques (c.à.d. ceux qui ne mutent pas) sont considérés dans la recherche de similarité. Les résultats expérimentaux que nous avons publiés dans Kelil *et al.* [69] montrent que l'utilisation de l'information de conservation dans SMS permet de traiter les séquences de

protéines difficiles à aligner mieux que les algorithmes basés sur d'alignement. Ces résultats montrent également que SMS est capable aussi de traiter efficacement les séquences de protéines faciles à aligner aussi bien que les algorithmes basés sur l'alignement. Ce qui suggère que l'utilité de l'information de conservation est probablement beaucoup plus importante qu'on ne le croit généralement.

Toutefois, d'autres travaux expérimentaux que nous avons publiés dans Kelil *et al.* [66] ont montré que, lorsque le nombre d'activités biochimiques ou le nombre de séquences de protéines augmente, la stratégie consistant à capturer uniquement l'information de conservation n'est plus suffisante pour obtenir des mesures de similarités satisfaisantes. Par conséquent, l'utilisation de l'information de mutabilité devient inévitable pour remédier à ce problème.

Pour pouvoir traiter de grands ensembles de protéines contenant de grands nombres d'activités biochimiques, nous avons développé tSMS, que nous avons publiée dans Kelil *et al.* [66], une version améliorée de SMS, plus efficace et plus rapide. Contrairement à SMS qui ne permet que les appariements identiques, tSMS permet aussi les mésappariements des acides aminés, ce qui lui permet de capturer l'information de mutabilité contenue dans les séquences de protéines. La mesure tSMS est calculée sur la base d'un nouvel algorithme d'appariement des pairs de motifs similaires, dont la complexité est moins élevée que celui utilisé dans SMS, et qui est la principale raison de l'efficacité de tSMS. En plus de cela, comme dans SMS, tSMS estime la longueur minimum des motifs importants dans les séquences en utilisant notre nouvelle méthode d'utilisation de la théorie de Karlin que nous présentons plus loin dans ce chapitre. Cependant, au lieu d'utiliser une seule longueur pour toutes les comparaisons comme dans SMS, tSMS calcule cette longueur pour chaque comparaison, ce qui lui confère une meilleure efficacité.

D'autre part, tSMS utilise la technique de décomposition spectrale, inspirée de l'approche « *analyse sémantique latente* » [12], généralement utilisée dans le traitement du langage naturel, pour représenter les protéines dans un espace vectoriel. La décomposition spectrale transforme chaque protéine en un vecteur dans le nouvel espace en utilisant l'ensemble des

protéines comme information de base, ce qui donne une portée globale à la mesure de similarité entre les différents vecteurs, au lieu de comparer juste des pairs de protéines.

4.4. SAF

Nos deux mesures de similarité, SMS et tSMS, sont basées sur l'utilisation des matrices de substitutions pour calculer les poids des motifs significatifs détectés dans les séquences de protéines. Le choix de l'une ou l'autre des matrices de substitutions a un impact direct sur leurs efficacités (résultats non publiés, disponibles sur le site internet de CLUSS^{4,5}). Ceci peut créer des ambiguïtés lors du choix de la matrice de substitution à utiliser, et peut aussi éventuellement compliquer la tâche du biologiste dans l'interprétation des résultats. Pour faire face à ce problème, nous avons développé SAF, que nous avons publié dans Kelil *et al.* [67], une nouvelle approche pour mesurer la similarité entre les séquences de protéines. Sans avoir recours ni à l'alignement des séquences ni à l'utilisation des matrices de substitutions, SAF permet de capturer les dépendances chronologiques et les caractéristiques structurelles partagées entre les séquences de protéines. SAF exploite notre nouvelle méthode d'utilisation de la théorie statistique de Karlin *et al.* [61] pour la détection des motifs les plus significatifs. En plus de cela, SAF utilise une méthode inspirée de l'approche *N-Grams* [132] combinée avec une méthode inspirée de « *latent semantic analysis* » [25] pour représenter les protéines dans un espace vectoriel multidimensionnel, où les similarités sont calculées entre les vecteurs au lieu des séquences de protéines. Les résultats obtenus lors de nos expérimentations [67] démontrent clairement l'efficacité de SAF et de son avantage sur les approches existantes, qu'elles soient basées sur l'alignement ou non.

4.5. SCS

Nos travaux de recherche sur la mesure de similarité entre les séquences de protéines nous ont conduit au développement d'une nouvelle mesure générale de similarité, appelée SCS [65], qui non seulement fonctionne avec les séquences de protéines, mais aussi sur plusieurs

⁴ http://prospectus.usherbrooke.ca/CLUSS/Results/CLUSS_2.0/COG/Results.htm

⁵ http://prospectus.usherbrooke.ca/CLUSS/Results/CLUSS_2.0/KOG/Results.htm

d'autres sortes de données qui ont une structure similaire à la structure primaire des protéines, communément appelées « *séquences catégoriques* ».

Très souvent, dans le traitement du texte en langage naturel [12], des méthodes basées sur l'analyse sémantique latente [127] sont utilisées pour extraire les relations cachées entre les documents, en capturant les relations sémantiques importantes en utilisant des informations globales extraites de grands nombres de documents plutôt que de simplement comparer des paires de documents. Ces méthodes passent généralement par l'application de l'algorithme *N-Gram* [25, 73, 132] sur un ensemble de documents pour la construction d'une matrice $T(W \times L)$, dite matrice d'occurrences ou matrice « *mots-documents* », dont les lignes correspondent aux mots et les colonnes correspondent aux documents, où W est le nombre de mots possibles ou la taille du dictionnaire des mots, et L est le nombre de documents. Le terme $T(i, j)$ représente la fréquence du mot i dans le document j . Bien que les séquences catégoriques ne contiennent pas de motifs distincts comme les mots dans le texte en langage naturel, l'analyse des séquences catégoriques est à bien des égards semblable à l'analyse du texte en langage naturel. Toutefois, le défi est d'être capable d'identifier dans les séquences catégoriques ces motifs qui joueront le rôle des mots dans les séquences, et de distinguer les motifs significatifs en termes de structure de ceux résultant de phénomènes aléatoires, ainsi on pourra alors construire la matrice d'occurrence.

En appliquant une nouvelle méthode d'appariement de paires de séquences inspirée du traitement du texte en langage naturel, en utilisant notre nouvelle méthode d'utilisation de la théorie de Karlin, SCS extrait d'un ensemble de séquences catégoriques un ensemble de motifs importants, et filtre les motifs qui ne représentent que du bruit. Ceci se fait en cherchant dans chaque paire de séquences les motifs identiques, ainsi que les motifs légèrement différents, connus en traitement du langage naturel sous le nom de « *paronymes* » et « *cognats* ». En langage naturel, par exemple l'anglais, les paronymes tels que « *affect* » et « *effect* », sont des mots qui sont liés et sont issues de la même racine, tandis que les cognats, comme « *shirt* » et « *skirt* », sont des mots différents mais qui ont une origine commune. Pour une revue détaillée, consultez Horst *et al.* [54].

L'une des originalités de la mesure de similarité SCS est que, l'algorithme *N-Gram* est appliqué seulement sur l'ensemble des motifs importants collectés pour construire la matrice d'occurrences, plutôt que sur toute la longueur des séquences catégoriques originales comme c'est le cas de l'algorithme *N-Gram* classique [14, 25]. Le fait que *N-Gram* ne soit appliqué que sur les motifs importants collectés permet d'éviter à SCS la collecte de *K-mots* qui constituent juste du bruit ou alors l'exclusion d'importants *K-mots* qui représentent des régions importantes dans les séquences. Les séquences catégoriques sont alors projetées sur un espace vectoriel de dimension réduite [38], dans lequel chaque séquence catégorique est représentée par un vecteur. Enfin, la mesure de similarité entre les différentes séquences est calculée simplement par le produit du cosinus entre les vecteurs correspondants.

Les résultats que nous avons obtenus avec SCS sur les différents types de séquences catégoriques des différents domaines d'applications, tel que *la caractérisation des protéines*, *la classification du langage naturel*, *la catégorisation de la musique*, *la détection des pourriels*, *la prédiction des faillites personnelles*, *la reconnaissance de la voix*, montrent clairement l'efficacité de notre nouvelle méthode et de son avantage et polyvalence comparés aux autres méthodes spécifiquement développées pour des domaines particuliers.

4.6. Méthode d'utilisation de la théorie de Karlin

Une caractéristique importante dans le calcul de la similarité entre deux séquences est la longueur minimale des motifs importants. La théorie statistique de Karlin *et al.* [60, 61] a été développée spécifiquement pour estimer une telle longueur. Cette théorie estime la longueur $K_{R,L}$ du plus long motif présent par chance au moins R fois dans L séquences, comme ceci :

$$K_{R,L} = \frac{\log n(|S_1|, \dots, |S_L|) + \log \lambda(1 - \lambda) + 0.577}{-\log \lambda}$$

$$n(|S_1|, \dots, |S_L|) = \sum_{1 \leq i_1 \leq \dots \leq i_R \leq L} \prod_{v=1}^R |S_{i_v}|$$

avec $\lambda = \max_{1 \leq i_1 \leq \dots \leq i_R \leq L} \left(\sum_{t=1}^m \prod_{j=1}^R p_i^{(v_j)} \right)$ et $\sigma \approx \frac{1.283}{|\log \lambda|}$

Dans les formules ci-dessus, S_i est la i^{eme} séquence de l'ensemble des L séquences, m est la taille de l'alphabet constituant ces séquences, $p_i^{(v_j)}$ est généralement spécifié comme la fréquence du i^{eme} caractère de la v^{eme} séquence, tandis que σ est la déviation standard de $K_{R,L}$. D'après cette théorie, un motif est considéré significatif si sa longueur dépasse $K_{R,L} + 2\sigma$. Ce critère garantit à un motif qui est considéré comme significatif d'avoir moins de 1% de probabilité d'apparaître par chance dans les L séquences.

La théorie statistique de Karlin offre une formulation simple et générale du problème de la recherche des motifs importants dans les séquences catégoriques en général. Cependant, cette théorie a été appliquée avec succès par ces auteurs sur les séquences ADN et protéines [60, 61]. Elle a été aussi utilisée avec succès dans plusieurs travaux de recherches dans le développement de nouveaux algorithmes en bioinformatique [96, 121]. Nous avons utilisé cette théorie dans nos travaux de recherches en raison de sa simplicité et de son efficacité.

Une utilisation conventionnelle de ce théorème serait de l'appliquer directement sur l'ensemble des L séquences afin d'estimer une seule valeur pour la longueur minimale des motifs à considérer comme significatifs. Cependant, nous avons trouvé qu'une telle utilisation du théorème augmente considérablement le risque de collecter des motifs présents par chance, surtout dans les séquences les plus similaires. Nous avons discuté ce problème en détails et l'avons illustré avec un exemple dans le troisième chapitre qui présente le papier SCS.

Pour remédier à ce problème, nous avons eu l'idée d'appliquer le théorème de Karlin sur chaque paire de séquences de l'ensemble des L séquences, au lieu de l'appliquer naïvement sur tout l'ensemble des séquences. Cette manière d'utiliser le théorème nous a permis de diminuer considérablement le risque de collecter des motifs présents par chance par les différents algorithmes de recherche de motifs que nous avons développés. Notre méthode d'utilisation de la théorie de Karlin est l'une des raisons les plus importantes – sinon la plus importante - derrière l'efficacité et le succès des algorithmes que nous avons développé pour la mesure de similarité entre les séquences de protéines et autres. Nous avons expliqué notre

méthode d'utilisation du théorème de Karlin en détails dans le travail que nous avons publié dans Kelil *et al.* [65], et que nous présentons dans le troisième chapitre de cette thèse.

Lors de nos travaux de recherches, nous avons appliqué cette théorie sur les séquences de protéines (alphabet à 20 caractères), le texte en langage naturel (alphabet à 26 caractères), la voix (alphabet à 256 caractères), ainsi que la musique (alphabet à 1024 caractères). Les différents résultats que nous avons obtenus montrent clairement l'efficacité et l'avantage de l'application de la théorie de Karlin sur des paires de séquences pour l'estimation de la longueur minimale des motifs importants dans les séquences. Il nous reste maintenant (c.à.d. dans des travaux future) à tester l'efficacité de cette méthode sur des séquences à alphabet plus petit, comme par exemple les séquences ADN et ARN, qui sont constitués d'alphabets à 4 caractères.

5. Clustering

5.1. État de l'art

Le clustering s'appelle aussi la « *classification non supervisée* », dont l'objectif est de découvrir les regroupements naturels d'un ensemble de motifs, de points ou d'objets quelconques, ou dans notre cas des séquences de protéines. En fait, il n'y a pas d'accord universel sur la définition standard du clustering [31]. En pratique, la plupart des chercheurs décrivent un cluster en tenant compte de son homogénéité interne et de son hétérogénéité externe [57]. En d'autres termes, les motifs au sein du même cluster devraient être similaires, tandis que les motifs dans différents clusters ne le devraient pas [119].

Le clustering est utilisé pour regrouper les séquences de protéines en familles en fonction de leurs similarités, ce qui fournira des indices importants sur les caractéristiques générales des familles de protéines. Il est utile pour déduire la fonction biologique d'une nouvelle protéine par son appartenance à une famille de protéines bien connues et annotées. Il peut être également utilisé pour faciliter la découverte des structures 2D/3D des protéines, ce qui est

très important pour la découverte des fonctions des protéines [19]. Nous trouvons dans la littérature deux types d’algorithmes pour le clustering des séquences de protéines.

Le premier type d’algorithmes est dédié au clustering des grandes banques de protéines. On peut citer par exemple l’algorithme BlastClust⁶ (Dondoshansky et Wolf, non publié), qui appartient à la célèbre collection de BLAST [2]⁷, ou alors les algorithmes SYSTERS [74], ProtClust [113] et ProtoMap [152], CluSTr [75], Hybrid [45], et bien d’autres. Ces algorithmes ont été conçus spécifiquement pour faire face aux très grands ensembles de protéines. Pour ce faire, ils utilisent diverses techniques et méthodes pour accélérer le clustering et l’examen des similitudes (c.à.d. des régions locales similaires) entre les structures primaires des protéines, au dépend bien entendu de la précision. Ceci a pour effet de les rendre peu sensibles aux subtiles similarités et différences qui existent entre les protéines de la même famille.

Pour le clustering des plus petits ensembles de protéines, on utilise généralement un deuxième types d’algorithmes qui sont beaucoup plus précis, mais bien entendu au dépend de la rapidité. À titre d’exemple, on peut citer l’algorithme TRIBE-MCL [30] basé sur la méthode des chaînes de Markov, ou gSPC [134] basé sur une méthode analogue au traitement des « *inhomogènes ferromagnétiques* » en physique, ou Secator [147] qui utilise la méthode du clustering hiérarchique ascendant avec la matrice de distances basée sur l’alignement multiple, ou COCO-CL [58] qui utilise le clustering hiérarchique basé sur l’exploration des corrélations évolutionnaires, ou celui de Sjölander [125] basé sur l’entropie relative en combinaison avec les mixtures de Dirichlet.

En générale, on peut classer les algorithmes de clustering des séquences de protéines dans trois catégories différentes, voir Tableau 2 et Figure 8. Les récents progrès et défis dans le domaine du clustering des séquences de protéines ont été revus en détails par Sjölander *et al.* [124], et aussi par Abdul Rahman *et al.* [1].

⁶ <http://toolkit.tuebingen.mpg.de/blastclust>

⁷ <ftp://ftp.ncbi.nih.gov/blast/>

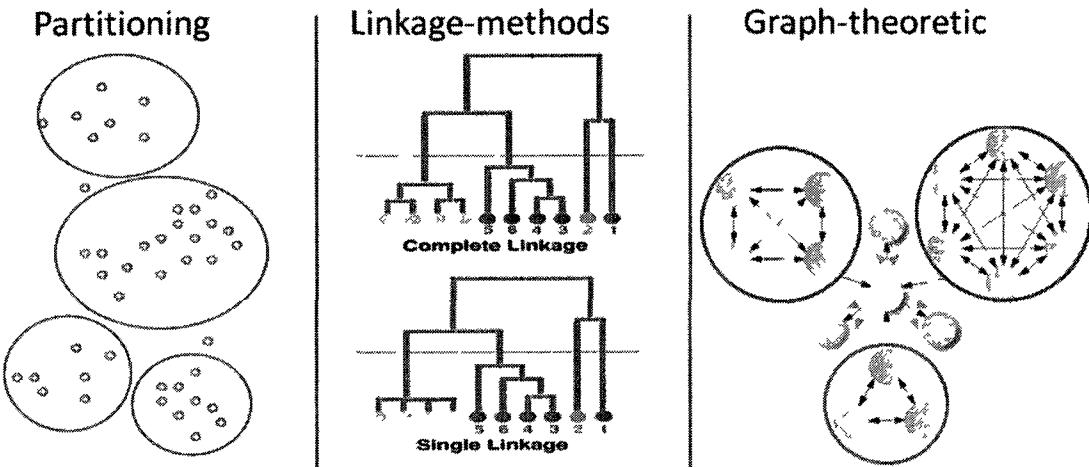


Figure 8. Les trois principales catégories d'algorithmes utilisées pour le clustering des séquences de protéines

5.1.1. Algorithmes de clustering hiérarchiques

Parmi toutes les approches utilisées en bioinformatique pour le clustering des séquences de protéines, les approches hiérarchiques sont sans doute celles qui sont le plus couramment utilisées, voir Tableau 3 et Tableau 4, et ceci est dû à plusieurs facteurs essentiels. Une approche de clustering hiérarchique typique produit un ensemble structuré de clusters, où à l'intérieur de chaque cluster les protéines sont regroupées de manière hiérarchique. Cette représentation des clusters et des protéines qui les composent, et qui est souvent utilisée pour l'étude phylogénétique des protéines, est biologiquement plus informative pour l'étude et la compréhension des protéines que l'ensemble des groupes non structurés retourné par les approches de clustering par partitionnement, connus sous le nom de « *flat-clustering* » [91], tels que les algorithmes cités dans le Tableau 5. Le clustering hiérarchique permet aussi de capturer avec plus de précision les formes naturelles des clusters, en particulier ceux dont les limites sont irrégulières. En outre, il ne nécessite pas le nombre de clusters comme paramètre d'entrée. Ces avantages se font bien entendu au détriment de la complexité. Les algorithmes les plus courants de clustering hiérarchique ont une complexité qui est au moins quadratique en termes de nombre d'objets par rapport à la complexité linéaire des algorithmes de clustering basés sur le partitionnement [91].

Tableau 2. Aperçu des différentes approches utilisées pour le clustering des séquences de protéines

Hierarchical clustering		Partitioning clustering
Linkage-methods	Graph-theoretic	
Single Linkage	Profile Hidden Markov Models	K-means
Recurrent Single Linkage	Markov Cluster Algorithm	K-medians
Average Linkage	Super Paramagnetic Clustering	Greedy Incremental
Markov Cluster Algorithm	Spectral Clustering	Algorithm
	Restricted Neighbourhood Search Clustering	Incremental Algorithm
	Ward's dissimilarity	

Le clustering hiérarchique peut être soit agglomératif dit aussi « *ascendant* », ou divisif dit aussi « *descendant* ». Le clustering hiérarchique ascendant commence avec chaque objet comme un seul groupe, et fusionne ensuite itérativement les groupes en de plus grands groupes. Tandis que le clustering hiérarchique descendant commence avec tous les objets comme un seul groupe, et subdivise ensuite itérativement chaque groupe en de plus petits groupes.

D'une part, les algorithmes de clustering ascendant prennent les décisions de fusionnement des objets et des groupes en se basant sur les tendances locales, sans tenir compte de la structure globale des objets. Avec ce type d'approche, les décisions de fusionnement faites lors des premières itérations ne peuvent pas être annulées dans le choix final du clustering [91]. D'autre part, les algorithmes de clustering descendant bénéficient d'une information plus complète sur la structure globale des objets lorsqu'ils prennent des décisions de partitionnement. Ceci fait en sorte que les algorithmes descendants produisent des clusters plus précis que les algorithmes ascendants. Toutefois, ils sont plus couteux en calcul et en espace. C'est la raison pour laquelle le clustering hiérarchique ascendant est plus répandu que le clustering hiérarchique descendant. Pour une discussion plus détaillée, consulter le livre référence intitulé « *introduction to information retrieval* » édité par Manning *et al.* [91].

La majorité des algorithmes basés sur le clustering hiérarchique dit « *linkage-methods* » examinent les relations entre les objets par paires, et ensuite représentent les objets par une structure hiérarchique, où les objets sont connectés par paires. Souvent il devient très difficile

Tableau 3. Algorithmes de clustering hiérarchiques, basés sur « linkage-methods »

Algorithmme	Mesure de similarité	Méthode de Clustering
BlastClust[2]	BLAST	Single Linkage
GeneRAGE[26]	Smith et Waterman	Recursive Single Linkage
Systers[58]	Smith et Waterman	Recursive Single Linkage
CluSTr[59]	Smith et Waterman	Single Linkage
ProClust[86]	Smith et Waterman	Profile Hidden Markov Models
Hybrid[36]	BLAST	Single Linkage + Markov Cluster Algorithm
CLUGEN[69]	Smith et Waterman	Average Linkage
FORCE[114]	BLAST	Single Linkage
COCO-CL[47]	ClustalW	Recursive Single Linkage

Tableau 4. Algorithmes de clustering hiérarchiques, basés sur la « théorie des graphs »

Algorithmme	Mesure de similarité	Méthode de Clustering
gSPC[101]	/	Step-Wise
TRIBE-MCL[25]	/	Markov Cluster Algorithm
TransClust[11]	BLAST	Transitivity clustering
SCPS[77]	FASTA	Spectral Clustering
Secator[112]	Smith et Waterman	Ward's dissimilarity

Tableau 5. Algorithmes de clustering basés sur le « partitionnement »

Algorithmme	Mesure de similarité	Méthode de Clustering
CD-HIT[44]	short word filtering	Greedy incremental clustering
Scalable[34]	K-words	K-means
Leaders-Subleaders[107]	Smith et Waterman	Incremental clustering
K-medians[107]	Smith et Waterman	K-medians

Les tableaux ci-dessus montrent la mesure de similarité et la méthode de clustering adoptée par chaque algorithme d'utiliser cette structure pour représenter les complexes réseaux d'interactions et systèmes biologiques. On utilise alors un autre type d'algorithmes basés sur la théorie des graphes.

Ces dernières années, la théorie des graphes, ainsi que les algorithmes impliqués ont ouvert de nouvelles voies dans la biologie contemporaine pour la compréhension de la structure, la fonction et l'évolution des systèmes biologiques complexes jusqu'ici impossible à concevoir

avec les méthodes classiques [9, 10]. Parmi les algorithmes de clustering hiérarchique des séquences de protéines, ceux basés sur la théorie des graphes sont sans doute, selon la tendance actuelle, ceux qui auront le plus de succès dans les prochaines années. Car, avec le déluge d'information biologiques qui ne cesse d'affluer, les graphes (ou réseaux) constituent un cadre très solide dans lequel les processus et systèmes biologiques, qui deviennent de plus en plus vastes et complexes, peuvent être convenablement modélisés et mieux appréhender. Nous citons quelques-uns des algorithmes les plus connus dans le Tableau 4. Par exemple, le récent projet TERAPROT (malheureusement qui a cessé d'exister) a pu faire l'usage des ordinateurs du CEA-DAM, construit initialement pour la simulation des essais nucléaires, pour comparer un ensemble de 240,000 séquences de protéines déduites de 67 génomes complets pour obtenir un graphe avec 625×10^8 d'arêtes. L'exploitation de ce type de graphe, entre autre en utilisant le clustering, aidera les scientifiques à progresser plus rapidement dans la compréhension des fonctions biologiques des protéines. Les connaissances ainsi acquises sur les organismes les plus élémentaires renseigneront sur l'organisation et le fonctionnement des organismes les plus évolués, dont celui de l'Homme.

Dans le Tableau 3 et le Tableau 4, on voit que malgré que les algorithmes de clustering hiérarchiques utilisent différentes approches pour le clustering des séquences de protéines, la plupart d'entre eux calculent la similarité entre les séquences de protéines en utilisant des méthodes basées sur l'alignement des séquences, comme BLAST [2], ou alors l'algorithme de « *programmation dynamique* » de Smith et Waterman [126]. Les algorithmes gSPC et TRIBE-MCL, quant à eux, n'ont pas de mesure de similarité propre à eux, et utilisent plutôt des mesures de similarité tierces. Les algorithmes de clustering qui se basent sur l'alignement des séquences font face à des difficultés majeures qui les rendent dans certaines situations incapables de produire des résultats biologiquement plausibles [49, 112]. Les algorithmes d'alignement sont fondés sur l'appariement des résidus dans des positions équivalentes, et supposent que les séquences sont généralement alignables, alors que souvent elles ne le sont pas, car elles contiennent fréquemment des régions conservées dans des positions non-équivalentes, surtout lorsqu'il s'agit de séquences qui contiennent des domaines répétés, inversés, supplémentaires ou manquants. En plus de ça, les résultats de l'alignement

dépendent fortement des paramètres d'entrée choisis par l'utilisateur (i.e., pénalités de « *gap* », matrice de substitution, etc.). Les difficultés les plus importantes qu'affrontent les algorithmes d'alignement ont été revues en détails par Higgins *et al.* [49], Mount [99], et aussi par Phuong *et al.* [112].

5.1.2. Algorithmes de clustering non-hiéronymiques

Il ne faut pas, cependant, négliger en bioinformatique le rôle des algorithmes de clustering des séquences de protéines non-hiéronymiques dits « *partitioning-methods* » ou plus encore « *flat-clustering* », voir Tableau 5. Ces algorithmes non-hiéronymiques qui sont développés souvent pour le clustering des grands ensembles de séquences de protéines, ne diffèrent pas seulement des algorithmes hiérarchiques par la méthode de clustering adoptée, mais aussi sur la manière de comparer les séquences de protéines. Tandis que les algorithmes de clustering hiérarchiques utilisent principalement l'alignement ou des méthodes basées sur l'alignement pour comparer les séquences de protéines (voir Tableau 3 et Tableau 4), les algorithmes non-hiéronymiques utilisent plutôt des méthodes pour capturer les caractéristiques essentielles des séquences de protéines, qui sont projetées ensuite dans des espaces multidimensionnelles, où les méthodes de calcul vectoriel sont appliquées pour comparer les séquences de protéines.

L'idée principale de ces algorithmes est de trouver un ensemble de caractéristiques, souvent appelés motifs ou « *bag of words* », capables de capturer la nature séquentielle des séquences de protéines [42]. Cette façon de faire a l'avantage d'être très efficace en termes de calcul et d'espace, mais malheureusement elle n'est pas aussi efficace sur les plus petits ensembles de séquences de protéines que les algorithmes hiérarchiques. Cependant, contrairement aux algorithmes hiérarchiques qui ont une complexité le plus souvent quadratique, ces algorithmes permettent de comparer un grand nombre de séquences de protéines dans un temps quasi-linéaire par rapport au nombre de séquences comparées. Par exemple, l'algorithme CD-HIT [55] qui utilise une méthode similaire dite « *short word filtering* », a été capable de comparer toutes les paires possibles d'un ensemble de 560,000 séquences de protéines en seulement 2 heures de temps, et cela sur une machine standard [84]. Ces algorithmes sont très pratiques pour filtrer les redondances dans les grandes bases de données

de séquences de protéines [84], ce qui permet de générer des sous-ensembles plus représentatifs qui sont plus rapides pour la recherche de similarités, et améliorent aussi la cohérence de l'annotation des séquences de protéines [53]. Par exemple, CD-HIT a été utilisé aussi pour générer des ensembles non-redondants de protéines par la banque de données UniProt⁸, qui est actuellement le catalogue d'information le plus complet sur les protéines.

En raison des difficultés citées plus haut, il existe un grand nombre de protéines que les biologistes ne peuvent pas étudier en utilisant les algorithmiques existantes. Ces protéines dont le nombre ne cesse d'augmenter chaque jour, sont soit laissées de côté, ou alors, dans de rares cas, sont étudiées avec des méthodes coûteuses en temps et en ressources, comme celles présentées par Boucher *et al.* [16], Marcotte *et al.* [92], Fukamizo *et al.* [37], Côté *et al.* [21], et Saito *et al.* [120]. Pour toutes ces raisons nous avons développé une série d'algorithmes hiérarchiques pour le clustering des séquences de protéines mais qui ne dépendent pas de l'alignement. Nous résumons ces algorithmes dans ce qui suit.

5.2. CLUSS

Nous avons développé un algorithme de clustering hiérarchique divisif, nommé CLUSS, qui est la première version d'une série d'algorithmes que nous avons conçus pour le clustering des séquences de protéines. Notre algorithme CLUSS tire avantage du fait qu'il utilise une approche hiérarchique divisif, ce qui lui permet de bénéficier d'une information plus complète sur la structure globale de la représentation hiérarchique des relations de similarités entre les séquences de protéines. Ceci fait en sorte que CLUSS est capable de produire des clusters plus précis. Par rapport aux algorithmes existants, CLUSS produit des clusters qui mettent en évidence de façon plus précise les caractéristiques structurelles et fonctionnelles des protéines. Il fournit aux biologistes un nouvel outil efficace pour l'analyse des protéines, en particulier celles qui causent des problèmes aux algorithmes basés sur l'alignement. Grâce à CLUSS, nous avons développé le premier serveur web capable d'effectuer le clustering des protéines sans faire appel à l'alignement (voir annexe 3).

⁸ <http://www.uniprot.org>

La nouveauté de CLUSS réside dans trois caractéristiques très importantes. Tout d'abord, CLUSS est appliqué directement sur des séquences de protéines non-alignées, éliminant ainsi le besoin de pré-aligner les séquences de protéines avant le clustering. Deuxièmement, il adopte notre nouvelle mesure de similarité SMS, qui est capable de détecter efficacement les motifs les plus importants qui représentent le mieux les propriétés séquentielles intrinsèques des protéines. Et finalement, CLUSS produit aussi les arbres phylogénétiques des ensembles de séquences de protéines.

CLUSS utilise une variante de la méthode de Ward [146] introduite par Batagelj [11] pour la représentation hiérarchique des relations entre les séquences de protéines. Il utilise aussi la méthode développée par Thompson *et al.* [137] pour évaluer l'importance de chaque protéine dans la structure hiérarchique. Il utilise également le théorème de Koenig-Huygens, qui donne la relation entre l'inertie totale (c.à.d. hétérogénéité entre les clusters) et l'inertie de chaque groupe (c.à.d. homogénéité dans les clusters) par rapport au centre de gravité de l'ensemble, pour trouver automatiquement les groupes de protéines les plus similaires qui constituent les clusters les plus homogènes. Nos expérimentations ont montré que, par rapport aux algorithmes de clustering classiques, CLUSS non seulement est plus efficace pour regrouper les séquences de protéines non-alignables selon leurs caractéristiques structurelles et fonctionnelles, mais est aussi performant que les algorithmes basés sur l'alignement, sur les séquences de protéines qui sont alignables.

Toutefois, vu que CLUSS utilise SMS pour mesurer la similarité entre les protéines, il hérite aussi de ses faiblesses. Donc, CLUSS a tendance à être moins efficace lorsqu'il est appliqué à de grands ensembles de protéines contenant de grands nombres de fonctions biologiques. En plus de cela, ajoutées à la complexité de SMS, les méthodes utilisées dans CLUSS pour la représentation hiérarchique et l'estimation de l'importance de chaque séquence de protéine parmi les autres protéines ont une complexité quadratique. Ce qui empêche CLUSS d'être efficace sur les grands ensembles de séquences de protéines.

5.3. CLUSS2

Nous avons développé CLUSS2, un nouvel algorithme pour le clustering des grands ensembles de séquences de protéines contenant de grands nombres de fonctions biologiques. CLUSS2 est un algorithme de clustering hiérarchique basé sur notre nouvelle mesure de similarité tSMS, qui est en même temps une extension et une amélioration de la mesure de similarité SMS utilisée dans CLUSS. Contrairement à SMS qui n'autorise que les appariements identiques, tSMS autorise l'appariement des motifs similaires qui est la principale raison de l'efficacité de CLUSS2.

Grâce à la technique de décomposition spectrale sur la matrice de similarité utilisée dans tSMS pour la construction d'un espace vectoriel où chaque séquence est représentée par un vecteur, CLUSS2 utilise les opérations vectorielles durant le processus de clustering. Ainsi chaque cluster est représenté par un candidat unique (i.e., centroid), ce qui accélère considérablement la phase de clustering. Un autre avantage est la possibilité d'utiliser des approximations pendant la décomposition spectrale pour réduire encore plus le temps de calcul. Nos différentes expérimentations ont montré que CLUSS2 est beaucoup plus rapide et efficace que ne l'est CLUSS, spécialement pour les grands ensembles de protéines contenant de grands nombres de fonctions biologiques. Nous avons aussi ajouté CLUSS2 au serveur web que nous avons développé, ainsi les biologistes ont accès dorénavant aux deux versions CLUSS et CLUSS2 (voir annexe 3).

6. Alignement

6.1. État de l'art

A partir de l'hypothèse que si des protéines comportent des régions conservées alors elles risquent aussi de partager certaines propriétés physico-chimiques, nous pouvons alors à partir de l'alignement des séquences de protéines, identifier les régions conservées et proposer des hypothèses sur le fonctionnement des protéines dont on ne connaît pas les mécanismes d'actions, et qui peuvent être vérifiées d'une manière expérimentale.

En bio-informatique, l'alignement des séquences de protéines est le processus d'apparier les acides aminés dans les séquences pour identifier les régions de similarités, notamment pour prédire :

- Les sites fonctionnels;
- Les régions conservées et les régions variables;
- Les fonctions des protéines;
- Les structures secondaires des protéines;
- Les relations de phylogénie;
- Les caractéristiques communes aux familles de protéines;
- Le lien entre la séquence, la structure, et à la fonction;

En pratique, pour aligner trois séquences de protéines de longueur 1000, il faut garder en mémoire 1000^3 scores de substitutions, ce qui correspond exactement à 1 Go de mémoire. Pour quatre séquences de protéines, il faut donc 1000 Go de mémoire. Ayant souvent plusieurs dizaines voire même des centaines de séquences à aligner, il est donc absolument hors de question d'utiliser un algorithme exact [93] pour aligner des séquences de protéines. C'est pourquoi plusieurs approches d'alignements approximatifs ont été développées à ce jour. On résume ces approches dans les sous-sections suivantes.

6.1.1. Alignement progressif

Dans la littérature, c'est l'approche la plus répandue dans le domaine de l'alignement des séquences [99]. D'ailleurs, elle a fait le succès des algorithmes d'alignement les plus célèbres, comme MUSCLE [29], T-COFFEE [104], MAFFT [63], et CLUSTAL [77], et bien d'autres. Cette méthode consiste en général à décomposer le problème de l'alignement de N séquences en $N(N-1)/2$ alignements de 2 séquences en utilisant souvent l'algorithme de « *programmation dynamique* » développé par Needleman et Wunsch [100]. Ensuite, les alignements ainsi obtenus sont alignés ensuite les uns aux autres par paires de manière progressive grâce à une structure hiérarchique ascendante des relations de similarités entre les protéines obtenues avec un clustering hiérarchique. L'inconvénient majeur de cette méthode

est qu'elle ne revient jamais en arrière pour réévaluer les alignements déjà effectués. Une erreur qui peut survenir à n'importe quelle itération de l'alignement ne peut jamais être corrigée dans les itérations suivantes, et se propagent à travers le processus de l'alignement, dégradant ainsi la qualité de l'alignement final. Cet inconvénient fait en sorte que cette méthode ne fonctionne pas bien sur les protéines qui contiennent de longues insertions internes ou des extensions N/C terminales, ou alors des répétitions en tandem [49]. Pour remédier à ce problème, souvent des méthodes itératives de raffinement de l'alignement sont utilisées, comme celles introduites par Hirosawa *et al.* [51] et Edgar [29], pour corriger les erreurs qui ont pu se produire dans les itérations du processus d'alignement, où l'alignement final est comparé itérativement à des alignements alternatifs, qui le remplacent s'ils sont jugés meilleurs.

6.1.2. Alignement itératif

Ce type d'alignement est assez populaire dans le domaine de l'alignement des séquences de protéines [99]. Par exemple, PRRP [39], qui utilise un algorithme itératif pour corriger l'alignement ainsi que les régions localement divergentes, obtient de meilleurs résultats quand il est appliqué pour le raffinement d'un alignement déjà construit par une méthode plus rapide [99]. Ou encore MUSCLE [29], bien qu'il soit un algorithme d'alignement progressif, utilise l'alignement itératif pour améliorer l'alignement obtenu par l'alignement progressif. L'alignement itératif a pour objectif de tenter d'améliorer le point faible de l'alignement progressif, qui est la forte dépendance à la qualité des alignements par paires obtenues initialement. Au lieu d'aligner les séquences progressivement, l'alignement itératif choisit plutôt des sous-groupes de séquences à aligner, ensuite ces sous-groupes sont alignés à leurs tours pour former un alignement global. L'alignement global est utilisé à son tour pour faire le choix des nouveaux sous-groupes de séquences à aligner lors de la prochaine itération. Le processus se termine quand il n'y a plus d'amélioration dans la qualité de l'alignement global. Différentes méthodes de sélection des sous-groupes de séquence et aussi pour l'évaluation de la qualité des alignements de séquences ont été revues par Hirosawa *et al.* [51].

6.1.3. Model de Markov caché

C'est un modèle probabiliste qui peut assigner une vraisemblance à toutes les combinaisons possibles de gaps, appariement, et mésappariements pour déterminer l'alignement le plus vraisemblable parmi tous les alignements possibles [109, 122]. En plus de produire un alignement le plus probable, le model de Markov caché peu produire également toute une famille d'alignements alternatifs possibles qui peuvent être alors évalués biologiquement. Malgré que les algorithmes basés sur le modèle de Markov caché aient été développés relativement récemment, ils offrent des améliorations significatives en termes de temps de calcul, spécialement pour les séquences qui contiennent des chevauchements [99]. Une méthode typique d'alignement basée sur le modèle de Markov caché représente un alignement sous une forme de graphe acyclique dirigé, connu comme un graphe d'ordre partiel, qui consiste en une série de nœuds représentant des entrées possibles dans les colonnes de l'alignement. Dans cette représentation, une colonne qui est parfaitement conservée (à savoir que toutes les séquences dans l'alignement partagent un résidu particulier à une position particulière) est codée par un seul nœud avec autant de connexions sortantes que de résidus dans la colonne suivante de l'alignement. Dans le modèle de Markov caché, les états « *observés* » sont les colonnes d'alignement individuel, tandis que les états « *cachés* » représentent les séquences ancestrales présumées à partir desquelles les séquences dans la requête sont hypothétiquement les descendants. Une variante de l'algorithme d'alignement de Needleman et Wunsch [100] connue sous le nom d'algorithme de « *Viterbi* » [34, 144] est généralement utilisée conjointement avec les méthodes d'alignements basées sur le modèle de Markov Caché [56]. L'avantage qui distingue vraiment les algorithmes d'alignements basés sur le modèle de Markov caché est que, les alignements préalables sont toujours mis à jour à chaque addition de nouvelles séquences à l'alignement global. Toutefois, cette technique peut être influencée par l'ordre dans lequel les séquences dans la requête sont intégrées dans l'alignement global, en particulier lorsque les séquences sont distantes [99].

6.1.4. Algorithmes génétique

Les techniques d'optimisation standard en informatique inspirées par des processus naturels ont également été utilisées pour tenter de produire des alignements de qualité. Une de ces techniques est l'algorithme génétique d'alignement [99], qui a été utilisé pour la production d'alignements en se basant sur la simulation de l'hypothèse faite sur le processus d'évolution qui a donné lieu à la divergence dans l'ensemble des séquences dans la requête. En principe, l'algorithme génétique d'alignement fonctionne en brisant l'alignement global, obtenu par une méthode plus rapide, en une série de fragments d'alignements possibles, qui sont à leurs tours réarrangés à plusieurs reprises pour donner un nouvel alignement global par l'introduction de gaps à des positions différentes. Une fonction objective est généralement optimisée durant la simulation. Le plus souvent c'est la fonction de score dite « *sum-of-pairs* » [138, 139], la même qui est fréquemment utilisée par l'algorithme de « *programmation dynamique* » dans les algorithmes d'alignements progressifs. En pratique, l'algorithme génétique d'alignement n'est pas une méthode couramment utilisée dans le domaine de l'alignement des séquences. Ceci est dû principalement au fait que c'est une méthode qui est particulièrement difficile à implémenter d'une manière efficace, en plus de son coût exorbitant en temps de calcul. Toutefois on peut citer deux exemples d'algorithmes, l'algorithme SAGA [103] développé pour l'alignement des séquences de protéines, et son équivalent RAGA [105] développé pour l'alignement des séquences d'ARN.

6.1.5. Recuit simulé

La méthode de l'alignement basée sur le recuit simulé a pour objectif d'améliorer un alignement déjà existant [99]. Un alignement qui a été produit par une autre méthode est raffiné par une série de réarrangements visant à trouver des régions d'alignements meilleures que celles de l'alignement d'entrée. Comme la méthode d'alignement basée sur l'algorithme génétique, la méthode basée sur le recuit simulé maximise une fonction objective, qui est le plus souvent la fonction de score communément connue sous le nom de « *sum-of-pairs* » [138, 139]. Le recuit simulé utilise la métaphore appelée « *facteur de température* » qui détermine la vitesse à laquelle les réarrangements se poursuivent et la probabilité de chaque

réarrangement. Un recuit simulé typique serait d'alterner les périodes de forts réarrangements avec une vraisemblance relativement faible (pour explorer les régions les plus éloignées de l'alignement) avec des périodes de faibles réarrangements avec des vraisemblances plus élevées pour explorer minutieusement les minima locaux à proximité des régions nouvellement colonisées. Le recuit simulé souffre des mêmes problèmes que l'algorithme génétique. Cette approche a été mise en œuvre dans l'algorithme MSASA introduit par Kim *et al.* [70], ou plus récemment dans AMAP introduit par Schwartz *et al.* [5].

6.1.6. Recherche de motifs

La recherche de motifs, connu aussi comme « *analyse de profils* », est une méthode de localisation des motifs importants dans les séquences déjà alignées, dans le but de produire de meilleurs alignements. Une variété de méthodes pour isoler les motifs ont été développés, mais toutes sont basées sur l'identification de courts motifs hautement conservés dans des alignements déjà existants, qui sont ensuite utilisés pour la construction de matrices de scores qui reflètent les tendances des acides aminés dans chaque position dans les motifs présumés. L'alignement original peut donc être raffiné en utilisant ces matrices [99]. La méthode d'*analyse des blocs*, connue aussi comme « *blocks analysis* », est une autre méthode pour trouver les motifs dans les séquences de protéines, mais elle restreint la recherche à des motifs dans les régions sans gaps. Les blocs peuvent être générés à partir d'alignements existants, ou encore être extraits de séquences non alignées en utilisant un ensemble de motifs déjà calculé à partir de familles de séquences connus [46-48]. Le serveur BLOCKS⁹ fournit une méthode interactive pour localiser de tels motifs dans les séquences non alignées. L'appariement statistique des motifs est aussi utilisé pour la recherche de motifs dans les séquences à aligner. Il utilise à la fois l'algorithme de maximisation de la vraisemblance dit « *expectation maximization* » et l'algorithme d'échantillonnage de Gibbs dit « *Gibbs sampler* ». Un des outils de recherche de motifs les plus courants est connu sous le nom MEME. Il utilise l'algorithme de maximisation de la vraisemblance avec la méthode du modèle de Markov caché pour générer des motifs qui sont ensuite utilisés comme paramètres

⁹ <http://blocks.fhcrc.org>

de recherche de motifs pour l'alignement des séquences de protéines par MAST dans la collection MEME [7, 8].

6.1.7. Alignement local et global

Indépendamment de la méthode d'alignement utilisé, la littérature nous rapporte deux types d'approches principales pour l'alignement des séquences de protéines, l'alignement « *global* » et l'alignement « *local* », voir Figure 9 et Figure 10. À titre d'exemple on peut citer les algorithmes d'alignement global tels que MUSCLE 3.7 [29], CLUSTALX [136], T-COFFEE [104], et les algorithmes d'alignement local tels que DIALIGN [97], DIALIGN-TX [130], SB-PIMA [114] and ML-PIMA [114]. D'une part, l'alignement global a pour objectif de couvrir la totalité de la longueur de toutes les séquences des protéines à aligner, en alignant tous les acides aminés dans chaque séquence, Figure 9. D'autre part, l'alignement local a pour objectif la recherche des motifs les plus conservés et cela en identifiant les régions similaires dans des séquences de protéines à aligner qui sont souvent très divergentes en général Figure 10. Il a été démontré par les deux études réalisées par McClure *et al.* [93] et Thompson *et al.* [139] que l'approche d'alignement la plus efficace dépend essentiellement de la nature structurelle des protéines à aligner. Ces deux études ont montré que, souvent l'alignement global produit les résultats les plus fiables biologiquement. Mais en présence,

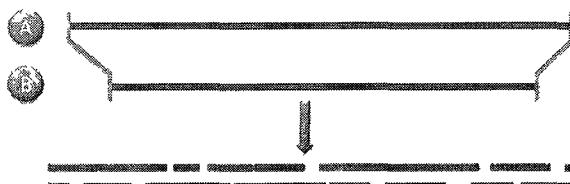


Figure 9. L'alignement global recherche les régions similaires sur la longueur des séquences

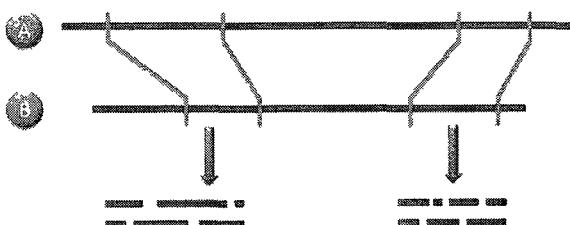


Figure 10. L'alignement local recherche les régions de similarités locales

dans les séquences de protéines, de grandes extensions N/C terminales ou alors de longues insertions internes, l'alignement local est le plus efficace pour trouver des alignements biologiquement valables. Cela est d'autant plus vrai quand il s'agit de séquences de protéines multi-modulaires [6, 135, 138].

Le problème le plus important avec ces deux types d'approches est que, sans connaissance préalable sur les propriétés structurelles et biochimiques de chacune des protéines à aligner, on ne peut pas choisir avec certitude l'approche à adopter pour effectuer l'alignement qui peut révéler les tendances les plus pertinentes fonctionnellement ou structurellement dans ces séquences de protéines. D'autre part, tous les algorithmes d'alignement existants, qu'ils soient basés sur l'alignement global ou local, supposent que les séquences de protéines à aligner sont généralement alignables. Alors, ils sont conçus de telle sorte que, pour un ensemble donné d'entrées de séquences de protéines, ils alignent toutes les séquences, et ignorent si l'ensemble comprend des protéines divergentes, celles qui ne partagent pas suffisamment de régions conservées pour produire des alignements biologiquement significatifs. Cela rend difficile l'identification des régions conservées. Pour faire face à ce problème, les biologistes sont souvent contraints à manipuler eux-mêmes les alignements ainsi obtenus en identifiant visuellement et en retirant les séquences de protéines qui semblent perturber les résultats des alignements, ce qui n'est pas toujours facile à effectuer en pratique, notamment lorsque les ensembles d'entrées comprennent plusieurs groupes de séquences de protéines divergents. Pour faire face à ces difficultés majeures, nous avons développé ALIGNER, un nouvel algorithme d'alignement de séquence de protéines.

6.2. ALIGNER

ALIGNER est un nouvel algorithme pour l'alignement des séquences de protéines. Contrairement à tous les algorithmes d'alignement qui existent déjà, notre algorithme est en mesure d'aligner de manière efficace autant les séquences de protéines qui nécessitent un alignement global que celles qui nécessitent un alignement local. Comme dans l'alignement global, ALIGNER s'étend sur toute la longueur des séquences à aligner en alignant tous les acides aminés dans chaque séquence. En même temps, ALIGNER donne une attention

particulière aux motifs significatifs partagés par les séquences de protéines (voir **Exemple 1** plus loin dans cette section). En plus, ALIGNER est capable de détecter dans l'ensemble d'entrée des protéines à aligner, les groupes de protéines qui partagent assez de régions similaires pour produire des alignements qui peuvent révéler d'importantes propriétés structurelles et fonctionnelles au sein de chaque groupe de protéines, et cela sans recourir à des manipulations par l'utilisateur sur l'ensemble d'entrée. Ce qui constitue en soi un avantage majeur pour les biologistes.

ALIGNER est un algorithme d'alignement de séquences de protéines qui intègre les trois différentes méthodes d'alignement suivantes, progressive, itératifs, et recherche de motifs. Ainsi ALIGNER tire avantage de la rapidité de la méthode progressive. Il utilise une méthode itérative efficace pour l'amélioration et le raffinement des alignements afin de corriger les faiblesses de la méthode progressive. Il utilise aussi une méthode de recherche de motifs pour la détection et l'alignement des régions les plus conservées, surtout celles qui se trouvent dans des régions éloignées dans les séquences qui sont souvent difficiles à aligner en utilisant l'alignement progressif.

Notre nouvel algorithme d'alignement ALIGNER tire un grand avantage de la dernière version de la méthode d'appariement des séquences de protéines que nous avons développée dans SMS, pour détecter automatiquement les motifs les plus significatifs partagés entre les protéines à aligner. En combinant cet algorithme d'appariement avec l'algorithme de programmation dynamique développé par Needleman and Wunsch [100], ALIGNER est capable de mesurer efficacement la similarité entre les protéines à aligner (voir **Exemple 2** plus loin dans cette section). De plus, en utilisant l'algorithme de clustering développé dans CLUSS2, ALIGNER est capable de détecter automatiquement parmi les protéines à aligner les groupes de protéines qui partagent assez de motifs significatifs pour produire des alignements qui peuvent révéler d'importantes propriétés structurelles et fonctionnelles.

Les résultats de nos tests expérimentaux ont montré clairement l'avantage d'ALIGNER face aux algorithmes existants. Notre nouvel algorithme s'est avéré plus efficace que les algorithmes d'alignement global sur les protéines qui nécessitent un alignement global, et

aussi plus efficace que les algorithmes d'alignement local sur les protéines qui nécessitent un alignement local. ALIGNER nous a permis de produire pour la première fois des alignements sur des familles de protéines jusque-là obtenues qu'avec des méthodes basées sur la structure tridimensionnelle des protéines.

ALIGNER est le premier algorithme d'alignement destiné à guider automatiquement les biologistes dans le choix des séquences de protéines à inclure dans les ensembles de séquences à aligner. Ceci permettra d'éviter le recours à des manipulations aléatoires ou arbitraires des ensembles de données d'entrée. En outre, notre algorithme permet d'aider et de réduire la charge de travail des biologistes, par le traitement automatique des séquences de protéines qui nécessitent un alignement global ou local. Il permet également d'éviter la manipulation des séquences de protéines qui ne partagent pas suffisamment de régions conservées pour produire des alignements biologiquement significatifs. Le serveur web de ALIGNER est situé à l'adresse <http://prospectus.usherbrooke.ca/ALIGNER> (voir annexe 4).

Exemple 1 : Dans cette exemple nous présentons la méthode utilisée dans ALIGNER pour aligner un ensemble de séquences de protéines. Dans l'exemple illustré dans la Figure 11, nous considérons le cas simple de l'alignement d'une séquence Y₁ avec trois autres séquences X₁, X₂, et X₃ qui sont déjà alignées. La méthode se résume suit :

- **Étape 1 :** D'abord, ALIGNER détecte les motifs les plus importants partagés entre toutes les paires de séquences de protéines (X_i, Y_j), voir Figure 11.A. Dans cette figure, les motifs détectés ont les couleurs suivantes : motifs rouges détectés entre X₁ et Y₁, motifs verts détectés entre X₂ et Y₁, motifs bleus détectés entre X₃ et Y₁.
- **Étape 2 :** Ensuite, ALIGNER aligne les séquences X₁, X₂, et X₃ avec la séquence Y₁ en tenant compte des motifs importants détectés dans l'étape précédente, voir Figure 11.B. Les motifs qui n'ont pas été considérés dans l'alignement ne sont pas perdus, ils seront plutôt utilisés pour le raffinement de l'alignement.

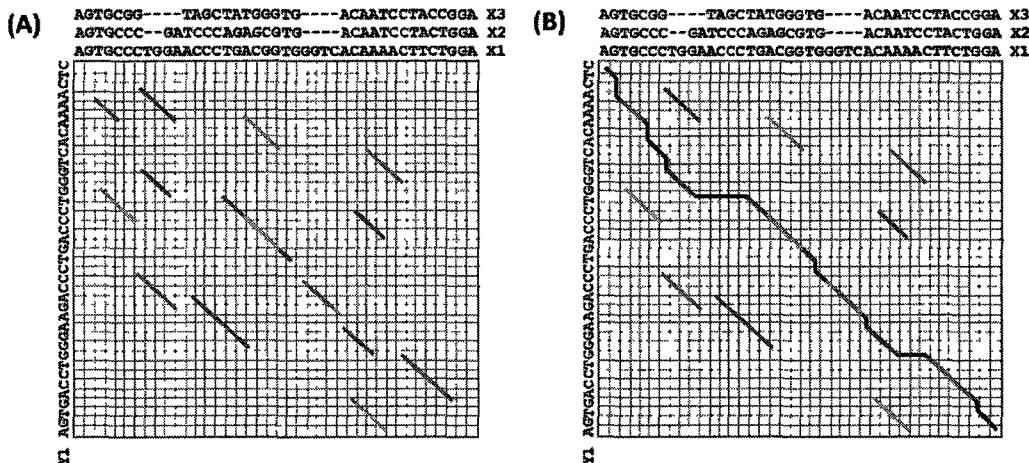


Figure 11. Exemple d'alignement d'un ensemble de séquences de protéines par ALIGNER
 (A) détection des motifs les plus importants partagés par les séquences de protéines
 (B) alignement des séquences de protéines tout en respectant le maximum de motifs

Exemple 2 : Dans cet exemple, nous présentons la méthode utilisée dans ALIGNER pour mesurer la similarité entre deux séquences de protéines, voir Figure 12.

- **Étape 1 :** D'abord, ALIGNER détecte les motifs les plus importants partagés par les deux séquences (motifs en rouge dans la Figure 12).
- **Étape 2 :** Ensuite ALIGNER aligne les deux séquences en tenant compte des motifs importants détectés dans l'étape précédente. La mesure de similarité sera égale au score de l'alignement obtenu normalisé par la longueur maximal des deux séquences.

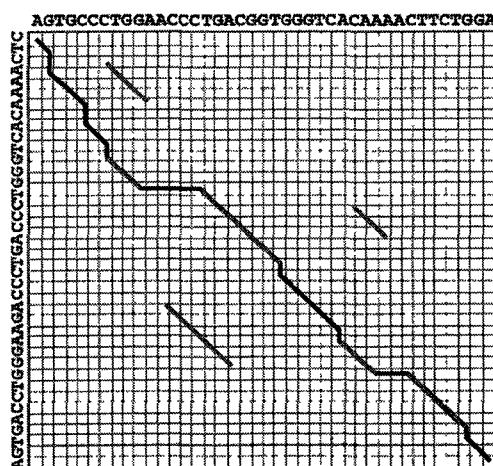


Figure 12. Exemple de mesure de similarité entre deux de séquences de protéines par ALIGNER

7. Conclusion

À travers cette introduction, nous avons présenté un bref aperçu sur les propriétés structurelles et fonctionnelles des protéines, et aussi leur rôle crucial au sein des organismes vivants. Nous avons aussi discuté de l'importance de pouvoir déterminer leurs fonctions biologiques. Ensuite nous avons présenté l'état de l'art des différentes méthodes qui ont été développé à ce jour pour l'analyse des protéines dans le but de prédire leurs fonctions biologiques, nous avons discuté alors des différents défis et problèmes que doivent surmonter ces méthodes. Enfin, nous avons présenté une série de méthodes et d'algorithmes, basés sur la similarité, le clustering, et l'alignement, que nous avons développés pour faire face à plusieurs défis majeurs dans l'analyse des séquences de protéines.

Dans le reste de cette thèse nous présentons les différentes publications dont ont fait les méthodes et algorithmes que nous avons développés.

Chapitre 1

CLUSTERING DES FAMILLES DE PROTÉINES

Le clustering des séquences de protéines en groupes qui partagent des fonctions ou des structures similaires est un problème important et encore non résolu en bioinformatique. Le clustering des séquences de protéines nécessite la résolution de deux problèmes distincts mais étroitement liés. Il faut tout d'abord définir une mesure qui permettra d'évaluer les similarités entre les séquences de protéines. Ensuite, il faut définir un critère ou un algorithme qui utilisera l'information de similarité pour grouper les séquences de protéines qui partagent le plus de propriétés structurelles ou fonctionnelles.

Dans ce chapitre nous présentons CLUSS, la première version d'une série d'algorithmes que nous avons conçus pour le clustering des familles de protéines. CLUSS adopte notre nouvelle mesure de similarité SMS qui est indépendante de l'alignement et capable de détecter efficacement les motifs représentant efficacement les propriétés séquentielles intrinsèques et biologiques des protéines. CLUSS est plus efficace que les algorithmes basés sur l'alignement pour regrouper les séquences de protéines non alignables selon leurs caractéristiques structurelles et fonctionnelles, en plus d'être aussi performant sur les séquences de protéines alignables. Nous avons développé un serveur Web de CLUSS situé à l'adresse <http://prospectus.usherbrooke.ca/CLUSS>. CLUSS a été publiée à « *BMC Bioinformatics* » en 2007. C'est cette publication que nous présentons dans ce chapitre.

Par la suite, CLUSS a subi plusieurs améliorations. Des améliorations portant sur le calcul de la longueur minimum des motifs collectés pour le calcul de la similarité ont été apportées à SMS, et aussi des améliorations portant sur le calcul de l'arbre phylogénétique basées sur l'analyse en composantes principales ont été apportées à CLUSS. Ainsi, la nouvelle version de CLUSS a été publiée à BIOKDD en 2007. D'autres améliorations plus importantes,

portant sur l'utilisation de l'analyse sémantique latente, ont été apportées à CLUSS ainsi que de nouvelles expériences sur de plus vastes ensembles de données. Ce qui a permis de publier une version ultime de CLUSS dans la conférence internationale BIBE en 2007.

Depuis sa publication, CLUSS a été utilisé et cité dans plusieurs travaux de recherches importants dans le domaine de la biologie cellulaire publiés dans des revues importantes comme « *Genomics* », « *Genome Biology and Evolution* », « *BMC Structural Biology* », et « *BMC Genomics* ». CLUSS est aussi présenté dans des ouvrages références dans le domaine de la bioinformatique et de la biologie, tel que « *Medicinal Protein Engineering* », « *Prediction of Protein Structures, Functions, and Interactions* », et « *Handbook of Research on Systems Biology Applications in Medicine, Vol. 2* ».

Ma contribution inclut, la conception, l'implémentation, et l'exécution de tous les tests impliquant CLUSS et SMS, ainsi que la rédaction des manuscrits et le développement du Serveur Web CLUSS. Le Dr. Shengrui Wang a supervisé le projet, a fourni les ressources, et a participé à la rédaction. Le Dr. Ryszard Brzezinski a aidé à la conception de SMS et à l'amélioration de CLUSS, et a participé à la rédaction des manuscrits. Alain Fleury a analysé certains résultats. Voici les publications dont a fait l'objet CLUSS:

- Abdellali Kelil, Shengrui Wang, Ryszard Brzezinski, Fleury Alain. **CLUSS: Clustering of protein sequences based on a new similarity measure.** *BMC Bioinformatics*, 8:286, 2007.
- Abdellali Kelil, Shengrui Wang, Ryszard Brzezinski. **Clustering of Non-Alignable Protein Sequences.** The 7th International Workshop on Data Mining in Bioinformatics, pp. 69-77. August 12th 2007. San Jose, CA, USA.
- Abdellali Kelil, Shengrui Wang, Ryszard Brzezinski. **A New Alignment-Independent Algorithm for Clustering Protein Sequences.** The 7th IEEE International Conference on BioInformatics and BioEngineering, pp. 27-34. October 14th-17th 2007. Conference Center at Harvard Medical School, Boston, Massachusetts, USA.

CLUSS: Clustering of protein sequences based on a new similarity measure

Abdellali Kelil¹§, Shengrui Wang¹, Ryszard Brzezinski², Alain Fleury²

¹Département d'informatique, Faculté des Sciences, Université de Sherbrooke,
Sherbrooke, (QC) Canada

²Département de Biologie, Faculté des Sciences, Université de Sherbrooke, Sherbrooke,
(QC) Canada

§Corresponding author

Email addresses:

AK: Abdellali.Kelil@USherbrooke.ca

SW: Shengrui.Wang@USherbrooke.ca

RB: Ryszard.Brzezinski@USherbrooke.ca

AF: Alain.Fleury@USherbrooke.ca

Abstract

Background

The rapid burgeoning of available protein data makes the use of clustering within families of proteins increasingly important. The challenge is to identify subfamilies of evolutionarily related sequences. This identification reveals phylogenetic relationships, which provide prior knowledge to help researchers understand biological phenomena. A good evolutionary model is essential to achieve a clustering that reflects the biological reality, and an accurate estimate of protein sequence similarity is crucial to the building of such a model. Most existing algorithms estimate this similarity using techniques that are not necessarily biologically plausible, especially for hard-to-align sequences such as proteins with different domain structures, which cause many difficulties for the alignment-dependent algorithms. In this paper, we propose a novel similarity measure based on matching amino acid subsequences. This measure, named SMS for Substitution Matching Similarity, is especially designed for application to non-aligned protein sequences. It allows us to develop a new alignment-free algorithm, named CLUSS, for clustering protein families. To the best of our knowledge, this is the first alignment-free algorithm for clustering protein sequences. Unlike other clustering algorithms, CLUSS is effective on both alignable and non-alignable protein families. In the rest of the paper, we use the term “*phylogenetic*” in the sense of “*relatedness of biological functions*”.

Results

To show the effectiveness of CLUSS, we performed an extensive clustering on COG database. To demonstrate its ability to deal with hard-to-align sequences, we tested it on the GH2 family. In addition, we carried out experimental comparisons of CLUSS with a variety of mainstream algorithms. These comparisons were made on hard-to-align and easy-to-align protein sequences. The results of these experiments show the superiority of CLUSS in yielding clusters of proteins with similar functional activity.

Conclusion

We have developed an effective method and tool for clustering protein sequences to meet the needs of biologists in terms of phylogenetic analysis and prediction of biological functions. Compared to existing clustering methods, CLUSS more accurately highlights the functional characteristics of the clustered families. It provides biologists with a new and plausible instrument for the analysis of protein sequences, especially those that cause problems for the alignment-dependent algorithms.

Background

With the rapid burgeoning of protein sequence data, the number of proteins for which no experimental data are available greatly exceeds the number of functionally characterized proteins. To predict a function for an uncharacterized protein, it is necessary not only to detect its similarities to proteins of known biochemical properties (i.e., to assign the unknown protein to a family), but also to adequately assess the differences in cases where similar proteins have different functions (i.e., to distinguish among subfamilies). One solution is to cluster each family into distinct subfamilies composed of functionally related proteins. Subfamilies resulting from clustering are easier to analyze experimentally. A subfamily member that attracts particular interest need be compared only with the members of the same subfamily. A biological function can be attributed with high confidence to an uncharacterized protein, if a well-characterized protein within the same cluster is already known. Conversely, a biological function discovered for a newly characterized protein can be extended over all members of the same subfamily. In the rest of the paper, we use the terms subfamily and cluster interchangeably.

The literature reports many algorithms that can be used to build protein clustering databases, such as the widely used algorithm BLAST [1] and its improved versions Gapped-BLAST and PSI-BLAST [2], as well as SYSTERS [3], ProtClust [4] and ProtoMap [5] (see [6] for a review). These algorithms have been designed to deal with large sets of proteins by using various techniques to accelerate examination of the relationships between proteins. However, they are not very sensitive to the subtle differences among similar proteins. Consequently, these algorithms are not effective for clustering protein sequences in closely related families. On the other hand, more specific algorithms have also been developed, for instance, the widely cited algorithms BlastClust [7], which uses score-based single-linkage clustering, TRIBE-MCL [8], based on the Markov cluster approach, and gSPC [9], based on a method that is analogous to the treatment of an inhomogeneous ferromagnet in physics, as well as others such as those introduced by Sjölander [10], Wicker *et al.* [11] and Jothi *et al.* [12]. Almost all of these algorithms are either based on sequence alignment or rely on alignment-dependent algorithms for computing similarity. As several alignments are often possible for a single

family, particularly for families which have not yet been definitively aligned and biologically approved, this will result in different clusterings. Such variable results create ambiguities and make biological interpretation of sequences a difficult task.

In this paper, we propose an efficient algorithm, CLUSS, for clustering protein families based on SMS, which is a new measure we propose for protein similarity. The novelty of CLUSS resides essentially in two features. First, CLUSS is applied directly to non-aligned sequences, thus eliminating the need for sequence pre-alignment. Second, it adopts a new measure of similarity, directly exploiting the substitution matrices generally used to align protein sequences and showing a great sensitivity to the relations among similar and divergent protein sequences. CLUSS can be summarized as follows (a detailed description of the algorithm is given later in the paper):

Given F , a family containing a given number of proteins:

- 1) Build a pairwise similarity matrix for the proteins in F using SMS our new similarity measure.
- 2) Create a phylogenetic tree of the protein family F using a hierarchical clustering approach.
- 3) Assign a co-similarity value to each node of the phylogenetic tree by applying a variant of Ward's formulas [13,14] introduced by Batagelj [15].
- 4) Calculate a critical threshold for identifying subfamily branches, by computing the interclass inertia [16].
- 5) Collect each leaf from its subfamily branch into a distinct subfamily (i.e., cluster).

Implementation

CLUSS was developed with standard C++, and tested in a basic desktop computer under Microsoft Windows XP. The source code, the application server, and all experimental results are available at CLUSS website.

The new similarity measure SMS

Many approaches to measuring the similarity between protein sequences have been developed. Prominent among these are alignment-dependent approaches including the well-known algorithm BLAST [1] and its improved versions Gapped-BLAST and PSI-BLAST [2], which the programs are available at [7], as well as several others such as the one introduced by Varré *et al.* [17] based on movements of segments, and the recent algorithm Scoredist introduced by Sonnhammer *et al.* [18] based on the logarithmic correction of observed divergence. These approaches often suffer from accuracy problems, especially for multi-domain, as well as circular permutation and tandem repeats protein sequences, which were well discussed by Higgins [19]. The similarity measures used in these approaches depend heavily on the quality of the alignment, which in turn depends on the alignability of the protein sequences. In many cases, alignment-free approaches can greatly improve protein comparison, especially for non-alignable protein sequences. These approaches have been reviewed in detail by several authors [20,21,22,23]. Their major drawback, in our opinion, is that they consider only the frequencies and lengths of similar regions within proteins and do not take into account the biological relationships that exist between amino acids. To correct this problem, some authors [22] have suggested the use of the Kimura correction method [24] or other types of corrections, such as that of Felsenstein [25]. However, to obtain an acceptable phylogenetic tree, the approach described in [22] performs an iterative refinement including a profile-profile alignment at each iteration, which significantly increases its complexity. Considering this, we have developed a new approach mainly motivated by biological considerations and known observations related to protein structure and evolution. The goal is to make efficient use of the information contained in amino acid subsequences in the proteins, which leads to a better similarity measurement. The principal idea of this approach is to use a substitution matrix such as BLOSUM62 [26] or

PAM250 [27] to measure the similarity between matched amino acids from the protein sequences being compared.

In this section, we will use the symbol $|.|$ to express the length of a sequence. Let X and Y be two protein sequences belonging to the protein family F . Let x and y be two identical subsequences belonging respectively to X and Y ; we use $\Gamma_{x,y}$ to represent the matched subsequence of x and y . We use l to represent the minimum length that $\Gamma_{x,y}$ should have; i.e., we will be interested only in $\Gamma_{x,y}$ whose length is at least l residues. We define $E_{X,Y}^l$, the key set of matched subsequences $\Gamma_{x,y}$ for the definition of our similarity function, as follows (see Figure 1 for an example):

$$E_{X,Y}^l = \left\{ \Gamma_{x,y} \mid \begin{array}{l} |\Gamma_{x,y}| \geq l, \\ (\forall \Gamma_{x',y'} \in E_{X,Y}^l) \wedge (\Gamma_{x',y'} \neq \Gamma_{x,y}) \Rightarrow (x' \not\subset x) \vee (y' \not\subset y) \end{array} \right\} \quad (1)$$

The symbols x' and y' in the formula are simply used as variables in the same way as x and y . The expression $(. \not\subset .)$ means that the first element is not included in the second one, either in terms of the composition of the subsequences or in terms of their respective positions in X . The matching set $E_{X,Y}^l$ contains all the matched subsequences of maximal length between the sequences X and Y . It will be used to compute the matching score of the sequence pair.

The formula $E_{X,Y}^l$ adequately describes some known properties of polypeptides and proteins. First, protein motifs (i.e., series of defined residues) determine the tendency of the primary structure to adopt a particular secondary structure, a property exploited by several secondary-structure prediction algorithms. Such motifs can be as short as four residues (for instance those found in β -turns), but the propensity to form an α -helix or a β -sheet is usually defined by longer motifs. Second, our proposal to take into account multiple (i.e., ≥ 2) occurrences of a particular motif reflects the fact that sequence duplication is one of the most powerful mechanisms of gene and protein evolution, and if a motif is found twice (or more) in a protein it is more probable that it was acquired by duplication of a segment from a common ancestor than by acquisition from a distant ancestor. The following pseudo-code describes how we can obtain the matching set $E_{X,Y}^l$:

Γ : matched subsequence.

E : matching set.

for $i=1$ to maximum of $|X|$ and $|Y|$

$k = 0, j = i$

while ($k < |X|$ and $j < |Y|$)

if ($X[k] = Y[j]$)

then Add the amino acid $X[k]$ to Γ

else If ($|\Gamma| \geq l$) Add the Γ to E

 Empty Γ

end else

 Increment k , Increment j

end while

if ($|\Gamma| \geq l$) Add Γ to E

 Empty Γ

$k = i, j = 0$

while ($k < |X|$ and $j < |Y|$)

if ($X[k] = Y[j]$)

then Add the amino acid $X[k]$ to Γ

else if ($|\Gamma| \geq l$) Add Γ to E

 Empty Γ

end else

 Increment k , Increment j

end while

if ($|\Gamma| \geq l$) Add Γ to E

end for

This algorithm for the construction of $E'_{X,Y}$ requires a CPU time proportional to $|X| * |Y|$. In practice, however, several optimizations are possible in the implementation, using encoding techniques to speed up this process. In our implementation of SMS, we used a technique that improved considerably the speed of the algorithm; we can summarize it as follows:

By the property that all possible matched subsequences satisfy $|\Gamma_{x,y}| \geq l$, we know that each $\Gamma_{x,y}$ in $E^l_{X,Y}$ is an expansion of a matched subsequence of length l . Thus, we first collect all the matched subsequences of length l , which takes linear time. Secondly, we expand each of the matched subsequences as much as possible on the both left and right sides. Finally, we select all the expanded matched sequences that are maximal according to the inclusion criterion. This technique is very efficient for reducing the execution time in practice. However, due to the variable lengths of the matched sequences, it may not be possible to reduce the worst-case complexity to a linear time. In the Results section, we provide a time comparison between our algorithm and several existing ones.

Let M be a substitution matrix, and Γ a matched subsequence belonging to the matching set $E^l_{X,Y}$. We define a weight $W(\Gamma)$ for the matched subsequence Γ , to quantify its importance compared to all the other subsequences of $E^l_{X,Y}$, as follows:

$$W(\Gamma) = \sum_{i=1}^{|\Gamma|} M[\Gamma[i], \Gamma[i]] \quad (2)$$

Where $\Gamma[i]$ is the i^{th} amino acid of the matched subsequence Γ , and $M[\Gamma[i], \Gamma[i]]$ is the substitution score of this amino acid with itself. Here, in order to make our measure biologically plausible, we use the substitution concept to emphasize the relation that binds one amino acid with itself. The value of $M[\Gamma[i], \Gamma[i]]$ (i.e., within the diagonal of the substitution matrix) estimate the rate at which each possible amino acid in a sequence keep unchanged over time. For the pair of sequences X and Y , we define the matching score $s_{X,Y}$, understood as representing the substitution relation of the conserved regions in both sequences, as follows:

$$s_{XY} = \frac{\sum_{\Gamma \in E^l_{XY}} W(\Gamma)}{\text{MAX}(|X|, |Y|)} \quad (3)$$

To define our similarity measure, we need to scale down $s_{X,Y}$. Let s_{max} be the matching score of the longest sequence belonging to the protein family F with itself, defined as follows:

$$s_{\max} = \{s_{X,X}; |X| = \max \{|Y|; Y \subset F\}\} \quad (4)$$

Finally, the similarity measure between the two sequences X and Y , $S_{X,Y}$ is obtained by dividing the matching score by the value of s_{\max} :

$$S_{X,Y} = \frac{s_{X,Y}}{s_{\max}} \quad (5)$$

Minimum length of matched subsequences “ l ”

In the CLUSS algorithm described in the following section, l , the minimum length of the matched subsequences in SMS, is set to 4. $l=4$ yields good results in all our experiments. Here we will attempt to provide an explanation of this choice.

Our aim is to detect and make use of the significant motifs best conserved during evolution and to minimize the influence of those motifs that occur by chance. This motivates one of the major biological features of our similarity measure, the inclusion of all long conserved subsequences in the matching (i.e., multiple occurrences), since it is well known that the longer the subsequences, the smaller the chance of their being identical by chance, and vice-versa. Here we make use of the theory developed by Karlin *et al.* in [28,29,30] to justify our choice of l . According to theorem 1 in [29] we have:

$$K_{r,N} = \frac{\log n(|Seq_1|, \dots, |Seq_N|) + \log \lambda(1-\lambda) + 0.577}{-\log \lambda} \quad (6)$$

where

$$n(|Seq_1|, \dots, |Seq_N|) = \sum_{1 \leq i_1 \leq \dots \leq i_r \leq N} \prod_{v=1}^r |Seq_{i_v}| \quad (7)$$

and

$$\lambda = \max_{1 \leq v_1 \leq \dots \leq v_r \leq N} \left(\sum_{i=1}^{20} \prod_{j=1}^r p_i^{(v_j)} \right) \quad (8)$$

These formulas calculates $K_{r,N}$, the *expected length of the longest common word present in at least r out of N sequences* [29] (i.e., Seq_1, \dots, Seq_N), where $p_i^{(v)}$ is generally specified as the i^{th} residue frequency of the observed v^{th} sequence.

By fixing $N=r=2$, we calculated $K_{2,2}$, the expected length of the longest matched subsequence present by chance at least 2 times out of each pair of sequences, for several protein datasets including the COG [31] database and the G-proteins [32], GH2 [33] and ROK [34] families. The results, presented in Table 1, show an average expected length very close to $K_{2,2}=4$ residues, with a relatively small standard deviation for each dataset. Thus, for lengths equal to or greater than four amino acids, identical protein subsequences are more likely to be conserved motifs. This choice of length was also made in previous protein sequence comparison contexts, such as Heringa [35] for secondary structure prediction and Leung *et al.* [36] for identifying matches in multiple long sequences.

The CLUSS algorithm

CLUSS is composed of three main stages. The first one consists in building a pairwise similarity matrix based on our new similarity measure SMS; the second, in building a phylogenetic tree according to the similarity matrix, using a hierarchical approach; and the third, in identifying subfamily nodes from which leaves are grouped into subfamilies.

Stage 1: Similarity matrix

Using one of the known substitution score matrices, such as BLOSUM62 [26] or PAM250 [27], and our new similarity measure, we compute S , the $(N \times N)$ pairwise similarity matrix, where N is the number of sequences of the protein family F to be clustered, and $S_{i,j}$ is the similarity measure between the i^{th} and the j^{th} protein sequences belonging to F . The construction of S takes CPU time proportional to $N(N-1)T^2/2$, with T the typical sequence length of the N sequences.

Stage 2: Phylogenetic tree

To build the phylogenetic tree, we have adopted the classical hierarchical approach. Starting from the protein sequences, each of which is considered as the root node of a (sub)tree containing only one node, we iteratively join a pair of root nodes in order to build a bigger subtree. At each iteration, a pair of root nodes is selected if they are the most similar root nodes in terms of a similarity measure derived from the above similarity matrix S . This process ends when there remains only one (sub)tree, which is the phylogenetic tree.

The similarity between two root nodes referred to above is computed in the following way. At the beginning of the iteration, the similarity between any pair of nodes is initialized by the similarity matrix computed in **Stage 1** (i.e., according to SMS). Let L and R be two nearest root nodes at a given iteration step; they are joined together to form a new subtree. Let P be the root node of the new subtree. P thus has two children, L and R . We assign a “length” value $D_{L,P}=D_{R,P}=(1-S_{L,R})/2$ to each of the two branches connecting L and R to P . This value is the estimate of the phylogenetic distance from either node L or R to their parent P in the tree. This distance has no strict mathematical sense; it is merely a measure of the evolutionary distance between the nodes. It is closer to the notion of dissimilarity. The similarity between the new root node P and any other root node K is defined as a weighted average of the similarity between the children of P and the node K :

$$S_{P,K} = \frac{d_L * S_{L,K} + d_R * S_{R,K}}{d_L + d_R} \quad (9)$$

Where $S_{L,K}$ and $S_{R,K}$ are in that order the similarity values between the nodes L and R with the node K before the joining, and d_L and d_R are the numbers of leaves in the subtree rooted at L and R , respectively. Note that in order to keep the notation simple, $S_{P,K}$ is retained here to represent the similarity between any pair of nodes that do not have any descendant relationships in the phylogenetic tree.

Stage 3: Clusters extraction

Given F , a family of N protein sequences, after computing their similarity matrix and phylogenetic tree, CLUSS locates subfamily nodes in this tree using Ward’s [13,14] approach. The main idea is to extract from the phylogenetic tree a number of subtrees, each of which corresponds to a cluster, while optimizing a validation criterion. The criterion is in fact a trade-off between the within-cluster compactness and the between-cluster separation [16]. The different steps are summarized as follows:

Step 1 (Computing the weight of each node): First, each leaf node is considered as a subtree in the phylogenetic tree. We assign to each subtree L (i.e., an individual leaf represents one protein sequence) a weight W_L according to its importance in F . W_L

depends on the number and closeness of the protein sequences that are in fact similar to L , and is thus intended to measure how well F is represented by this particular sequence. For this purpose, we make use of the Thompson [37] method in the definition of W_L :

$$W_L = \sum_{i \in \{branch(L \rightarrow P) - \{P\}\}} \frac{D_{Parent(i),i}}{d_{Parent(i)}} \quad (10)$$

Where P is the root of the phylogenetic tree, L a leaf in this tree, $branch(L \rightarrow P) - \{P\}$ the subset of nodes on the branch from L to P excluding P , $Parent(i)$ the parent of the node i , $D_{Parent(i),i}$ is the length of the branch connecting the node i to its parent (as defined in the previous phase), and $d_{Parent(i)}$ the number of leaves in the subtree rooted at the parent of i . According to this definition, the value of W_L is small if L is very representative and is large if L is not very representative. Iteratively, we assign to each internal subtree P the weight value W_P equal to the sum of the weights of its children $W_L + W_R$.

Step 2 (Computing co-similarity for all internal nodes): Iteratively, until the root of the phylogenetic tree is reached, we assign to the subtree rooted at each non-leaf node P the co-similarity value C_P (between its two child nodes), which is calculated according to the generalized Ward dissimilarity formula [13,14] introduced by Batagelj [15], as follows:

$$C_P = \frac{W_L * W_R}{W_L + W_R} * S_{L,R} \quad (11)$$

Where W_L and W_R are the weights of L and R , respectively, and $S_{L,R}$ is the similarity between L and R computed in Stage 2.

By taking into account information about the neighbourhood around each of the nodes L and R , the concept of co-similarity reflects the cluster compactness of all the sequences (leaf nodes) in the subtree. In fact, its value is inversely proportional to the within-cluster variance. When the subtree becomes larger, the co-similarity tends to become smaller, which means that the sequences within the subtree become less similar and the difference (separation) between sequences in different clusters becomes less significant.

Step 3 (Separating high co-similarity nodes from low co-similarity nodes): The CLUSS algorithm makes use of a systematic method for deciding which subtrees to retain as a trade-off between searching for the highest co-similarity values and searching for the largest possible clusters. We first separate all the subtrees into two groups, one being the group of high co-similarity subtrees and the other the low co-similarity subtrees. This is done by sorting all possible subtrees in increasing order of co-similarity and computing a separation threshold according to the method based on the maximum interclass inertia [11].

Step 4 (Extracting clusters): From the group of high co-similarity subtrees, we extract those that are largest. A high co-similarity subtree is largest if the following two conditions are satisfied: 1) it does not contain any low co-similarity subtree; 2) if it is included in another high co-similarity subtree, the latter contains at least one low co-similarity subtree. Each of these (largest) subtrees corresponds to a cluster and its leaves are then collected to form the corresponding cluster (see Figure 2 for an example).

Results

To illustrate its efficiency, we tested CLUSS extensively on a variety of protein datasets and databases and compared it with several mainstream clustering algorithms. We analyzed the results obtained for the different tests with support from the literature and functional annotations. Most important data and results are provided with this paper as supplementary material, the others are available at CLUSS Website.

The clustering quality measure

To highlight the functional characteristics and classifications of the clustered families, we introduce the *Q-measure*, which quantifies the quality of a clustering by measuring the percentage of correctly clustered protein sequences based on their known functional annotations. This measure can be easily adapted to any protein sequence database. The *Q-measure* is defined as follows:

$$Q\text{-measure} = \frac{\left(\sum_{i=1}^C P_i \right) - U}{N} \cdot 100 \quad (12)$$

Where N is the total number of clustered sequences, C is the number of clusters obtained, P_i is the largest number of sequences in the i^{th} cluster obtained belonging to the same function group according to the known reference classification, and U is the number of unclustered sequences. For the extreme case where each cluster contains one protein with all proteins classified as such, the *Q-measure* is 0, since C becomes equal to N , and each P_i the largest number of obtained sequences in the i^{th} cluster is 1.

COG database

To illustrate the efficiency of CLUSS in grouping protein sequences according to their functional annotation and biological classification, we performed extensive tests on the phylogenetic classification of proteins encoded in complete genomes, commonly named the Clusters of Orthologous Groups of proteins database (COG) [31]. As mentioned in the website for the database, the COG clusters were delineated by comparing protein sequences encoded in complete genomes, representing major phylogenetic lineages. Each

COG consists of individual proteins or groups of paralogs from at least 3 lineages and thus corresponds to an ancient conserved domain. In order to evaluate CLUSS in a statistical manner, we randomly generated 1000 different subsets from the COG database. Each subset contains between 59 and 1840 non-orphan protein sequences (i.e., each selected protein sequence has at least one similar protein sequence from the same functional classification of the COG database).

We tested CLUSS on the 1000 subsets using each of the substitution matrices BLOSUM62 [26] and PAM250 [27] to compute SMS. The average *Q-measure* value of the clusterings obtained is superior to **92%** with a standard deviation of **3.57%** (see Figure 3), while the minimum *Q-measure* value is **80.03%** and the maximum value is **99.35%**. This result shows that CLUSS is indeed very effective in grouping sequences according to the known functional classification of COG.

In the aim of comparing the efficiency of CLUSS to that of alignment-dependent clustering algorithms, we performed tests using CLUSS, BlastClust [7], TRIBE-MCL [8] and gSPC [9] on the COG database. In all performed comparisons, we used the default parameters of compared algorithms. We also used the widely known algorithm to compare protein sequences ClustalW [38] to calculate similarity matrices used by TRIBE-MCL [8] and gSPC [9]. Due to the complexity of alignment, these tests were done on six randomly generated subsets, named SS1 to SS6. The FASTA files of these subsets are provided as supplementary material [see Additional files 1, 2, 3, 4, 5 and 6]. The experimental results of each algorithm are summarized in Figure 4 for the obtained *Q-measures*, and Table 2 for the obtained numbers of clusters and the execution times. The detailed results using CLUSS are available as supplementary material [see Additional files 7, 8, 9, 10, 11 and 12]. BlastClust [7] yielded better results than TRIBE-MCL [8] and gSPC [9]. TRIB-MCL [8] obtained just one cluster for subsets SS1, SS2, SS4 and SS6. For each of the six subsets, the results show clearly that CLUSS obtained the best *Q-measure* compared to the other algorithms tested. Globally, the clusters obtained using our new algorithm CLUSS correspond better to the known characteristics of the biochemical activities and modular structures of the protein sequences. In Table 2 it can be seen that the fastest algorithm is BLAST, closely followed by our algorithm

CLUSS, while TRIBE-MCL and gSPC, which use ClustalW [38] as similarity measures, are much slower than BLAST.

G-proteins family

The G-proteins [32] (guanine nucleotide binding proteins) belong to the larger family of the GTPases. Their signalling mechanism consists in exchanging guanosine diphosphate (GDP) for guanosine triphosphate (GTP) as a general molecular function to regulate cell processes (reviewed extensively in [39]). This family has been the subject of a considerable number of publications by researchers around the world, so we considered it a good reference classification to test the performance of CLUSS. The sequences belonging to this family and the obtained clustering result are provided as supplementary material [see Additional files 13 and 14]. The experimental results obtained using the algorithms CLUSS, BlastClust [7], TRIBE-MCL [8] and gSPC [9], are summarized in Figure 5 for the obtained *Q-measures*, and Table 3 for the corresponding numbers of clusters and the execution time. The clustering results for the G-protein family show clearly that although this family is known to be easy to align, which should have facilitated the clustering task of the alignment-dependent algorithms, CLUSS yields a clustering with *Q-measure* value of **87.09%**, the highest of all the algorithms tested. Thus, the results obtained by CLUSS are much closer to the known classification of the G-protein family than those of the other tested algorithms are. In Table 3, we can make the same observation about the execution times of the different algorithms as in Table 2.

Glycoside Hydrolase family 2 (GH2)

To show the performances of CLUSS with multi-domain protein families which are known to be hard-to-align and have not yet been definitively aligned, experimental tests were performed on 316 proteins belonging to the Glycoside Hydrolases family 2 from the CAZy database (version of January 2006), the FASTA file is provided as supplementary material [see Additional file 15]. The CAZy database describes the families of structurally related catalytic and carbohydrate-binding modules or functional domains of enzymes that degrade, modify, or create glycosidic bonds. Among proteins included in CAZy database, the Glycoside Hydrolases are a widespread group of enzymes that hydrolyse the glycosidic bond between two or more carbohydrates or between a

carbohydrate and a non-carbohydrate moiety. Among Glycoside Hydrolases families, the GH2 family, extensively studied at the biochemical level includes enzymes that perform five distinct hydrolytic reactions. Only complete protein sequences were retained for this study. In our experimentation, the GH2 proteins were subdivided into 28 subfamilies [see Additional file 16], organized in four main branches (see Figure 6). Three branches correspond perfectly to enzymes with known biochemical activities. The first branch (subfamilies 1–7) includes enzymes with “ β -galactosidase” activity from both Prokaryotes and Eukaryotes. The third branch (subfamilies 18 to 22) groups enzymes with “ β -mannosidase” activity, while the fourth branch (subfamilies 23 to 28) includes “ β -glucuronidase”.

The clustering scheme obtained warrants further comment. The “orphan” subfamily 17 includes nineteen sequences labelled as “ β -galactosidase” in databases. While the branch 1 “ β -galactosidase” are composed of five modules, known as the “sugar binding domain”, the “immunoglobulin-like β -sandwich”, the “($\alpha\beta$)₈-barrel”, the “ β -gal small_N domain” and the “ β -gal small_C domain”, the members of subfamily 17 lack the last two of these domains, which makes them more similar to “ β -mannosidase” and “ β -glucuronidase”. These enzymes are distinct from those of branch 1 [40] and their separate localization is justified.

The second branch is the most heterogeneous in terms of enzyme activity. However, most of the subfamilies (9 to 16) group enzymes that are annotated as “putative β -galactosidase” in databases. To the best of our knowledge, none of these proteins, identified through genome sequencing projects, have been characterized by biochemical techniques, so their enzymatic activity remains hypothetical. At the beginning of this branch, subfamily 8 (shown in detail in Figure 7) groups enzymes characterized very recently: “*exo*- β -glucosaminidase” [41,42] and “*endo*- β -mannosidase” [43]. Again, these enzymes share only three modules with the enzymes from branches 1, 3 and 4. The close proximity among “*exo*- β -glucosaminidase” and “*endo*- β -mannosidase” emerging from this work has not been described so far. Furthermore, subfamily 8 includes closely related plant enzymes with “*endo*- β -mannosidase” activity and bacterial enzymes

produced by members of the genus *Xanthomonas*, including several plant pathogens. This could be an example of horizontal genetic transfer between members of these two taxa.

Subfamily 22 (see Figure 8), also found at the beginning of a branch, has been recently analyzed by Côté *et al.* [41] and Fukamizo *et al.* [44], using structure-based sequence alignments and biochemical structure-function studies. It was shown that proteins from this subfamily have a different catalytic doublet and could recognize a new substrate not yet associated with GH2 members.

Globally, the clustering result for the GH2 proteins corresponds well to the known characteristics of their biochemical activities and modular structures. The results obtained with the CLUSS algorithm were highly comparable with those of the more complex analysis performed by Côté *et al.* [41] and Fukamizo *et al.* [44] using clustering based on structure-guided alignments, an approach which necessitates prior knowledge of at least one 3D protein structure.

The 33 (α/β)₈-barrel proteins from the GH2 family

The 33 (α/β)₈-barrel proteins are a group within the GH2 family, studied recently by Côté *et al.* [41] and Fukamizo *et al.* [44]. The periodic character of the catalytic module known as “(α/β)₈-barrel” makes these sequences hard-to-align using classical alignment approaches. The difficulties in aligning these modules are comparable to the problems encountered with the alignments of tandem-repeats, which have been exhaustively discussed [19]. The FASTA file and full clustering results of this subfamily are reported as supplementary material [see Additional files 17 and 18]. This group of 33 protein sequences includes “ β -galactosidase”, “ β -mannosidase”, “ β -glucuronidase” and “exo- β -D-glucosaminidase” enzymatic activities, all of them extensively studied at the biochemical level. These sequences are multi-modular, with various types of modules, which complicate their alignment. Thus, the clustering of such protein sequences using the alignment-dependent algorithms becomes problematic. In our experiments, we tested quite a few known algorithms to align the 33 protein sequences, such as MUSCLE [45], ClustalW [38], MAFFT [46] and T-Coffee [47], etc. The alignment results of all these algorithms are in contradiction with those presented by Côté *et al.* [41], which in turn are

supported by the structure-function studies of Fukamizo *et al.* [44]. This encouraged us to perform a clustering on this subfamily, to compare the behaviour of CLUSS with BlastClust [7], TRIBE-MCL [8] and gSPC [9] to validate the use of CLUSS on the hard-to-align proteins. The experimental results with the different algorithms are summarized in Table 4, which shows the cluster correspondence of each of the sequences by algorithm used. An overview of the results is given below.

CLUSS results

The 33 (α/β)₈-barrel proteins were subdivided by CLUSS into five subfamilies, organized in four main branches (see Table 5 and Figure 9). The first branch corresponds to the first cluster, which includes the enzymes with “ β -galactosidase” activity; the second branch corresponds to the second and the third clusters, which include the enzymes with “ β -mannosidase” activity; the third branch corresponds to the fourth cluster, which includes the enzymes with “*exo*- β -D-glucosaminidase” activity; and the fourth branch corresponds to the fifth cluster, which includes the enzymes with “ β -glucuronidase” activity.

BLAST results

The 33 (α/β)₈-barrel proteins were subdivided into five subfamilies. Almost all the enzymes were clustered in the appropriate clusters, except for seven proteins that were unclustered, among which we find the following well-classified enzymes: “ β -galactosidase” enzymes: GenBank:AAA69907, GenBank:AAA35265 and GenBank:AAA23216; “ β -mannosidase” enzyme: NCBI:ZP_00425692; “*exo*- β -D-glucosaminidase” enzyme: GenBank:AAX62629.

TRIBE-MCL results

The 33 (α/β)₈-barrel proteins were subdivided by TRIBE-MCL into two mixed subfamilies. We find the “ β -mannosidase” enzymes EMBL:CAB63902, GenBank:AAD42775 and EMBL:CAD33708 grouped in the “ β -galactosidase” subfamily. Furthermore, the “*exo*- β -D-glucosaminidase” enzymes and the “ β -glucuronidases” enzymes are grouped in the same subfamily.

gSPC results

The 33 $(\alpha/\beta)_8$ -barrel proteins were subdivided by gSPC into three subfamilies. Almost all the enzymes were grouped in the appropriate subfamily, except for the “ β -galactosidases” and the “ β -glucuronidases” which were grouped in the same subfamily.

Globally, the clustering of the 33 $(\alpha/\beta)_8$ -barrel proteins generated by CLUSS corresponds better to the known characteristics of their biochemical activities and modular structures than do those yielded by the other algorithms tested. The results obtained with our new algorithm were highly comparable with those of the more complex, structure-based analysis performed by Côté *et al.* [41] and Fukamizo *et al.* [44].

Other clustering tests

In our benchmarking (i.e., COG and G-proteins), we compared the execution times of SMS and ClustalW [38]; these results are provided as supplementary materials [see Additional file 19]. We also compared the performance of CLUSS with two other alignment-dependent algorithms, Secator [11] and COCO-CL [12]; the results again show the clear superiority of CLUSS. We also tested CLUSS on a variety of protein families and databases, such as the Clusters of Orthologous Groups for eukaryotic complete genomes database (KOG) [31], Glycoside Hydrolase family 8 (GH8) from the CAZy database [33] and the protein family known as the “Repressor, ORF, Kinases” (ROK) family [34]. Similarly to the results shown in this section, all of these clusterings were highly concordant with their respective reference classifications. The FASTA files and the clustering results for the protein families and databases tested are available at the CLUSS website.

Discussion

The alignment of protein sequences often provides information on conserved and mutated motifs, which is a good approach to measure the similarity between two protein sequences. The problem with this approach is that the result depends primarily on the alignability of the protein sequences, also on the algorithm selected and the parameters set by the user depending on the alignment algorithm used (e.g., gap penalties), which implies several different alignments with each algorithm. Such variations may create difficulties in measuring similarity between sequences and consequently complicate the clustering task. For the case of easy-to-align protein families, such as the G-protein family, almost all alignment algorithms find the same alignment for the conserved regions; however, the alignments of the less conserved regions are significantly different. On the other hand, for the case of hard-to-align protein families, such as the GH2 family, each alignment algorithm tends to diverge to its own, distinct results. Thus, in all cases, there is a significant need to develop efficient and robust alignment-independent approaches to clustering protein sequences.

The SMS developed in this paper makes it possible to measure the similarity between protein sequences based solely on the conserved motifs. The major advantage of SMS compared to the alignment-dependent approaches is that it gives significant results with protein sequences independent of their alignability, which allows SMS to be effective on both easy-to-align and hard-to-align protein families. This property is inherited by CLUSS, our new clustering algorithm, which uses SMS as its similarity measure. CLUSS used jointly with SMS is an effective clustering algorithm when used on protein sets with a restricted number of functions, which is the case of almost all protein families. It more accurately highlights the characteristics of the biochemical activities and modular structures of the clustered protein sequences than do the alignment-dependent algorithms.

So far, our similarity measure has been based on pre-determined substitution matrices. A possible future development is to propose an approach to automatically compute the weights of the conserved motifs instead of relying on pre-calculated substitution scores. There is also a need to speed up the extraction of the conserved motifs and the clustering

of the phylogenetic tree, to scale the algorithm on datasets that are much larger in size with many more biological functions.

Conclusions

Clustering of protein families into phylogenetically correct groups is a difficult problem, especially for those whose alignment is not biologically validated and not definitively performed. In this paper, we have proposed a new similarity measure, SMS, based on which we develop the new clustering algorithm CLUSS. CLUSS is applied directly to non-aligned sequences. Compared to existing clustering methods, CLUSS more accurately reflects the functional characteristics of the clustered families. It provides biologists with a new and plausible instrument for the analysis of protein sequences, especially those that cause problems for the alignment-dependent algorithms.

We believe that CLUSS can become an effective method and tool for clustering protein sequences to meet the needs of biologists in terms of phylogenetic analysis and function prediction. In fact, CLUSS gives an efficient evolutionary representation of the phylogenetic relationships between protein sequences. This algorithm constitutes a significant new tool for the study of protein families, the annotation of newly sequenced genomes and the prediction of protein functions, especially for proteins with multi-domain structures whose alignment is not definitively established. Finally, the tool can also be easily adapted to cluster other types of genomic data. The application server and the implementation are available at CLUSS website.

Availability and requirements

Project name: CLUSS
Project home page: <http://prospectus.usherbrooke.ca/CLUSS>
Operating system(s): MS Windows
Programming language: C++
Other requirements: /
License: Freely offered
Any restrictions to use by non-academics: /

Abbreviations

GH2: Glycoside Hydrolase family 2
GH8: Glycoside Hydrolase family 8
COG: Clusters of Orthologous Groups of proteins
ROK: Repressor, ORF, Kinases

Authors' contributions

AK designed, programmed and executed all experimentations with CLUSS and SMS, created the CLUSS web site, and wrote most of the manuscript. SW supervised the whole project, provided resources and wrote part of the manuscript. RB helped to design SMS and improve CLUSS through links with biological aspects, analyzed the results of clustering methods and wrote part of the manuscript. AF analyzed some results of the clustering method and helped in writing the manuscript. All authors read and approved the final manuscript.

Acknowledgements

The authors thank Samy Metari (MOIVRE laboratory, Université de Sherbrooke) for helpful discussions and Michel Benoit (Département d'informatique, Université de Sherbrooke) for valuable benchmarking contributions.

References

1. Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ: **Basic local alignment search tool.** *J. Mol. Bio.* 1990, **215**:403–410.
2. Altschul SF, Madden TL, Schaffer AA, Zhang J, Zhang Z, Miller W, Lipman DJ: **Gapped BLAST and PSI-BLAST: A new generation of protein database search programs.** *Nucl. Acids Res.* 1997, **25**:3389–3402.
3. Krause A, Stoye J, Vingron M: **The SYSTERS protein sequence cluster set.** *Nucl. Acids Res.* 2000; **28**:270–272.
4. Pipenbacher P, Schliep A, Schneckener S, Schonhuth A, Schomburg D, Schrader R: **ProClust: Improved clustering of protein sequences with an extended graph-based approach.** *Bioinformatics* 2002, **18**:S182–S191.
5. Yona G, Linial N, Linial M: **ProtoMap: Automatic classification of protein sequences and hierarchy of protein families.** *Nucl. Acids Res.* 2000, **28**:49–55.
6. Sjölander K: **Phylogenomic inference of protein molecular function: Advances and challenges.** *Bioinformatics* 2004, **20**:170–179.
7. **Basic Local Alignment Search Tool :** [<http://www.ncbi.nlm.nih.gov/BLAST>].
8. Enright AJ, Van Dongen S, Ouzounis CA: **An efficient algorithm for large-scale detection of protein families.** *Nucl. Acids Res.* 2002, **30**:1575–1584.
9. Tetko IV, Facius A, Ruepp A, Mewes HW: **Super Paramagnetic Clustering of Protein Sequences.** *BMC Bioinformatics* 2005, **6**:82.
10. Sjölander K: **Phylogenetic inference in protein superfamilies: Analysis of SH2 domains.** *Intell. Syst. Mol. Biol.* 1998, **6**:165–174.
11. Wicker N, Perrin GR, Thierry JC, Poch O: **Secator: A Program for Inferring Protein Subfamilies from Phylogenetic Trees.** *Mol. Biol. Evol.* 2001, **18**:1435–1441.
12. Jothi R, Zotenko E, Tasneem A, Przytycka TM: **COCO-CL: Hierarchical clustering of homology relations based on evolutionary correlations.** *Bioinformatics* 2006, **22**:779–788.
13. Ward JH: **Hierarchical Grouping to Optimize an Objective Function.** *J. Am. Stat. Assoc.* 1963, **58**:236–244.

14. Ward JH, Hook ME: **Application of a Hierarchical Grouping Procedure to a Problem of Grouping Profiles.** *Educ. Psychol. Meas.* 1963, **23**:69-82.
15. Batagelj V: **Generalized Ward and related clustering problems.** In *Classification and Related Methods of Data Analysis*, edited by H. H. Bock, Amsterdam: Elsevier 1988, 67–74.
16. Duda RO, Hart PE, Stork DG: *Pattern Classification*, second edition, John Wiley and Sons, 2001.
17. Varré JS, Delahaye JP, Rivals E: **The transformation distance: A dissimilarity measure based on movements of segments.** *Bioinformatics* 1999, **15**:194–202.
18. Sonnhammer ELL, Hollich V: **Scoredist: A simple and robust sequence distance estimator.** *BMC Bioinformatics* 2005, **6**:108.
19. Higgins D: **Multiple alignment.** In *The Phylogenetic Handbook*. Edited by Salemi M, Vandamme A.M. Cambridge University Press 2004, **45**:45-71.
20. Reinert G, Schbath S, Waterman MS: **Probabilistic and statistical properties of words: An overview.** *J. Comp. Biol.* 2000, **7**:1-46.
21. Rocha J, Rossello F, Segura J: **The Universal Similarity Metric does not detect domain similarity.** *Q-bio.QM* 2006, **1**:0603007.
22. Edgar RC: **Local homology recognition and distance measures in linear time using compressed amino acid alphabets.** *Nucl. Acids Res.* 2004, **32**:380-385.
23. Vinga S, Almeida J: **Alignment-free sequence comparison – A review.** *Bioinformatics* 2003, **19**:513-523.
24. Kimura M: **Evolutionary rate at the molecular level.** *Nature*, 1968 **217**:624–626.
25. Felsenstein J: **An alternating least squares approach to inferring phylogenies from pairwise distances.** *Syst. Biol.* 1997, **46**:101.
26. Henikoff S, Henikoff JG: **Amino acid substitution matrices from protein blocks.** *Proceedings of the National Academy of Sciences of the United States of America* 1992, **89**:10915-10919.
27. Dayhoff MO, Schwartz RM, Orcutt BC: **A model of evolutionary change in proteins.** *Atlas of Protein Sequence and Structure vol. 5* 1978, *suppl.* **3**:345-352.
28. Karlin S, Ost F: **Maximal length of common words among random letter sequences.** *The Annals of Probability* 1988, **16**:535-563.

29. Karlin S, Ghandour G: **Comparative statistics for DNA and protein sequences: Single sequence analysis.** *Proc. Natl. Acad. Sci. USA* 1985, **82**:5800-5804.
30. Karlin S, Ghandour G: **Comparative statistics for DNA and protein sequences: Multiple sequence analysis.** *Proc. Natl. Acad. Sci. USA* 1985, **82**:6186-6190.
31. Phylogenetic classification of proteins encoded in complete genomes: [<http://www.ncbi.nlm.nih.gov/COG/>].
32. GPCRPDB: Information system for GPCR interacting proteins: [<http://www.gpcr.org>].
33. The carbohydrate-active enzymes (CAZy) database: [<http://www.cazy.org/>].
34. Titgemeyer F, Reizer J, Reizer A, Saier Jr MH: **Evolutionary relationships between sugar kinases and transcriptional repressors in bacteria.** *Microbiology* 1994, **140**:2349-2354.
35. Heringa J: **Computational methods for protein secondary structure prediction using multiple sequence alignments.** *Current Protein & Peptide Science* 2000, **1**:273-301.
36. Leung MY, Blaisdell BE, Burge C, Karlin S: **An Efficient Algorithm for Identifying Matches with Errors in Multiple Long Molecular Sequences.** *J. Mol. Biol.* 1991, **221**:1367-1378.
37. Thompson JD, Higgins DG, Gibson TJ: **Improved sensitivity of profile searches through the use of sequence weights and gap excision.** *Comput. Appl. Biosci.* 1994, **10**:19-29.
38. Thompson JD, Higgins DG, Gibson TJ: **CLUSTAL W: Improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice.** *Nucl. Acids Res.* 1994, **22**:4673-4680.
39. Lodish H, Berk A, Matsudaira P, Kaiser CA, Krieger M, Scott MP, Zipursky L, Darnell J: *Molecular Cell Biology*, 5th ed. New York and Basingstoke: W.H. Freeman and Co., 2004.
40. Fanning S, Leahy M, Sheehan D: **Nucleotide and deduced amino acid sequences of Rhizobium meliloti 102F34 lacZ gene: Comparison with prokaryotic beta-galactosidases and human beta-glucuronidase.** *Gene* 1994, **141**:91-96.

41. Côté N, Fleury A, Dumont-Blanchette E, Fukamizo T, Mitsutomi M, Brzezinski R: **Two exo- β -D-glucosaminidases/exochitosanases from actinomycetes define a new subfamily within family 2 of glycoside hydrolases.** *Biochem. J.* 2006, **394**:675–686.
42. Ike M, Isami K, Tanabe Y, Nogawa M, Ogasawara W, Okada H, Morikawa Y: **Cloning and heterologous expression of the exo- β -D-glucosaminidase-encoding gene (gls93) from a filamentous fungus, *Trichoderma reesei* PC-3-7.** *Appl. Microbiol. Biotechnol.* 2006, **72**: 687–695.
43. Ishimizu T, Sasaki A, Okutani S, Maeda M, Yamagishi M, Hase S: **Endo-beta-mannosidase, a plant enzyme acting on N-glycan: Purification, molecular cloning and characterization.** *J. Biol. Chem.* 2004, **279**:3855-3862.
44. Fukamizo T, Fleury A, Côté N, Mitsutomi M, Brzezinski R: **Exo- β -D-glucosaminidase from *Amycolatopsis orientalis*: Catalytic residues, sugar recognition specificity, kinetics, and synergism.** *Glycobiology* 2006, **16**:1064-1072.
45. Edgar RC: **MUSCLE: A multiple sequence alignment method with reduced time and space complexity.** *BMC Bioinformatics* 2004, **5**:113.
46. Katoh K, Misawa K, Kuma K, Miyata T: **MAFFT: A novel method for rapid multiple sequence alignment based on fast Fourier transform.** *Nucl. Acids Res.* 2002, **30**:3059-3066.
47. Notredame C, Higgins D, Heringa J: **T-Coffee: A novel method for multiple sequence alignments.** *Journal of Molecular Biology* 2000, **302**:205-217.

Figures

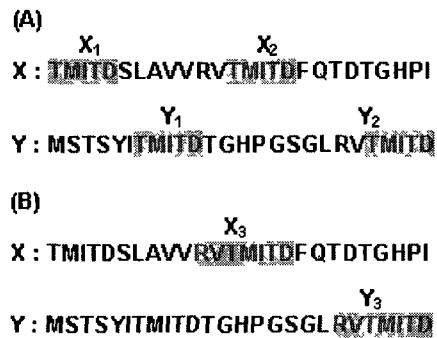


Figure 1 - Matching sequences

Let X and Y be two protein sequences, as illustrated in figures A and B.

- (A). For the pair of subsequences x_1 and y_1 we add a matching subsequence Γ_1 , identical to x_1 and y_1 , to the matching set $E^4_{X,Y}$. Similarly, we add Γ_2 identical to x_1 and y_2 , and Γ_3 identical to x_2 and y_1 . However, since $x_2 \subset x_3$ and $y_2 \subset y_3$, (x_3 and y_3 are shown in figure B) we do not add Γ_4 , identical to x_2 and y_2 , to $E^4_{X,Y}$.
- (B). For the pair of subsequences x_3 and y_3 we add a matching subsequence Γ_5 , identical to x_3 and y_3 , to the set $E^4_{X,Y}$, even if x_3 overlaps with x_2 .

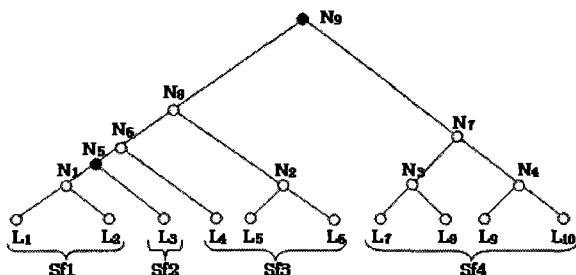


Figure 2 - Merging leaves

Let us take a rooted phylogenetic tree with $L_1, L_2 \dots L_{10}$ as leaves, and $N_1, N_2 \dots N_9$ as internal nodes, where N_5 and N_9 are identified as low co-similarity nodes (black nodes). Leaves are merged until a black node is reached, except for L_3, L_4, L_5 and L_6 , which need special consideration. All leaves connected between N_5 and N_9 are merged into a distinct subfamily. L_3 is connected directly to N_5 so it constitutes a distinct subfamily. We thus obtain the subfamilies $Sf1, Sf2, Sf3$ and $Sf4$, while $Sf2$ contains the orphan sequence represented by leaf $L3$.

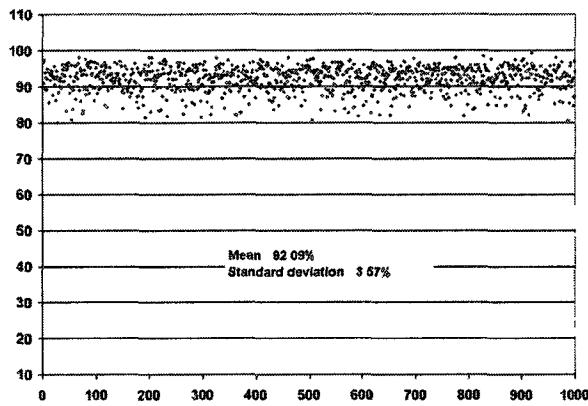


Figure 3 – Clustering results for the 1000 subsets from COG

Each red point is a quality measure (*Q-measure*) of a clustering result of one of the 1000 randomly generated subsets from the COG database. As shown, the obtained results are in good concordance with the functional reference characterization of COG. The average of the quality measure of the 1000 clusterings is equal to **92.09%** with a standard deviation equal to **3.57%**. More than **75%** of the 1000 clusterings obtained a quality measure superior to **90%**, and more than **21%** of the clusterings obtained a quality measure superior to **95%**. The minimum value of the quality measure is **80.03%** and the maximum value is **99.35%**.

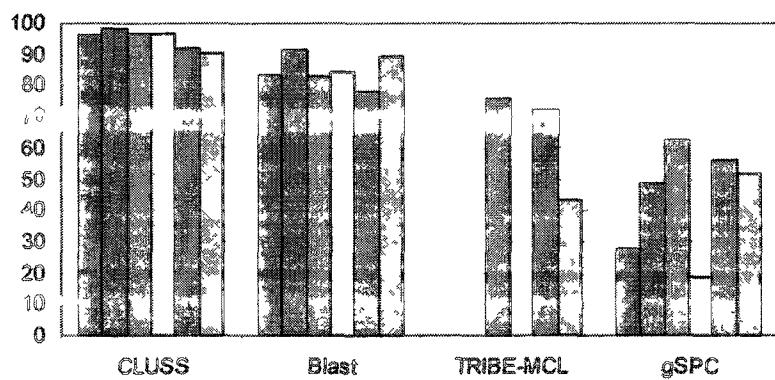


Figure 4 – Clustering results for the six subsets from COG

For each algorithm (reading horizontally), the bars represent the *Q-measure* of the clustering results obtained on six randomly generated subsets: SS1, red; SS2, blue; SS3, green; SS4, yellow; SS5, gray; SS6, amber.

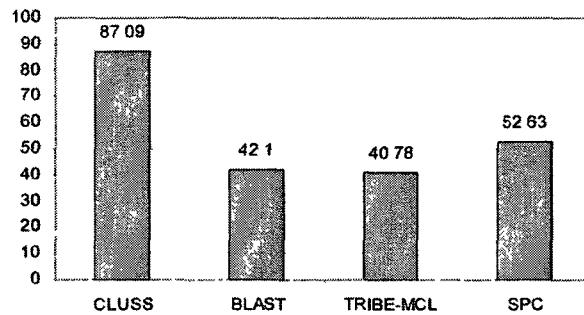


Figure 5 – Clustering results for the G-proteins

For each algorithm (reading horizontally), the bars represent the *Q-measure* of the clustering results obtained on the members of the G-protein family. CLUSS obtained the highest quality measure of all the clustering results for this family, which shows that the CLUSS grouping is nearest to the functional reference classification for the G-protein family.

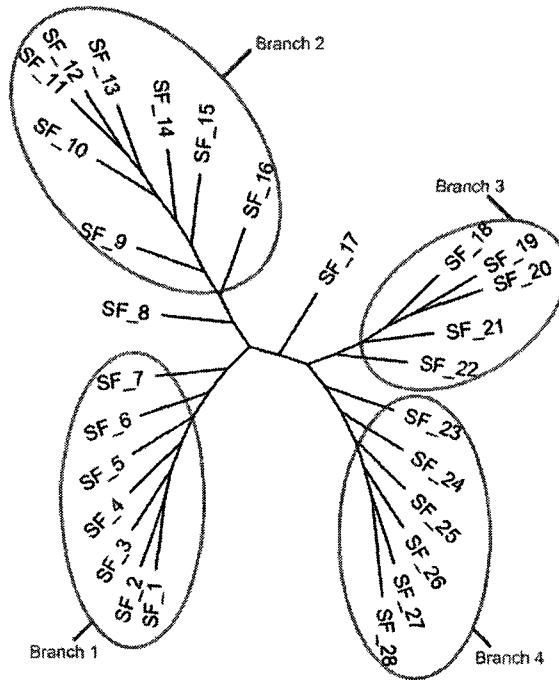


Figure 6 - CLUSS phylogenetic analysis of GH2 family

The 316 enzymes of the GH2 family are clustered by CLUSS into 28 subfamilies (**SF_1** to **SF_28**), in a phylogenetic tree composed of four main branches. Branches 1, 3 and 4

correspond to “ β -galactosidase”, “ β -mannosidase” and “ β -glucuronidase” activities, respectively. Most enzymes in branch 2 are labelled as “putative β -galactosidases” in databases. The “orphan” subfamily SF_17 includes nineteen sequences labelled as “ β -galactosidases” in databases. Subfamily SF_8 contains “*exo-glucosaminidase*” and “*endo-mannosidase*” activities.

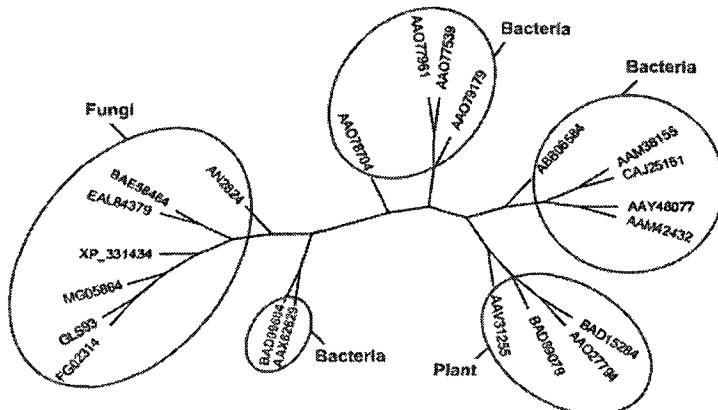


Figure 7 - Subfamily SF_8 phylogenetic analysis

The phylogenetic tree of the 22 enzymes of subfamily SF_8 is grouped into (DDBJ:BAD89079, DDBJ:BAD15284) “endo- β -mannosidasee” and (GenBank:AAX62629, DDBJ:BAD99604) “exo- β -D-glucosaminidase” activities. Subfamily SF_8 also includes closely related plant enzymes and bacterial enzymes produced by members of the genus Xanthomonas, including several plant pathogens.

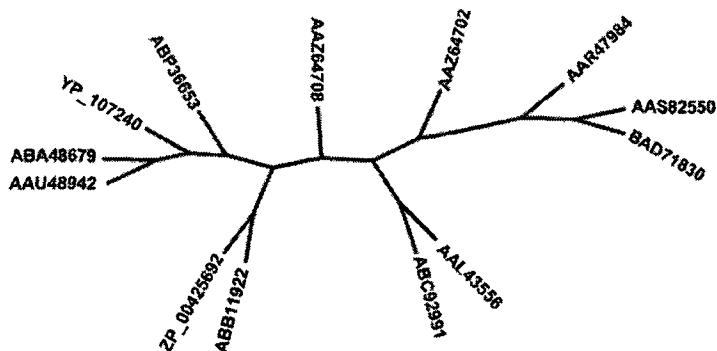


Figure 8 - Subfamily SF_22 phylogenetic analysis

CLUSS has clustered in the same subfamily the enzymes GenBank:AAU48942 “*Burkholderia mallei*”, NCBI:YP_107240 “human”, GenBank:AAZ64708 “*Ralstonia*

eutropha", GenBank:AAL43556 "*Agrobacterium tumefaciens*", GenBank:ABB11922 "*Burkholderia*" and NCBI:ZP_00425692 "*Burkholderia vietnamensis*", which were recently analyzed by Côté *et al.* [41] and Fukamizo *et al.* [44] and characterized by their ability to recognize a substrate not yet associated with GH2 members.

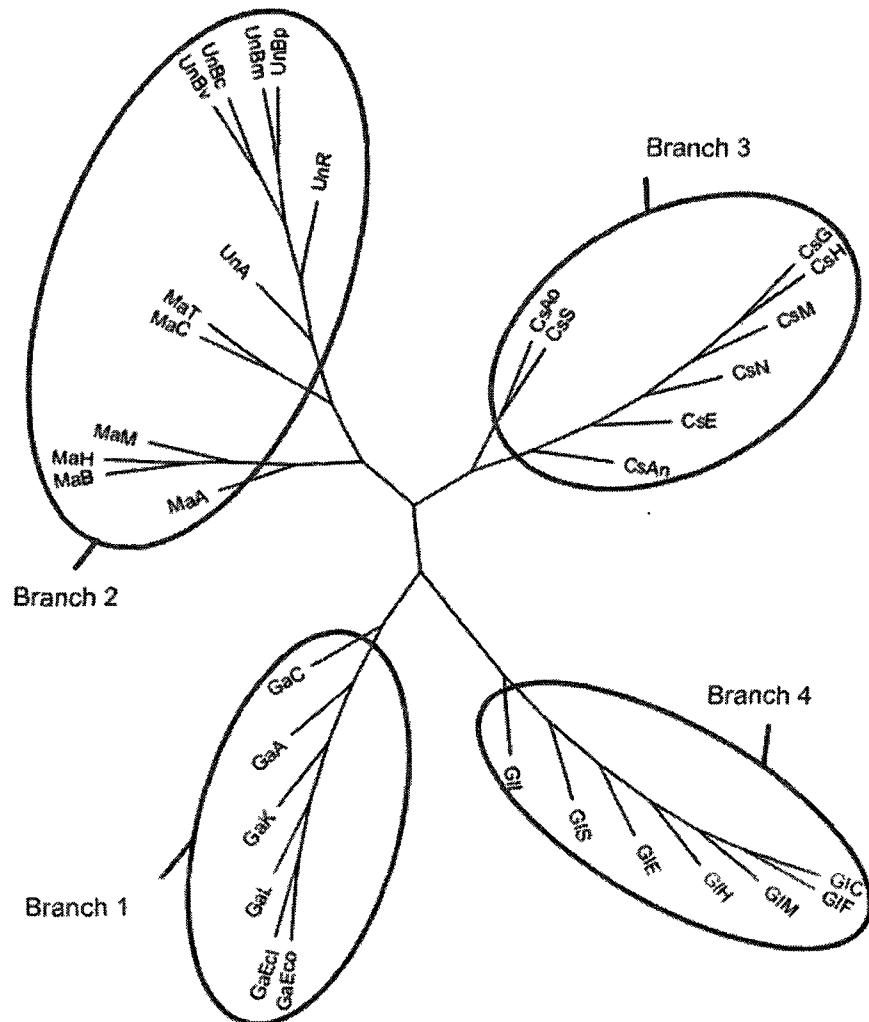


Figure 9 – 33 (α/β)₈-barrel group phylogenetic analysis

The database entries of the 33 (α/β)₈-barrel group are indicated at :
<http://prospectus.usherbrooke.ca/CLUSS/Results/Data/33%20barrel/Names.txt>

Tables

Table 1 - Expected length of longest common subsequence computed for several protein datasets

The columns represent respectively, DS: the tested protein datasets, NS: number of tested protein sequences, AEL: average of the expected length of the longest common subsequence and finally SD: the standard deviation.

DS	NS	AEL	SD
COG database	144298	3.934	0.363
KOG database	60748	4.062	0.458
G-proteins family	381	3.718	0.200
GH2 family	316	4.355	0.232
ROK family	730	4.074	0.324

Table 2 – Clustering results of the six subsets from the COG database

Number of clusters obtained by clustering the protein sequences of the six randomly generated subsets from the COG database (rows) with each of the clustering algorithms tested (columns). To each execution time of TRIBE-MCL [8] and gSPC [9], we added the corresponding execution time of ClustalW [38] used to compute the similarity matrix. Time is indicated in seconds.

Protein subsets	CLUSS		BLAST		MCL+ClustalW		SPC+ClustalW	
	Nbr	Time	Nbr	Time	Nbr	Time	Nbr	Time
SS1 (469 proteins)	30	106	114	14	1	495	9	499
SS2 (743 proteins)	15	234	102	58	1	1272	33	1275
SS3 (455 proteins)	30	114	132	18	8	586	27	588
SS4 (409 proteins)	19	82	125	11	1	452	4	454
SS5 (564 proteins)	35	103	172	15	6	538	30	540
SS6 (6444 proteins)	225	4272	732	583	1	95895	77	98880

Table 3 – Clustering results of the G-protein family

Number of clusters obtained by clustering the protein sequences of the G-protein family (rows) with each of the tested clustering algorithms (columns). Time is indicated in seconds. (The same remark applies as in Table 2 concerning TRIBE-MCL [8] and gSPC [9]).

Protein subsets	CLUSS		BLAST		MCL		SPC	
	Nbr	Time	Nbr	Time	Nbr	Time	Nbr	Time
G-proteins (381 proteins)	51	85	24	14	2	419	20	432

Table 4 – Clustering results of the 33 (α/β)8-barrel protein sequences

The clustering correspondence of each of the 33 (α/β)8-barrel protein sequences (rows), obtained by Côté *et al.* [41] and Fukamizo *et al.* [44] and each of the clustering algorithms tested (columns). Each number in the table represents the corresponding cluster of the row's protein sequence obtained with the column's method. They are bold when they correspond to Côté *et al.* [41] and Fukamizo *et al.* [44] classification. The symbol “/” means that the row's protein sequence is unclustered.

Proteins	Côté/Fuka.	CLUSS	BLAST	MCL	SPC
GaEco	1	1	1	1	1
GaA	1	1	/	1	1
GaK	1	1	/	1	1
GaC	1	1	/	1	1
GaEcl	1	1	1	1	1
GaL	1	1	1	1	1
MaA	2	2	2	1	2
MaB	2	2	2	2	2
MaH	2	2	2	2	2
MaM	2	2	2	2	2
MaC	2	3	2	1	2
MaT	2	3	2	1	2
UnA	3	3	3	2	2
UnBv	3	3	3	2	2
UnBc	3	3	/	2	2
UnBm	3	3	3	2	2
UnBp	3	3	3	2	2
UnR	3	3	3	2	2
CsAo	4	4	/	1	3
CsS	4	4	4	1	3
CsG	4	4	4	1	3
CsM	4	4	4	1	3
CsN	4	4	/	1	3
CsAn	4	4	/	1	3

CsH	4	4	4	1	3
CsE	4	4	4	1	3
GIC	5	5	5	1	1
GIE	5	5	5	1	1
GIH	5	5	5	1	1
GIL	5	5	5	1	1
GIM	5	5	5	1	1
GIF	5	5	5	1	1
GIS	5	5	5	1	1

Chapitre 2

CLUSTERING DES GRANDES FAMILLES DE PROTÉINES

Dans le chapitre précédent, nous avons présenté CLUSS un algorithme de clustering des familles de protéines. CLUSS est basée sur SMS conçue spécifiquement pour mesurer la similarité entre les séquences de protéines qu'elles soient alignables ou non alignables, une propriété qui joue un rôle clé dans CLUSS. Nous avons montré que, par rapport aux algorithmes de clustering basés sur l'alignement, CLUSS capture plus efficacement les caractéristiques des activités biochimiques et des structures modulaires des séquences de protéines, surtout lorsqu'il s'agit de séquences de protéines non alignables. Toutefois, CLUSS et SMS ont tendance à être moins efficaces lorsqu'ils sont appliqués sur de grands ensembles de protéines qui incluent de grands nombres d'activités biochimiques. Par ailleurs, dans le calcul de SMS, malgré l'utilisation de techniques d'optimisation pour accélérer la mise en correspondance des motifs dans le calcul de la similarité, il ne nous a pas été possible de réduire sa complexité maximale qui est quadratique par rapport aux longueurs des séquences. En plus de cela, dans CLUSS, le coût engendré par la méthode hiérarchique de construction de l'arbre phylogénique ainsi que le calcul des poids des nœuds dans cet arbre, a une complexité quadratique par rapport au nombre de séquences. Tous ces facteurs empêchent CLUSS et SMS d'être efficaces sur des grands ensembles de séquences de protéines. Afin de surmonter ces problèmes majeurs, nous présentons dans ce chapitre CLUSS2 une version améliorée de CLUSS dont la complexité est linéaire par rapport au nombre de protéines, et aussi tSMS une version améliorée de SMS, dont la complexité est linéaire par rapport aux longueurs des séquences. En plus de cela, contrairement à CLUSS et SMS, CLUSS2 utilisé conjointement avec tSMS sont capable de traiter de grands ensembles de séquences de protéines avec de grands nombres d'activités biochimiques.

CLUSS2 diffère sensiblement de CLUSS sur deux points essentiels. Tout d'abord, CLUSS2 est basé sur une nouvelle mesure tSMS qui est une amélioration significative de SMS pour le calcul de similarité entre les séquences de protéines. La mesure tSMS est basée sur un nouvel algorithme d'appariement qui, au contraire de SMS, autorise les mésappariements par la mise en correspondance des motifs identiques et similaires, plutôt que d'imposer seulement les appariements identiques comme dans SMS, et c'est la principale raison de l'efficacité de tSMS. De plus, dans tSMS la longueur minimum des motifs significatifs collectés est calculée spécifiquement pour chaque paire de séquences de protéines appariées, au lieu d'utiliser la même longueur minimum pour tous les appariements comme dans SMS. Ceci permet à tSMS d'être plus efficace dans la collecte des motifs importants et dans l'exclusion des motifs constituants du bruit. Par ailleurs, le nouvel algorithme d'appariement utilisé dans tSMS est basé sur l'utilisation de la structure de données connue sous le nom de « *Suffix Tray* », ce qui confère à tSMS une complexité maximal linéaire par rapport aux longueurs des séquences au lieu de la complexité maximal quadratique de SMS. En plus de cela, CLUSS2 utilise la technique de décomposition en valeurs singulières (SVD) de la matrice de similarités obtenue à partir de tSMS, afin de créer un espace vectoriel où chaque séquence est représentée par un vecteur. Cette transformation permet l'application des opérations vectorielles au cours du processus de clustering. Ceci permet d'obtenir un représentant (centroid) pour chaque groupe, et aussi la possibilité de réduire encore le temps d'exécution en utilisant des représentations approximatives plus efficaces.

Tout ceci rend CLUSS2 beaucoup plus rapide et plus efficace que CLUSS, en particulier pour les ensembles de données de protéines de grandes tailles et spécialement ceux qui contiennent beaucoup d'activités biologiques. L'algorithme utilisé dans CLUSS2 pour la représentation hiérarchique des similarités entre les protéines nous a permis de développer un nouvel algorithme pour l'identification des relations de parentés entre les individus en utilisant les profils d'ADN, que nous avons d'ailleurs publié dans la revue « *Journal of Forensic Sciences* » en 2010.

CLUSS2 a été présenté dans un article publié dans la revue « *International Journal of Computational Biology and Drug Design* » en 2008. En plus des nouveaux fondements et concepts derrières CLUSS2 et tSMS, ce papier présente aussi une panoplie de nouvelles expérimentations effectuées avec CLUSS2 et CLUSS, ainsi que d'autres algorithmes courants. Ces expérimentations ont montré que, contrairement à CLUSS, et aux autres algorithmes, les performances de CLUSS2 restent relativement stable avec l'augmentation du nombre de fonctions biologiques dans les ensembles de séquences de protéines. À ce jour, CLUSS2 a été utilisé par plusieurs chercheurs. On peut citer par exemple les travaux du Dr. Dmitri A. Petrov, affilié à « *Department of Biology, University of Stanford* » publiés en 2009 dans la revue « *Genome Biology and Evolution* », pour étudier le mode de sélection des gènes impliqués dans des maladies humaines. On peut citer aussi les travaux du Dr. Robert P. Hirt, affilié à « *Institute for Cell and Molecular Biosciences Newcastle* » publiés en 2010 dans la revue « *BMC Genomics* », sur l'étude de la diversité fonctionnelle et l'organisation structurelle et transcriptomique de la famille de gènes « *Trichomonas vaginalis* vast BspA-like ». CLUSS2 ainsi que tSMS ont été utilisés dans une étude publiée dans « *International Conference on Future BioMedical Information Engineering 2009* » par un groupe de recherche affilié à « *University of Electronic Science and Technology of China* » comme plateforme pour le développement d'une méthode d'analyse de clustering à large échelle des séquences de protéines. Malgré que CLUSS semble être plus connue que CLUSS2 (probablement due à la meilleure visibilité du journal *BMC Bioinformatics*), dans l'historique de notre serveur web nous enregistrons beaucoup plus de requêtes avec CLUSS2 que celles avec CLUSS. Voici la publication dont a fait l'objet CLUSS2:

- Abdellali Kelil, Shengrui Wang, Ryszard Brzezinski. **CLUSS2: An Alignment-Independent Algorithm for Clustering Protein Families with Multiple Biological Functions.** International Journal of Computational Biology and Drug Design, 1(2); 122-140; 2008.

CLUSS2: An Alignment-Independent Algorithm for Clustering Protein Families with Multiple Biological Functions

Abdellali Kelil¹, Shengrui Wang¹, Ryszard Brzezinski²,

¹ProspectUS Laboratory, Department of Computer Sciences, Faculty of Sciences, University of Sherbrooke, Sherbrooke, QC J1H 3Z3 Canada

²Microbiology and Biotechnology Laboratory, Faculty of Sciences, Department of Biology, University of Sherbrooke, Sherbrooke, QC J1H 3Z3 Canada

Abstract: In a previous paper, we developed an alignment-independent algorithm, CLUSS, for clustering protein sequences. We have shown that CLUSS can deal with both alignable and non-alignable protein sequences. Compared to existing clustering methods, CLUSS more accurately captures the functional characteristics of the clustered protein sequences and has several advantages over alignment-dependent algorithms. However, CLUSS tends to be ineffective on protein datasets that include a large number of biochemical activities. In order to overcome this major difficulty, this paper presents an improved algorithm named CLUSS2, whose efficiency scales well with the increase of the number of biochemical activities. CLUSS2 differs significantly from CLUSS in many ways, including the protein sequence representation, the approach for extracting conserved motifs and the time efficiency. Our experiments show that, compared to existing clustering algorithms, CLUSS2 more accurately highlights the functional characteristics of the clustered families, especially for those with a large number of biochemical activities. In terms of runtime, it is also much more efficient than CLUSS and several other existing algorithms.

Keywords: clustering; similarity measure; biological function; non-alignable.

Biographical notes: **Abdellali Kelil** is currently a Ph.D. candidate at University of Sherbrooke, and a member of the ProspectUS data mining and bioinformatics laboratory at the same university. His current research interests include protein analysis and classification and functional prediction.

Shengrui Wang received his Ph.D. from the Institut National Polytechnique in Grenoble, France. He is director of ProspectUS laboratory at University of Sherbrooke. His research interests include pattern recognition, data mining, artificial intelligence and information retrieval.

Ryszard Brzezinski received his Ph.D. from the University of Warsaw, Poland. He is director of a laboratory of molecular biotechnology, environmental microbiology and bioinformatics at the University of Sherbrooke.

1 Introduction

To predict the biochemical activity of a newly sequenced or not yet characterized protein sequence, it is necessary to compare its biochemical properties to those of functionally well-characterized protein sequences, to assign this protein to one of the protein families. However, this is not sufficient to attribute a biochemical activity to the protein with a high degree of confidence, since a single family can include a number of biochemical activities. A possible solution for assessing the differences in cases where protein sequences from the same family have different activities is clustering. The literature reports many clustering approaches to the task of grouping protein families into subfamilies of protein sequences that are functionally more closely related. However, clustering protein sequences remains a difficult challenge, especially for sequences whose alignment is not biologically validated (i.e., hard-to-align or totally non-alignable sequences), such as tandem-repeat, multi-domain and circular-permutation proteins, for which alignment-dependent algorithms do not yield biologically plausible clustering results. The main reason is that these algorithms use an alignment process based on matching motifs in corresponding positions, whereas non-alignable proteins often have similar or conserved domains in non-corresponding positions. A more detailed discussion on why these proteins are difficult to align and hard to cluster is given in (Keil, et al., 2007a). To the best of our knowledge, the only alignment-independent clustering algorithm which is effective on both alignable and non-alignable protein sequences is the CLUSS algorithm which we proposed recently in (Keil, et al., 2007b).

CLUSS is based on a measure named SMS which we designed specifically to compute the similarity between two protein sequences. The SMS measure depends on identical matched motifs and is effective for both alignable and non-alignable protein sequences, a property that plays a key role in CLUSS. Compared to alignment-dependent algorithms, CLUSS highlights the characteristics of the biochemical activities and modular structures of the clustered protein sequences. However, it has a tendency to be less effective when applied to large protein datasets with many biochemical activities. CLUSS also suffers from another problem. Despite the use of optimization techniques to speed up the matching of motifs, it is still not possible to reduce the worst-case complexity to a linear time in the SMS computation, which remains slow, especially for large protein datasets. All these factors prevent CLUSS from being effective on large protein datasets.

In this paper we propose a new algorithm for clustering protein sequences, which we have named CLUSS2. CLUSS2 is similar to CLUSS in that both are hierarchical clustering algorithms and both aim primarily to cluster hard-to-align sequences. However, CLUSS2 differs significantly from CLUSS in two main respects. First, CLUSS2 is based on a new measure tSMS that extends SMS for computing similarity between protein sequences. The tSMS measure allows the matching of similar motifs, rather than imposing identical matches as in SMS. tSMS is computed based on a new algorithm for extracting matched motifs, which is the main reason for its increased efficiency. The second major difference from CLUSS is that CLUSS2 applies Singular Value Decomposition (SVD) techniques to the similarity matrix obtained from tSMS, to create a representation of each protein sequence in a vector space. This transformation allows the application of vector operations during the clustering process. One important advantage is that this yields a representative (centroid) for each

cluster; another is the possibility of further reducing the runtime by using approximate representations. CLUSS2 is much faster and more effective than CLUSS, especially for large protein datasets with a large number of biological activities.

To show the effectiveness of CLUSS2, we performed extensive clustering experiments on the COG and KOG databases, which contain phylogenetic classifications of proteins encoded in complete genomes (Tatusov, et al., 2003), and also on reference sequence proteins encoded by complete prokaryotic and chloroplast plasmids and genomes, known as the Protein Clusters (PC) database, available at <ftp://ftp.ncbi.nih.gov/genomes/Bacteria/CLUSTERS/>. To demonstrate its ability to deal with hard-to-align sequences, we tested it on the $(\alpha/\beta)_8$ barrel proteins group, belonging to the Glycoside Hydrolases (GH) family (Coutinho, et al., 1999). In addition, we carried out experimental comparisons with a variety of mainstream algorithms including the BlastClust program (Dondoshansky and Wolf, unpublished), which belongs to the standalone BLAST package used to cluster either protein or nucleotide sequences, available from the NCBI website <ftp://ftp.ncbi.nih.gov/blast/>, and the well-known algorithms TRIBE-MCL (Enright, et al., 2002) and gSPC (Tetko, et al., 2005). These comparisons were made on hard-to-align and easy-to-align protein sequences. The results of these experiments show advantages of CLUSS2 in yielding more significant clusters of proteins with similar functional activities, especially for large protein datasets with a variety of biochemical activities.

2 The New Similarity Measure tSMS

The measure SMS, used in CLUSS to measure the similarity of a pair of protein sequences X and Y , was defined based on a key set of strictly matched subsequences (i.e., identical amino acids) of maximal length between the sequences X and Y , denoted by $E_{X,Y}$. Unlike other word-counting methods, which measure similarity by detecting multiple occurrences and handling them according to their matching scores, as in the well-known algorithm Blast (Altschul, et al., 1990), which uses the SHP criterion, SMS takes into account both the position and the inclusion of the matched subsequences.

The fact that we utilize a single similarity value which includes all of the identical matches as well as matched motifs from positions which, while non-equivalent according to the primary structure, might well be equivalent when viewed in terms of secondary and tertiary structure, allows us to take advantage of certain information included in the secondary and tertiary structure. Certainly, taking into account only identical motifs may lead us to overlook some important information in computing similarity. But at the same time, it also filters out noise (i.e., similarities due to chance) from our similarity measure. We believe that for protein datasets which include a small number of biochemical activities, the overlooked information is relatively insignificant compared with the noise-filtering effect.

The experimental results reported in our recent studies (Kelil, et al., 2007a) and (Kelil, et al., 2007b) seem to confirm the advantage of this strategy for such datasets. On the other hand, these studies show that the strategy is not very efficient on protein datasets with a large number of biochemical activities. This suggests that the proportion of overlooked similarity

information may become more significant as the number of biochemical activities increases, undermining the noise-filtering advantage.

The similarity measure SMS also suffers from problems of speed. Although we utilized a technique to speed up the extraction of significant motifs, the variable length of the matched sequences made it not impossible to reduce the worst-case complexity to a linear time using this technique.

In this paper, we propose a new similarity measure named tSMS (for “tolerant SMS”) which generalizes SMS in terms of tolerance to mismatches and scales well with increase in the number of biochemical activities. Also, tSMS is much faster than SMS; this is made possible by the optimization techniques used, which have reduced the worst-case complexity to a linear time.

2.1 The matching set

We will use $| \cdot |$ to express the length of a sequence. Let X and Y be two protein sequences whose similarity we want to measure, belonging to the protein family F which contains N protein sequences. Let x and y be two subsequences of the same length, belonging to X and Y , respectively. We use $\Gamma_{x,y}$ to represent the matched subsequence of x and y . We use l to represent the minimum number of matched residues between x and y that $\Gamma_{x,y}$ must include; at the same time, l is also the maximum number of non-matched residues allowed in $\Gamma_{x,y}$. A detailed discussion on the choice of the value of l was given in (Keil, et al., 2007a). The length l is used with the aim of detecting and utilizing the significant motifs best conserved during evolution and minimizing the influence of motifs that occur by chance. We use m (chosen by the user) to represent the minimum substitution score that two matched residues must have in order to be considered similar, or to be considered allowable in $\Gamma_{x,y}$. For X and Y , we define the set of all matched subsequences $\Gamma_{x,y}$ denoted by $E_{X,Y}^{l,m}$, as follows:

$$E_{X,Y}^{l,m} = \left\{ \Gamma_{x,y} \mid \begin{array}{l} |\Gamma_{x,y}| = |x| = |y| \\ \text{card}(\{\Gamma_{x_i,y_i} \mid x_i = y_i\}) \geq l \\ \text{card}(\{\Gamma_{x_i,y_i} \mid x_i \neq y_i\}) \leq l \\ \forall i \leq |\Gamma_{x,y}|, (x_i \in x) \wedge (y_i \in y) \Rightarrow M(x_i, y_i) \geq m \\ (\forall \Gamma_{x',y'} \in E_{X,Y}^{l,m}) \wedge (\Gamma_{x',y'} \neq \Gamma_{x,y}) \Rightarrow (x' \not\subset x) \vee (y' \not\subset y) \end{array} \right\} \quad (1)$$

Here M is one of the substitution matrices (chosen by the user) and i is used to identify the i^{th} position in a subsequence. The variables x_i and y_i are simply the i^{th} amino acids belonging to subsequences x and y , respectively. $M(x_i, y_i)$ is the substitution score of the i^{th} amino acids of the subsequences x and y . The constant m is a minimum value that the score between amino acids x_i and y_i must have to be considered as matched. The symbols x' and y' in the formula are simply used as variables, in the same way as x and y . The expression $(. \not\subset .)$ means that the element to the left of the symbol \subset is not included in the one to the right, either in terms of the composition of the subsequences or in terms of their respective positions in their protein sequence. The role of l is to detect and make use of the significant motifs best conserved during evolution and to minimize the influence of the motifs that occur by chance.

The matching set $E_{x,y}^{l,m}$ thus includes the significant motifs that correspond to matched protein subsequences that are more likely to be similar due to conservation phenomena and not due to chance. The matching set will be used to compute the matching score of the pair of sequences. Here are a few detailed explanations about Formula 1:

- $|\Gamma_{x,y}| = |x| = |y|$ means that the matched motif $\Gamma_{x,y}$ as well as the matched subsequences x and y include the same number of amino acids.
- $\text{card}(\{\Gamma_{x_i,y_i} | x_i = y_i\}) \geq l$ means that the matched motif $\Gamma_{x,y}$ must include at least l identical similar residues according to the threshold m .
- $\text{card}(\{\Gamma_{x_i,y_i} | x_i \neq y_i\}) \leq m$ means that the matched motif $\Gamma_{x,y}$ can include at most l non-identical residues according to the threshold m .
- $|\Gamma_{x,y}| \geq l$ means that the matched subsequences $\Gamma_{x,y}$ must have the minimum length l .
- $\forall i \leq |\Gamma_{x,y}|, (x_i \in x) \wedge (y_i \in y) \Rightarrow M(x_i, y_i) \geq m$ means that the subsequences x and y must not include matched residues with a substitution score less than a threshold m .
- $(\forall \Gamma'_{x',y'} \in E_{x,y}^{l,m}) \wedge (\Gamma'_{x',y'} \neq \Gamma_{x,y}) \Rightarrow (x' \not\subset x) \vee (y' \not\subset y)$ means that for any matched subsequences $\Gamma_{x,y}$ and $\Gamma'_{x',y'}$ belonging to $E_{x,y}^{l,m}$, $\Gamma'_{x',y'}$ and $\Gamma_{x,y}$ being different implies that $\Gamma'_{x',y'}$ is not included in $\Gamma_{x,y}$ either in terms of the composition of their corresponding subsequences or in terms of their respective positions in their protein sequences according to the partial order induced by set inclusion. In other words, each of the $\Gamma_{x,y}$ in $E_{x,y}^{l,m}$ is maximal.

To summarize, the formula means that the matching set $E_{x,y}^{l,m}$ contains all the matched subsequences $\Gamma_{x,y}$ of maximal length (i.e., at least l identical matched residues and at most l non-identical matched residues) between the sequences X and Y , with a tolerance to mismatches determined by m .

The formula $E_{x,y}^{l,m}$ adequately describes some known properties of polypeptides and proteins. First, protein motifs (i.e., series of defined residues) determine the tendency of the primary structure to adopt a particular secondary structure, a property exploited by several secondary-structure prediction algorithms. Such motifs can be as short as four residues (for instance, those found in β -turns), but the propensity to form an α -helix or a β -sheet is usually defined by longer motifs. Second, our proposal to take into account multiple occurrences of a particular motif reflects the fact that sequence duplication is one of the most powerful mechanisms of gene and protein evolution. If a motif is found twice or more in a protein, it is more probable that it was acquired by duplication of a segment from a common ancestor than by acquisition from a distant ancestor.

2.2 Definition of the similarity measure tSMS

Our primary concern is to develop an approach that will enable us to cluster hard-to-align protein sequences such as circularly-permuted, multi-domain and tandem-repeat protein

sequences. For such sequences, the alignment-dependent approaches usually fail to yield biologically suitable results. In fact, the hard-to-align proteins often have similar and conserved domains in non-equivalent positions in the primary structure, which makes them difficult to align. However, these domains might well be in equivalent positions when viewed in terms of secondary and tertiary structure. In the absence of explicit identification of such positions in our alignment-free approach to similarity computation, we adopted the strategy of matching all the conserved domains, even those on non-equivalent positions. The reason is that, with a suitable value of the minimum threshold “ l ” for matched motifs, which allows us to detect and make use of the significant motifs best conserved during evolution and to minimize the influence of those motifs that occur by chance, it is more probable that we will effectively match motifs that are similar due to conservation rather than to random phenomena.

For a protein sequence that comprises a number of significant motifs that were better conserved during evolution, each motif contributes in a complex way to provide one or more biological functions. A mutation in one of the conserved motifs can significantly alter or even eradicate the biological activity of the protein, while in another conserved motif it might only slightly decrease the expression of the biological function. So, we make use of a substitution matrix to emphasize the fact that each conserved motif can be involved to a different degree in a biological activity.

Let M be a substitution matrix, and Γ a matched subsequence belonging to the matching set $E_{X,Y}^{l,m}$. We define a weight $W(\Gamma)$ for the matched subsequence Γ , to quantify its importance compared to all the other matched subsequences of $E_{X,Y}^{l,m}$, as follows:

$$W(\Gamma) = \sum_{i=1}^{|l|} M(\Gamma[i], \Gamma[i]) \quad (2)$$

Where $\Gamma[i]$ is the i^{th} amino acid of the matched subsequence Γ , and $M(\Gamma[i], \Gamma[i])$ is the substitution score of this amino acid with itself. Here, in order to make our measure biologically plausible, we use the substitution concept to emphasize the relation that binds one amino acid with itself. The value of $M(\Gamma[i], \Gamma[i])$ (i.e., within the diagonal of the substitution matrix) estimates the rate at which each possible amino acid in a sequence remains unchanged over time. For the pair of sequences X and Y , we define the matching score $S_{X,Y}$, understood as representing the substitution relation of the conserved regions in both sequences, as follows:

$$S_{X,Y} = \frac{\sum_{\Gamma \in E_{X,Y}^{l,m}} W(\Gamma)}{\max(|X|, |Y|)} \quad (3)$$

Which is our new similarity measure tSMS for a pair of protein sequences X and Y .

2.3 Conservability versus mutability

The scoring of identical matches with a substitution matrix in SMS reflects the conservability of matched residues. The term conservability is more appropriate than

mutability. The nuance is significant for SMS. In fact, protein sequences to be compared contain conservability and mutability information. In the case of easy-to-align protein sequences, both conservability and mutability information can be obtained, while in the case of hard-to-align protein sequences mutability information is difficult to obtain. This is due to some known problems, such as the problem of repeats and the problem of substitutions; for details see (Higgins, 2004). To the best of our knowledge, existing alignment-based algorithms fail to effectively capture conservability and mutability information in hard-to-align protein sequences. On the other hand, the experimental results reported in (Kelil, et al., 2007a) show that the use of only conservability information allows SMS to deal with hard-to-align sequences better than the alignment-based algorithms. Experimental results also show that SMS handles easy-to-align protein sequences equally well as the alignment-based algorithms. This suggests that the utility of conservability might be much more significant than is generally believed. However, the experiments conducted showed that, as the number of biochemical activities increases, the strategy of capturing only the conservability information becomes increasingly insufficient to obtain an accurate similarity measure. Therefore, the use of mutability information becomes inevitable to overcome this drawback. In tSMS, both conservability and mutability information are captured and used to measure the similarity.

2.4 Computational complexity

To compute tSMS, we have made use of a variant of the data structure known as the “*Suffix Tree*” (Weiner, 1973), developed by (Cole, et al., 2006) and named the “*Suffix Tray*”. The Suffix Tree is a well-known approach to solving the problem of string matching in linear time. Given the question of how many occurrences of a pattern P there are in a string T and where they occur, the Suffix Tree allows an answer to be generated in $O(|P| + z|T|)$ time and with $O(|T|)$ space, where z is the number of occurrences of the pattern P in the text T . With the Suffix Tray, on the other hand, the same task can be performed in $O(|P| + \log \Sigma)$ with the same space complexity $O(|T|)$ as for the Suffix Tree. Here Σ is the alphabet size. For our case $\Sigma=20$, which is the number of amino acids. The fact that the Suffix Tray performs the matching in a time independent of $|T|$ is very advantageous for speeding up our algorithm.

Let X and Y be a pair of protein sequences to be compared. We start by building the Suffix Trays corresponding to the individual sequences, T_X and T_Y , which takes time and space $O(|X|)$ and $O(|Y|)$, respectively. These Suffix Trays are trees of $O(|X|)$ and $O(|Y|)$ nodes, containing all the suffixes of the protein sequences X and Y , respectively. Instead of matching X and Y , which takes time $O(|X| \times |Y|)$, we perform the same task by matching only the suffixes of T_X with those of T_Y , or vice-versa, as follows:

Let $T_X = \{x_1, x_2, \dots, x_t\}$ be the set of all suffixes of T_X , where t is the number of possible suffixes. Finding all the occurrences (i.e., exact matching) of a suffix x_i out of the Suffix Tray T_Y takes time $O(|P| + \log 20)$. Let k be the average number of possible matches of all amino acids according to the chosen value of m (in Formula 1) and the chosen substitution matrix. If we consider that we allow a restricted number of matches per residue (see Table 1) and a restricted number of mismatches per matched motif (i.e., $\leq l$), in the worst case, there exist k^l possible transformations of x_i , which implies that the pattern x_i will have to be matched k^l

times with the Suffix Tray T_Y . This has a time complexity of $\bar{k}^l O(|P| + \log 20)$. Since both k and l are constants, and are usually small values, the coefficient \bar{k}^l is also a constant. Performing the matching between all T_X suffixes and the Suffix Tray T_Y thus takes time $\bar{k}^l O(|x_1| + \log 20) + \bar{k}^l O(|x_2| + \log 20) + \dots + \bar{k}^l O(|x_t| + \log 20) = \bar{k}^l O(|X|)$, which is also linear.

Table 1. Number of possible matches for each amino acid with different values of m

Amino acids	BLOSUM 62			PAM250		
	$m=0$	$m=1$	$m=2$	$m=0$	$m=1$	$m=2$
A	6	2	1	10	5	1
C	2	1	1	3	1	1
D	5	3	2	10	6	4
E	8	4	3	10	5	3
F	6	3	2	6	4	3
G	4	1	1	7	4	1
H	6	3	2	9	6	4
I	5	4	3	6	5	4
K	6	4	2	10	4	2
L	5	4	3	5	5	5
M	6	4	2	7	4	4
N	10	4	1	11	7	3
P	1	1	1	7	3	1
Q	9	4	2	9	7	4
R	6	3	2	9	5	4
S	9	4	1	11	6	1
T	5	2	1	11	3	1
V	6	4	2	6	4	4
W	3	3	2	4	2	2
Y	4	4	4	5	2	2
Average \bar{k}	5.6	3.1	1.9	7.8	4.4	2.7

Depending on the m value (i.e., column), each amino acid (i.e., row) has a limited number of possible matches; each \bar{k} value is the average of the corresponding column values.

3 The New Clustering Algorithm CLUSS2

CLUSS2 is composed of three main stages. The first one consists in building a pairwise similarity matrix S using our new similarity measure tSMS. The second consists in building a phylogenetic tree according to this matrix, using a new hierarchical clustering approach based on spectral decomposition. The third consists in identifying subfamily nodes from which leaves are grouped into subfamilies.

In the algorithm CLUSS (Kelil, et al., 2007b), we used a classical clustering approach by directly making use of the pairwise similarity matrix. In the present version we have developed a new and original hierarchical algorithm, inspired by the LSA approach, for more details see (Berry, et al., 1996). We take advantage of this approach by extracting global information from a large number of protein sequences rather than carrying out a pairwise comparison. We have chosen to keep the name CLUSS, since both versions have the same basic principles, and they are inspired from the same idea.

3.1 Stage 1: Similarity matrix

Using one of the known substitution score matrices, such as BLOSUM62 or PAM250, and our new similarity measure tSMS, we compute S , the $N \times N$ pairwise similarity matrix, where N is the number of sequences of the protein family F to be clustered, and $S_{i,j}$ is the similarity between the i^{th} and the j^{th} protein sequences of F . By using tSMS, the construction of the pairwise similarity measure matrix S becomes much faster, since we transform all the N protein sequences into Suffix Trays only once before the pairwise matching of the protein sequences. Both the transformation of each protein sequence and the matching of two protein sequences take linear time with respect to sequence length, as seen in Section 2.1.

3.2 Stage 2: Phylogenetic tree

Using spectral decomposition on the pairwise similarity matrix S , we obtain a set of vectors. Each of the vectors is used to represent a protein sequence in the new vector space resulting from the decomposition of S . Such a representation is valid in the sense that the similarity between each pair of sequences from the original similarity matrix S is equal or approximately equal to the similarity between the corresponding vectors measured by the inner product function. This representation facilitates the subsequent (hierarchical) clustering. In fact, a cluster will be represented by only one vector; cluster merging can be easily performed by adding two vectors; and the similarity between two clusters can then be estimated by the cosine similarity function. This stage is composed of three steps, as follows.

3.2.1 Step 1: Spectral decomposition of the similarity matrix S

We will utilize the theorem in linear algebra, which states that any $R \times C$ matrix A whose number of rows R is greater than or equal to its number of columns C can be written as the product of an $R \times C$ column-orthogonal matrix U , a $C \times C$ diagonal matrix Z with non-negative elements, which are the singular values, and the transpose of an $C \times C$ orthogonal

matrix V . This decomposition is named the singular value decomposition (SVD). The matrix A can be written as follows:

$$A = U \times Z \times V^T \quad (4)$$

We apply the SVD to the squared pairwise similarity matrix $S_{N \times N}$, which is decomposed into the product of three $N \times N$ matrices U , Z and V . The first of these, U , is a left singular matrix describing the original row entities as vectors of derived orthogonal factor values; the second, Z , is a diagonal matrix containing non-negative scaling values; and the third, V , is a right singular matrix describing the original column entities in the same way as the first matrix. Since the matrix Z contains non-negative singular values, the SVD of S can be written in the following form:

$$S = (U \times \sqrt{Z}) \times (\sqrt{Z} \times V^T) \quad (5)$$

For the special case where S is a square and symmetric matrix with a diagonal including much larger values than the rest of the matrix (as is the case here), the matrix S is very likely to be a semi-definite positive matrix, or at least very close to that. We can thus write Formula 6 in the form:

$$S \simeq (U \times \sqrt{Z}) \times (\sqrt{Z} \times U^T) \quad (6)$$

We can write:

$$S \simeq (U \times \sqrt{Z}) \times (U \times \sqrt{Z})^T \quad (7)$$

We define an $N \times N$ matrix $E = U \times \sqrt{Z}$, for which each row $E_i = U_i \times \sqrt{Z}$. Now each protein sequence i belonging to the protein family F to be clustered is represented by the vector E_i in the new vector space, mapped by the matrix E . Therefore, the similarity measure $S_{X,Y}$ between a pair of sequences X and Y is now equal or approximately equal to the inner product $\langle E_X, E_Y \rangle$. The idea of mapping the protein sequences onto a vector space is based on the conservability of distance. This transformation allows us to apply vector operations during the clustering process and obtain (and maintain) a representative for each subcluster. The transformation, as discussed in LSA, also allows us to take advantage of transitivity in the similarities between pairs of proteins (documents, in LSA).

It is possible to take further advantage of this representation. In fact, by taking into account only the K (where $K \leq N$) largest non-negative singular values from the $N \times N$ matrix Z , and their corresponding singular vectors from the $N \times N$ matrices U and V , we get the rank K approximation of S with the smallest error according to the Frobenius norm (Golub, et al., 1996). The matrices U , Z and V are reduced to $N \times K$, $K \times K$ and $N \times K$ matrices, respectively. Thus, the spectral decomposition approach maps the protein vectors onto a new multidimensional space in which the corresponding vectors are the rows of the $N \times K$ matrix E . Reducing the K value significantly speeds up the clustering process. In the experiments carried out in this paper, we have not exploited the strategy of reducing the value of K , since we set it to $K=N$ because we wanted to concentrate our efforts on the accuracy of the new clustering approach adopted in CLUSS2. However, we will do it extensively in a future work.

In our implementation of the singular value decomposition, we made use of the fast, incremental, low-memory and large-matrix singular values decomposition algorithm recently developed by (Brand, 2006), that for a K rank matrix $N \times N$, the singular value decomposition can be performed in $O(KN^2)$ time with $K \leq \sqrt{N}$.

3.2.2 Step 2: Phylogenetic tree

Starting from vectors E_1, E_2, \dots, E_N , each of which is considered as the root node of a subtree containing only one node, we initialize the similarity between any pair of nodes by the cosine product of corresponding vectors. We iteratively join a pair of root nodes in order to build a bigger subtree. At each iteration, a pair of root nodes is selected if they are the most similar root nodes (i.e., corresponding vectors have the largest cosine product). This process ends when there remains only one subtree, which is the phylogenetic tree.

Now we introduce the concept of co-similarity for ranking the nodes in the phylogenetic tree. Let L and R be a pair of nodes (L for left and R for right) belonging to the phylogenetic tree. By taking into account information about the neighbourhood around each of the nodes L and R , the concept of co-similarity reflects the cluster compactness of all the sequences (i.e., leaf nodes) in the subtree. In fact, its value is inversely proportional to the within-cluster variance. As the subtree becomes larger, the co-similarity tends to become smaller, which means that the sequences within the subtree become less similar and the difference (i.e., separation) between sequences in different clusters becomes less significant. In simpler terms, the co-similarity of a particular node is a measure of the balance between its two child nodes. Before the construction of the phylogenetic tree, all co-similarities (of the leaves) are initialized to zero.

Let L and R be the two most similar root nodes at a given iteration step; they are joined together to form a new subtree P (P for parent), which thus has two children, L and R , such that E_P is its corresponding vector. The new root node P has the following definitions:

$$E_P = E_L + E_R \quad \text{and} \quad c_P = \frac{\|E_L\| \times \|E_R\|}{\|E_L\| + \|E_R\|} \quad (8)$$

where E_L , E_R and E_P are vectors corresponding to the root nodes L , R and P , respectively, and c_P is the co-similarity of P . The norms $\|E_L\|$ and $\|E_R\|$ depend on the number and proximity of leaves belonging to the subtrees L and R , respectively, and they measure how well F is represented by each one of these particular subtrees. According to this definition, the value of a norm is large if the corresponding subtree is more representative and small if it is less representative.

We assign a “length” value to each of the two branches connecting L and R to P . These values are the estimate of the phylogenetic distance from the individual nodes L and R to their parent P in the tree. This distance has no strict mathematical sense; it is merely a measure of the evolutionary distance between the nodes. It is comparative to the notion of dissimilarity. We calculate it as follows:

$$d_{L,P} = \frac{\|E_R\|}{\|E_L\| + \|E_R\|} \quad \text{and} \quad d_{R,P} = \frac{\|E_L\|}{\|E_L\| + \|E_R\|} \quad (9)$$

3.2.3 Step 3: Separating nodes

This step is exactly the same as in the CLUSS algorithm. However, we give more details about this step here. The CLUSS2 algorithm makes use of a systematic method for deciding which subtrees to retain as a trade-off between searching for the highest co-similarity values and searching for the largest possible clusters. We first separate all the subtrees into two groups, one being the group of low co-similarity subtrees, and the other the high co-similarity subtrees. This is done by sorting all possible subtrees in increasing order of co-similarity and computing a separation threshold according to the maximum interclass inertia method, based on the Koenig-Huygens theorem, which gives the relationship between the total inertia and the inertia of each group relative to the centre of gravity. In our case we have just two groups, the high co-similarity group and the low co-similarity group. The procedure is described as follows:

Let D be the set of subtrees, D_{Low} the subset of low co-similarity subtrees, and D_{High} the subset of high co-similarity subtrees, such that:

$$D_{Low} \cup D_{High} = D \quad \text{and} \quad D_{Low} \cap D_{High} = \emptyset \quad (10)$$

$$\forall L, R \in D \mid L \in D_{Low}, R \in D_{High} \Rightarrow c_L < c_R \quad (11)$$

The symbols D_{Low} and D_{High} are simply used as variables representing all possible separations of D according to equations 10 and 11. According to the Koenig-Huygens theorem, we calculate the total inertia as follows:

$$I_{Total} = \sum_{i \in D_{Low}} (c_i - \bar{c}_{D_{Low}})^2 + \sum_{j \in D_{High}} (c_j - \bar{c}_{D_{High}})^2 + (\bar{c}_{D_{Low}} - \bar{c}_{D_{High}})^2 \quad (12)$$

where c_i and c_j are co-similarity values of subtrees i and j belonging to the subsets D_{Low} and D_{High} , all respectively; and $\bar{c}_{D_{Low}}$ and $\bar{c}_{D_{High}}$ are means (i.e., centres of gravity) of subsets D_{Low} and D_{High} , respectively. The best separation of D , the set of sorted subtrees on two subsets D_{Low} and D_{High} , is given by the maximum value of I_{Total} .

3.3 Stage 3: Extracting clusters

From the subset of high co-similarity subtrees belonging to D_{High} , we extract those that are largest. A high co-similarity subtree is largest if the following two conditions are satisfied:

- It does not contain any low co-similarity subtree belonging to the subset D_{Low} .
- If it is included in another high co-similarity subtree, the latter contains at least one low co-similarity subtree from the subset D_{Low} .

Each of these largest subtrees corresponds to a cluster and its leaves are then collected to form the corresponding cluster.

4 Experiments

To illustrate its efficiency, we tested CLUSS2 extensively on a variety of protein datasets and compared it both with CLUSS and with several mainstream clustering algorithms. We analyzed the results obtained for the different tests with support from the literature and functional annotations. All the data and results cited in this section are available on the CLUSS website <http://prospectus.usherbrooke.ca/CLUSS>. To evaluate the quality of the clustering results obtained, in our experiments we used the Q-measure that we introduced in (Kelil, et al., 2007b).

4.1 Benchmarking

To illustrate the efficiency of CLUSS2 in grouping protein sequences according to their functional annotations and biological classifications, we performed extensive tests on the widely known databases COG (unicellular organisms), KOG (eukaryotic organisms) and PC (microbial protein clusters). The COG and KOG databases include clusters of orthologous groups of proteins that were delineated by comparing protein sequences encoded in complete genomes, representing major phylogenetic lineages. The PC database is a compilation of proteins from the complete genomes of prokaryotes, plasmids and organelles that have been grouped and manually curated and annotated based on sequence similarity and protein function.

Table 2. Generated datasets

Benchmark	<i>A</i>		<i>B</i>		<i>C</i>	
	Av. Nbr.	Av. Length	Av. Nbr.	Av. Length	Av. Nbr.	Av. Length
$A_1 B_1 C_1$	298	1087	487	1102	678	1198
$A_2 B_2 C_2$	230	2024	458	2043	696	2076
$A_3 B_3 C_3$	256	815	449	895	628	912

Av. Nbr. is the average number and **Av. Length** is the average length, of all protein sequences within each set (row), in each benchmark (column).

In order to evaluate CLUSS2 in a statistical manner, we generated three benchmarks named *A*, *B* and *C*, each containing three different large sets, such that $A=\{A_1, A_2, A_3\}$, $B=\{B_1, B_2, B_3\}$ and $C=\{C_1, C_2, C_3\}$. The nine sets in these benchmarks have been generated in this way; A_1 , B_1 and C_1 from the COG database, A_2 , B_2 and C_2 from the KOG database and A_3 , B_3 and C_3 from the PC database. Each set contains 1000 different, large, randomly generated subsets of protein sequences. Each subset contains a large number of non-orphan protein sequences (i.e., each protein sequence has at least one similar protein sequence from the same functional classification). Each subset in the benchmark *A* contains a number of proteins with at least 5 biochemical activities. In the benchmark *B*, each subset contains a number of proteins with at least 10 biochemical activities. And finally, in the benchmark *C*, each subset

contains a number of proteins with at least 20 biochemical activities. Details about the generated benchmarks are given in Table 2. We tested CLUSS2 and CLUSS on the three benchmarks using both substitution matrices BLOSUM62 and PAM250. The obtained results for both matrices were very similar. The results obtained are shown in Table 3, and discussed below.

Table 3. Benchmarking results (Time in seconds)

Benchmark	Algorithm	A			B			C		
		Qm	SD	Time	Qm	SD	Time	Qm	SD	Time
$A_1 - B_1 - C_1$	CLUSS2	90.32	3.56	11	86.74	3.41	5	96.61	3.86	15
	CLUSS	90.56	4.04	27	86.15	4.63	21	96.28	4.67	60
$A_2 - B_2 - C_2$	CLUSS2	92.25	5.12	16	87.34	5.76	9	94.45	5.71	18
	CLUSS	90.81	7.87	49	85.16	7.10	39	91.64	7.14	68
$A_3 - B_3 - C_3$	CLUSS2	91.85	7.92	18	86.56	8.45	14	96.11	7.81	21
	CLUSS	81.39	9.60	61	77.91	11.09	55	88.68	10.94	92

Qm is the average Q-measure, **SD** the standard deviation and **Time** the average execution time, of the clustering results of each set (row) in each benchmark (main column) using each CLUSS version (child row).

4.1.1 Benchmark A with 5 biological activities (Table 3)

The average Q-measure (Qm) and the standard deviation (SD) values of the clustering results obtained for each database (COG, KOG and PC) are essentially equal with CLUSS2 and CLUSS. However, the execution times (Time) for each database clearly show that CLUSS2 is definitely faster than CLUSS.

4.1.2 Benchmark B with 10 biological activities (Table 3)

The Qm and SD values of the clustering results obtained for each of the databases show a small advantage of CLUSS2 compared to CLUSS. However, the Time values for each database show once again that CLUSS2 is faster than CLUSS.

4.1.3 Benchmark C with 20 biological activities (Table 3)

The Qm values of the clustering results obtained for each of the databases using CLUSS2 are clearly higher than those obtained with CLUSS. Also, the SD values of the clustering results obtained for each database using CLUSS2 are visibly lower than those obtained with CLUSS. The Time values for each database using CLUSS2 increase much more slowly than those obtained using CLUSS.

The results obtained clearly show that CLUSS2 is indeed effective in grouping sequences according to the known functional classification of COG, KOG and PC databases more efficiently than CLUSS. Contrary to what was observed for CLUSS, the efficiency of the new algorithm CLUSS2 does not notably decrease with an increase in the number of biochemical functions included in the clustered protein datasets. Another important fact to note is that the optimization techniques used in the new similarity measure tSMS have significantly improved the time efficiency of the clustering process.

Table 4. Clustering results on the COG database (Time in seconds)

Protein subsets	CLUSS2		CLUSS		BlastClust		TRIBE-MCL		gSPC	
	Qm	Time	Qm	Time	Qm	Time	Qm	Time	Qm	Time
C1 (509)	96.02	33	80.01	109	67.01	20	30.02	422	38.01	451
C2 (448)	98.07	35	68.13	94	42.07	19	35.01	406	31.02	386
C3 (546)	92.01	36	72.03	114	40.01	22	55.08	479	44.01	492
C4 (355)	98.04	23	86.04	69	69.01	16	40.01	273	42.01	280
C5 (508)	96.01	29	63.04	137	35.03	16	57.01	446	36.10	440
C6 (509)	96.02	33	80.01	109	67.01	20	30.02	422	38.01	451

Table 5. Comparison on the KOG database (Time in seconds)

Protein Subsets	CLUSS2		CLUSS		BlastClust		TRIBE-MCL		gSPC	
	Qm	Time	Qm	Time	Qm	Time	Qm	Time	Qm	Time
K1 (317)	97.02	61	82.02	242	33.13	41	54.05	790	40.02	843
K2 (419)	95.02	86	69.02	279	55.02	63	60.01	371	50.01	450
K3 (383)	91.01	161	76.02	381	69.01	134	30.02	1244	30.02	1348
K4 (458)	95.02	54	76.05	310	37.01	37	59.02	1315	47.01	1349
K5 (480)	95.06	60	79.33	324	50.02	34	46.03	1425	43.02	1409
K6 (388)	93.02	76	80.03	441	32.01	49	49.01	1269	55.04	1336

Table 6. Clustering results on the PC database (Time in seconds)

Protein Subsets	CLUSS2		CLUSS		BlastClust		TRIBE-MCL		gSPC	
	Qm	Time	Qm	Time	Qm	Time	Qm	Time	Qm	Time
P1 (538)	91.02	29	65.01	84	44.01	16	31.01	447	41.02	441
P2 (392)	94.01	23	73.01	79	31.01	18	35.02	250	57.01	264
P3 (442)	93.02	31	70.01	84	34.06	14	32.01	316	39.01	390
P4 (595)	95.02	46	60.01	152	66.01	35	58.50	711	30.02	633
P5 (561)	91.17	39	81.02	97	68.08	18	54.02	433	34.01	435
P6 (427)	94.02	22	77.08	75	34.01	16	43.03	410	49.02	399

4.2 Comparisons

To compare the efficiency of CLUSS2 to that of alignment-dependent clustering algorithms, we performed tests using CLUSS2, CLUSS, BlastClust, TRIBE-MCL and gSPC

on the COG, KOG and PC databases. In all of the tests performed, we used the widely known protein sequence comparison algorithm ClustalW (Thompson, et al., 1994) to calculate the similarity measure matrices used by TRIBE-MCL and gSPC. Due to the complexity of alignment, these tests were done on three sets of six randomly generated subsets, named C1 to C6 for COG, K1 to K6 for KOG and P1 to P6 for PC; each generated protein subset includes protein sequences with at least 20 biological activities.

The results obtained are summarized in Table 4, Table 5 and Table 6. The experiments show clearly that CLUSS2 obtained the best Q-measure, compared to the other algorithms tested. Even if we compare the results of CLUSS2 with those of CLUSS, we can see that CLUSS2 has obtained better clustering results. This is because each of the subsets tested contains a number of proteins with a large number of biological functions (each subset includes protein sequences with at least 20 biological functions). Globally, the clusters obtained using our new algorithm CLUSS2 correspond better to the known characteristics of the biochemical activities and modular structures of the protein sequences according to the COG, KOG and PC classifications. The execution times reported in Table 4, Table 5 and Table 6 for algorithm comparison, show clearly that the fastest algorithm is BlastClust, closely followed by the CLUSS2 algorithm, and then by CLUSS, while TRIBE-MCL and gSPC, which use ClustalW as a similarity measure, are much slower.

4.3 G-Proteins family

The G-Proteins (for guanine nucleotide binding proteins) that are available at <http://www.gpcr.org/> belong to the larger family of GTPases. Their signalling mechanism consists in exchanging guanosine diphosphate (GDP) for guanosine triphosphate (GTP) as a general molecular function to regulate cell processes, reviewed extensively in (Lodish, et al., 2004). This family has been the subject of a considerable number of publications by researchers around the world, so we considered it a good reference classification to test the performance of CLUSS2. The sequences belonging to this family (version of October 6, 2007), including the 2604 sequences used in our experiments, are available on the CLUSS website. The experimental results obtained using both the CLUSS2 and CLUSS algorithms as well as the algorithms BlastClust, TRIBE-MCL and gSPC are summarized in Table 7.

The clustering results for the G-Proteins family show clearly that although this family is known to be easy to align, which should have facilitated the clustering task of the alignment-dependent algorithms, CLUSS2 yields a clustering with the highest Qm value of all the algorithms tested, nearly followed by CLUSS. Thus, the results obtained by CLUSS2 are much closer to the known classification of the G-Proteins family than are those of the other algorithms tested. In Table 7, we can make the same observation about the execution times of the different algorithms as in Table 4, Table 5 and Table 6.

Table 7. Clustering results of the G-Proteins family (Time in seconds)

Protein set	CLUSS2		CLUSS		BlastClust		TRIBE-MCL		gSPC	
	Qm	Time	Qm	Time	Qm	Time	Qm	Time	Qm	Time
G-Proteins	91.78	402	89.32	2199	57.78	372	50.89	32654	61.45	36751

4.4 The 33 (α/β)₈-barrel proteins

To show the performance of CLUSS2 with multi-domain protein families which are known to be hard to align and have not yet been definitively aligned, experimental tests were performed on the 33 (α/β)₈-barrel proteins studied recently by (Côté, et al., 2006) and (Fukamizo, et al., 2006), which form a group in Glycoside Hydrolases family 2 (GH2) from the Carbohydrate Active Enzymes database (CAZy) located at <http://www.cazy.org/>. The periodic character of the catalytic module known as “(α/β)₈-barrel” makes these sequences hard to align using classical alignment approaches. The difficulties in aligning these modules are comparable to the problems encountered with the alignment of tandem-repeats, which have been exhaustively discussed by (Higgins, 2004). The FASTA file and full clustering results of this subfamily are available on the CLUSS website. This group of 33 protein sequences includes “ β -galactosidase”, “ β -mannosidase”, “ β -glucuronidase” and “exo- β -D-glucosaminidase” enzymatic activities, all extensively studied at the biochemical level. These sequences are multi-modular, with various types of modules, which complicate their alignment. Clustering such protein sequences using the alignment-dependent algorithms thus becomes problematic. This encouraged us to perform a clustering on this particular group of the GH2 subfamily, to compare the behaviour of both algorithms CLUSS2 and CLUSS with BlastClust, TRIBE-MCL and gSPC in order to validate the use of CLUSS2 on the hard-to-align proteins. An overview of the results is given in Table 8, with a detailed discussion below. The corresponding names and database entries of the 33 (α/β)₈-barrel proteins group are indicated on the CLUSS website.

The 33 (α/β)₈-barrel proteins were subdivided by CLUSS2 and CLUSS into five subfamilies, corresponding to their known biochemical activities. However, contrarily to CLUSS, which has classified the two proteins MaC and MaT with the first cluster, CLUSS2 classified all the 33 (α/β)₈-barrel proteins in the same subfamilies obtained by (Côté, et al., 2006) that in turn are supported by the structure-function studies of (Fukamizo, et al., 2006). This shows the superiority of CLUSS2 comparing to CLUSS in clustering protein sequences. The first cluster includes enzymes with “ β -mannosidase” activities; the second cluster includes enzymes with “ β -mannosidase” activities; the third cluster includes enzymes with “ β -glucuronidase” activities; the forth cluster includes enzymes with “ β -galactosidase” activities; the fifth cluster includes enzymes with “exo- β -D-glucosaminidase” activities. While the other algorithms do not succeed to obtain clustering results that correspond to the functional classification of the 33 (α/β)₈-barrel proteins group obtained by (Côté, et al., 2006) and (Fukamizo, et al., 2006). Since, there are a number of well classified proteins (i.e., GaA, GaK, GaC, CsAo, CsN and CsAn) which could not be classified by BlastClust, and a number of proteins which were wrongly classified by TRIBE-MCL and gSPC, for details see Table 8.

Table 8. Clustering results of the 33 (α/β)₈-barrel proteins group

#	Proteins	Côté/Fukamizo	CLUSS2	CLUSS	BlastClust	TRIBE-MCL	gSPC
1	UnA	1	1	1	1	1	1
2	UnBv	1	1	1	1	1	1
3	UnBc	1	1	1	/	1	1
4	UnBm	1	1	1	1	1	1
5	UnBp	1	1	1	1	1	1
6	UnR	1	1	1	1	1	1
7	MaA	2	2	2	2	2	1
8	MaB	2	2	2	2	1	1
9	MaH	2	2	2	2	1	1
10	MaM	2	2	2	2	1	1
11	MaC	2	2	1	2	2	1
12	MaT	2	2	1	2	2	1
13	GIC	3	3	3	3	2	2
14	GIE	3	3	3	3	2	2
15	GIH	3	3	3	3	2	2
16	GIL	3	3	3	3	2	2
17	GIM	3	3	3	3	2	2
18	GIF	3	3	3	3	2	2
19	GIS	3	3	3	3	2	2
20	GaEco	4	4	4	4	2	2
21	GaA	4	4	4	/	2	2
22	GaK	4	4	4	/	2	2
23	GaC	4	4	4	/	2	2
24	GaEcl	4	4	4	4	2	2
25	GaL	4	4	4	4	2	2
26	CsAo	5	5	5	/	2	3
27	CsS	5	5	5	5	2	3
28	CsG	5	5	5	5	2	3
29	CsM	5	5	5	5	2	3
30	CsN	5	5	5	/	2	3
31	CsAn	5	5	5	/	2	3
32	CsH	5	5	5	5	2	3
33	CsE	5	5	5	5	2	3

The symbol “/” means that the corresponding algorithm (column) was not able to classify the corresponding protein (row) with any one of the other proteins (i.e., orphan protein).

5 Conclusion

Our new similarity measure tSMS makes it possible to measure the similarity between protein sequences much more quickly and effectively than SMS – especially for protein datasets that include proteins with a relatively large number of biochemical activities – based

solely on the conserved motifs, with a certain tolerance to mismatches. Its major advantage over the alignment-dependent approaches is that it gives significant results with protein sequences independent of their alignability, making it effective on both easy-to-align and hard-to-align protein sequences. These properties are inherited by CLUSS2, our new clustering algorithm, which uses tSMS as its similarity measure. Compared to CLUSS, our new clustering algorithm CLUSS2 is a much more effective clustering algorithm for protein sets with respect to the number of biological activities. It more accurately highlights the characteristics of the biochemical activities of the clustered protein sequences than do CLUSS and several mainstream alignment-dependent algorithms.

So far, our similarity measure tSMS has been based on pre-determined substitution matrices. A possible future development is to propose an approach to automatically compute the weights of the matched motifs instead of relying on pre-calculated substitution scores.

References

- Altschul, S. F.; Gish, W.; Miller, W.; Myers, E. W. & Lipman, D. J. (1990), 'Basic local alignment search tool', *J Mol Biol* **215**(3), 403-410.
- Berry, M. W. & Fierro, R. D. (1996), 'Low-Rank Orthogonal Decompositions for Information Retrieval Applications', *Numerical Linear Algebra with Applications* **3**(4), 301-327.
- Brand, M. (2006), 'Fast Low-Rank Modifications of the Thin Singular Value Decomposition', *Linear Algebra and Its Applications* **415**(1), 20-30.
- Cole, R.; Kopelowitz, T. & Lewenstein, M. (2006), *Automata, Languages and Programming*, 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part I, chapter Suffix Trays and Suffix Trists: Structures for Faster Text Indexing, pp. 358-369.
- Coutinho, P. M. & Henrissat, B. (1999), *Recent Advances in Carbohydrate Bioengineering*, The Royal Society of Chemistry, Cambridge, chapter Carbohydrate-Active Enzymes: An Integrated Database Approach, pp. 312.
- Côté, N.; Fleury, A.; Dumont-Blanchette, E.; Fukamizo, T.; Mitsutomi, M. & Brzezinski, R. (2006), 'Two exo-beta-D-glucosaminidases/exochitosanases from actinomycetes define a new subfamily within family 2 of glycoside hydrolases', *Biochem J* **394**(Pt 3), 675-686.
- Enright, A. J.; Dongen, S. V. & Ouzounis, C. A. (2002), 'An efficient algorithm for large-scale detection of protein families', *Nucleic Acids Res* **30**(7), 1575-1584.
- Fukamizo, T.; Fleury, A.; Côté, N.; Mitsutomi, M. & Brzezinski, R. (2006), 'Exo-beta-D-glucosaminidase from Amycolatopsis orientalis: Catalytic residues, sugar recognition specificity, kinetics, and synergism', *Glycobiology* **16**(11), 1064-1072.
- Golub, G. H. & Loan, C. F. V. (1996), *Matrix computations* (3rd ed.), Johns Hopkins University Press, Baltimore, MD, USA.
- Higgins, D. (2004), *The Phylogenetic Handbook - A Practical Approach to DNA and Protein Phylogeny*, Marco Salemi and Anne-Mieke Vandamme, chapter Multiple alignment, pp. 45-71.
- Kelil, A.; Wang, S. & Brzezinski, R. (2007a), 'A New Alignment-Independent Algorithm', IEEE 7th BIBE, Conference Center at Harvard Medical School, Cambridge, Boston, Massachusetts, USA.
- Kelil, A.; Wang, S.; Brzezinski, R. & Fleury, A. (2007b), 'CLUSS: clustering of protein sequences based on a new similarity measure.', *BMC Bioinformatics* **8**, 286.

- Lodish, H.; Berk, A.; Matsudaira, P.; Kaiser, C. A.; Krieger, M.; Scott, M. P.; Zipursky, L. & Darnell, J. (2005), *Molecular Cell Biology*, W.H. Freeman and Co., New York and Basingstoke.
- Tatusov, R. L.; Fedorova, N. D.; Jackson, J. D.; Jacobs, A. R.; Kiryutin, B.; Koonin, E. V.; Krylov, D. M.; Mazumder, R.; Mekhedov, S. L.; Nikolskaya, A. N.; Rao, B. S.; Smirnov, S.; Sverdlov, A. V.; Vasudevan, S.; Wolf, Y. I.; Yin, J. J. & Natale, D. A. (2003), 'The COG database: An updated version includes eukaryotes', *BMC Bioinformatics* 4, 41.
- Tetko, I. V.; Facius, A.; Ruepp, A. & Mewes, H. (2005), 'Super paramagnetic clustering of protein sequences', *BMC Bioinformatics* 6, 82.
- Thompson, J. D.; Higgins, D. G. & Gibson, T. J. (1994), 'CLUSTAL W: Improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice', *Nucleic Acids Res* 22(22), 4673-4680.
- Weiner, P. (1973), 'Linear Pattern Matching Algorithm', in IEEE Computer Society, ed., '14th Symposium on Switching and Automata Theory', Los Alamitos, CA, 1-11.

Chapitre 3

SIMILARITÉ DES SÉQUENCES CATÉGORIQUES

L'analyse sémantique latente, ou « *Latent Semantic Analysis* » (LSA), est une méthode utilisée pour le traitement du langage naturel. Elle permet d'établir les relations entre un ensemble de textes et les mots qui les composent. La LSA utilise une matrice qui décrit la présence ou l'absence des mots dans les textes. C'est une matrice creuse dont les lignes correspondent aux mots et les colonnes correspondent aux textes. La LSA transforme la matrice d'occurrences en un produit de trois matrices en utilisant la décomposition en valeurs singulières. Cette décomposition permet de représenter les textes dans un espace vectoriel où les textes sont représentés par des vecteurs, ainsi on peut utiliser les opérations vectorielles pour trouver les relations de similarités entre les textes. Cette transformation permet, entre autre, de réduire la taille de la matrice d'occurrence quand les moyens de calcul et de mémoire physique l'exigent, et aussi de filtrer la matrice d'occurrence lors de présence de bruit important. La LSA permet aussi de :

- Comparer des textes (clustering et classification);
- Comparer des textes de différentes langues (sémantique);
- Comparer des mots dans un ensemble de textes (synonymie et polysémie);
- Recherche d'information dans les textes sans recalculer la matrice d'occurrence.

Bien que la LSA ait été initialement développée pour l'analyse du langage naturel, elle a souvent été appliquée pour traiter différents types de séquences catégoriques [25, 38, 44, 154], et ce malgré qu'elles ne contiennent pas de motifs distincts comme les mots dans les textes. Les séquences catégoriques sont des données structurées sous forme de chaînes de symboles (c.à.d. catégories), pour lesquelles l'ordre chronologique ainsi que les caractéristiques structurelles (c.-à-d. les motifs qui caractérisent la nature intrinsèque des

séquences catégoriques apparentées) sont informatives et importantes. De nombreux types de données scientifiques et commerciales sont sous forme de séquences catégoriques: par exemple, les séquences biologiques comme les séquences d'ADN, les séquences d'ARN, et les séquences de protéines, ou alors le texte en langage naturel, et aussi les transactions de vente au détail et les transactions bancaires, etc.

Pour rendre possible le traitement des séquences catégoriques par la LSA, la méthode *N-Gram* est souvent utilisée pour transformer chaque séquence catégorique en un ensemble de mots dit « *bag of words* » [25, 95, 132]. Cette méthode a pour objectif de collecter l'ensemble de tous les motifs de longueur fixe N, qui est choisi par l'utilisateur, dans chaque séquence. Ces motifs sont ensuite utilisés pour construire la matrice d'occurrences qui décrit la présence ou l'absence de chaque motif dans chaque séquence. Les lignes de la matrice d'occurrences correspondent aux motifs et les colonnes correspondent aux séquences. Cette transformation des séquences rend possible l'utilisation de la LSA sur ce type de données. Toutefois, l'utilisation de la méthode *N-Gram* pour construire la matrice d'occurrences souffre d'un inconvénient majeur lié au choix de N par l'utilisateur [95]. Il s'agit de collecter des motifs présents par hasard dans les séquences si N est choisi trop petit, ou alors de rater des motifs importants présents dans les séquences si N est choisi trop grand. Par ailleurs, la méthode *N-Gram* a pour objectif de collecter tous les motifs sans exception, sans faire distinction entre les motifs importants ou ceux qui ne le sont pas, ce qui provoque souvent la présence de bruit excessif dans la matrice d'occurrences.

Dans ce chapitre nous présentons une nouvelle méthode, nommée SCS basée sur l'utilisation de l'analyse sémantique latente, pour mesurer la similarité entre les séquences catégoriques. SCS utilise d'abord un nouvel algorithme d'appariement des séquences catégoriques, inspiré principalement de l'algorithme d'appariement utilisé dans tSMS pour détecter les motifs qui représentent le mieux les caractéristiques structurelles et séquentielles des séquences catégoriques. Ensuite, la méthode *N-Gram* est appliquée sur l'ensemble des motifs ainsi collectés pour construire la matrice d'occurrences, plutôt que sur toute la longueur des séquences originales comme c'est le cas avec la méthode *N-Gram* conventionnelle. Ceci est

l'une des originalités importantes de notre méthode. Le fait que la méthode *N-Gram* soit appliquée que sur les motifs importants collectés permet d'éviter à SCS de collecter des motifs qui constituent juste du bruit ou alors d'exclure des motifs importants qui représentent des régions structurellement significatives. Ensuite, la LSA est appliquée sur la matrice d'occurrence obtenue pour calculer la similarité entre les séquences catégoriques. Nos résultats expérimentaux ont montré que SCS n'est pas seulement efficace sur les séquences de protéines mais aussi sur différentes variétés de séquences catégoriques. Ces résultats ont montré aussi clairement l'efficacité et la polyvalence de SCS dans différents domaines d'applications, tel que : *caractérisation des protéines, classification du langage naturel, catégorisation de la musique, détection des pourriels, prédition des faillites personnelles, reconnaissance de la voix*, etc.

La version préliminaire de SCS a été publiée à la conférence internationale « *IEEE International Conference on Data Mining* » en 2008. En 2010, une version plus détaillée et allongée a été publiée dans le journal « *Knowledge and Information Systems* », et c'est le papier que nous présentons dans ce chapitre. Ma contribution inclut le développement de SCS, l'exécution des tests expérimentaux, la rédaction des manuscrits. Mes superviseurs Dr. Shengrui Wang et Dr. Ryszard Brzezinski ont supervisé et validé tout le projet. Dr. Qingshan Jiang a analysé certains résultats. Voici donc les publications dont a fait l'objet SCS:

- Abdellali Kelil, Shengrui Wang. *SCS: A New Similarity Measure for Categorical Sequences*. The 8th IEEE International Conference on Data Mining, Pisa, Italy. December 15th-19th 2008.
- Abdellali Kelil, Shengrui Wang, Qingshan Jiang, Ryszard Brzezinski. *A general measure of similarity for categorical sequences*. Springer Journal of Knowledge and Information Systems, 24(2); 197-220; 2010.

A general measure of similarity for categorical sequences

Abdellali Kelil^{1,4}, Shengrui Wang^{1,4,*}, Qingshan Jiang^{2,+†}, and Ryszard Brzezinski^{3,4,*}

¹ ProspectUS Laboratory, Department of Computer Science

² School of Software, Xiamen University, Xiamen 361005, China

³ Microbiology and Biotechnology Laboratory, Department of Biology

⁴ Faculty of Sciences, University of Sherbrooke, Sherbrooke, QC J1H 3Z3 Canada

+ Supported by the National Natural Science Foundation of China

* Supported by research grants from the Natural Sciences and Engineering Research Council of Canada

Abdellali.Kelil@USherbrooke (first and corresponding author)

Shengrui.Wang@USherbrooke.ca

QJiang@xmu.edu.cn

Ryszard.Brzezinski@USherbrooke.ca

Abstract

Measuring the similarity between categorical sequences is a fundamental process in many data mining applications. A key issue is extracting and making use of significant features hidden behind the chronological and structural dependencies found in these sequences. Almost all existing algorithms designed to perform this task are based on the matching of patterns in chronological order, but such sequences often have similar structural features in chronologically different order.

In this paper we propose SCS, a novel, effective and domain-independent method for measuring the similarity between categorical sequences, based on an original pattern matching scheme that makes it possible to capture chronological and non-chronological dependencies. SCS captures significant patterns that represent the natural structure of sequences, and reduces the influence of those which are merely noise. It constitutes an effective approach to measuring the similarity between data in the form of categorical sequences, such as biological sequences, natural language texts, speech recognition data, certain types of network transactions, and retail transactions. To show its effectiveness, we have tested SCS extensively on a range of datasets from different application fields, and compared the results with those obtained by various mainstream algorithms. The results obtained show that SCS produces results that are often competitive with domain-specific similarity approaches.

Keywords: Categorical Sequences, Similarity Measure, Chronological Order, Matching, Significant Patterns

1. Introduction

Categorical sequences are data structured as strings of related or unrelated categories, for which both chronological order and structural features (i.e., subsequences characterizing the intrinsic sequential nature of related sequences) are informatively important. Many types of scientific and business data are in the form of categorical sequences: for instance, biological sequences, natural language texts, retail transactions, etc.

The similarity between categorical sequences is measured through the detection of chronological dependencies and structural features hidden within these sequences. This measure can lead to a better understanding of the nature of these sequences, in the context of their application fields. For instance:

- In biochemistry, each protein has its own unique linear chain made up of 20 possible amino acids, containing structural features, known as conserved domains, which precisely define its biochemical activity. Many different proteins are involved in the same biochemical activities, since they share similar structural features.
- In linguistics, despite the fact that each written work has its own unique sequence of words, structural features that reveal a certain literary style can be pinpointed, making it possible to identify the author, since each author marks his written work with some structural characteristics definitive of his own style.
- In finance, each credit card holder has his own spending behaviour, from which it is possible to extract some sequential factors describing his unique profile. From these sequential factors, it is possible to extract structural features that might predict customers who have a potential risk of bankruptcy.

In the past few years, with the emergence of research areas such as computational biology and text processing we have seen an increasing need to develop similarity measures that deal efficiently with categorical sequences. The most important known challenges presented by these data, which are only partially addressed by existing methods, are the following:

- It is difficult to extract the information underlying the chronological dependencies of structural features which may have significant meaning.
- Very often, categorical sequences are infected with significant quantities of noise. Unlike numerical sequences, for which we can filter out noise by applying signal processing techniques, categorical sequences require the use of a different, specific set of approaches to handle the non-dependency between the categories making up these data.
- The absence of a measurable similarity relation between the values of the different categories forming these data makes it difficult to measure the similarity between the categorical sequences.
- The high computational cost involved is also an important problem.
- Categorical sequences may include similar structural features with significant meaning in chronologically different positions. This has been ignored by almost all the existing approaches.

The literature reports a number of approaches to measuring the similarity between categorical sequences. One example is the very common Levenshtein distance (Levenshtein. 1966), usually named the “*Edit Distance*”, which is calculated by finding the minimum cost required to transform one sequence into another using “*insertion*”, “*deletion*” and “*replacement*” operations. Sequence alignment (Needleman *et al.* 1970) is another commonly used approach that finds the best matching for a pair of categorical sequences by inserting “*gaps*” in appropriate positions, so that the positions where identical or similar categories occur in the two sequences are aligned.

Both of these approaches have a major drawback due to the fact that they are based on matching of subsequences in chronological order. They break down when applied to sequences comprising similar structural features in chronologically different positions. For instance, protein sequences often have similar conserved domains located in non-equivalent positions when viewed in terms of primary structure, making them difficult to match in chronological order. However, these domains might well be in equivalent positions when viewed in terms of three-dimensional structure (Kelil *et al.* 2007a). Another drawback the two approaches share is that they yield similarity measures which depend heavily on the costs the user assigns to the “*insertion*”, “*deletion*” and “*replacement*” operations in the case of the edit distance, or the “*opening gap*” and “*extension gap*” costs in the case of sequence alignment. This creates ambiguities and complicates the similarity measurement task, especially for sequences of significantly different lengths.

The literature also reports the *N*-Gram approach (Suen. 1979) for measuring the similarity between categorical sequences. The *N*-Gram approach is popular for its speed and simplicity. The *N*-Gram are the set of all possible grams (i.e., patterns) of a fixed length *N* for which, with an *m*-letter alphabet, we obtain m^N possible patterns.

It is generally believed that in the *N*-Gram approach, the restriction to a fixed length *N* in collecting patterns from the sequences is a major drawback (Mhamdi *et al.* 2006). The value of *N* is set independently of the intrinsic structure of the sequences, as in the example of the *m*-letter alphabet, and the length of the sequences. Depending on the value of *N*, this results in either the collection of patterns representing noise or the exclusion of significant patterns. Moreover, all patterns of length *N* are collected, without distinguishing between significant and non-significant patterns, which increases the probability of collecting a number of motifs representing noise.

To the best of our knowledge, the literature does not report any approach that simultaneously addresses all of the challenges cited above. To rectify this shortcoming, in this paper we propose a new general similarity measure named SCS. Our new similarity measure allows us to extract hidden relations between categorical sequences, by capturing structural relations using global information extracted from a large number of sequences rather than merely comparing pairs of sequences. SCS detects and makes use of the significant patterns underlying the chronological dependencies of the structural features, filtering out noise by collecting the significant patterns that best represent the properties of categorical sequences and discarding those patterns that occur by chance and represent only noise. Moreover, SCS measures similarity in a way that more efficiently reflects the structural relationships between categorical sequences, with a worst-case computational cost that is linear with respect to sequence length. In addition, by utilizing an efficient subsequence matching scheme, SCS simultaneously handles the chronological and non-chronological order of the structural features. This allows it to deal with categorical sequences that include similar structural features with significant meaning in chronologically non-equivalent positions. Our experiments showed that the patterns used in SCS are more significant in terms of representing the natural structural features of categorical sequences and capturing chronological and non-chronological dependencies.

SCS constitutes an effective method for measuring the similarity of categorical sequences. To show this, we have tested it extensively on different data types and compared the results with those obtained by many existing mainstream approaches.

2. The new similarity measure SCS

By applying a new pairwise sequence matching scheme, SCS extracts from a set of categorical sequences a set of patterns with significant meaning, and filters out noise patterns. This is done by examining each pair of sequences for common identical patterns, as well as for patterns that are slightly different, known as “*paronyms*” and “*cognates*”. In natural language text, “*paronyms*” such as “*affect*” and “*effect*” are words that are related and derive from the same root, while “*cognates*” such as “*shirt*” and “*skirt*” are words that have a common origin. For a detailed review see (Horst. 1999). Taking identical patterns, “*paronyms*” and “*cognates*” into account improves the extraction of significant patterns.

After that, the *N*-Gram algorithm is applied directly on the set of extracted significant patterns, rather than on the original input categorical sequences. The categorical sequences are then mapped onto a new vector space of reduced dimension (Ganapathiraju *et al.* 2004), in which each categorical sequence is represented by a vector. Finally, the measure of the similarity between different sequences is computed simply by calculating the cosine product between the corresponding vectors. This idea is developed in the following sections.

2.1. The main idea of SCS

Very often, in natural language text processing (Berry *et al.* 1996), methods such as Latent Semantic Analysis (Song *et al.* 2009) are used to extract hidden relations between documents, by capturing semantic relations using global information extracted from a large number of documents rather than merely comparing pairs of documents. These methods usually make use of a word-document matrix $T(W \times L)$, in which rows correspond to words and columns correspond to documents, where W is the number of possible words and L is the number of documents. The term $T_{i,j}$ represents the occurrence of word i in document j . Although categorical sequences do not contain distinctive patterns like words in natural language text, categorical sequence data analysis is in many respects similar to natural language text analysis. However, the challenge is to be able to identify those patterns that map to a specific meaning in terms of sequence structure and to distinguish significant patterns from patterns resulting from random phenomena.

In much the same way that a word-document matrix is used in natural language text processing to extract the hidden relations between documents, we use a pattern-sequence matrix on the categorical sequences to extract the hidden relations between these sequences. This is done by capturing structural relations using global information extracted from a large number of sequences rather than merely comparing pairs of sequences. Henceforth, we use $T(W \times L)$ to denote the pattern-sequence matrix, in which the term $T_{i,j}$ represents the number of occurrences of pattern i in sequence j , while W is the number of possible patterns, and L is the number of sequences. The significant patterns used to construct T are detected and collected using the matching scheme described in the next subsection.

2.2. The matching scheme

2.2.1. Collection of significant patterns

In this work, a significant pattern is obtained from the matching of a pair of sequences. Let C be a set of categorical sequences, from which X and Y are a pair of sequences. Let x and y be a pair of subsequences belonging respectively to X and Y . Here, the symbol x or y is simply used as a variable, representing any subsequence belonging to the sequence X or Y . A significant pattern is a significantly long matched subsequence of the two sequences X and Y . In this paper, it is any sequence belonging to a matching set $E_{X,Y}$ that is built by collecting all the possible pairs of subsequences x and y that satisfy the following conditions:

$$E_{X,Y} = \left\{ \begin{array}{l} x \in X \\ y \in Y \\ \forall x', y' \in E_{X,Y} \Rightarrow (x \not\subset x') \vee (y \not\subset y') \end{array} \middle| \begin{array}{l} |x| = |y| \\ |x \cap y| > N_{X,Y} \\ |x \setminus y| < N_{X,Y} \end{array} \right\}$$

The symbols x' and y' in the formula are simply used as variables, in the same way as x and y . The expression $(. \not\subset .)$ means that the element to the left of the symbol $\not\subset$ is not included in the one to the right, either in terms of the composition of the patterns or in terms of their positions in their respective sequences. We use the parameter $N_{X,Y}$ to represent the minimum number of matched positions with similar categories between x and y ; at the same time, $N_{X,Y}$ is also used to represent the maximum number of matched positions with different categories allowed. A detailed discussion on the choice of $N_{X,Y}$ is provided in the next subsection. Here below are a few explanations about the previous formula:

- $|x| = |y|$: means that x and y have the same length.
- $|x \cap y| > N_{X,Y}$: means that x and y include more than $N_{X,Y}$ matched positions with similar categories.

- $|x \setminus y| < N_{X,Y}$ means that x and y include fewer than $N_{X,Y}$ matched positions with different categories.
- $\forall x', y' \in E_{X,Y} \Rightarrow (x \not\subset x') \vee (y \not\subset y')$: means that, for any pair of matched subsequences x' and y' belonging to $E_{X,Y}$, at least one of x and y is not included in x' or y' , respectively, either in terms of their compositions or in terms of their respective positions in their corresponding sequences, according to the partial order induced by set inclusion.

By looking for similar patterns in X and Y , the aim of the matching set $E_{X,Y}$ is to capture information shared by X and Y , related to certain chronological dependencies in their structural features. At the same time, by taking into account multiple occurrences of patterns in non-equivalent positions, the matching set $E_{X,Y}$ seeks to capture the structural features in non-chronological order. In fact, with this formula, $E_{X,Y}$ captures pairs of patterns x and y that show a “*within*” chronological similarity, even if they are in non-chronological order from the standpoint of their respective positions within the sequences X and Y .

As an example of the matching scheme, let X , Y , and Z be three categorical sequences for which we want to measure the pairwise similarity, as illustrated in Figure 1. In this example, by assuming that $N_{X,Y} = N_{X,Z} = N_{Y,Z} = 3$, the matching scheme will match, from each pair of sequences, the pairs of similar patterns with length >3 that also contain a number of mismatches <3 (specified by the matching condition). The pairs of patterns satisfying the matching condition are collected, as shown in Table 1.

x_1	x_2
x_1	x_2
y_1	y_2
z_1	z_2
x	TMSITADSLAVVRV TMMITEDF QTDTGHPI
y	MSTSYITMGITCD TGHPGSGGLRQ TMRITED
z	QMTMGATEDDRVSLAVQHSPTL RITADAMNR

Figure 1. Example showing matching of categorical sequences
Shaded patterns correspond to matched significant patterns

Table 1. Pairwise matching scheme

Letters in bold correspond to mismatches – Shaded patterns correspond to collected pairs of patterns

Pairwise Matching		X		Y		Z	
		x_1	x_2	y_1	y_2	z_1	z_2
X	x_1	-	-	TMGITCD TMSITAD	TMRITED TMSITAD	TMGATED TMSITAD	TLRITAD TMSITAD
	x_2	-	-	TMGITCD TMMITED	TMRITED TMMITED	TMGATED TMMITED	TLRITAD TMMITED
Y	y_1	TMSITAD TMGITCD	TMMITED TMGITCD	-	-	TMGATED TMGITCD	TLRITAD TMGITCD
	y_2	TMSITAD TMRITED	TMMITED TMRITED	-	-	TMGATED TMRITED	TLRITAD TMRITED
Z	z_1	TMSITAD TMGATED	TMMITED TMGATED	TMGITCD TMGATED	TMRITED TMGATED	-	-
	z_2	TMSITAD TLRITAD	TMMITED TLRITAD	TMGITCD TLRITAD	TMRITED TLRITAD	-	-

2.2.2. Minimum length of significant patterns

An important feature of our similarity measure is the determination of the length of patterns to be considered significant. It is well known that the longer the patterns, the smaller the chance of their being identical by chance, and vice versa. In fact, according to Karlin's theorem 1 in (Karlin *et al.* 1985), the expected length $K_{R,L}$ of the longest common pattern present by chance at least $R\%$ times out of L m -category sequences S_1, S_2, \dots, S_L , is calculated as follows:

$$K_{R,L} = \frac{\log n(|S_1|, \dots, |S_L|) + \log \lambda(1 - \lambda) + 0.577}{-\log \lambda}$$

$$n(|S_1|, \dots, |S_L|) = \sum_{1 \leq i_1 \leq \dots \leq i_R \leq L} \prod_{v=1}^R |S_{i_v}|$$

$$\lambda = \max_{1 \leq i_1 \leq \dots \leq i_R \leq L} \left(\prod_{l=1}^m \prod_{j=1}^R p_i^{(v_j)} \right)$$

$$\sigma \approx \frac{1.283}{|\log \lambda|}$$

Where $p_i^{(v_j)}$ is generally specified as the i^{th} category frequency of the observed v^{th} sequence, while σ is the asymptotic standard deviation of $K_{R,L}$.

For a given set of categorical sequences for which we want to detect and extract the significant patterns, we make use of Karlin's theorem on each pair of matched sequences X and Y to estimate one specific and appropriate value for the minimum length of significant patterns $N_{X,Y}$ for the pair. A conventional way of using the theorem would be to estimate, for the whole set of sequences, a single value for the minimum length of significant patterns. Applying Karlin's theorem to each pair of sequences tends to yield a large value for $N_{X,Y}$ and consequently reduces the chance of collecting patterns occurring by chance. Figure 2 gives a typical example to illustrate this idea.

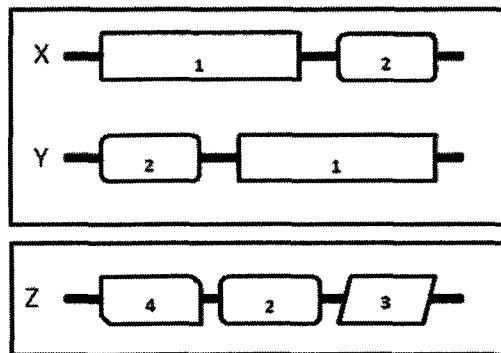


Figure 2. Application of Karlin's theorem

First, let us take a pair of related categorical sequences X and Y , from which we want to extract the significant patterns. Within these sequences, let us suppose that pattern 2 occurs by chance and pattern 1 has a significant meaning in terms of the structures of X and Y . Applying Karlin's theorem, we likely get the longest common pattern present by chance equal to the length of pattern 2. By matching X and Y , this threshold allows us to select pattern 1 as significant and filter out pattern 2. Now, let us consider the case where there is an additional sequence Z that is unrelated to X and Y . In this case, because of the

fact that X and Y are less similar to Z , Karlin's theorem will lead to a shorter length threshold, significantly increasing the risk of considering pattern 2 as significant for X and Y . This is why we have chosen to adopt a pairwise application of Karlin's theorem. In this example, pairwise matching of X , Y , and Z highlights the pattern with significant meaning shared by X and Y ; at the same time, considering sequence Z within the set of sequences does not disturb the relation between sequences X and Y .

However, due to the fact that Z is unrelated to X and Y , Karlin's theorem will yield a shorter length threshold for the pairs X with Z , and Y with Z , than for the pair X with Y , which increases the risk of considering pattern 2 as significant for the pairs X with Z , and Y with Z . But even if this happens, the similarity between the pair of related sequences X with Y is more significant than the similarity between other unrelated pairs.

By reformulating the theorem proposed by Karlin we can thus say that the expected length $K_{2,2}$ of the longest common pattern present by chance at least 2 times (i.e., $R=2$) out of 2 m -category sequences (i.e., $L=2$), X and Y is calculated as follows:

$$K_{2,2} = \frac{\log(|X|^2 + |Y|^2) + \log \lambda(1 - \lambda) + 0.57}{-\log \lambda}$$

$$\lambda = \max \left(\sum_{i=1}^m (p_i^X)^2, \sum_{i=1}^m (p_i^Y)^2 \right)$$

Where p_i^X and p_i^Y are generally the i^{th} category frequency of the observed sequences X and Y respectively. For each pair of compared sequences X and Y , we use these formulas to calculate the minimum length of matched significant patterns, which is the value to be assigned to $N_{X,Y}$.

According to the conservative criterion proposed by Karlin we can say that, for a pair of categorical sequences X and Y , a pattern observed 2 times is designated statistically significant if it has a length that exceeds $K_{2,2}$ by at least two standard deviations. Thus, in building the matching set $E_{X,Y}$, we extract all the common patterns that satisfy this criterion. This means that, for the pair of sequences X and Y , we calculate a specific and appropriate value of $N_{X,Y} = K_{2,2} + 2\sigma$. In practice, $N_{X,Y}$ is rounded to the largest integer inferior to $K_{2,2} + 2\sigma$. This criterion guarantees that a matched pattern designated as statistically significant (i.e., a pattern that maps to a specific meaning in terms of sequence structure) has less than a 1% probability of occurring by chance.

2.3. Application of the N -Gram algorithm

2.3.1. The pattern-sequence matrix

Let C be a set of categorical sequences. Let X and Y be two different sequences of C , $N_{X,Y}$ the minimum length of the significant patterns, and $E_{X,Y}$ the set of collected pairs of significant patterns. Let E be the set of all possible matching sets, and N_{min} the minimum value that $N_{X,Y}$ can have, such that:

$$E = \bigcup_{X,Y \in C} E_{X,Y}$$

$$N_{min} = \min_{X,Y \in C} N_{X,Y} + 1$$

To compute an initial pattern-sequence matrix T , we collect and order all the N_{min} grams from each significant pattern included in E . For a set of sequences made up of m categories, there could be as many as $m^{N_{min}}$ possible N_{min} grams, alt-

hough the number is much smaller in practice. Let E_X be the set of all possible matching sets involving the categorical sequence X (i.e., all the significant patterns that are parts of X), such that:

$$E_X = \bigcup_{Y \in C} E_{X,Y}$$

In Table 2, we show as an example the composition of E_X , E_Y , and E_Z , the sets of all possible matching sets involving respectively the categorical sequences X , Y , and Z shown as an example in Figure 1, for which the collected significant patterns are shown in Table 1.

Table 2. Sets E_X , E_Y , and E_Z involving sequences X , Y , and Z , respectively

Set	Composition
$E_X = E_{X,Y} \cup E_{X,Z}$	$\{(x_1, y_1), (x_1, y_2), (x_1, z_2), (x_2, y_1), (x_2, y_2), (x_2, z_1)\}$
$E_Y = E_{Y,X} \cup E_{Y,Z}$	$\{(y_1, x_1), (y_1, x_2), (y_1, z_1), (y_2, x_1), (y_2, x_2), (y_2, z_2)\}$
$E_Z = E_{Z,X} \cup E_{Z,Y}$	$\{(z_1, x_2), (z_1, y_1), (z_2, x_1), (z_2, y_2)\}$

The initial value of the term $T_{i,X}$ is defined as the number of occurrences of the i^{th} N_{min} gram belonging to the subset E_X . The final matrix T is obtained by removing the rows with only zero elements. In other words, we discard those N_{min} grams that are specific to only one sequence. In our experiments, we found that the number of remaining rows W is much smaller than $m^{N_{min}}$ (i.e., $\ll m^{N_{min}}$). This property is very important for the next section.

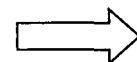
The most important advantage with this new approach is that each sequence in the set of sequences contributes to the capture of the structural features and chronological dependencies of all other sequences in the set. And the more frequently a pattern occurs in the sequences, the more heavily it is represented in the pattern-sequence matrix T . Moreover, the matrix T

Table 4. Collected N_{min} -grams

Sequences	Patterns	N_{min} grams	E_X	E_Y	E_Z
X	x_1 TMSITAD	TMSI	3	2	2
		MSIT	3	2	2
	x_2 TMMITED	SITA	3	2	2
		ITAD	3	2	2
Y	y_1 TMGITCD	TMMI	3	2	2
		MMIT	3	2	2
	y_2 TMRITFD	MITE	3	2	2
		ITED	3	2	2
Z	z_1 TMGATED	TMGI	2	3	2
		MGIT	2	3	2
	z_2 TLRITAD	GITC	2	3	2
		ITCD	2	3	2

Table 3. The pattern-sequence matrix

Sequences	X	Y	Z
ATED	1	1	2
GATE	1	1	2
GITC	2	3	2
ITAD	4	3	4
ITCD	2	3	2
ITED	3	2	2
ITFD	2	3	2
LRIT	1	1	2
MGAT	1	1	2
MGIT	2	3	2
MITE	3	2	2
MMIT	3	2	2
MRIT	2	3	2
MSIT	3	2	2
RITA	1	1	2
RTTF	2	3	2
SITA	3	2	2
TLRI	1	1	2
TMGA	1	1	2
TMGI	2	3	2
TMMI	3	2	2
TMRI	2	3	2
TMSI	3	2	2



is filled by using only the grams corresponding to the significant patterns collected, and not all the possible patterns from the original input categorical sequences as in the classical N -Gram approach.

As an example, we build the pattern-sequence matrix of the significant patterns detected in the categorical sequences X , Y , and Z given in Figure 1, for which the collected significant patterns are shown in Table 1. In this example, the minimum lengths of significant patterns $N_{X,Y} = N_{X,Z} = N_{Y,Z} = 3$, and thus $N_{min} = 3 + 1 = 4$. Table 4 shows the collected N_{min} grams from each detected significant pattern from each categorical sequence. In this table, we also show the number of occurrences of the N_{min} grams in each of the sets E_X , E_Y , and E_Z . After merging the rows corresponding to identical N_{min} grams, we obtain the pattern-sequence matrix shown in Table 3.

2.3.2. Range of possible values of N_{min}

Because of the risk that the value of N_{min} may be too small or too large (i.e., problems of feasibility), we evaluated the possible values of N_{min} for a variety of categorical sequence datasets from different research fields. We used two widely known collections of well-characterized protein sequences, the COG and KOG databases (Tatusov *et al.* 2003), both in their 9 November 2008 versions, comprising 192,987 proteins from unicellular organisms and 112,920 proteins from eukaryotic organisms, respectively. We also used the Reuters-21578 text categorization test collection, one of the most widely used test collections for text categorization research, which comprises 21,578 manually categorized articles that appeared on the Reuters newswire in 1987. We also used the 1500 categorical sequences generated via a bio-inspired processing from an in-house speech database used in (Loiselle *et al.* 2005), including a number of recorded human voices made up of isolated French letters and numbers. More details about the datasets are provided in the Experiments section. For each dataset, we computed, for all possible pairs of categorical sequences X and Y , the length of collected significant patterns $N_{X,Y}$. Table 5 and Figure 3 present some statistics on the results obtained with the different datasets.

Table 5 shows, for each dataset, the number of sequences (Nbr) included, the average length of the sequences (AV length), the value of N_{min} obtained, and also the average (AV $N_{X,Y}$) and the standard deviation (SD $N_{X,Y}$) of the value of $N_{X,Y}$. In Figure 3, we also show the percentage of the possible pairs of matched sequences with the same minimum length of significant patterns $N_{X,Y}$.

Table 5. The value of N_{min} with different categorical sequence datasets

Dataset	Nbr	AV Length	N_{min}	AV $N_{X,Y}$	SD $N_{X,Y}$
COG	192,987	305	3	5.4	3.2×10^{-7}
KOG	112,920	420	3	5.5	5.6×10^{-6}
Reuters	21,578	17,083	3	5.2	3.2×10^{-6}
Speech	1,310	1,480	6	7.9	8.2×10^{-5}

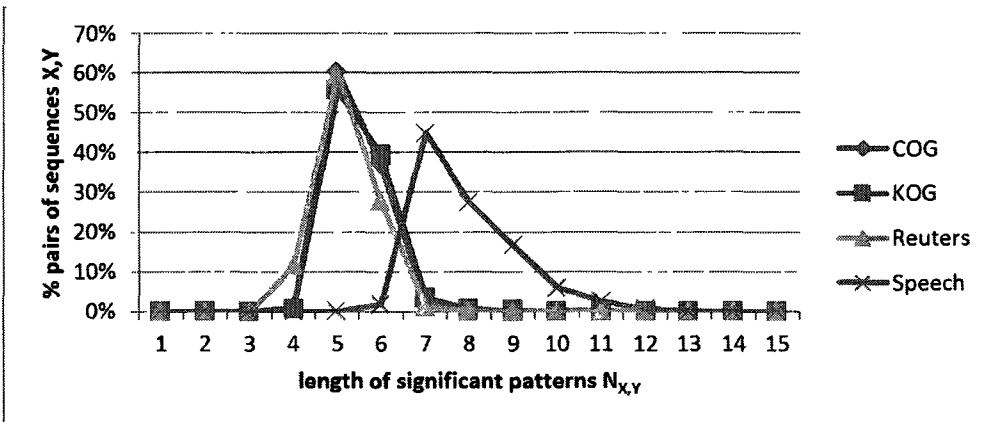


Figure 3. Percentage of possible pairs of sequences by length of significant patterns

From Table 5 we see that for all the datasets we obtained N_{min} values which, in the worst case, allow the building of pattern-matrices with practicable dimensions. The worst case occurs when the pattern-sequence matrix has $m^{N_{min}}$ rows (i.e., maximum number of possible patterns) with m possible categories. The average value of $N_{X,Y}$ (AV $N_{X,Y}$) and its standard deviation SD $N_{X,Y}$, show that the variation in the length of collected significant patterns is rather small. This is confirmed by the results shown in Figure 3, where we can see that for all datasets, all values obtained for the minimum length of significant patterns $N_{X,Y}$ are comprised within a relatively small restricted range of values.

2.4. The similarity measure

In the pattern-sequence matrix T , each sequence is expressed as a column vector and each pattern as a row vector. This representation is known as a vector space model. Represented in this way, the sequences are seen as points in the multidimensional space spanned by patterns. However, this representation does not recognize related patterns or sequences, and the dimensions are too large (Ganapathiraju *et al.* 2004). To take advantage of this representation, we perform a singular value decomposition (SVD) on the pattern-sequence matrix T . Let $L = |C|$ and R be the total ranks of T . Thus the matrix T can be decomposed into the product of three matrices, as follows:

$$T = U \times \Sigma \times V^T$$

where U is a $W \times R$ left singular matrix, Σ is an $R \times R$ diagonal matrix of positive singular values, and V is an $L \times R$ right singular matrix. By taking into account only the R' (where $R' \ll R$) largest singular values from the matrix Σ (the choice of R' is discussed in (Berry *et al.* 1996)), and their corresponding singular vectors from the matrices U and V , we get the matrix T' , the rank R' approximation of T with the smallest error according to the Frobenius norm (Golub *et al.* 1996). Thus, the matrices U , Σ and V are reduced to the $W \times R'$ matrix U' , the $R' \times R'$ matrix Σ' and the $L \times R'$ matrix V' , respectively, such that:

$$T \approx T' = U' \times \Sigma' \times V'^T$$

Utilizing the singular value decomposition theory (Berry *et al.* 1996), the sequences expressed as column vectors in the matrix T' are projected via spectral decomposition onto a new multidimensional space spanned by the column vectors of

the matrix V'^T with reduced dimension $R' \ll R$. The only parts of U' , Σ' , and V'^T that contribute to the value of the i^{th} column of T' are the whole matrix Σ' and $V'_{:,i}^T$ the i^{th} column of the matrix V'^T . Thus, the representation of the sequences in the new R' -dimension space corresponds to the column vectors of the $R' \times L$ matrix $\Sigma' \times V'^T$.

Finally, the similarity measure $S_{X,Y}$ for the pair of sequences X and Y is simply computed by calculating the cosine product of their corresponding column vectors on the $R' \times L$ matrix resulting from the product $\Sigma' \times V'^T$.

The most important advantage of the strategy of transforming the pattern-sequence matrix by spectral decomposition into a new vector representation is that, the similarity measure between different categorical sequences can be computed in the new space using global information extracted from the whole set of sequences rather than merely comparing pairs of sequences. This advantage is made possible by the spectral decomposition that transforms each column-vector in the pattern-sequence matrix into a vector in the new multidimensional space by using the whole set of sequences which gives a global scope to the similarity measure between different vectors.

As an example, we compute the pairwise similarity measures of the categorical sequences X , Y , and Z shown in Figure 1. First, we perform singular value decomposition of the pattern-sequence matrix shown in Table 3. Next, we compute the similarity measures for each pair of sequences by using the cosine product of the corresponding column vectors of the resulting product matrix $\Sigma \times V^T$ (i.e., we choose $R' = R$). The computed pairwise similarity measures are shown in Table 6.

Table 6. Example of similarity measures

Similarity	Measure
$S_{X,Y}$	0.984
$S_{X,Z}$	0.914
$S_{Y,Z}$	0.929

2.5. Time complexity of SCS

At the stage of collecting the significant patterns, we made use of the fast string matching approach developed by (Amir *et al.* 2004), which allows us to find all the locations of any pattern from a sequence X in a sequence Y in time $O(|Y| \sqrt{N_{X,Y} \log N_{X,Y}})$. For the singular value decomposition, we utilized the fast, incremental, low-memory and large-matrix SVD algorithm recently developed by (Brand. 2006), which performs the SVD for a R rank matrix $W \times L$ in $O(WLR)$ time with $R \leq \sqrt{\min(W,L)}$.

3. Experiments

To evaluate the theoretical time complexity of SCS experimentally, and to compare its time efficiency to that of existing approaches, we executed SCS on a selection of four subsets of related categorical sequences. Each of the four subsets includes a large number of related sequences of average length 10^2 , 10^3 , 10^4 , and 10^5 , respectively. Then, we compared the execution time obtained with those yielded by a variety of mainstream similarity measure approaches, including those introduced by (Kohonen. 1985), (Kondrak. 2005), (Oh *et al.* 2004), (Cai *et al.* 2004), and (Li *et al.* 2007). More details on these approaches are provided in the rest of this section. We report the different execution times for each approach with each subset in Figure 4.

Figure 4 shows that the execution times obtained by SCS confirm the theoretical time complexity of SCS presented above. From a practical point of view, we see that among the approaches tested, the one developed by (Kondrak. 2005) obtains the fastest execution time, while that developed by (Li *et al.* 2007) obtains the slowest one. This test shows that

efficiency is not the main strength of SCS, even though it obtains a relatively good execution time. However, as we will see in the experiment section, SCS outclasses all of these approaches in terms of effectiveness.

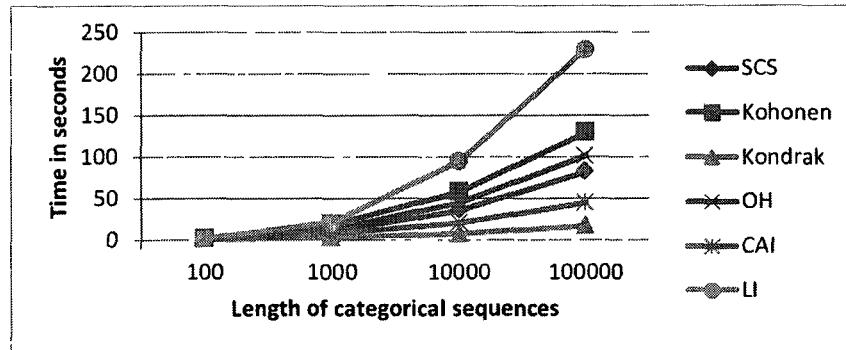


Figure 4. Execution times obtained with different approaches

To illustrate the effectiveness of our new similarity measure approach, we have tested SCS extensively on a variety of datasets from different research fields and compared the results with those obtained by several domain-specific mainstream algorithms. In all our experiments, we used these algorithms with their default input parameters. These experiments include tests of SCS on categorical sequences generated from speech data to assess its ability to recognize spoken words and speakers, comparing the results with those obtained by several mainstream algorithms designed to deal with categorical sequences. We also tested SCS more extensively on different protein databases, and compared the results with those of algorithms designed specifically to deal with such data. The aim of the protein data experiments was to illustrate the effectiveness of the general SCS approach in identifying protein sequences according to their functional annotations and biological classifications. Finally, to evaluate its ability to identify related natural language texts, we also tested SCS on the entire Reuters-21578 text categorization test collection, and compared the results with those obtained by algorithms specifically designed to deal with texts and documents.

To evaluate and compare similarity measures, we need an objective quality index. Given that our experiments are performed on data with known classes, it is possible to make use of the well-known ROC Curve approach. Intuitively, a good similarity measure should result in a high similarity value for two sequences belonging to the same class and a low similarity value for those belonging to different classes. For a given sequence X , its class is considered as the positive class and all the other classes together are considered as the negative class. If all the data are classified according to their similarity to X , then an ROC curve can be built. The area under the ROC curve can be used as a quality index of the corresponding similarity measure with respect to the data point X and its class. Obviously, the larger the area under the ROC curve is, the greater the discriminative power of the similarity measure approach. Our quality index is defined as the average area under the ROC curve with respect to each data point X . Again, a larger quality index value means greater discriminative power.

3.1. Speech recognition

Speech recognition is the technology that allows computers to automatically identify who says what, by converting the human voice to a type of data much easier to comprehend and analyze using computers. Our aim in making use of these data is to show the effectiveness of SCS on the categorical sequences produced especially for speech recognition. The speech data used in this section come from the in-house speech database used in (Loiselle *et al.* 2005), made up of isolated French letters (i.e., vowels: “*a*”, “*e*”, “*i*”, “*o*”, “*u*”) and numbers (i.e., “*1*”, …, “*9*”) spoken by 5 men and 5 women, with each symbol

pronounced 10 times by each speaker. Each recorded speech was used to produce a sequence made up of 20 different events, based on a bio-inspired processing approach (Loiselle *et al.* 2005). Each pronunciation is thus represented by a categorical sequence made up of 20 different categories. The details about the average lengths of the sequences produced for each letter and number by each speaker are shown in Table 7. The first row contains the list of the different speakers; the symbol “*M*” designates “*male*” and “*F*” designates “*female*”. The first column contains the pronounced letters and numbers. The rest of the table contains the average lengths of the sequences produced for each letter and number by each speaker. The produced sequences can be classified either by speakers (i.e., 10 classes) or by words (i.e., 14 classes). In this experiment, we computed the quality index of the results obtained for all classes.

Table 7. Average length of produced sequences

	M1	M2	M3	M4	M5	F1	F2	F3	F4	F5
“a”	822	923	1245	722	892	1006	1187	2280	1750	1539
“e”	799	1008	883	1921	690	1047	2195	1773	1994	1560
“i”	330	612	578	1361	245	605	1532	1469	1705	804
“o”	335	414	1157	2056	579	503	2925	599	794	749
“u”	512	543	757	1285	447	523	1652	1365	1606	785
“1”	1372	1368	1393	1598	1292	1502	1377	1461	1358	1501
“2”	1201	1020	1252	970	1134	841	1257	994	1227	930
“3”	1306	1118	1378	1216	1274	1306	1413	1115	1404	1227
“4”	1402	1470	1336	1672	1387	1533	1351	1652	1465	1616
“5”	2032	1935	2006	2327	2132	2051	2114	2021	1958	1964
“6”	2036	1991	1974	2293	1950	2227	2173	2255	2030	2211
“7”	1584	1359	1569	1589	1384	1485	1731	1490	1619	1314
“8”	992	999	1200	1206	1089	1050	1156	1102	1128	1177
“9”	1481	1525	1608	1580	1422	1497	1542	1627	1523	1480

We compared the results obtained using SCS with those yielded by several mainstream approaches. The comparison approaches were the one proposed by (Kohonen. 1985), based on the set median that has the smallest sum of distances from the other elements; the one proposed by (Kondrak. 2005), based on the *N*-Gram approach with a predefined value of *N*; the one proposed by (Oh *et al.* 2004), based on a matching scheme that takes into account the non-chronological order of matched subsequences; the one proposed by (Cai *et al.* 2004), based on the longest common subsequences similarity model; and the one proposed by (Li *et al.* 2007), based on sequence alignment.

In Table 8 and Table 9 we summarize the results obtained by each algorithm. Each table shows the quality index obtained by each approach (i.e., column) for each subset of sequences belonging to the same class (i.e., row). The last row in each table contains the average quality index obtained by each approach. In Table 8, words are used as known classifications, while in Table 9, speakers are used as known classifications. In Table 8 and Table 9 we can see that our approach obtained the best quality indices for both types of classifications, by words as well as by speakers.

We conclude that SCS is able to effectively recognize related categorical sequences generated from the pronounced letters and numbers, whether categorized by pronounced words or speakers, and does so better than the other approaches. We strongly believe that this is because SCS more effectively highlights the significant unseen information behind the chronological dependencies and structural features within these sequences, thanks to its detection and use of the significant patterns that best represent the natural structure of these sequences, thereby minimizing the influence of those patterns that occur by chance and represent only noise. In addition, the matching technique, which allows us to simultaneously handle the “*within*” chronological order and the “*between*” non-chronological order of the structural features, also plays an important role in reaching these conclusive results.

Table 8. Quality index with words as classes

	SCS	KOH	KON	OH	CAI	LI
“a”	0.92	0.78	0.78	0.75	0.84	0.74
“e”	0.95	0.74	0.85	0.84	0.82	0.76
“i”	0.97	0.71	0.84	0.75	0.81	0.80
“o”	0.94	0.88	0.88	0.82	0.81	0.74
“u”	0.99	0.79	0.87	0.85	0.78	0.74
“l”	0.97	0.85	0.78	0.80	0.74	0.82
“2”	0.99	0.82	0.90	0.75	0.85	0.73
“3”	0.94	0.83	0.82	0.85	0.82	0.70
“4”	0.92	0.82	0.85	0.75	0.75	0.75
“5”	0.91	0.85	0.76	0.83	0.81	0.80
“6”	0.99	0.79	0.77	0.80	0.85	0.76
“7”	0.97	0.77	0.78	0.84	0.73	0.71
“8”	0.96	0.89	0.80	0.85	0.81	0.82
“9”	0.96	0.79	0.75	0.81	0.78	0.70
Av.	0.96	0.81	0.82	0.81	0.80	0.76

Table 9. Quality index with speakers as classes

	SCS	KOH	KON	OH	CAI	LI
M1	0.93	0.83	0.84	0.80	0.77	0.78
M2	0.90	0.83	0.81	0.75	0.82	0.76
M3	0.95	0.80	0.83	0.76	0.83	0.79
M4	0.98	0.79	0.83	0.73	0.81	0.68
M5	0.92	0.75	0.81	0.77	0.75	0.72
F1	0.95	0.74	0.86	0.78	0.71	0.78
F2	0.96	0.81	0.86	0.77	0.83	0.71
F3	0.96	0.81	0.86	0.77	0.82	0.75
F4	0.98	0.81	0.78	0.76	0.73	0.80
F5	0.95	0.81	0.75	0.81	0.82	0.80
Av.	0.95	0.80	0.82	0.77	0.79	0.76

3.2. Protein sequences

In biochemistry, a protein sequence is a linear chain made up of 20 possible amino acids. Thus, a protein is a categorical sequence made up of 20 possible categories. An important open problem in computational biology is to automatically predict the biochemical activity of a newly sequenced or not yet characterized protein sequence. To achieve this, biologists often compare the non-characterized protein sequence to those that are biochemically well-characterized, and assign to this protein the biochemical activity of the most similar proteins.

In this experiment, we applied SCS to predict the biochemical activities of protein sequences. We tested SCS on a variety of protein datasets and compared the results with those obtained by different mainstream algorithms designed specifically to deal with protein sequences. For instance, we considered SMS, introduced by (Kelil *et al.* 2007b) based on a strict matching scheme that captures the most significant patterns in chronological and non-chronological order; tSMS, introduced by (Kelil *et al.* 2008), which is an improved version of SMS that allows mismatches; one of the most commonly used bioinformatics programs, Blast, introduced by (Altschul *et al.* 1990) based on the local sequence alignment; the approach introduced by (Wu *et al.* 2003) based on short patterns used analogously to the index terms in information retrieval; and the one introduced by (Bogdan-Marta *et al.* 2005) based on the cross-entropy measure applied over the collected *N*-Gram patterns with a fixed value of *N*. Below, we report the results obtained for the different datasets, with support from the literature and functional annotations.

To demonstrate the effectiveness of SCS in measuring the similarity between protein sequences according to their functional annotations and biological classifications, we have performed extensive tests on the widely known databases COG and KOG (Tatusov *et al.* 2003), and PC (i.e., from the NCBI website). The COG and KOG databases include a classification of proteins encoded in complete genomes. COG and KOG contain 192,987 and 112,920 well-classified protein sequences, respectively. The PC database is a compilation of proteins from the complete genomes of different organisms that have been grouped and manually classified and annotated based on sequence similarity and protein function.

To perform a biological and statistical evaluation of our new similarity measure, we used the three ensembles of randomly generated datasets from (Kelim *et al.* 2008): C1 to C6 generated from the COG database, containing respectively 509, 448, 546, 355, 508 and 509 protein sequences; K1 to K6 generated from the KOG database, containing respectively 317, 419, 383, 458, 480 and 388 protein sequences; and finally P1 to P6 generated from the PC database, containing respectively 538, 392, 442, 595, 561 and 427 protein sequences. Each generated subset includes protein sequences with at least 20 biochemical activities, within which each biochemical activity defines a particular class of proteins.

In Table 10, Table 11 and Table 12 we summarize the results obtained by each algorithm on each subset. Each table shows the average quality index obtained by each algorithm (i.e., column) for each subset of protein sequences (i.e., row). The last row in each table contains the global average quality index obtained by each algorithm. The results given in Table 10, Table 11 and Table 12 show that tSMS obtains the best similarity measures over all generated subsets. The results obtained with tSMS are closely followed by those of SCS and SMS, while Wu and Bogan obtained less good results. A bit farther behind we find Blast, which obtains the poorest results.

Table 10. Average quality index on COG

	SCS	tSMS	SMS	Blast	Wu	Bogan
C1	0.96	0.97	0.93	0.70	0.78	0.84
C2	0.95	0.96	0.95	0.61	0.84	0.88
C3	0.91	0.98	0.95	0.77	0.88	0.82
C4	0.93	0.98	0.89	0.74	0.77	0.82
C5	0.92	0.95	0.93	0.60	0.81	0.84
C6	0.94	0.97	0.95	0.68	0.77	0.86
Av.	0.94	0.97	0.93	0.68	0.81	0.84

Table 11. Average quality index on KOG

	SCS	tSMS	SMS	Blast	Wu	Bogan
K1	0.91	0.92	0.91	0.65	0.68	0.66
K2	0.91	0.94	0.91	0.55	0.67	0.71
K3	0.92	0.96	0.93	0.58	0.74	0.69
K4	0.86	0.92	0.86	0.54	0.62	0.61
K5	0.88	0.94	0.84	0.70	0.68	0.71
K6	0.88	0.91	0.84	0.75	0.58	0.69
Av.	0.89	0.93	0.88	0.63	0.66	0.68

Table 12. Average quality index on PC

	SCS	tSMS	SMS	Blast	Wu	Bogan
P1	0.94	0.96	0.93	0.78	0.81	0.76
P2	0.95	0.98	0.92	0.76	0.90	0.79
P3	0.93	0.95	0.94	0.62	0.68	0.83
P4	0.94	0.95	0.91	0.79	0.80	0.80
P5	0.93	0.95	0.92	0.73	0.79	0.78
P6	0.91	0.98	0.94	0.80	0.87	0.93
Av.	0.93	0.96	0.93	0.75	0.81	0.82

Although it is not designed especially to handle protein sequences (i.e. it does not take into account the substitution relations between different amino acids), the results yielded by our new approach SCS are very close in quality to the best results obtained by tSMS. Furthermore, the results obtained by SCS are comparable to those of SMS, and much better than those obtained by Blast, Wu, and Bogan. This performance is especially remarkable if we consider that tSMS and SMS need a substitution matrix as input parameter in order to decide which amino acids should be matched and compute the weights of the significant patterns. In our experiments, the results obtained by tSMS and SMS were made possible by the use of the substitution matrix that maximizes the quality index for each test. This means that one needs prior knowledge about the classes of the protein sequences in order to choose the appropriate matrix for tSMS and SMS. This is precisely the reason for proposing SCS in this paper: SCS is a general measure that does not depend on the use of a substitution matrix.

3.3. Texts and documents

Measuring the similarity between two texts or documents is a fundamental process in many areas in natural language processing, such as text classification and information retrieval. The key issue is to measure this similarity without explicit knowledge of the statistical nature of these texts. The literature reports a number of approaches developed to measure the similarity between texts and documents. Some of the most recent examples are the one introduced by (Chim *et al.* 2007) based on a suffix tree document model, the one introduced by (Wan. 2007) based on the earth mover's distance, and the one introduced by (Aslam *et al.* 2003) based on an information-theoretic approach. These different approaches have demonstrated their ability to measure the similarity between natural language texts effectively. For this reason, and in the aim of evaluating the performance of our new similarity measure, we decided to perform extensive tests to compare the results obtained by SCS to those of the approaches cited above.

To effectively evaluate the performance of our new approach, we tested SCS on the entire Reuters-21578 text categorization test collection, the most widely used test collection for text categorization research. It comprises 21,578 articles which appeared on the Reuters newswire in 1987. Each article was manually indexed (i.e., classified) according to which categories, from which sets, it belonged to. The category sets (i.e., classes) are as follows: Exchanges (39 classes), Orgs (56 classes), People (267 classes), Places (175 classes) and Topics (135 classes). To make these articles accessible to SCS, they were transformed into categorical sequences by withdrawing spaces and newline marks. This pre-processing concerns only SCS, since the other tested algorithms are designed to handle texts, phrases and words as they are. In this experiment, we computed all quality indices for all Reuters-21578 categories (i.e., classes).

In Table 13 we summarize the results obtained by each algorithm on each of the category sets. The table shows the quality index obtained by each approach (i.e., column) for each subset of articles belonging to the same category (i.e., row). The last row contains the average quality index obtained by each approach. The results summarized in Table 13 show that the approach introduced by (Chim *et al.* 2007) obtains the best quality indices over all category sets, followed relatively closely by SCS, while the approaches developed by (Wan. 2007) and (Aslam *et al.* 2003) obtain less good results.

Table 13. Quality index with Reuters-21578 categories

	SCS	CHIM	WAN	ASLAM
Exchange	0.77	0.83	0.58	0.64
Orgs	0.67	0.82	0.43	0.62
People	0.72	0.75	0.45	0.53
Places	0.75	0.78	0.51	0.58
Topics	0.80	0.85	0.59	0.71
Av.	0.74	0.81	0.51	0.62

In this experiment, despite the fact that the approaches of (Chim *et al.* 2007), (Wan. 2007) and (Aslam *et al.* 2003) were all designed especially to handle natural language texts by taking into account the semantic concepts underlying words and phrases, and despite the fact that the data used in this experiment were transformed by withdrawing spaces and newline marks to make them accessible to SCS, the results yielded by our new approach are very close in quality to the best obtained results, in comparison with the results obtained by the other approaches.

3.4. Application of SCS for the prediction of biochemical activity of proteins

In this section, our new similarity approach is used to predict the biochemical activities of two sets of selected protein sequences from different organisms, obtained from the NCBI website. The first set includes well-characterized proteins, all extensively studied at the biochemical level. This set is used to evaluate the ability of SCS to predict biochemical activities of well-characterized protein sequences. The second set includes not yet characterized proteins whose biochemical activities we sought to predict in this paper, subsequently providing the obtained prediction results to the NCBI staff in the aim of adequately annotating the concerned protein sequences. The database entries and the corresponding organisms for the selected protein sequences are indicated in Table 14.

To be able to predict the biochemical activities of the target protein sequences, SCS was used to measure the similarity between each of these sequences with all the protein sequences included in the nr database (version of 8 December 2008), the non-redundant protein database maintained by NCBI as a target for BLAST search services, including more than 7 million protein sequences (i.e., very few of which are annotated). Then, an approach called *SNN*, for Significant Nearest Neighbors, inspired by the *KNN* classification approach, was used to systematically select the most significant similar sequences. These were used as the input dataset for the alignment-free clustering algorithm CLUSS, developed by (Kelil *et al.* 2007b), given that it recently proved to be more accurate in highlighting the biochemical activities of proteins than the alignment-based algorithms, especially for sequences that are hard to align. A biochemical activity can thus be attributed with high confidence to the uncharacterized protein sequence, if a well-characterized protein within the same cluster is already known. More details about the technique used to select the most similar sequences for each target sequence are provided below.

The *SNN* approach is mainly inspired by the widely known *KNN* classification algorithm (Wu *et al.* 2008). The *KNN* algorithm is a classification method based on statistical theory. It is among the simplest and most widely used pattern classification methods, and is usually employed when there is insufficient prior knowledge about the data distribution. The *KNN* algorithm classifies an object by the vote of its neighbors belonging to the class most common among its K nearest neighbors. For a detailed review of the classification problem, see (Cieslak *et al.* 2009). The major drawback of *KNN* is that the sensitivity of the results varies depending on the value selected for the parameter K . In particular, difficulties arise when an object from a given class has fewer than K real neighbors from the same class. We can see this clearly in the following example.

The example in Figure 5 represents the distribution of 3 objects from the class “black” (i.e., black circles) and 9 objects from class “white” (i.e., white circles) according to their pairwise closeness. In Figure 5, it is clear that black and white circles are well separated. However, *KNN* is not able to distinguish which candidates are actual neighbors. As a clear example of this, if we select $K = 10$, objects within the region bounded by the red line are considered to be neighbors of x as well as of y . This assigns to x and y the same classification, which is clearly not the case. This directly increases the rate of false positives and false negatives in the classification process.

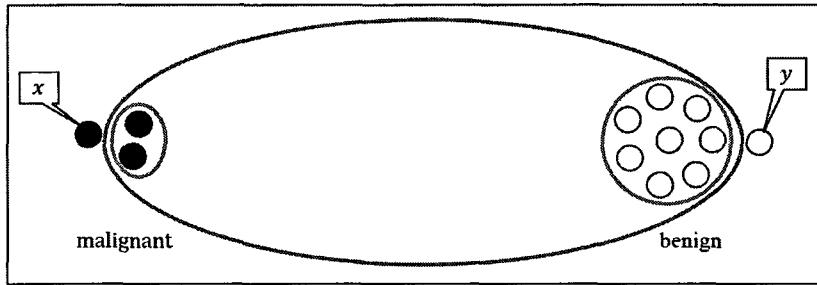


Figure 5. Difference between KNN and SNN

To deal with these drawbacks, here we present *SNN* that dynamically adjusts the value of K in the *KNN* approach. One of the major advantages of *SNN* compared to *KNN* is that *SNN* is able to detect the genuine nearest neighbors instead of choosing a fixed number of neighbours that may poorly reflect the data distribution. This has a direct impact on classification precision, which is explained in the following example.

In Figure 5, *SNN* is able to distinguish which objects are really neighbours. Objects within the region bounded by the blue line are considered neighbours of x , while objects within the region bounded by the green line are considered neighbours of y . This has the advantage of classifying the object x as 100% in class “black” and y as 100% in class “white”. One of the major advantages of *SNN* compared to *KNN* is thus that *SNN* discriminates more accurately between neighbourhoods of different sizes.

The *SNN* approach makes use of a systematic method for deciding which object in a given set of objects to retain as most similar to a target object. We first separate all the objects belonging to this set into two groups, one being the group of candidates highly similar to the target object, and the other of low similarity. This is done by sorting all objects in decreasing order of similarity and computing a separation threshold according to the maximum interclass inertia method, based on the Koenig-Huygens theorem, which gives the relationship between the total inertia and the inertia of each group relative to the center of gravity. In our case we have just two groups, the high similarity group and the low similarity group. The procedure is described as follows:

Let R be the uncharacterized protein sequence to be predicted, and let F be the set of obtained similarity measures between the sequence R and all the sequences from the nr database, with F_L the subset of low similarity measures, and F_H the subset of high similarity measures, such that:

$$\begin{aligned} F_L \cup F_H &= F \\ F_L \cap F_H &= \emptyset \\ \forall X, Y \in F | X \in F_L, Y \in F_H \Rightarrow S_{R,X} &< S_{R,Y} \end{aligned}$$

where $S_{R,X}$ and $S_{R,Y}$ are the similarity measures obtained between sequences R and X , and R and Y , respectively. The symbols F_L and F_H are simply used as variables representing all possible separations of F according to previous equations. By making use of the Koenig-Huygens theorem, the total inertia I is calculated as follows:

$$I = \sum_{i \in F_L} (S_{R,i} - \bar{S}_{F_L})^2 + \sum_{j \in F_H} (S_{R,j} - \bar{S}_{F_H})^2 + (\bar{S}_{F_L} - \bar{S}_{F_H})^2$$

where $S_{R,i}$ and $S_{R,j}$ the obtained similarity measures of sequences R and i , and R and j , such that i and j belong to the subsets F_L and F_H , all respectively; and \bar{S}_{F_L} and \bar{S}_{F_H} are the means (i.e., centers of gravity) of subsets F_L and F_H , respectively. The best separation of F is the subsets F_L and F_H that maximize the value of the total inertia I in the previous equation. Then, the most significant similar sequences to be used as input data for the clustering process are the subset of protein sequences corresponding to the subset F_H maximizing I the total inertia.

Table 14. Prediction of biochemical activities of the selected protein sequences

	Protein	Organism	Known Activity	Predicted activity
Protein Sequences with Known Activities	AAA24053	Bacteria		
	AAA69907	Bacteria		
	AAA35265	Eukaryota		
	AAA23216	Bacteria	β -Galactosidase	β -Galactosidase
	BAA07673	Bacteria		
	AAK06078	Bacteria		
Protein Sequences with Unknown Activities	AAC48809	Eukaryota		
	AAC74689	Bacteria		
	AAA52561	Eukaryota		
	AAK07836	Bacteria	β -Glucuronidase	β -Glucuronidase
	AAA37696	Eukaryota		
	AAD01498	Eukaryota		
Protein Sequences with Unknown Activities	AAV32104	Eukaryota	Unknown	
	XP_960828	Eukaryota	Unknown	Ribonucleotide-Diphosphate Reductase
	NP_249831	Bacteria	Unknown	
	ACB94306	Bacteria	Unknown	
	XP_001675807	Eukaryota	Unknown	FMRFamide
	YP_869103	Bacteria	Unknown	
Protein Sequences with Unknown Activities	ABK18067	Bacteria	Unknown	
	YP_846502	Bacteria	Unknown	ATP-Binding Cassette
	XP_001636168	Eukaryota	Unknown	
	ABS67555	Bacteria	Unknown	
	YP_429591	Bacteria	Unknown	Neuropeptide Precursor
	YP_001417212	Bacteria	Unknown	
Protein Sequences with Unknown Activities	YP_605034	Bacteria	Unknown	
	YP_342594	Bacteria	Unknown	Methyltransferase Type 12
	YP_049838	Bacteria	Unknown	

Table 14 shows the predicted biochemical activities of the target protein sequences. For the set of well-characterized sequences, the clustering has predicted exactly the appropriate biochemical cluster for each protein. For the set of target protein sequences with unknown biochemical activities, the clustering has put each uncharacterized sequence in a cluster containing an already well-characterized protein. Consequently, the activity of the well-characterized protein is assigned to the uncharacterized protein sequence. (See Table 14)

4. Discussion

The excellent results obtained in this paper on different types of categorical sequences from different application fields clearly show the effectiveness of our new general method and its advantage over existing domain-specific mainstream methods for measuring the similarity between categorical sequences. First, the results obtained with speech data show that SCS measures the similarity between pronounced letters and numbers more effectively than other approaches designed to perform the same task on categorical sequences. Second, the results obtained with the protein sequences show that, despite the fact that SCS does not take into account the substitution relations between different amino acids, it is competitive with approaches designed especially to deal with protein sequences. Third, the results obtained with natural language texts show that, even though the data used in this experiment were handled by SCS blindly by withdrawing spaces and newline marks, SCS was able to highlight the related texts as well as the approaches designed to deal with these data by taking into account the semantic relations between words and phrases.

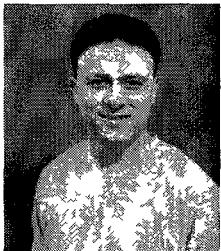
In conclusion, SCS effectively highlights the significant unseen information behind the chronological dependencies and structural features within different types of categorical sequences from different application fields. This is possible because it detects and uses the significant patterns that best represent the natural structures of these sequences, and minimizes the influence of those patterns that occur by chance and represent only noise. In addition, the matching technique, which allows us to simultaneously handle the “*within*” chronological order and the “*between*” non-chronological order of the structural features, plays an important role in reaching these conclusive results.

5. References

- Altschul, S.F., Gish, W., et al. (1990) Basic local alignment search tool. *J. Mol. Biol.*, 215, pp. 403-410.
- Amir, A., Lewenstein, M., et al. (2004) Faster algorithms for string matching with k mismatches. *J. Algorithms*, 50, pp. 257-275.
- Aslam, J.A. and Frost, M. (2003) An information-theoretic measure for document similarity. Proceedings of the 26th annual international conference on research and development in information retrieval, pp. 449-450.
- Berry, M.W. and Fierro, R.D. (1996) Low-rank orthogonal decompositions for information retrieval applications. *Numerical Linear Algebra Applications*, 1, pp. 1-27.
- Bogdan-Marta, A., Laskaris, N., et al. (2005) A novel efficient protein similarity measure based on n-gram modeling. *CIMED* 2005.
- Brand, M. (2006) Fast low-rank modifications of the thin singular value decomposition. *Linear Algebra and its Applications*, 415, 20.
- Cai, K., Chen, C., et al. (2004) Efficient similarity matching for categorical sequence based on dynamic partition. International conference on software engineering and applications, Cambridge, MA, USA, pp. 13-18.
- Chim, H. and Deng, X. (2007) A new suffix tree similarity measure for document clustering. Proceedings of the 16th international conference on World Wide Web, pp. 121-130.
- Cieslak, D. and Chawla, N. (2009) A framework for monitoring classifiers' performance: When and why failure occurs? *Knowledge and Information Systems*, Vol. 18, Num 1, pp. 83-108.

- Ganapathiraju, M., Klein-Seetharaman, J., et al. (2004) Characterization of Protein Secondary Structure Using Latent Semantic Analysis. IEEE signal processing magazine, may 2004 issue 15, pp. 78-87.
- Golub, G.H. and Van Loan, Charles F. (1996) Matrix Computations (Johns Hopkins Studies in Mathematical Sciences). The Johns Hopkins University Press.
- Horst, S. (1999) Symbols and computation: A critique of the computational theory of mind. *Minds Mach.*, 9, pp. 347-381.
- Karlin, S. and Ghodsi, G. (1985) Comparative statistics for DNA and protein sequences: Single sequence analysis. *Proc. Natl. Acad. Sci. USA*, 82, pp. 5800-5804.
- Kelil, A., Wang, S., et al. (2008) CLUSS2: An alignment-independent algorithm for clustering protein families with multiple biological functions. *IJCBDD*, 1, pp. 122-140.
- Kelil, A., Wang, S., et al. (2007a) CLUSS: Clustering of protein sequences based on a new similarity measure. *BMC Bioinformatics*, 8, 286.
- Kelil, A., Wang, S., et al. (2007b) A new alignment-independent algorithm for clustering protein sequences. *BIBE 2007*, 27-34.
- Kohonen, T. (1985) Median strings. *Pattern Recognition Letters*, 3, pp. 309-313.
- Kondrak, G. (2005) N-gram similarity and distance. *Proceedings of the 12th Conference on String Processing and Information Retrieval*. pp. 115-126.
- Levenshtein, V.I. (1966) Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10, pp. 707-710.
- Li, C. and Lu, Y. (2007) Similarity measurement of Web sessions by sequence alignment. *International conference on network and parallel computing workshops*, pp. 716-720.
- Loiselle, S., Rouat, J., et al. (2005) Exploration of rank order coding with spiking neural networks for speech recognition. *Proceedings of the IEEE International Joint Conference on Neural Networks*, pp. 2076-2080.
- Mhamdi, F., Rakotomalala, R., et al. (2006) A hierarchical n-grams extraction approach for classification problem. *Proceedings of the IEEE International conference on signal-image technology and internet-based systems*, Tunisia, pp. 310-321.
- Needleman, S.B. and Wunsch, C.D. (1970) A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.*, 48, pp. 443-453.
- Oh, S. and Kim, J. (2004) A hierarchical clustering algorithm for categorical sequence data. *Inf. Process. Lett.*, 91, pp. 135-140.
- Song, W. and Park, S. (2009) Latent semantic analysis for vector space expansion and fuzzy logic-based genetic clustering. *Knowledge and Information Systems*.
- Suen, C.Y. (1979) N-gram statistics for natural language understanding and text processing. *IEEE TPAMI, PAMI-1*, pp. 164-172.
- Tatusov, R.L., Fedorova, N.D., et al. (2003) The COG database: An updated version includes eukaryotes. *BMC Bioinformatics*, 4, 41.

- Wan, X. (2007) A novel document similarity measure based on earth mover's distance. *Inf. Sci.*, 177, pp. 3718-3730.
- Wu, K.P., Lin, H.N., et al. (2003) A new similarity measure among protein sequences. *Proc. IEEE Comput. Soc. Bioinform. Conf.*, 2, pp. 347-352.
- Wu, X., Kumar, V., et al. (2008) Top 10 algorithms in data mining. *Knowledge and Information Systems*, 14, pp. 1-37.



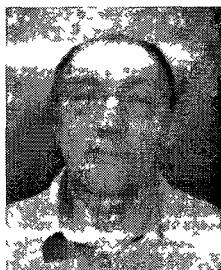
Abdellali Kelil received his engineering degree from the University of Annaba, Algeria. He is currently a Ph.D. candidate in Bioinformatics at the University of Sherbrooke, and a member of the ProspectUS Data Mining and Bioinformatics laboratory at the same University. He is about to submit his thesis and to start his postdoctoral studies at the University of Montreal within the laboratory of Dr. Stephen Michnick Canada Research Chair in Integrative Genomics. His research interests include Bioinformatics, Genomics, Proteomics, Phylogenetic, Sequence Analysis, Data Mining, Pattern Recognition, and Information Retrieval. During his PhD, he worked extensively, to tackle different challenges and solve different problems related to the area of bioinformatics and computational biology. He was able to develop many new effective algorithms and approaches that meet the needs of biologists. Many of his contributions have been successful and rewarded by honorary mentions by the scientific community.



Shengrui Wang received his Ph.D. from the National Polytechnic Institute of Grenoble, France. He is director of ProspectUS laboratory at University of Sherbrooke. His research interests include Pattern Recognition, Data Mining, Artificial Intelligence Information Retrieval, Neural Networks, Image Processing, Remote Sensing, GIS. His current projects include Graph Matching and Graph Clustering for Content-Based Image Retrieval, Fuzzy Clustering and Variable Selection for Data Mining, Construction of Integrated Information Infrastructure for Automobile Driving. He is co-leader of a project within NCE Auto21. He is also a member of the Group MOIVRE at the University of Sherbrooke.



Qingshan Jiang is a professor at Chengdu University and Xiamen University, China. He received a Ph.D. in mathematics from Chiba Institute of Technology, Japan, in 1996, and a Ph.D. in computer science from University of Sherbrooke, Canada, in 2002. During his over 25 years of study and research, he has published over 90 scientific papers at international journals and conference proceedings. His expertise lies in system development with a strong focus in the areas of image processing, statistical analysis, fuzzy modeling and data mining.



Ryszard Brzezinski received his Ph.D. from the University of Warsaw, Poland. He is director of a laboratory of molecular biotechnology, environmental microbiology and bioinformatics at the University of Sherbrooke. His research interests include Molecular Biotechnology, Enzymology, Genetic Expression, Environmental Microbiology and Bioinformatics. He is a member of the CEVDM the Center for Study and Recovery in Microbial Diversity at the University of Sherbrooke. He is also a member of SEVE the Inter-institutional research center in Plant Breeding.

Chapitre 4

ALIGNEMENT DES PROTÉINES APPARENTÉES

Selon l'évolution, plusieurs types de mutations peuvent se produire dans les gènes qui codent pour les protéines [145]. Les mutations ponctuelles substituent un résidu unique pour un autre. Les insertions et suppressions de résidus peuvent également se produire, impliquant un seul résidu jusqu'à plusieurs centaines. D'autres mécanismes de l'évolution sont à l'œuvre dans la nature notamment la recombinaison génétique, où les brins d'ADN sont rompus et puis rejoints pour reformer de nouvelles combinaisons de gènes. Seuls les résidus qui sont essentiels pour la fonction d'une protéine, ou qui sont nécessaires à la protéine pour se replier correctement, sont conservés [88]. En comparant les protéines d'une même famille, et en recherchant les résidus qui sont conservés dans tous les membres de la famille, nous pouvons alors apprendre beaucoup de choses sur la structure et la fonction de cette famille [49, 99]. L'une des méthodes les plus utilisées par les chercheurs pour accomplir cette tâche est l'alignement multiple des séquences. Ainsi, l'alignement est devenu un outil fondamental et souvent le point de départ des analyses dans de nombreux domaines de la biologie moléculaire moderne, de l'étude de l'évolution des protéines à la prédiction de leurs fonctions et structures 2D/3D. En plaçant la séquence de protéine dans le cadre de la famille, l'alignement multiple non seulement peut identifier d'important motifs liés à la structure ou la fonction qui ont été conservés à travers l'évolution, mais il peut aussi mettre en évidence les caractéristiques particulières non conservées à la suite d'événements spécifiques ou des perturbations [79, 102].

Une vaste gamme d'algorithme d'alignement ont été développés à ce jour dans le but de construire des alignements de haute qualité dans des délais raisonnables, afin de permettre le traitement du grand nombre de protéines séquencées à ce jour. Cependant, tous ces

algorithmes sont basés sur la supposition que les séquences d'entrées sont globalement alignables, alors que souvent elles ne le sont pas, surtout les séquences qui contiennent des domaines répétés, inversés, supplémentaires ou manquants. Ceci oblige souvent les biologistes à traiter manuellement les ensembles de séquences à aligner lorsque les alignements obtenus ne sont pas satisfaisants. La marche à suivre pour obtenir un alignement biologiquement satisfaisant est généralement la même. Après avoir sélectionné manuellement un ensemble de séquences de protéines à aligner, ou en utilisant des outils comme BLAST ou FASTA, un algorithme d'alignement est choisi parmi ceux qui sont adaptés aux propriétés structurelles des protéines à aligner (information qui n'est pas toujours disponible), ou alors il est choisi arbitrairement parmi les algorithmes les plus connus. Après quoi, l'algorithme est utilisé pour produire l'alignement de l'ensemble des séquences de protéines. Ensuite, le résultat est analysé et évalué visuellement afin de déterminer si l'alignement a besoin de quelques améliorations, ce qui est habituellement le cas dans la pratique, surtout pendant les premières tentatives d'alignements. Dans ce cas, des séquences sont manuellement retirées, ou ajoutées à l'ensemble de séquences d'entrée dans le but d'obtenir un alignement qui permettra de mieux mettre en évidence les régions conservées dans les séquences d'intérêt. Ce processus est répété plusieurs fois jusqu'à ce qu'un tel alignement est atteint.

L'inconvénient majeur des algorithmes d'alignement existants est qu'ils ignorent si l'ensemble de séquences d'entrée comprend des protéines qui partagent assez de régions similaires pour produire des alignements biologiquement intéressants, et laisse plutôt à l'utilisateur le soin de procéder à l'analyse visuelle et les manipulations nécessaires sur l'ensemble d'entrée.

Dans ce chapitre nous présentons ALIGNER, un algorithme que nous avons développé pour aligner de manière efficace autant les séquences de protéines qui nécessitent un alignement global que celles qui nécessitent un alignement local. En plus de cela, ALIGNER est conçu pour détecter automatiquement parmi les protéines à aligner les groupes de protéines dont l'alignement peut révéler d'importantes propriétés structurelles et fonctionnelles.

ALIGNER est le premier algorithme qui soit conçue pour guider automatiquement les biologistes dans le choix des séquences de protéines à inclure dans les ensembles d'entrée, et aussi pour éviter de recourir aux manipulations aléatoires ou arbitraires de l'utilisateur sur les ensembles d'entrées. En outre, notre algorithme est destiné à aider et à réduire la charge de travail des biologistes, par le traitement automatique des groupes de séquences de protéines qui ne partagent pas suffisamment de régions conservées pour produire des alignements satisfaisants.

ALIGNER a été soumis à « *BMC Bioinformatics* » en 2010. Nous avons aussi lancé le serveur web de ALIGNER situé à <http://prospectus.usherbrooke.ca/ALIGNER/>, qui offrira aux biologistes la possibilité d'utiliser interactivement notre nouvel algorithme.

Ma contribution inclut la conception de ALIGNER, l'exécution des tests expérimentaux, le développement du serveur web, et la rédaction du manuscrit. Mes superviseurs Dr. Shengrui Wang et Dr. Ryszard Brzezinski ont supervisé, participé à la rédaction, et validé tout le projet. Voici donc la publication dont a fait l'objet ALIGNER:

- Abdellali Kelil, Ryszard Brzezinski, Shengrui Wang. ***ALIGNER: Detecting and Aligning Related Protein Sequences***. Soumis à : BMC Bioinformatics, 2010.

ALIGNER: Detecting and Aligning Related Protein Sequences

Abdellali Kelil^{1,3}, Ryszard Brzezinski^{2,3} and Shengrui Wang^{1,3}

¹Department of Computer Sciences, Faculty of Sciences, University of Sherbrooke, J1K 2R1 Canada.

²Department of Biology, Faculty of Sciences, University of Sherbrooke, J1K 2R1 Canada.

³2500, boul. de l'Université, Sherbrooke (Québec) CANADA J1K 2R1

ABSTRACT

We introduce ALIGNER, an algorithm capable of effectively aligning protein sequences that need either global or local alignment. ALIGNER is the first algorithm devised to detect and align protein sequences that share enough conserved regions to produce biologically meaningful alignments. ALIGNER automatically handles some steps performed manually using existing alignment algorithms, and avoids resorting to user random or arbitrary manipulation of the input datasets. Extensive experimentations show that ALIGNER outperforms almost all alignment algorithms.
<http://prospectus.usherbrooke.ca/ALIGNER>

1 PROBLEMATIQUE

Protein sequence alignment is the process of finding the best matching between the sequences by inserting “gaps” in the appropriate positions in each sequence, so that the positions where the sequences have identical or similar residues are aligned. The alignment aims to identify regions of similarity that might reveal significant patterns of functional, structural, or evolutionary significance in a given set of protein sequences. The literature reports two types of alignment approaches, global and local. On one hand, global alignment approaches span the entire length of all protein sequences by aligning every residue in every sequence. On other hand, local alignment approaches look for most conserved patterns by identifying regions of similarity within long protein sequences that are often widely divergent overall. It has been shown by McClure *et al.* [35] and Thompson *et al.* [53] that which alignment approach is most effective depends essentially on the structural nature of the protein sequences to be aligned. Often, global alignment produces the most accurate and reliable alignments, but in the presence of large N/C-terminal extensions and internal insertions, an example is shown in Figure 1, local alignment is the most successful. This is even truer in the case of multi-modular protein sequences illustrated in Figure 1 for an example. The most important problem with both of these approaches is that, without prior knowledge about the biochemical and structural properties of each of these proteins, we cannot choose with high certainty the appropriate approach to perform the alignment that can reveal the patterns in these sequences that are the most relevant, functionally or structurally. This is the main reason for proposing ALIGNER.

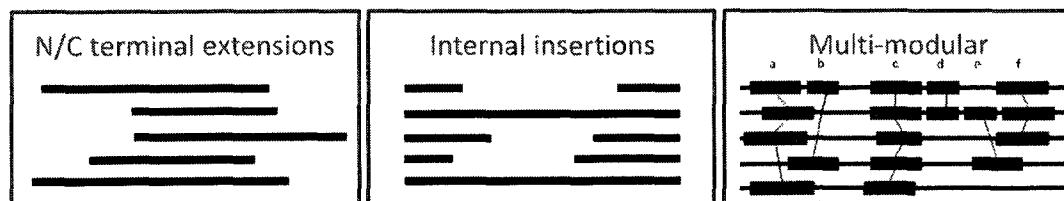


Figure 1. Examples of N/C-terminal extensions, internal insertions and multi-modular protein sequences

On the other hand, existing alignment approaches are devised specifically to produce, for a given input dataset, the alignment of all the protein sequences and ignore if the input dataset includes divergent protein sequences, those that do not share enough conserved regions to produce biochemically significant alignments, which can complicate considerably the identification of regions of similarity. In real case studies, to deal with this problem, input datasets are often manually handled by discarding protein sequences that may disturb the alignment. However, this process is not always possible, especially when input datasets include several divergent groups of protein sequences. This is the second reason for the approach proposed.

In this paper, we present ALIGNER, a new and effective alignment algorithm capable of effectively aligning protein sequences that need either global or local alignment. Like the global alignment algorithms, ALIGNER spans the entire length of all protein sequences, by aligning every residue in every sequence. At the same time and like the local alignment algorithms, ALIGNER pays particular attention to the significant patterns shared between protein sequences. In addition, ALIGNER detects groups of related protein sequences in input protein datasets that share enough significant patterns to produce alignments that can reveal important structural and functional properties within each group of proteins, without resorting to user manipulation of the input datasets. ALIGNER is freely donated to the scientific community. The implementation and the webserver are available at <http://prospectus.usherbrooke.ca/ALIGNER>. More details about the use of the webserver are provided at the end of this paper.

2 IMPLEMENTATION

The main strategy of ALIGNER is based on a progressive alignment approach. For a given set of protein sequences Figure 2.1, the ALIGNER algorithm performs the alignment via the following steps (a detailed description of the algorithm is given later in the paper):

1. Significant patterns within each sequence are detected and collected using a new pairwise matching scheme, Figure 2.2
2. A pairwise matrix of the similarity between all possible pairs of protein sequences is built, using a new approach, Figure 2.3.

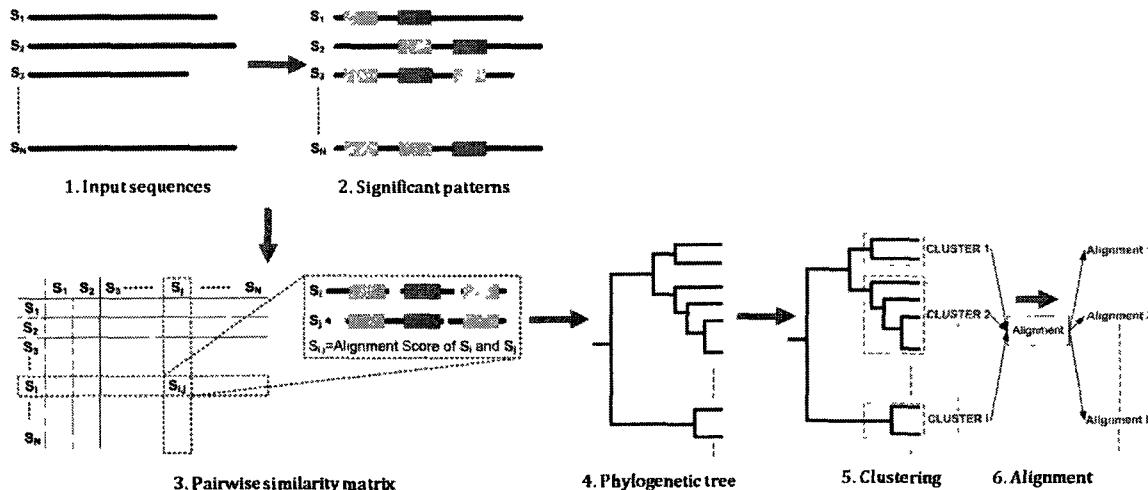


Figure 2. Illustration of the different steps of ALIGNER

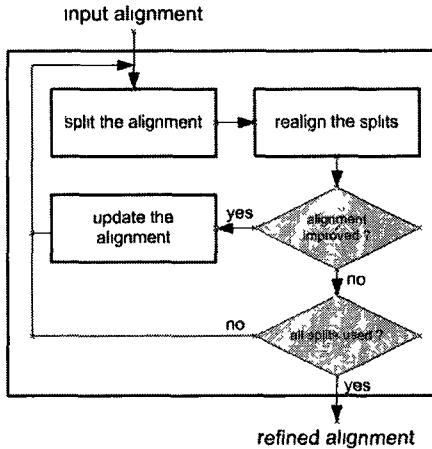


Figure 3. Flowchart of the refinement algorithm

3. A phylogenetic tree is built, using the approach introduced by Kelil *et al.* [26] (Figure 2.4).
4. The phylogenetic tree is partitioned into smaller subtrees, using the approach introduced by Kelil *et al.* [26]. The leaves of each subtree represent a distinct subset of protein sequences (Figure 2.5).
5. Protein sequences in each subset are aligned hierarchically (i.e., by progressive alignment), using their corresponding subtree as the guide tree. A new alignment objective function is used (Figure 2.6).
6. An iterative bootstrap refinement process is applied on the alignment obtained with each subset of protein sequences (Figure 3).

A key issue in this method is the detection of significant patterns across protein sequences that might reveal functional, structural, or evolutionary properties. Below, we provide a detailed description of how this is performed.

2.1 Significant patterns

A key issue in ALIGNER is the detection of significant patterns across protein sequences that might reveal functional, structural, or evolutionary properties. This is performed using the method we introduced in Kelil *et al.* [27, 28] based on a new pairwise matching scheme and the statistical theory of Karlin *et al.* [24], that guarantees a pattern designated as statistically significant (i.e., a pattern that maps to a specific meaning in terms of protein sequence structure) has less than a 1% probability of occurring by chance.

2.2 Similarity measure

In ALIGNER, the similarity measure between protein sequences is calculated by using the classical *dynamic programming* algorithm developed by Needleman and Wunsch [38] in conjunction to the affine gap penalty scheme described in [11]. This algorithm spans the entire length of compared sequences by aligning every residue in every sequence in the aim to produce global alignments usually used to calculate overall similarity between protein sequences. For a review see the reference papers [51] and [11]. However, unlike the classical *dynamic programming* algorithm that uses only residue-to-residue substitution scores, ALIGNER pays particular attention to local similarities, by taking into account significant patterns (collected in subsection 0) and shared between pairs of protein sequences in the calculation of the similarity measure. This makes our similarity measure more successful than the classical *dynamic programming* algorithm on problematic cases, such as multi-

domain protein sequences with remote similarities detailed by Higgins [20]. We describe below how the similarity measure is calculated in ALIGNER.

Let X and Y be two protein sequences for which we want to measure the similarity, and $E_{X,Y}$ the set of pairs of significant patterns (x,y) collected from X and Y in the previous subsection, with $x \in X$ and $y \in Y$. For the rest of this section, X^p and Y^q are the p^{th} and q^{th} residues in X and Y , respectively. Now, we define $R_{X,Y}$ the set of significant pairs of residues belonging to the set of collected pairs of significant patterns in $E_{X,Y}$, such that:

$$R_{X,Y} = \left\{ (X^p, Y^q) \mid \begin{array}{l} (X^p \in x) \vee (Y^q \in y) \\ (x, y) \in E_{X,Y} \end{array} \right\}$$

Then, by using the residue-to-residue substitution score in conjunction to $R_{X,Y}$ the set of significant pairs of residues, we define $RR_{X,Y}^{p,q}$ a new residue-to-residue score between the p^{th} residue in X and the q^{th} residue in Y as follows, with M a substitution matrix:

$$RR_{X,Y}^{p,q} = M(X^p, Y^q) + \sum_{(X^p, Y^q) \in R_{X,Y}} M(X^p, Y^q)$$

Finally, by using this new residue-to-residue score in combination with the *dynamic programming* alignment algorithm, the similarity measure between X and Y , is then computed as follows:

$$S_{X,Y} = \frac{\text{Alignment Score of } X \text{ and } Y}{\max(|X|, |Y|)}$$

2.3 Phylogenetic tree

Now, let be a set of protein sequences to be aligned. By using one of the known substitution matrices, and the similarity measure defined in the previous subsection, we compute the pairwise similarity matrix S . Then we build the phylogenetic tree of the set of protein sequences to be aligned, using the approach introduced by Kelil *et al.* [26] in CLUSS2. A spectral decomposition on S is applied to obtain a set of vectors, each of which is used to represent a protein sequence in the new vector space resulting from the decomposition of S . Such a representation is valid in the sense that the similarity between each pair of sequences from the original similarity matrix S is approximately equal to the similarity between the corresponding vectors measured by the inner product function (i.e., preservation of similarity). This representation facilitates the use of hierarchical clustering. In fact, a cluster will be represented by only one vector, thus cluster merging can be easily performed by adding two vectors, and the similarity between two clusters can then be estimated by the cosine similarity function.

2.4 Sequence weighting

In the literature, several sequence alignment algorithms such as CLUSTAL [51] and MUSCLE [11] often use different methods for weighting protein sequences in the aim of improving alignments, such as those introduced by Henikoff *et al.* [19], Altschul *et al.* [2], Thompson *et al.* [51], and Gotoh [17]. The weight of a sequence depends generally on the number and the closeness of the protein sequences that are in fact similar to the protein sequence, and is thus intended to measure how well a set of protein sequences is represented by this particular sequence. However, there is no consensus on the most effective way to calculate such weights [11]. Nevertheless, the method introduced by Thompson *et al.* [51] enables a signifi-

cant saving in running time, and is therefore the weighing method adopted in ALIGNER. The implementation of this method used in ALIGNER is given below.

Let T be the phylogenetic tree, built as described in the previous subsection, of an input set of protein sequences to be aligned. Each leaf node in the phylogenetic tree represents a particular protein sequence. We define W_X as the weight of a protein sequence X , as follows:

$$W_X = \sum_{i \in [branch(L_X \rightarrow R) - \{R\}]} \frac{D_{Parent(i), i}}{d_{Parent(i)}}$$

In this formula, R is the root of the phylogenetic tree, L_X is the leaf belonging to this tree and representing the protein sequence X , $branch(L_X \rightarrow R) - \{R\}$ is the subset of internal nodes on the branch from L_X to R excluding R , $Parent(i)$ is the parent of the node i , and $D_{Parent(i), i}$ is the length of the branch connecting the node i to its parent, and $d_{Parent(i)}$ is the number of leaves in the subtree rooted at the parent of i . In our experiments, the use of this method for weighting protein sequences has allowed ALIGNER to lead 2% to 5% better alignment results.

2.5 Clustering

One of the principal aims of this work is to be able to detect, within input datasets of protein sequences to be aligned, subsets including sequences that are more likely to share significant intrinsic features underlying important structural and functional properties, making their alignment useful and informative in the case of when the alignment of the whole input dataset is not. To this end, we have adopted the strategy of clustering phylogenetic trees developed by Kelil *et al.* [28]. A simple and systematic method is used to decide which subtrees within the phylogenetic tree to retain as the largest possible clusters (i.e., clusters each of which includes the largest possible number of related protein sequences). This is done by computing a separation threshold according to the *maximum interclass inertia* method, based on the well-known *Koenig-Huygens* theorem, which gives the relationship between the total inertia and the inertia of each group relative to the centre of gravity. This simple clustering method has been successfully applied to the clustering of several protein datasets. When used with an accurate similarity measure, it has been shown to be capable of grouping protein sequences according to their functional annotations and biological classifications. For a detailed discussion, see the papers published by Kelil *et al.* in [26] and [28].

2.6 Alignment objective function

In this work, our primary concern is to develop an approach that will be able to align protein sequences that are known to be hard to align, such as NC-terminal extension, internal insertion, and multi-domains protein sequences. For such sequences, the classical sequence-based alignment approaches usually fail to yield biologically suitable results. For a review, see the detailed description presented by Higgins in [20] and by Kelil in [28] and [27]. In fact, hard-to-align protein sequences often have similar and conserved domains in non-equivalent positions when viewed in terms of primary structure, which makes them difficult to align. However, these domains might well be in equivalent positions when viewed in terms of secondary and tertiary structures. In the absence of an explicit identification of such positions in our alignment approach, we adopt the strategy of matching all the conserved domains, even those in non-equivalent positions. The reason is that, with a suitable length of significant patterns, it is more probable that we will effectively match patterns that are similar due to conservation rather than to random phenomena. This is why, in our new alignment objective function described below, we consider all the residues involved in one or more collected significant patterns, as described in section 0, even if some of these patterns are in non-equivalent positions.

In our alignment approach, we have adopted the classical *dynamic programming* alignment algorithm developed by Needleman and Wunsch [38] with the affine gap penalty scheme described in [11]. To find a global and optimal alignment between two sequences, the dynamic programming algorithm optimizes an objective function defined as the sum of all matching scores between aligned positions. The objective function aims to maximize the overall score of a multiple alignment, where the scores of each pair of matched positions of the alignment are added up to yield the overall score. In our approach, we have employed the well-known *Sum-of-Pairs* score as the overall score: for more details please see Edgar [11], Thompson *et al.* [53], and Thompson *et al.* [52]. The *Sum-of-Pairs* score can take into account the chemical/physical properties of residues, by making use of the substitution relations between different residues, which in its turn estimates the rate at which each possible amino acid in a sequence keeps unchanged or substituted by another residue over time. The *Sum-of-Pairs* score of a given position in the alignment is defined as the sum of all substitution scores between all pairs of matched residues at this position.

Like the *Sum-of-Pairs* score, our new alignment objective function spans the entire length of all protein sequences, aligning every residue in every sequence, but also pays particular attention to residues that are involved in one or more of the significant patterns collected in section 0. This new objective function is described below.

At this stage, we consider the general case of aligning two profiles (note that a profile may be either an individual sequence or a set of aligned sequences treated as one sequence by regarding each column as a symbol).

Now, let A and B be two profiles that we want to align. Here, the symbols A and B are used simply as variables, and they can express multiple alignments or individual sequences. For the rest of this section, we use the following notation, A^p and B^q are the p^{th} and q^{th} positions (or columns) in A and B ; A_i and B_j are the i^{th} and j^{th} sequences in A and B ; while A_i^p and B_j^q are the p^{th} and q^{th} residues in the i^{th} and j^{th} sequences in A and B . First of all, we define $E_{A,B}$, the set of all matching sets including all pairs of significant patterns between all pairs of protein sequences from A and B , respectively, such that:

$$E_{A,B} = \bigcup_{X \in A, Y \in B} E_{X,Y}$$

We also define, for each pair of positions A^p and B^q in A and B , the set $E_{A,B}^{p,q}$ that includes all pairs of residues belonging to a pair of patterns in the set of collected significant patterns $E_{A,B}$, such that:

$$E_{A,B}^{p,q} = \left\{ (A_i^p, B_j^q) \middle| \begin{array}{l} \forall i,j \\ (A_i^p \in x) \vee (B_j^q \in y) \\ (x, y) \in E_{A,B} \end{array} \right\}$$

For a protein sequence comprising a number of significant patterns that were highly conserved during evolution, each of these patterns contributes in a complex way to provide one or more biological functions. A point mutation in one of the conserved patterns may significantly alter or even eradicate the biological activity of the protein, while in another conserved pattern it might only slightly decrease the expression of the biological function. So, we make use of a substitution matrix to emphasize the fact that each conserved pattern can be involved to a different degree in a biological activity. To emphasize this phenomenon in the computation of our new alignment matching score, we define the *Sum-of-Residues* score $SR_{A,B}^{p,q}$ for each pair of positions A^p and B^q in A and B , as the sum of the substitution scores of all the pairs of residues belonging to the set $E_{A,B}^{p,q}$, such that:

$$SR_{A,B}^{p,q} = \sum_{\substack{x \in X, y \in Y \\ x \in A, y \in B \\ (x,y) \in E_{A,B}^{p,q}}} M(x,y) \cdot W_X \cdot W_Y$$

In this formula, x and y are used as variables and simply represent a pair of matched residues from a pair of collected significant patterns from a pair of protein sequences X belonging to A and Y belonging to B , while M is one of the known substitution matrices. The symbols W_X and W_Y represent the weights of the protein sequences X and Y calculated by the method described in subsection 2.4.

In formula above, we use the substitution concept to emphasize the relation that binds one amino acid with itself. The value of $M(x,y)$ (i.e., within the diagonal of the substitution matrix) estimates the degree which each possible amino acid in a sequence remains unchanged over time.

At this stage, we define $SP_{A,B}^{p,q}$ the *Sum-of-Pairs* score of the profiles A and B at the positions A^p and B^q as follows:

$$SP_{A,B}^{p,q} = \frac{1}{|A| \cdot |B|} \sum_{i \leq |A|} \sum_{j \leq |B|} M(A_i^p, B_j^q) \cdot W_{A_i} \cdot W_{B_j}$$

The symbols W_{A_i} and W_{B_j} represent the weights of the i^{th} and the j^{th} protein sequences belonging to the profiles A and B , respectively. Note that in the formula above there is no need to consider the scores of pairs of sequences in A or in B as their scores are unchanged under all possible alignments of A and B .

Now, we can define $MS_{A,B}^{p,q}$, our new matching score for A^p and B^q at the p^{th} and q^{th} positions in A and B , as the sum of $SP_{A,B}^{p,q}$, the *Sum-of-Pairs* score, and $SR_{A,B}^{p,q}$, the *Sum-of-Residues* score, defined as follows:

$$MS_{A,B}^{p,q} = SP_{A,B}^{p,q} + SR_{A,B}^{p,q}$$

The most important advantage with this matching score is that, in addition to the use of the *Sum-of-Pairs* score that evaluates the similarity between p^{th} and q^{th} positions in A and B , we emphasize the residues that are involved in one or more collected significant patterns shared by the p^{th} and q^{th} positions over all sequences in A and B , respectively. Now, by considering \bar{A} and \bar{B} the two profiles resulting from the alignment of the two profiles A and B , we can define $AS_{A,B}$ the alignment objective function, as follows:

$$AS_{A,B} = \max_{\bar{A}, \bar{B}} \left(\sum_{\forall i} MS_{\bar{A}, \bar{B}}^{i,i} \right)$$

The objective function $AS_{A,B}$ can be maximized effectively by applying the *dynamic programming* alignment algorithm [38], with user predefined *opening gap* and *extension gap* penalty costs.

2.7 Alignment

At this stage, we make use of the most widely employed alignment approach, known as *progressive alignment*, to align the individual subsets (clusters) of protein sequences belonging to the phylogenetic subtrees obtained during the clustering stage

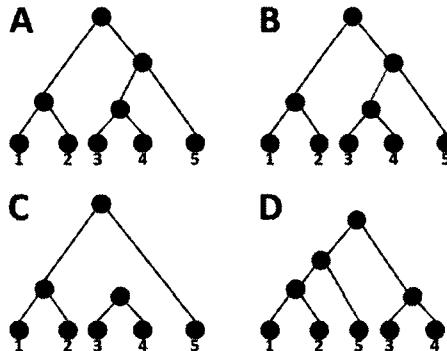


Figure 4. Example of an iterative tree-based alignment refinement.
(A) Given an input phylogenetic tree of a set of aligned protein sequences, **(B)** the red edge is deleted, **(C)** which subdivided the phylogenetic tree into two distinct subtrees,

described in subsection 2.5 above.

Given a subset of protein sequences belonging to one of the obtained phylogenetic subtrees, the *progressive alignment* approach builds up the alignment of the protein sequences by starting with the alignment of the closest pair of profiles (note that a profile can be either a sequence or a group of aligned sequences) as determined by the phylogenetic subtree, and progressively aligns the next closest pair of profiles, following the hierarchical structure of the phylogenetic subtree. This process continues in an iterative fashion until only one profile remains, which is the alignment result. At each iteration, the new alignment objective function introduced in subsection 2.6 is used conjointly with the *dynamic programming* algorithm to obtain the best alignment result of each pair of profiles. The online version of ALIGNER allows also the selection and the alignment of the protein sequences belonging to two or more of the obtained clusters.

2.8 Refinement

The main drawback of the progressive alignment approach is that errors which may occur at any iteration of the alignment can never be corrected in later iterations, and are propagated through the alignment process, degrading the quality of the final alignment. In addition, a particular problem should be noted with progressive alignment approach, which is the effect of introducing a single divergent sequence into a set of closely related sequences, causing the iteration to diverge away from the best possible alignment [53].

To overcome this drawback, we adopted the iterative bootstrap refinement approach introduced by Hirosawa *et al.* [21] to correct any errors that may have occurred in the alignment process. This approach is based on a tree-dependent, restricted partitioning technique that is able to improve the quality of the final alignment by iteratively refining the alignment whenever two sub-alignments are merged in a tree-based way. An edge is deleted from the phylogenetic tree, subdividing the original tree into two disjoint subtrees each of which represents a distinct sub-alignment. The two sub-alignments are realigned to obtain the refined alignment, which is compared to the original alignment to decide if it is accepted or rejected. An example is shown in Figure 4. The edges are picked in order of decreasing distance from the root.

3 RESULTS

To illustrate its efficiency, we tested ALIGNER extensively on a variety of protein datasets and compared the results with those of several mainstream algorithms. We analyzed the results with support from the literature and functional annotations.

All the details of the results obtained in the experiments performed in this section are provided with this manuscript as supplementary material, and available also at <http://prospectus.usherbrooke.ca/ALIGNER>. In all our experiments, for each algorithm tested we utilized the last known to date version of the original implementation of the authors with the default input parameters. ALIGNER is used with the following affine gap penalty parameters, gap extension penalty = -1, opening gap penalty = -11, substitution matrix = BLOSUM62.

First, to illustrate the effectiveness of ALIGNER in grouping protein sequences according to their functional annotations and biological classifications, we performed extensive tests on the widely known databases COG (for unicellular organisms) and KOG (for eukaryotic organisms) [49]. The COG and KOG databases contain phylogenetic classifications of proteins encoded in complete genomes, in which clusters contain orthologous groups of proteins that were delineated by comparing protein sequences encoded in complete genomes representing major phylogenetic lineages.

Second, to show the effectiveness of ALIGNER in the clustering of hard-to-align multi-domain proteins, experimental tests were performed on the $(\alpha/\beta)_n$ -barrel proteins studied by Côté *et al.* [9] and Fukamizo *et al.* [14], which form a group in the Glycoside Hydrolases family 2 (GH2) from the Carbohydrate Active Enzymes database (CAZy) located at <http://www.cazy.org/>.

Third, to illustrate the effectiveness of ALIGNER in aligning protein sequences, we performed extensive tests on the widely used multiple alignment benchmark BAliBASE [52], a database of high quality, which is composed of manually refined multiple sequence alignments specifically designed for testing, evaluating, and comparing multiple sequence alignment algorithms.

Fourth, we tested ALIGNER on the Glycoside Hydrolase family 46 (GH46) from the Carbohydrate-Active Enzymes database (CAZy), July 2009 version [7]. GH46 includes 39 enzymes with Chitosanase activity, for which alignment is still problematic using the classical, sequence-based alignment approaches. In all of these experiments, the various algorithms tested were executed with their default input parameters.

In addition, in the aim of evaluating the effectiveness of our new alignment objective function introduced in subsection 2.6, and in order to show the usefulness of our pairwise matching scheme introduced in subsections 0 and **Error! Reference source not found.** in the detection of significant patterns, and also demonstrate the contribution of the captured significant patterns in all the results obtained by ALIGNER presented in this section, we performed all the experiments presented in this work with two versions of ALIGNER. The first version is used with the matching score $MS_{A,B}^{p,q}$ as presented in the section 2.6, while in the second version the matching score $MS_{A,B}^{p,q}$ is defined as equal to the *Sum-of-Pairs* score $SP_{A,B}^{p,q}$ without considering $SR_{A,B}^{p,q}$ the *Sum-of-Residues* score, as follows: (in all this section we identify this version as ALIGNER*)

$$MS_{A,B}^{p,q} = SP_{A,B}^{p,q}$$

3.1 Clustering

COG and KOG databases

To illustrate the effectiveness of ALIGNER in grouping protein sequences according to their functional annotation and biological classification, we tested it the two ensembles of six randomly generated subsets used in [26]. The subsets from the COG database, numbered C1 to C6, contain 336, 214, 288, 355, 676, 309 sequences; and K1 to K6 from the KOG database contain 363, 425, 441, 360, 326, 590 sequences. Each of these subsets includes non-orphan sequences (each sequence has at

least one similar sequence from the same functional classification) with at least 10 different biochemical activities. We evaluated the results using the quality measure Q_m [27], which evaluates the quality of a clustering by measuring the percentage of correctly clustered protein sequences based on their known functional annotations.

In addition, we compared the results with those obtained by different existing algorithms that have been successfully applied on the clustering of different protein datasets: for instance, CLUSS [27] based on alignment-free similarity measure using a strict matching scheme and hierarchical clustering based on Koenig-Huygens theorem, CLUSS2 [26] improved version of CLUSS based on a permissive matching and a spectral clustering based on latent semantic analysis, BlastClust [1] based on all-against-all BLAST for measuring the pairwise similarity and a score-based single-linkage agglomerative algorithm for the clustering, and CD-HIT [22] based on a greedy incremental method in which sequences are sorted in decreasing length and clusters are delimited by sequences with a similarity below a given threshold. All these algorithms incorporate different methods for the calculation of pairwise similarities between protein sequences, thus they accept protein sequences as input data. Nevertheless, we tested also several algorithms for which third-party similarity measure methods are needed to compute the pairwise similarities, for instance: Tribe-MCL [12] version 09.308 a widely known algorithm based on Markov cluster approach, and gSPC [50] based on a method that is analogous to the treatment of an inhomogeneous ferromagnet in physics, as well as more recent algorithms, FORCE [57] based on transitive graph projection and clusters arbitrary sets of objects for a given pairwise similarity measures, TransClust [54] an improved version of FORCE based on weighted transitive graph projection that is underlying model reflects hidden transitive substructures, SCPS [39] based on spectral transformation of the pairwise similarity matrix so that objects are mapped onto a vector space after which they are clustered using K-means. In our experiments, the last versions of the widely-known protein comparison algorithms ClustalW [30] version 2.0.12 and Blast [1] version 2.2.23 were used as third-party similarity measure methods to calculate the input pairwise similarity matrices used by these algorithms. In the results shown in Table 1 and Table 2, Tribe-MCL and gSPC are used with ClustalW while FORCE, TransClust, and SCPS are used with Blast. In other hand, the algorithm CD-HIT was originally developed to cluster highly homologous protein sequences; therefore the default value of its input parameter which is “*sequence identity cut-off*” is fixed at the large value of 0.9. In the aim of decreasing the discriminative power of CD-HIT the value of the input parameter is reduced to 0.3 the minimum value authorized by the algorithm.

In Table 1 and Table 2, we summarize the clustering results obtained by each algorithm on each protein dataset. Each table shows the Q_m obtained by each algorithm on each of the generated protein datasets. The two last columns in each table show respectively the average of the Q_m and the running time obtained by each algorithm on each ensemble of datasets.

The experimental results show that for COG database the algorithms CLUSS2 and FORCE obtained in average the best clustering results compared to the rest of algorithms, followed closely by ALIGNER then CD-HIT after that TransClust, SCPS, and ALIGNER*, while the other algorithms obtained relatively less good results. Nevertheless, for KOG database, the algorithms TransClust, CLUSS2 and ALIGNER showed higher performance in comparison to the other algorithms for all protein datasets, followed by FORCE, while CD-HIT, ALIGNER*, TRIBE-MCL, and gSPC obtained less good results compared to COG database. On the other hand, the running times in both tables show that CD-HIT and BlastClust are the fastest algorithms tested, followed by CLUSS2, while ALIGNER is clearly slower than ALIGNER*, while Tribe-MCL and gSPC are much slower than the rest. These results warrant further comment.

Table 1 Q_m and running time (T) averages on the COG database

Algorithm	C1	C2	C3	C4	C5	C6	Q_m av.	T av.
ALIGNER	94	95	93	97	94	95	95	156
ALIGNER*	79	81	84	87	83	79	82	109
CLUSS2	96	98	92	98	96	96	96	67
CLUSS	83	77	72	86	73	80	79	105
BlastClust	83	90	86	82	93	86	87	17
TRIB-MCL	70	75	82	69	78	76	75	408
gSPC	74	71	74	75	70	72	73	417
FORCE	99	100	86	96	98	98	96	593
TransClust	95	96	88	91	94	98	94	247
SCPS	93	93	72	89	87	86	86	144
CD-HIT	91	93	91	85	95	92	91	12

Table 2 Q_m and running time (T) averages on the KOG database

Algorithm	K1	K2	K3	K4	K5	K6	Q_m av.	T av.
ALIGNER	95	96	98	96	97	97	97	491
ALIGNER*	67	63	71	75	70	63	68	295
CLUSS2	98	97	99	98	96	98	98	183
CLUSS	85	79	76	86	82	87	83	330
BlastClust	64	31	46	38	77	50	51	64
TRIB-MCL	68	65	54	68	51	57	61	1069
gSPC	56	63	71	67	58	54	62	1123
FORCE	97	91	86	91	96	93	92	802
TransClust	100	100	99	99	100	97	99	575
SCPS	85	78	84	88	88	92	85	588
CD-HIT	91	85	69	68	93	70	79	28

* Alignment objective function used without "Sum-of-Residues" score

In Table 1 and Table 2, CLUSS performed less well than CLUSS2 because, as shown in the study presented by Kelil *et al.* [26], CLUSS tends to be less effective in clustering large protein datasets with large numbers of biochemical activities, which is the case with the generated protein subsets used in this experiment. In addition, ALIGNER obtained by far better clustering results than ALIGNER*, which confirms the effectiveness of the new alignment objective function used in ALIGNER, but with some cost in terms of efficiency compared to ALIGNER* due to the use of the pairwise matching scheme in ALIGNER.

The algorithms ALIGNER*, TRIBR-MCL, gSPC based on only global sequence alignment for measuring the pairwise similarity clearly performed better on COG than on KOG. This is because KOG includes protein sequences from eukaryotic organisms, which are known to contain more multi-domain proteins than prokaryotic organisms [56]. In fact, prokaryotes contain 40% to 65% of multi-domain proteins while eukaryotes contain ~65% to 80% [5]. And according to the findings of McClure *et al.* [35] and Thompson *et al.* [53] that have shown global alignment is less successful on multi-domain protein sequences, it becomes clear why algorithms based on only global alignment are less effective on KOG database. In contrast, although based on global alignment, the results obtained by ALIGNER on both COG and KOG databases are very comparable in quality, which confirms the significant contribution of detecting significant patterns in the capture of crucial local and remote similarities in concert with global alignment in the capture of important overall similarities between protein sequences.

es. Though ALIGNER and CLUSS utilize a strict matching scheme, ALIGNER obtains better results. This is because of the improvement in the method used to measure pairwise similarities: unlike CLUSS, ALIGNER supplements the strict matching scheme used to detect significant patterns with a pairwise alignment that captures important overall similarity information. However, there is some cost in terms of efficiency, since ALIGNER is slower than CLUSS.

Both of BlastClust and CD-HIT are approximated and simplified clustering algorithms allowing them to run much faster than other algorithms, which explain the running time they obtained, but at some cost of sensitivity, however surprisingly CD-HIT obtained competitive clustering results on COG database. This may be because CD-HIT is more adapted for protein sequences from unicellular organisms.

It is certain that ClustalW and Blast have impacted directly on the large disparity between good and less good results obtained by Tribe-MCL, gSPC, FORCE, TransClust and SCPS. The fact that the algorithms used with Blast outperform the algorithms used with ClustalW especially on KOG database show clearly that Blast outperforms ClustalW in the estimation of the similarity measure between protein sequences. This confirms the findings of Sauder *et al.* [45] that have performed large-scale comparison of protein sequence alignment algorithms, and shown that ClustalW is less effective than Blast in the detection of remote homologies between protein sequences, especially those with low sequence identity.

Globally, the clustering results obtained by ALIGNER correspond well to the known functional and structural properties of the protein sequences, according to the COG, KOG biochemical classifications. In addition, the performances shown by ALIGNER still competitive to the best mainstream clustering algorithms for the recognition of functionally and structurally related protein.

The $(\alpha/\beta)_8$ -barrel proteins group

The CAZy database [7] describes families of structurally related catalytic and carbohydrate-binding modules or functional domains of enzymes that degrade, modify, or create Glycosidic bonds. Among proteins included in CAZy database, the Glycoside Hydrolases are a widespread group of enzymes that hydrolyse the Glycosidic bond between two or more carbohydrates or between a carbohydrate and a non-carbohydrate moiety. Among Glycoside Hydrolases families, the GH2 family, extensively studied at the biochemical level includes enzymes known as the $(\alpha/\beta)_8$ -barrel proteins group that perform five distinct hydrolytic reactions. Only well-characterized and complete protein sequences were retained for this study. The database entries and the names of the $(\alpha/\beta)_8$ -barrel group are provided as supplementary material. The periodic character of the catalytic module known as “ $(\alpha/\beta)_8$ -barrel” makes this group of proteins sequences hard-to-align using classical alignment approaches [9],[14]. The difficulties in aligning these modules are comparable to the problems encountered with the alignment of tandem-repeats protein sequences [20]. For this reason that this group of proteins has been analyzed by Côté *et al.* [9] and Fukamizo *et al.* [14] using structure-based sequence alignments and biochemical structure-function studies.

So far, the group of $(\alpha/\beta)_8$ -barrel protein sequences includes “ β -galactosidase”, “ β -mannosidase”, “ β -glucuronidase” and “*exo*- β -D-glucosaminidase” enzymatic activities, all extensively studied at the biochemical level. These sequences are multi-modular, with various types of modules, which complicate their alignment [9]. Thus, the clustering of such protein sequences using alignment-dependent algorithms is seriously compromised. This encouraged us to cluster this particular group of the GH2 subfamily in order to validate the use of ALIGNER on hard-to-align protein sequences. An overview of the results is given in Table 3, with a detailed discussion below. The FASTA file as well as the corresponding names and database entries of these proteins are provided as supplementary material.

Table 3. Clustering results of the $(\alpha/\beta)_8$ -barrel proteins group

#	Proteins	Enzymatic activities									
β -mannosidase											
1	MaA	1	1	1	1	1	1	1	1	1	1
2	MaB	1	1	1	1	1	1	1	1	1	1
3	MaH	1	1	1	1	1	1	1	1	1	1
4	MaM	1	1	1	1	1	1	1	1	1	2
5	MaC	1	5	1	5	1	1	1	1	1	3
6	MaT	1	5	1	5	1	1	1	1	1	3
7	GIC	2	2	2	2	2	1	2	1	1	1
8	GIE	2	2	2	2	2	1	2	1	1	4
9	GIH	2	2	2	2	2	1	2	1	1	4
10	GIL	2	2	2	2	2	1	2	1	1	4
11	GIM	2	2	2	2	2	1	2	1	1	4
12	GIF	2	2	2	2	2	1	2	1	1	4
13	GIS	2	2	2	2	2	1	2	1	1	4
14	GaEco	3	3	3	3	3	2	2	1	1	5
15	GaA	3	3	3	3	5	3	2	1	1	5
16	GaK	3	3	3	3	6	3	2	1	1	5
17	GaC	3	3	3	3	7	3	2	1	1	5
18	GaEcl	3	3	3	3	3	2	2	1	1	5
19	GaL	3	3	3	3	3	3	2	1	1	5
20	CsAo	4	4	4	4	8	2	3	1	1	6
21	CsS	4	4	4	4	4	3	3	1	1	7
22	CsG	4	4	4	4	4	3	3	1	1	7
23	CsM	4	4	4	4	4	2	3	1	1	7
24	CsN	4	4	4	4	9	2	3	1	1	7
25	CsAn	4	4	4	4	10	2	3	1	1	7
26	CsH	4	4	4	4	4	2	3	1	1	7
27	CsE	4	4	4	4	4	2	3	1	1	7
28	UnA	5	5	5	5	11	5	1	1	1	1
29	UnBv	5	5	5	5	11	5	1	1	1	1
30	UnBc	5	5	5	5	12	5	1	1	1	1
31	UnBm	5	5	5	5	11	5	1	1	1	1
32	UnBp	5	5	5	5	11	5	1	1	1	1
33	UnR	5	5	5	5	11	5	1	1	1	1

In Table 3, the $(\alpha/\beta)_8$ -barrel proteins were successfully subdivided by ALIGNER, ALIGNER*, CLUSS2, and CLUSS corresponding to their known biochemical activities. However, contrarily to ALIGNER* and CLUSS, which have classified the two proteins MaC and MaT with the fifth cluster, ALIGNER classified all the 33 $(\alpha/\beta)_8$ -barrel proteins in the same subfamilies obtained by the more complex analysis performed by Côté *et al.* [9] that in turn are supported by the structure-function studies of Fukamizo *et al.* [14] using clustering based on structure-guided alignments, an approach which necessitated prior knowledge of at least one 3D protein structure. This shows the advantage of ALIGNER comparing to CLUSS in clustering protein sequences, and also the advantage of using the new alignment objective function used in ALIGNER. However, the rest of the algorithms failed to obtain clustering results with the same quality and precision even by tuning the values of their input parameters (i.e., Table 3 shows results obtained by default values of input parameters). In one hand, BlastClust suc-

ceeded to cluster “ β -mannosidase” and “ β -mannosidase” proteins but failed on “ β -galactosidase” and “exo- β -D-glucosaminidase” proteins. At the same time, Tribe-MCL mixed together proteins with different enzymatic activities “ β -mannosidase” and “ β -glucuronidase” and partitioned “ β -galactosidase” and “exo- β -D-glucosaminidase” enzymatic groups. In other hand, surprisingly CD-HIT succeeded to cluster almost all proteins in the appropriate groups, except for “ β -mannosidase” proteins that were subdivided into three clusters. In contrast, FORCE, TransClust, and SCPS seem to be not sufficiently sensitive to discriminate the subtle differences between the members of the $(\alpha/\beta)_8$ -barrel group from the GH2.

We conclude by, in contrast to the experimental results obtained on COG and KOG databases, the results obtained on the group of $(\alpha/\beta)_8$ -barrel proteins belonging to the GH2 family, show that a clustering algorithm such as FORCE, TransClust, or SCPS that is generally effective on easy-to-align protein sequences will not necessarily be on hard-to-align special cases such as tandem-repeat protein sequences. Moreover, it may even be outperformed by algorithms that are generally less effective on easy-to-align protein sequences such as BlastClust or CD-HIT. Therefore, it is primordial to have an algorithm that is effective on both easy-to-align and hard-to-align protein sequences such as ALIGNER. This property is very important for ALIGNER in the detection of groups of protein sequences in input protein datasets that share structural and functional properties to produce meaningful alignments.

3.2 Alignment

BALiBASE version 3.0

To evaluate the performance of ALIGNER in the alignment of protein sequences, we tested it extensively on the BALiBASE database [4] a well-known and widely utilized in the literature as a standard benchmark for testing and evaluating protein sequence alignment algorithms. The version 3.0 of BALiBASE database [52] which is to date the last version comprises more difficult alignment cases than the previous versions, corresponding better to the genuine challenges now encountered in the alignment of complex protein sequences.

The BALiBASE version 3.0 includes 6255 protein sequences divided into 744 alignments, unlike the previous versions all provided in full-length of protein sequences for all test cases, which complicates drastically the alignment task for both global and local sequence alignment algorithms. The reference alignments in BALiBASE version 3.0 are grouped into nine reference sets, each reference set includes a number of reference alignments representing a different multiple alignment problem either in terms of primary structure of proteins or in terms of percentage sequence identity. However, the first five reference sets are the most frequently used as benchmark by the alignment algorithms published, certainly because these reference sets comprise the most commonly encountered alignment problems in the real case studies. Thus, in our experiments we will focus only on the alignment of these first five reference sets that are organized as follows:

1. RV10: alignment of equidistant sequences with different percentages of identity RV11 (<20%), RV12 (20-40%).
2. RV20: alignments of families with orphans.
3. RV30: alignments of divergent subfamilies.
4. RV40: alignments of sequences with large extensions.
5. RV50: alignments of sequences with large insertions.

In this experiment, we have evaluated the different results using the two different alignment scores “*Sum-of-Pairs*” and “*Total-Columns*” described by Thompson *et al.* [53]. The *Sum-of-Pairs* score assesses the percentage of correctly aligned pairs of residues, while the *Total-Columns* score assesses the percentage of correctly aligned columns. Both scores are re-

stricted to the *core-blocks* [4], which define in each alignment those regions that can be reliably aligned. These two scores are calculated by comparing the produced alignment to the corresponding BAliBASE reference alignment by using the “*bali_score*” program for which the source code is publically available with the BAliBASE database at the website <http://bips.u-strasbg.fr/fr/Products/Databases/BAliBASE/>.

In addition, we compared ALIGNER to a large number of alignment algorithms from which some are the most utilized in the literature. Table 4 shows the list of algorithms selected, in which the reference, the version used, and the website hosting the program are provided for each algorithm. Since unlike other alignment algorithms tested ALIGNER produces for each input dataset multiple sub-alignments, therefore we compute the total alignment score of each input dataset as the weighted average of the scores obtained by the sub-alignments produced by ALIGNER weighted by the number of the sequences in each sub-alignment divided by the total number of the sequences in the input dataset. We assign to protein sequences clustered by ALIGNER as orphans an alignment score of zero. In Table 5 and Table 6, in which the top five results obtained in each column are bolded, we show the average of the *Sum-of-Pairs* and *Total-Columns* scores obtained by each alignment algorithm tested on each reference alignment set. The last column in each table shows the overall average for the alignment scores obtained by each algorithm. In addition, in Figure 5 we show also the histograms of the distributions of all *Sum-of-Pairs* scores in Figure 5.A and *Total-Columns* scores in Figure 5.B as well as running times in Figure 5.C obtained by each alignment algorithm tested. The plots in Figure 5 were built using the well-known and freely available “R” software for statistical computing <http://www.r-project.org/>. In Figure 5, for each of *Sum-of-Pairs*, *Total-Columns*, and running times, the histogram of the distribution of the results obtained by each algorithm tested is represented by , a gray box representing the

Table 4 Tested Alignment algorithms and their web-links

Algorithm	Ver.	Availability
MUSCLE [11]	3.7	http://www.drive5.com/muscle
TCOFFEE [40]	8.69	http://www.tcoffee.org
MAFFT [25]	6.809	http://align.bmri.kyushu-u.ac.jp/mafft
PRRP [16]	3.5.1	http://www.genomeist.kyoto-u.ac.jp/~aln_user
ClustalW2 [30]	2.0.12	ftp://ftp.ebi.ac.uk/pub/software/clustalw2
DIALIGN [36][37]	2.2.1	http://bibiserv.techfak.uni-bielefeld.de/dialign
DIALIGN-T [47]	0.2.2	http://dialign.tugraz.at
DIALIGN-TX [46]	1.0.2	http://dialign.tugraz.at
MULTALIN [8]	5.4.1	http://multalin.toulouse.inra.fr/multalin
PROBCONS [10]	1.12	http://probcons.stanford.edu
POA [18][32]	2	http://bioinfo.mbi.ucla.edu/poa
PSALIGN1 ⁽¹⁾ [48]		http://salilab.org/salign
PSALIGN2 ⁽²⁾ [48]		http://salilab.org/salign
KALIGN [31]		http://msa.sbc.suse.cgi-bin/msa.cgi
PROBALIGN [43]		http://probalign.nit.edu/standalone.html
PCMA [42]		ftp://iole.swmed.edu/pub/PCMA
ALIGN-M [55]	2.3	http://bioinformatics.vub.ac.be
PRANK [34]		http://www.ebi.ac.uk/goldman-srv/prank
MUMMALS [33]	1.01	http://prodatabiomed.edu/mummals/mummals.php
SAM [41]	3.5	http://compbio.soe.ucsc.edu/sam.html
AMAP [23]	2.0	http://code.google.com/p/amap-align

⁽¹⁾ PSALIGN with a modified version of PROBCONS

⁽²⁾ PSALIGN with modified version of TCOFFEE

second and the third quartiles, one horizontal line inside each box representing the median, and two vertical dashed lines above and below each gray box representing respectively the first and the fourth quartile.

In all our experiments presented in this work, the most difficult task was to execute the alignment algorithm PSALIGN based on the use of a modified version of TCOFFEE. This is because, apart the fact that it is very slow compared to all algorithms tested (see Figure 5.C), it suffers from many programming bugs that make it very difficult for an automatic benchmarking. This has forced us to execute it manually many times on each alignment to avoid bugs.

The results obtained in Table 5 and Figure 5.A show that, except the three local alignment algorithms SAM, MULTALIN, and POA that obtained globally less good results, all the algorithms tested obtained in average relatively good *Sum-of-Pairs* scores, with a clear advantage of ALIGNER, closely followed by PROBCONS, TCOFFEE, PROBALIGN, and MUMMALS, in this order. Surprisingly, DIALIGN obtained in average better *Sum-of-Pairs* scores than DIALIGN-TX, which is supposed to be an improved version of DIALIGN. In addition, we can see in Table 5 that, ALIGNER obtained better results than the other algorithms tested on the RV11, RV20, RV30, and RV50 reference sets, and also obtained results among the top five for RV12 reference set. These results show that ALIGNER is able to align correctly a high percentage of pairs of residues within the *core-blocks*, performing as well as or better than the top best algorithms.

The results obtained in Table 6 and Figure 5.B show that all the algorithms tested obtained relatively less good *Total-Columns* scores than *Sum-of-Pairs* scores obtained in Table 5. However, the results obtained in Table 6 confirm those obtained in Table 5, since again the three local alignment algorithms SAM, MULTALIN, and POA obtained globally less good results than the algorithms tested, while ALIGNER obtained in average better results followed by PROBCONS, TCOFFEE, PROBALIGN, and MAFFT. In addition, in Table 6 we can see that, ALIGNER obtained better results than the other algorithms tested on the RV20, RV30, RV40, and RV50 reference sets, and also obtained results among the top five for RV11 and RV12 reference sets. These results show that ALIGNER is by far more successful in aligning entire columns within the *core-blocks*, which makes ALIGNER more effective in discovering local conserved regions in aligned sequences.

The results obtained in Table 5 and Table 6 as well as those obtained in Figure 5.A and Figure 5.B, show two things very important. In one hand, the algorithms tested obtained less good *Total-Columns* scores than *Sum-of-Pairs* scores, which conforms that in practice it is always more difficult to correctly align entire columns than to merely correctly align pairs of residues. This makes sense when we know that, to obtain a good *Total-Columns* score an alignment algorithm should not only align pairs of residues but also align all residues belonging to the same column, which is sensibly more difficult. In other hand, although the values of the results obtained in Table 6 are smaller than those obtained in Table 5, the difference between the results obtained by the algorithms tested is more accentuated in Table 6 and Figure 5.B (i.e., $sd \approx 0.3$) than in Table 5 and Figure 5.A (i.e., $sd \approx 0.2$). This means that some algorithms have more or less difficulties to align all residues belonging to the same column.

As expected local alignment algorithms DIALIGN, DIALIGN-T, DIALIGN-TX, POA, and SAM did not outperformed global alignment algorithms in producing accurate alignments, even in the presence of N/C-terminal extensions and internal insertions. These results can be explained by the fact that local approaches are unreliable in obtaining suitable alignments outside the most conserved regions, and all of the sequences in BALiBASE version 3.0 are provided in full-length, which enlarges the alignments outside the conserved regions, which may make the alignment more difficult for local alignment algorithms. In Table 5 and Table 6 as well as in Figure 5.A and Figure 5.B, ALIGNER outperformed popular and effective algorithms, as MUSCLE, TCOFFEE, MAFFT, DIALIGN, and PROBALIGN. In addition, ALIGNER obtained much better alignment results than ALIGNER*. In Table 5 and Figure 5.A, and especially in Table 6 and Figure 5.B, we can see the

contrast between the results obtained by ALIGNER and those obtained by ALIGNER*.

It's very interesting also to remark that the most important improvements of ALIGNER in comparison to ALIGNER* concern the “*Total-Column*” scores, which means that ALIGNER is more capable to correctly align entire columns within the *core-blocks*. This confirms again the significant contribution of the method utilized in ALIGNER for detecting significant patterns in the capture of crucial local and remote similarities in concert with global alignment in the capture of important overall similarities between protein sequences.

Table 5 Sum-of-Pairs scores obtained on BALiBASE version 3.0

Algorithm	RV11	RV12	RV20	RV30	RV40	RV50	Av.
ALIGNER	0.722	0.873	0.900	0.878	0.746	0.832	0.831
ALIGNER*	0.586	0.805	0.818	0.779	0.668	0.728	0.737
MUSCLE	0.559	0.858	0.855	0.750	0.769	0.738	0.761
TCOFFEE	0.623	0.886	0.878	0.785	0.802	0.787	0.798
MAFFT	0.581	0.870	0.877	0.785	0.816	0.789	0.788
PRRP	0.537	0.852	0.858	0.756	0.687	0.708	0.744
CLUSTALW	0.490	0.820	0.820	0.689	0.698	0.668	0.707
DIALIGN	0.581	0.870	0.848	0.682	0.734	0.692	0.748
DIALIGN-T	0.438	0.822	0.822	0.686	0.730	0.704	0.704
DIALIGN-TX	0.456	0.824	0.830	0.688	0.746	0.705	0.712
MULTALIN	0.304	0.539	0.620	0.450	0.218	0.345	0.440
PROBCONS	0.626	0.885	0.879	0.788	0.813	0.790	0.801
POAV	0.321	0.451	0.587	0.436	0.203	0.373	0.414
PSALIGN1	0.592	0.880	0.872	0.754	0.799	0.774	0.783
PSALIGN2	0.533	0.860	0.841	0.722	0.769	0.739	0.749
KALIGN	0.530	0.845	0.852	0.743	0.777	0.701	0.748
PROBALIGN	0.596	0.886	0.877	0.780	0.829	0.781	0.795
PCMA	0.546	0.868	0.864	0.778	0.807	0.776	0.775
ALIGN-M	0.465	0.823	0.836	0.721	0.790	0.781	0.732
PRANK	0.474	0.816	0.818	0.693	0.670	0.685	0.701
MUMMALS	0.606	0.889	0.871	0.778	0.777	0.785	0.790
SAM	0.145	0.626	0.771	0.551	0.669	0.522	0.547
AMAP	0.468	0.846	0.829	0.677	0.665	0.688	0.706

Table 6 Total-Columns scores obtained on BALiBASE version 3.0

Algorithm	RV11	RV12	RV20	RV30	RV40	RV50	Av.
ALIGNER	0.366	0.742	0.632	0.664	0.519	0.569	0.591
ALIGNER*	0.271	0.550	0.540	0.444	0.398	0.428	0.448
MUSCLE	0.323	0.696	0.307	0.331	0.359	0.340	0.412
TCOFFEE	0.389	0.744	0.363	0.392	0.420	0.423	0.471
MAFFT	0.345	0.715	0.348	0.430	0.454	0.400	0.461
PRRP	0.313	0.681	0.351	0.344	0.294	0.326	0.408
CLUSTALW	0.241	0.642	0.263	0.264	0.308	0.265	0.351
DIALIGN	0.345	0.715	0.290	0.259	0.337	0.277	0.398
DIALIGN-T	0.209	0.621	0.254	0.271	0.346	0.299	0.347
DIALIGN-TX	0.221	0.634	0.262	0.270	0.345	0.311	0.355
MULTALIN	0.108	0.251	0.104	0.077	0.014	0.056	0.119
PROBCONS	0.388	0.747	0.379	0.421	0.431	0.413	0.480
POAV	0.137	0.177	0.115	0.089	0.007	0.077	0.113
PSALIGN1	0.350	0.738	0.352	0.312	0.406	0.382	0.443
PSALIGN2	0.278	0.698	0.301	0.356	0.390	0.374	0.412
KALIGN	0.283	0.669	0.291	0.348	0.381	0.284	0.395
PROBALIGN	0.343	0.745	0.338	0.417	0.467	0.374	0.463
PCMA	0.294	0.707	0.332	0.407	0.407	0.388	0.436
ALIGN-M	0.266	0.658	0.251	0.266	0.410	0.374	0.379
PRANK	0.228	0.604	0.256	0.269	0.269	0.268	0.334
MUMMALS	0.356	0.749	0.356	0.371	0.371	0.395	0.453
SAM	0.038	0.403	0.169	0.117	0.258	0.151	0.198
AMAP	0.242	0.678	0.275	0.250	0.281	0.303	0.359

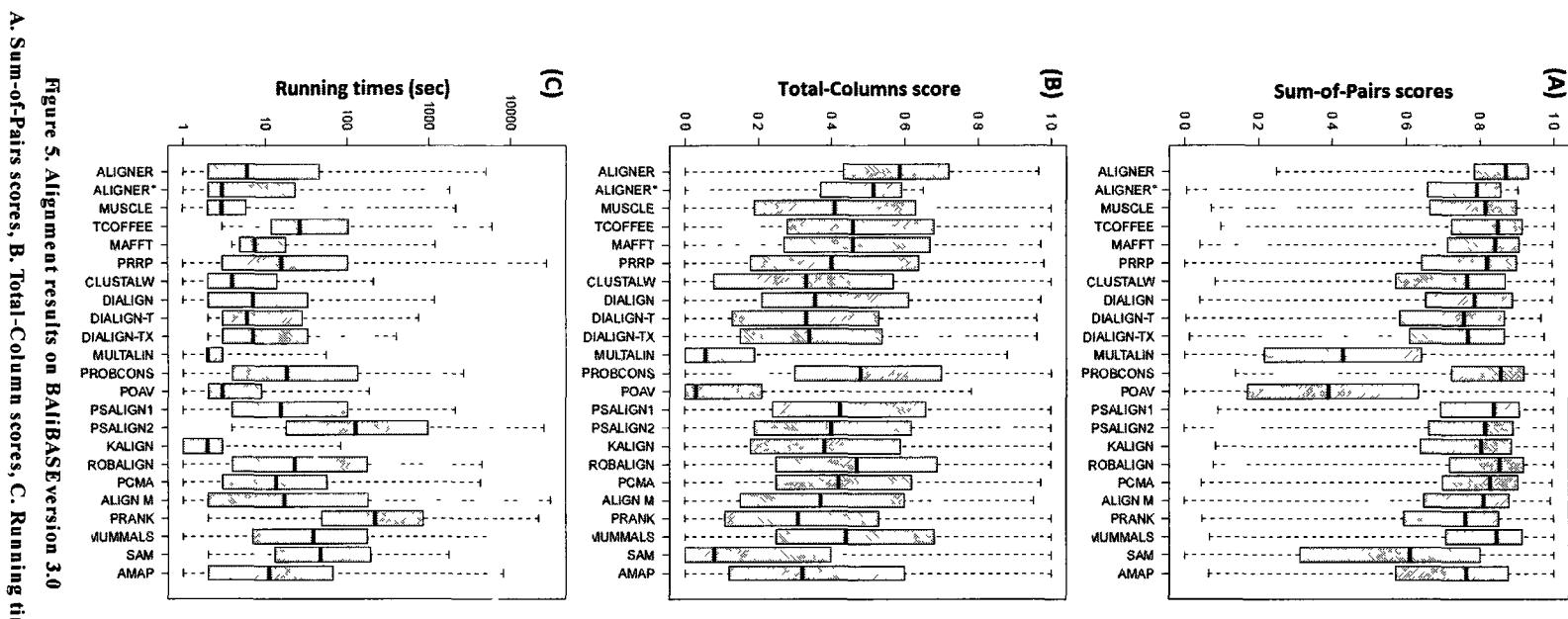


Figure 5. Alignment results on BAliBASE version 3.0
A. Sum-of-Pairs scores, B. Total-Column scores, C. Running times

In Figure 5.C, we see that the fastest algorithms are KALIGN, and MULTALIN. In contrast to MULTALIN that obtained weak alignment scores, it's very interesting to see that although it is the fastest algorithm, KALIGN still very competitive to the top best alignment algorithms in the obtaining of good alignment scores. For example, even though is more than 100 times faster than PSALIGN2 and PRANK, and more than 10 times faster than TCOFFEE and PRRP, KALIGN still obtaining comparable alignment scores to these algorithms, and very close to the top five algorithms such PROBCOMS and MUMMALS that are more than 10 time slower. In addition, while PSALIGN1 that are based on a modified version of PROBCONS obtained running times comparable to those obtained by the original PROBCONS, PSALIGN2 that is based on a modified version of TCOFFEE obtained clearly slower running times than the original TCOFFEE. This is surprising when we know that both of PSALIGN1 and PSALIGN2 have been developed to improve both of PROBCONS and TCOFFE, while the score they obtained in Figure 5.A and Figure 5.B are not better than those obtained by the original PROBCONS and TCOFFE. At the same time, the execution times obtained by ALIGNER are relatively good in comparison to those obtained by the algorithms that obtained the top five best alignment results PROBCONS, TCOFFEE, PROBALIGN, and MUMMALS, and MAFFT. We consider that the execution times obtained by ALIGNER are a good trade-off between its accuracy and efficiency.

GH46 family

Among the glycoside Hydrolase families included in the CAZy database [15], the GH46 family, extensively studied at the biochemical level, includes enzymes with Chitosanase activity. The GH46 family is difficult to align by sequence-based approaches, as the alignments obtained with various algorithms fail to reflect the biological relationships among residues established by crystallographic and biochemical approaches. To handle this type of protein family, more complex structure-

M H-K1	N174	10	29	38	48
A S P D D N F S P E T, L Q F L R N N T - G L D G E Q W N N I - M K L I N K P	- - - - - A G A G L D D' P H K K E I A M E L V ' S S A	Q D D L N W I K Y Y G Y C E D I			
		3	13	23	33
58	68	78	88	98	108
M H-K1	E B E R G Y T I G L' F G A T T G G S R D T H P D G P D L F K A Y D A A K G A S N P s, A, D G A L K R L G I N G K				
N174	G B G R G Y T G G I I G F C S G -	T G - D M L F Z V O H Y T D ' L E P G N T L A K Y L P A L K K V N G S A			
	43	53	58	68	78
118	128	138	148	158	
M H-K1	M K G S I L E I K D S E K V F C G K I K' K L Q N D A A W R K A M N E T F Y N V Y I R Y S V E Q A R' Q R G F T ' S				
N174	S H S G L - - - - G T P F T K D W A T A A K D T V F Q Q A Q N D E R D R V Y F D P A V S Q A K A D G L R' -				
	102	112	122	132	
168	178	188	195	201	211
M H-K1	P V T I G ' S F V T A L N Q G A T G G S D T L Q G L' L A R S - - - G S S' - - - S N E K T F M K N F H A K E T				
N174	A L Q G O F A Y ' Y A I V M H G P G N D P T S F G G I R K T A M K K A R T P A Q, G G D E T T Y L N A F L D A K				
	141	151	161	171	181
221		# #	250		
M H-K1	L V V D T N K Y N K P P N G K N R V K - Q W D T L V D M G K M N L K N V D S E, I A Q V T D W E M K -				
N174	A A M L T E A H D - - D - T S R V D T E Q R V F L K A G N L D L N P P L K W K T Y G D P Y V I N S				
	201	208	218	228	238

Figure 6. The alignment of Chitosanase MH-K1 and N174 obtained by Saito *et al.* [3].

Identical pairs of residues are shaded in yellow, green, and blue.

Catalytic residues are shaded in blue.

Residues forming the interaction network revealed by Fukamizo *et al.* [44] are shaded in green.

Numbers above correspond to MH-K1 residues. Numbers below correspond to N174 residues.

based alignment methods are used. However, in practice this type of method is usually expensive in time and resources and often requires prior knowledge of the 3D structure for at least one of the proteins from the studied family, as in the work of Saito *et al.* [7], Lacombe-Harvey *et al.* [44], and Fukamizo [29]. Here we aim to develop an alignment approach that would respect most if not all of the functional relationships among residues established by biochemical and structural studies. For instance, a study by Boucher *et al.* [13] revealed a conserved N-terminal module of \approx 50 residues, including two invariant carboxylic residues, Glu22 and Asp40, directly involved in the catalytic activity of the Chitosanase from *Streptomyces* sp. N174. This result was confirmed more recently by several studies on other proteins belonging to the GH46 family. After that, Saito *et al.* [6] presented an alignment between two primary structures of Chitosanase based on the best superimposition of their respective 3D structures. And recently, Lacombe-Harvey *et al.* [44] revealed a number of structurally conserved residues essential for enzyme activity. Furthermore, a theoretical and experimental study performed by Fukamizo *et al.* [29] revealed that a number of structurally conserved residues outside the catalytic module have a critical role in Chitosanase, building a network of interactions highly conserved in GH46 members which is not detected by most alignment algorithms. In the Chitosanase from *Streptomyces* sp. N174, this network includes residues Asp145, Arg190 and Arg205. So far, two 3D structures have been published for the GH46 members: that of the Chitosanase from *Bacillus circulans* MH-K1, composed of 259 residues (DDBJ accession number BAA01474; PDB file 1QGI), and that of *Streptomyces* sp. N174, composed of 238 residues (GenBank accession number AAA19865; PDB file 1CHK). Their overall three-dimensional folding is very similar even though they share only 20% identity at the sequence level. Both MH-K1 and N174 Chitosanase have a structure with two globular upper and lower domains, which generate the active site cleft for the substrate binding. However, the backbone helices that connect the two domains in the two enzymes are different. These two Chitosanase thus differ in the size and shape of the active site cleft. This structural difference explains why these enzymes differ slightly in their substrate specificity.

In our experiments, none of the alignment algorithms cited in Table 4, even ALIGNER, was able to produce an alignment identical with the one based on direct structural comparison obtained by Saito *et al.* [15], shown in Figure 6. All of them break down in aligning the two Chitosanase MH-K1 and N174, whether alone or among the other members of the GH46 family. In addition, while almost all of the algorithms were able to align the *N*-terminal section including the catalytic module, none of them could correctly align all of the structurally conserved residues outside the catalytic module, and especially the backbone helices that connect the two globular upper and lower domains revealed by Saito *et al.* [44].

In this section, our goal is to evaluate the usefulness and the effectiveness of our new alignment algorithm with the GH46 family as a case study. To this end, we will attempt to obtain an alignment of the two Chitosanase MH-K1 and N174 that would respect the functional relationships among residues established by biochemical and structural studies, and to do so without resorting to prior knowledge about the 3D structures.

First, we use ALIGNER to perform the full-length alignment of all the members of the GH46 family. After that, if ALIGNER groups MH-K1 and N174 in the same cluster, we extract and evaluate the alignment of MH-K1 and N174 from the corresponding cluster alignment. Otherwise, if ALIGNER assigns MH-K1 and N174 to two different clusters, we align the protein sequences within the corresponding clusters using ALIGNER, and extract and evaluate the alignment of MH-K1 and N174 from the alignment obtained. In this experiment, the GH46 protein sequences were retrieved from the CAZy database [44], July 2009 version. After removing all duplicates and fragments, we obtained only 35 protein sequences to align. The performed alignment yielded the clustering and the phylogenetic tree shown in Figure 7.

As shown in Figure 7, ALIGNER has assigned the two Chitosanase N174 and MH-K1 to two separate clusters, “CLUSTER2” and “CLUSTER5”, respectively. Among all the members of GH46, ALIGNER has recognized those that come from viruses and grouped them together in “CLUSTER4”, distinct from those that come from bacteria, which were separated into different clusters according to their taxonomies. For instance, all the protein sequences belonging to “CLUSTER1” include a module known as “peptidoglycan-binding” domain, while in the other clusters we find protein sequences with different and distinct taxonomies, “Actinomycetales”, “Bacillus”, and “Bacteria”.

To continue, we performed the alignment of the only two clusters that include the reference Chitosanase. Figure 8 shows the alignment obtained with ALIGNER for the protein sequences from “CLUSTER2” and “CLUSTER4”. The full-length alignment is provided with this paper as supplementary material. In Figure 8, columns coloured with yellow, blue, and green correspond to pairs of residues in N174 and MH-K1 aligned by ALIGNER and also aligned and identified as identical residues by Saito *et al.* [7], while columns coloured in gray correspond to pairs of residues in N174 and MH-K1 belonging to well aligned columns by ALIGNER but have not been assigned as identical pairs of residues by Saito *et al.* [44]. In Figure 6, it is clearly apparent that all of the 47 residues in MH-K1 and N174 considered as identical by Saito *et al.* [44] (i.e., yellow, blue, and green columns) were aligned by ALIGNER; among other things, it also aligned the 2 catalytic residues (shaded in blue) as well as the 3 residues (shaded in green) forming the interaction network revealed by Fukamizo *et al.* [44]. In addition, in Figure 8 we can also see that each of the 47 pairs of identical residues in MH-K1 and N174 was remarkably aligned by ALIGNER with identical or very similar residues from protein sequences belonging to “CLUSTER2” and “CLUSTERS”, again highlighting the important role of the identical residues revealed by Saito *et al.* [15] within the two Chitosanase MH-K1 and N174. However, well aligned columns by ALIGNER including pairs of residues between MH-K1 and N174 missed by Saito *et al.* [44] represent probably a new discovery of well conserved residues among protein sequences of the GH46 family with possibly important structural and functional role.

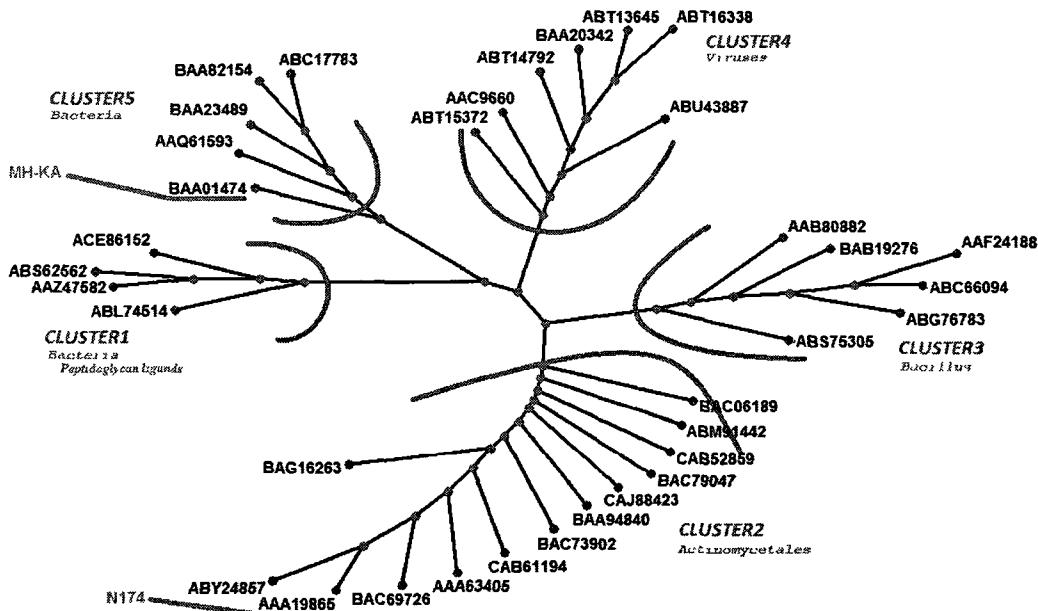


Figure 7. The phylogenetic tree and clustering result of the GH46 family obtained by ALIGNER.
Chitosanase N174 and MH-K1 are indicated in green

To summarize, ALIGNER was able to obtain an alignment that reflects the structure-function relationships revealed in structural and biochemical studies to a degree not attained by any of the other classical alignment algorithms tested, without resorting to the prior knowledge provided by the crystallographic studies, and without resorting to user random or arbitrary manipulation of the input datasets. Moreover, ALIGNER revealed a new set of residues among the members of the GH46 family probably involved in the Chitosanase.

3.3 A new alignment methodology

As a result of this real case experiment, we propose in this work a new and useful alignment methodology based on the use of ALIGNER for the alignment of protein sequences, especially those that cause problems for the classical algorithms. Usually, a conventional alignment methodology consists in, for a given input dataset of protein sequences, an alignment algorithm is used to align the input dataset, after which the result is analyzed and evaluated visually to decide if the alignment needs some improvements, which is usually the case in practice especially during first attempts. In this case, sequences are manually eliminated, retained, or even added to the input dataset in the aim to obtain an alignment that will better highlight conserved regions. This process is repeated many times until such alignment is reached. In the case when such alignment is reached with difficulties the protein sequences are called “*hard-to-align*”, or “*non-alignable*” when it is not possible to reach such alignment.

The most important drawback with such alignment methodology is, it ignores if input datasets includes “*divergent*”, “*hard-to-align*”, or “*non-alignable*” protein sequences, and leaves rather to the user the task of making the visual analysis and the necessary manipulations.

To deal with this drawback, we draw inspiration from the experimental test performed on the protein sequences from the GH46 family to introduce a new alignment methodology that will guide automatically biologists in the choice of suitable protein sequences to include in input datasets to be aligned, and avoid resorting to user random or arbitrary manipulation of the input datasets. In addition, our methodology is intended to assist and reduce the workload of biologists, by automatically handling groups of protein sequences that do not share enough conserved regions to produce significant alignments. This new methodology consists on the use of ALIGNER as follows:

1. Identify an input dataset of protein sequences with interest.
2. If needed, add a number of related protein sequences.
3. Cluster the protein sequences in the input dataset.
4. Discard orphan protein sequences.
5. If the protein of interest is an orphan, go to step 1.
6. Align clusters including protein sequences of interest.

These steps are made possible with the webserver of ALIGNER. At a first stage, we can input a preselected set of protein sequences, and after tuning the input parameters and the output formats we submit the request. We obtain then a set of clusters, and for each cluster we obtained the phylogenetic tree and the alignment of the protein sequences, in which the significant patterns in each sequence are highlighted. We obtain also the phylogenetic tree of all the clusters. The second stage consists of analyzing the obtained results (i.e., alignments, phylogenetic trees, and significant patterns), and according to our needs, (i) if the protein sequences of interest are clustered as orphans, then we should preselect another set of protein sequences to use as input dataset for the webserver; (ii) if the protein sequences of interest are clustered in the same cluster, then these clusters should be analyses in priority and may be isolated from the input dataset; (iii) if the protein sequences of

interest are in different clusters, then we select the appropriate clusters to be aligned in the second stage. In the second stage, we can also tune the *input* parameters and the *output* formats before submit the request. As a result, we obtain the phylogenetic tree and the alignment of the protein sequences from the selected clusters, in which the significant patterns in each sequence are highlighted.

4 CONCLUSION

In this paper, we have proposed an effective algorithm capable of performing well in aligning protein sequences that need either global or local alignment. Moreover, ALIGNER is able to detect groups of protein sequences that share enough significant patterns to produce alignments that can reveal important structural and biochemical properties. Compared to existing algorithms, ALIGNER yields alignments that more accurately highlight the functional characteristics of the aligned sequences. It provides biologists with a new and plausible instrument for the analysis of protein sequences, especially those that cause problems for the classical algorithms.

In addition, we present in this paper a new alignment methodology based on the use of ALIGNER algorithm, which will not only be a gain of time during alignment task (i.e., no more fumbling to remove or add sequences in input datasets), but also more effective, because ALIGNER automatically handles some steps that are performed manually using the existing algorithms.

Furthermore, the matching technique presented in this work is the key issue behind the performance of ALIGNER, and played an important role in reaching the conclusive results presented in this work, especially with “*hard-to-align*” case studies. The matching technique allowed to effectively detecting the significant unseen information behind the chronological dependencies and structural features in protein sequences. This is made possible by the use of a strong statistical theory that allowed the detection of the significant patterns that best represent the natural structure of protein sequences, and the minimization of the influence of those patterns that occur by chance and represent only noise.

So far, our alignment algorithm is based on the use of a strict pairwise matching scheme for the detection and the collection of the significant patterns. A possible future development of our alignment algorithm is the use of a permissive matching scheme that will allow the mismatches and the indels.

5 ACKNOWLEDGMENT

The authors would like to thank Dr. Miklós Csürös Professor at the University of Montreal for having taking the time to evaluate the paper, as well as for all his comments and suggestions, for which Dr. Csürös has a great contribution in the quality of this paper.

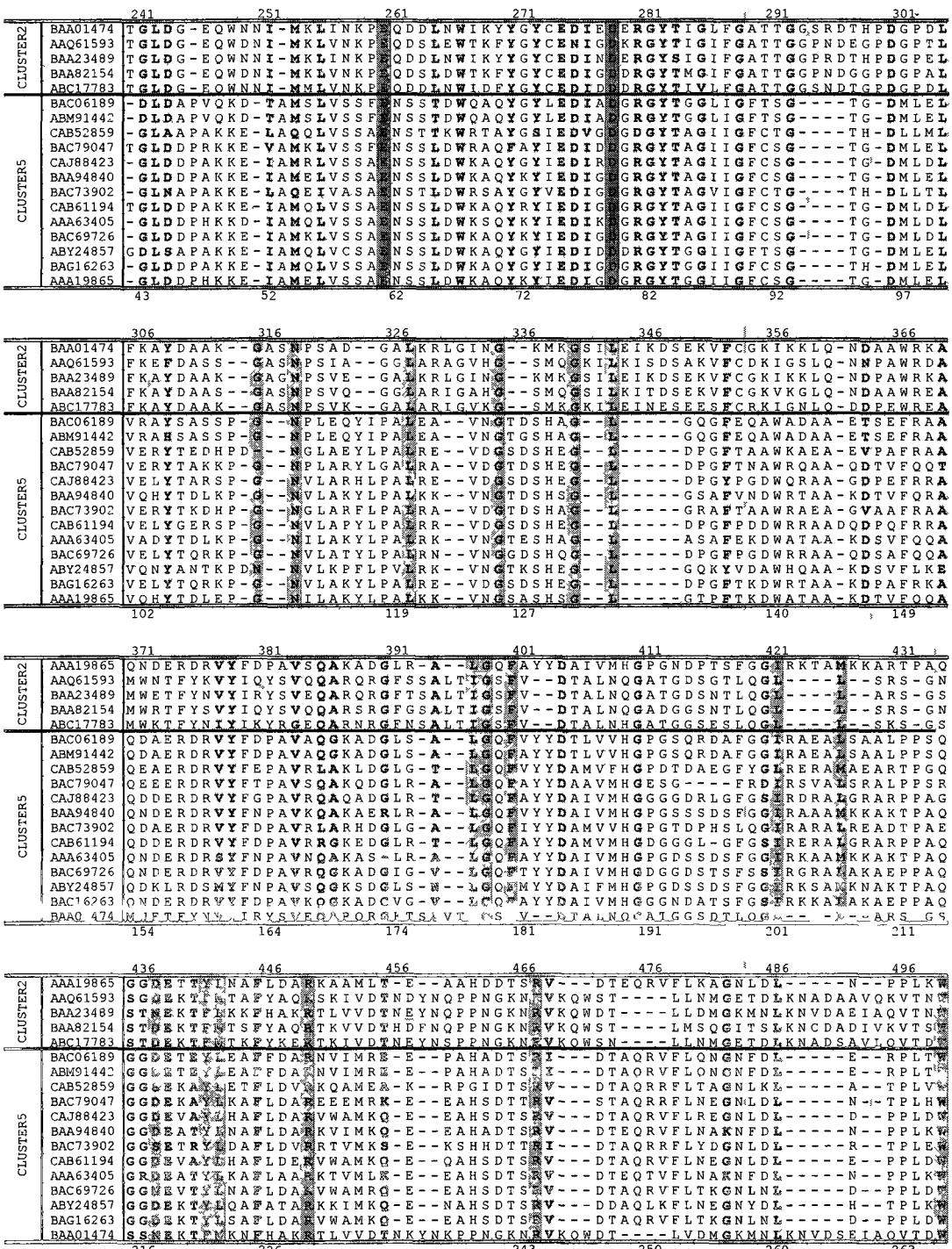


Figure 8. Alignment of CLUSTER 2 including N174 (AAA19865) with CLUSTER 5 including MH-K1 (BAA01474)

Identical pairs of residues are shaded in yellow, green, and blue. Catalytic residues are shaded in blue

Residues forming the interaction network revealed by Fukamizo et al [44] are shaded in Green

Residues shaded in gray are well aligned residues missed by Saito et al. [15]

Numbers above identify alignment columns. Numbers below correspond to N174 residues

REFERENCES

- [1] Altschul, S F , Gish, W , Miller, W , Myers, E W and Lipman, D J Basic local alignment search tool *J Mol Biol*, 215(3), 1990, 403-410
- [2] Altschul, S F , Madden, T L , Schäffer, A A , Zhang, J , Zhang, Z , Miller, W and Lipman, D J Gapped BLAST and PSI-BLAST: a new generation of protein database search programs *Nucl Acids Res*, 25(17), 1997, 3389-3402
- [3] Arel, S S and Lior, P Multiple alignment by sequence annealing *Bioinformatics*, 23(2), 2007, e24-29
- [4] Bahr, A , Thompson, J D , Thierry, J C and Poch, O BALiBASE (Benchmark Alignment dataBASE) enhancements for repeats, trans membrane sequences and circular permutations *Nucl Acids Res*, 29(1), 2001, 323-326
- [5] Batey, S , Nickson, A A and Clarke, J Studying the folding of multidomain proteins *HFSP Journal*, 2(6), 2008, 365-377
- [6] Boucher, I , Fukamizo, T , Honda, Y , Willick, G E , Neugebauer, W A and Brzezinski, R Site-directed Mutagenesis of Evolutionary Conserved Carboxylic Amino Acids in the Chitosanase from Streptomyces sp N174 Reveals Two Residues Essential for Catalysis *J Biol Chem*, 270(52), 1995, 31077-31082
- [7] Cantarel, B L , Coutinho, P M , Rancurel, C , Bernard, T , Lombard, V and Henrissat, B The Carbohydrate-Active EnZymes database (CAZy): an expert resource for Glycogenomics *Nucl Acids Res*, 37(D233-D238), 2009
- [8] Corpet, F Multiple sequence alignment with hierarchical clustering *Nucl Acids Res*, 16(22), 1988, 10881-10890
- [9] Côte, N , Fleury, A , Dumont-blanchette, E , Fukamizo, T , Mitsutom, M and Brzezinski, R Two exo- β -D-glucosaminidases/exochitosanases from actinomycetes define a new subfamily within family 2 of glycoside hydrolases *Biochem J*, 394(3), 2006, 675-686
- [10] Do, C B , Mahabhashyam, M S P , Brudno, M and Batzoglou, S ProbCons: Probabilistic consistency-based multiple sequence alignment *Genome Res*, 15(2), 2005, 330-340
- [11] Edgar, R C MUSCLE: multiple sequence alignment with high accuracy and high throughput *Nucl Acids Res*, 32(5), 2004, 1792-1797
- [12] Enright, A J , Van Dongen, S and Ouzounis, C A An efficient algorithm for large-scale detection of protein families *Nucl Acids Res*, 30(7), 2002, 1575-1584
- [13] Fukamizo, T , Amano, S , Yamaguchi, K , Yoshikawa, T , Katsumi, T , Saito, J , Suzuki, M , Miki, K , Nagata, Y and Ando, A *Bacillus circulans* MH-K1 Chitosanase Amino Acid Residues Responsible for Substrate Binding *J Biochem*, 138(5), 2005, 563-569
- [14] Fukamizo, T , Fleury, A , Côte, N , Mitsutom, M and Brzezinski, R Exo-beta-D-glucosaminidase from *A mycolatopsis orientalis*: catalytic residues, sugar recognition specificity, kinetics, and synergism *Glycobiology*, 16(11), 2006, 1064-1072
- [15] Fukamizo, T , Juffer, A H , Vogel, H H , Honda, Y , Tremblay, H , Boucher, I , Neugebauer, W A and Brzezinski, R Theoretical Calculation of pKa Reveals an Important Role of Arg205 in the Activity and Stability of *Streptomyces* sp N174 Chitosanase *J Biol Chem*, 275(33), 2000, 25633-25640
- [16] Gotoh, O Significant improvement in accuracy of multiple protein sequence alignments by iterative refinement as assessed by reference to structural alignments *J Mol Biol*, 264(4), 1996, 823-838
- [17] Gotoh, O A weighting system and algorithm for aligning many phylogenetically related sequences *Computer Applications in the Biosciences*, 11(5), 1995, 543-551
- [18] Grasso, C and Lee, C Combining partial order alignment and progressive multiple sequence alignment increases alignment speed and scalability to very large alignment problems *Bioinformatics*, 20(10), 2004, 1546-1556
- [19] Henikoff, S and Henikoff, J G Amino acid substitution matrices from protein blocks *Proc Natl Acad Sci USA*, 89(22), 1992, 10915-10919
- [20] Higgins, D G Multiple alignment In Salemi, M and Vandamme, A eds *The Phylogenetic Handbook, A Practical Approach to DNA and Protein Phylogeny* Cambridge University Press, 2004, 45-71
- [21] Hiroswa, M , Totoki, Y , Hoshida, M and Ishikawa, M Comprehensive study on iterative algorithms of multiple sequence alignment *Comput Appl Biosci*, 11(1), 1995, 13-18
- [22] Huang, Y , Niu, B , Gao, Y , Fu, L and Li, W CD-HIT Suite: a web server for clustering and comparing biological sequences *Bioinformatics*, 26(5), 2010, 680-682
- [23] Hughey, R and Krogh, A SAM Sequence Alignment and Modeling Software System University of California at Santa Cruz, Santa Cruz, CA, USA, 1995
- [24] Karlin, S and Ghassan, G Comparative statistics for DNA and protein sequences: single sequence analysis *Proc Natl Acad Sci USA*, 82(17), 1985, 5800-5804
- [25] Katoh, K and Toh, H Recent developments in the MAFFT multiple sequence alignment program *Brief Bioinform*, 9(4), 2008, 286-298
- [26] Kelil, A , Wang, S and Brzezinski, R CLUSS2: an alignment-independent algorithm for clustering protein families with multiple biological functions *IJCDD*, 1(2), 2008, 122-140
- [27] Kelil, A , Wang, S and Brzezinski, R A New Alignment-Independent Algorithm for Clustering Protein Sequences In *The 7th IEEE International Conference on Bioinformatics and Bioengineering*, Conference Center at Harvard Medical School, Cambridge, Boston, Massachusetts., 27-34
- [28] Kelil, A , Wang, S , Brzezinski, R and Fleury, A CLUSS: clustering of protein sequences based on a new similarity measure *BMC Bioinformatics*, 8, 2007, 286
- [29] Lacombe-Harvey, M , Fukamizo, T , Gagnon, J , Ghinet, G M , Dennhart, N , Letzel, T and Brzezinski, R Accessory active site residues of *Streptomyces* sp N174 chitosanase *FEBS Journal*, 276(3), 2009, 857-869
- [30] Larkin, M A , Blackshields, G , Brown, N P , Chenna, R , McGettigan, P A , McWilliam, H , Valentin, F , Wallace, I M , Wilm, A , Lopez, R , Thompson, J D , Gibson, T J and Higgins, D G Clustal W and Clustal X version 2.0 *Bioinformatics*, 23(21), 2007, 2947-2948
- [31] Lassmann, T and Sonnhammer, E Kalign - an accurate and fast multiple sequence alignment algorithm *BMC Bioinformatics*, 6(1), 2005, 298
- [32] Lee, C , Grasso, C and Sharlow, M F Multiple sequence alignment using partial order graphs *Bioinformatics*, 18(3), 2002, 452-464
- [33] Loyer, A and Goldman, N Phylogeny-Aware Gap Placement Prevents Errors in Sequence Alignment and Evolutionary Analysis *Science*, 320(5883), 2008, 1632-1635
- [34] Loyer, A and Goldman, N An algorithm for progressive multiple alignment of sequences with insertions *Proc Natl Acad Sci U S A*, 102(30), 2005, 10557-10562
- [35] McClure, M A , Vasi, T K and Fitch, W M Comparative analysis of multiple protein-sequence alignment methods [published erratum appears in *Mol Biol Evol* 1994 Sep, 11(5) 811] *Mol Biol Evol*, 11(4), 1994, 571-592
- [36] Morgenstern, B DIALIGN 2: improvement of the segment-to-segment approach to multiple sequence alignment *Bioinformatics*, 15, March 1999, 211-218
- [37] Morgenstern, B DIALIGN: multiple DNA and protein sequence alignment at BiBiServ *Nucl Acids Res*, 32(suppl_2), 2004, W33-36

- [38] Needleman, S B and Wunsch, C D A general method applicable to the search for similarities in the amino acid sequence of two proteins *J Mol Biol*, 48(3), 1970, 443-453
- [39] Nepusz, T, Sasidharan, R and Paccanaro, A SCPS a fast implementation of a spectral method for detecting protein families on a genome-wide scale *BMC Bioinformatics*, 11(1), 2010, 120
- [40] Notredame, C, Higgins, D G and Heringa, J T-Coffee A novel method for fast and accurate multiple sequence alignment *J Mol Biol*, 302(1), 2000, 205-217
- [41] Pei, J and Grishin, N V MUMMALS multiple sequence alignment improved by using hidden Markov models with local structural information *Nucl Acids Res*, 34(16), 2006, 4364-4374
- [42] Pei, J, Sadreyev, R and Grishin, N V PCMA fast and accurate multiple sequence alignment based on profile consistency *Bioinformatics*, 19(3), 2003, 427-428
- [43] Roshan, U and Livesay, D R Probalign multiple sequence alignment using partition function posterior probabilities *Bioinformatics*, 22(22), 2006, 2715-2721
- [44] Saito, J, Kita, A , Higuchi, Y , Nagata, Y , Ando, A and Miki, K Crystal Structure of Chitosanase from *Bacillus circulans* MH-K1 at 1.6-A Resolution and Its Substrate Recognition Mechanism *J Biol Chem*, 274(43), 1999, 30818-30825
- [45] Sauder, J M, Arthur, J W and Jr, R L D Large-scale comparison of protein sequence alignment algorithms with structure alignments *Proteins Structure, Function, and Genetics*, 40(1), 2000, 6-22
- [46] Subramanian, A R , Kaufmann, M and Morgenstern, B DIALIGN-TX greedy and progressive approaches for segment-based multiple sequence alignment *Algorithms for Molecular Biology*, 3, 2008, 6
- [47] Subramanian, A R , Weyer-Menkhoff, J , Kaufmann, M and Morgenstern, B DIALIGN-T An improved algorithm for segment-based multiple sequence alignment *BMC Bioinformatics*, 6(1), 2005, 66
- [48] Sze, S , Lu, Y and Yang, Q A Polynomial Time Solvable Formulation of Multiple Sequence Alignment *Journal of Computational Biology*, 13(2), 2006, 309-319
- [49] Tatusov, R L , Koonin, E V and Lipman, D J A Genomic Perspective on Protein Families *Science*, 278(5338), 1997, 631-637
- [50] Tetko, I, Faturis, A , Ruepp, A and Mewes, H Super paramagnetic clustering of protein sequences *BMC Bioinformatics*, 6(1), 2005, 82
- [51] Thompson, J D , Higgins, D G and Gibson, T J CLUSTAL W improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice *Nucl Acids Res*, 22(22), 1994, 4673-4680
- [52] Thompson, J D , Koehl, P , Ripp, R and Poch, O BALIBASE 3.0 latest developments of the multiple sequence alignment benchmark *Proteins*, 61(1), 2005, 127-136
- [53] Thompson, J D , Plewniak, F and Poch, O A comprehensive comparison of multiple sequence alignment programs *Nucl Acids Res*, 27(13), 1999, 2682-2690
- [54] Tobias, W , Dorothea, E , Sita, L , Sven, R , Mario, A , John, H M , Sebastian, B , Jens, S and Jan, B Partitioning biological data with transitivity clustering *Nature Methods*, , 2010, 419-420
- [55] Van Walle, I , Lasters, I and Wyns, L Align-m-a new algorithm for multiple alignment of highly divergent sequences *Bioinformatics*, 20(9), 2004, 1428-1435
- [56] Vogel, C , Bashton, M , Kerrison, N D , Chothia, C and Teichmann, S A Structure, function and evolution of multidomain proteins *Curr Opin Struct Biol*, 14(2), 2004, 208
- [57] Wittkop, T , Baumbach, J , Lobo, F and Rahmann, S Large scale clustering of protein sequences with FORCE -A layout based heuristic for weighted cluster editing *BMC Bioinformatics*, 8(1), 2007, 396

CONCLUSION

À travers les différents chapitres de cette thèse, nous avons présenté une série de méthodes et d'algorithmes pour faire face à plusieurs problématiques majeures dans le domaine de la bioinformatique et de la biologie cellulaire.

Dans le premier chapitre, nous avons présenté CLUSS, le premier algorithme dans la littérature conçu pour le clustering des protéines qu'elles soient alignables ou non. De plus, CLUSS est le premier algorithme à pouvoir effectuer le clustering des protéines sans faire appel à l'alignement. CLUSS est basé sur l'utilisation de notre nouvelle mesure de similarité SMS qui est capable de détecter les motifs les plus importants, ceux qui reflètent le mieux les caractéristiques structurelles et fonctionnelles des protéines. SMS est basée sur une nouvelle approche d'appariement de pairs de motifs identiques que nous avons développée. Tout d'abord, CLUSS mesure les similarités entre les protéines en utilisant SMS, qu'il utilise ensuite pour construire une représentation hiérarchique des relations entre les protéines, par la suite il évalue automatiquement l'importance de chaque protéine parmi toutes les autres, et il utilise une méthode systématique pour découvrir les clusters dans la structure hiérarchique. Pour montrer l'efficacité de CLUSS en comparaison avec les autres algorithmes de clustering existants, nous avons effectué un grand nombre de tests expérimentaux, sur différents ensembles de protéines, qui ont montré clairement que CLUSS est plus efficace pour regrouper les protéines selon leurs caractéristiques structurelles et fonctionnelles, spécialement celles qui causent des problèmes aux algorithmes basés sur l'alignement.

Dans le deuxième chapitre nous avons présenté CLUSS2, une version améliorée et plus performante de CLUSS, qui est capable de traiter plus efficacement et plus rapidement un plus grand nombre de protéines contenant beaucoup plus de caractéristiques structurelles et fonctionnelles. CLUSS2 est basé sur l'utilisation de la mesure de similarité tSMS, qui est une

version améliorée et plus performante de SMS. tSMS utilise un algorithme d'appariement des paires de séquences basé sur l'utilisation de la structure de donnée « *Suffix Tray* » ce qui permet à tSMS d'avoir une complexité moyenne linéaire par rapport aux longueurs des séquences, contrairement à la complexité quadratique de SMS. Plus encore, plutôt que de comparer juste des pairs de protéines comme dans SMS, tSMS représente les protéines dans un espace vectoriel, où chaque protéine est transformée en un vecteur en utilisant l'ensemble des protéines comme information, ce qui donne une portée globale à la mesure de similarité. Tout ceci permet à CLUSS2 d'utiliser les opérations vectorielles durant le processus de clustering, ce qui accélère considérablement la phase de clustering. CLUSS2 tire aussi avantage des approximations pendant la décomposition spectrale pour réduire encore plus le temps de calcul. Pour montrer l'efficacité de CLUSS2 en comparaison à CLUSS et aussi aux autres algorithmes de clustering existants, nous avons effectué un grand nombre de tests expérimentaux sur différents types de protéines. Les résultats ont montré clairement que CLUSS2 est plus efficace que CLUSS ainsi que les autres algorithmes testés, spécialement pour les grands ensembles de protéines contenant de grands nombres de fonctions biologiques.

Dans le troisième chapitre nous avons présenté SCS une version généralisée de SMS et tSMS, qui est une nouvelle méthode pour la mesure de similarité, qui non seulement fonctionne avec les protéines, mais aussi sur toutes les sortes de données qui ont une structure similaire à la structure primaire des protéines, communément appelées « *Séquences Catégoriques* », telles que « *les séquences biologiques* », « *le langage naturel* », « *la musique* », « *les transactions bancaires* », et « *les communications réseaux* », etc. SCS est capable de détecter dans les séquences catégoriques les dépendances chronologiques et les caractéristiques structurelles. En plus de cela, SCS est capable de faire face aux séquences catégoriques qui comprennent des caractéristiques structurelles importantes dans des positions non-équivalentes. Les excellents résultats obtenus lors de nos différents tests expérimentaux sur les différents types de séquences catégoriques des différents domaines d'applications, tels que : *la caractérisation des protéines*, *la classification du langage naturel*, *la catégorisation de la musique*, *la détection de pourriels*, *la reconnaissance de la parole*, *la*

prédition des faillites personnelles (résultats non publiés), etc. ont montré sans équivoque l'efficacité et la précision de notre nouvelle mesure de similarité ainsi que de son avantage et de sa polyvalence comparé aux autres méthodes qui sont spécifiquement développées pour des domaines bien particuliers.

Dans le quatrième chapitre nous avons présenté ALIGNER, un nouvel algorithme pour l'alignement des séquences de protéines, qui est en mesure d'aligner de manière efficace aussi bien les séquences de protéines qui nécessitent un alignement global que celles qui nécessitent un alignement local. De plus, contrairement aux algorithmes d'alignement existants, ALIGNER est capable de détecter dans les ensembles de protéines à aligner, les sous-ensembles de protéines qui partagent assez de motifs significatifs pour produire des alignements qui peuvent révéler d'importantes propriétés structurelles et fonctionnelles. ALIGNER utilise l'algorithme d'appariement que nous avons développé dans SMS pour détecter efficacement les motifs les plus significatifs partagés entre les protéines à aligner. ALIGNER utilise aussi l'algorithme de clustering développé dans CLUSS2 pour détecter automatiquement les groupes de protéines qui partagent suffisamment de propriétés structurelles et fonctionnelles pour produire des alignements biologiquement significatifs. Ensuite, il utilise une approche hiérarchique progressive et itérative en combinaison avec une nouvelle fonction objective pour l'alignement de chaque groupe de protéines. ALIGNER ne sera pas seulement un gain de temps au cours de la tâche d'alignement (c.à.d. plus besoin de tâtonner pour supprimer ou ajouter des séquences des ensembles de séquences à aligner), mais aussi plus efficace, car il gère automatiquement certaines étapes qui sont effectuées manuellement à l'aide des algorithmes existants. Lors de nos tests expérimentaux nous avons clairement démontré la supériorité de ALIGNER face aux algorithmes d'alignement existants, qu'ils soient basés sur l'alignement global ou local. Parmi les tests effectués, ALIGNER a été le seul algorithme basé sur la séquence à pouvoir obtenir un alignement biologiquement valable.

À ce jour, nos travaux ont porté sur la similarité, le clustering, et l'alignement des séquences de protéines et autres. Une prochaine étape de nos travaux serait d'adapter les différentes

méthodes que nous avons développées pour qu'elles puissent traiter aussi d'autres types de séquences biologiques, tel que les séquences ADN et ARN. Nous allons aussi mettre en profit notre expertise dans le domaine de la recherche de motifs pour la recherche des régions importantes dans les séquences biologiques comme :

- Les sites d'interactions ou de fixations dans les protéines;
- Les sites de restriction dans l'ADN;
- Les régions codantes dans le génome;
- Les facteurs de transcription dans les protéines.

Un autre aspect de nos travaux que nous allons explorer est l'utilisation des méthodes que nous avons développées pour des analyses de séquences de protéines à grande échelle. Cela nécessitera la parallélisation de nos méthodes pour qu'elles puissent s'exécuter sur des machines à multiple processeurs. L'objectif sera d'accélérer la recherche de similarités entre une protéine requête avec toute une banque de données. Le résultat d'une telle comparaison pourra alors être utilisé à des fins de clustering ou d'alignement. Ceci donnera une portée plus globale au clustering et à l'alignement, plutôt que d'utiliser un nombre restreint de séquences de protéines choisi arbitrairement par l'utilisateur.

ANNEXE 1 : ÉVALUATION DU CLUSTERING

Dans la littérature, les méthodes pour évaluer la qualité des résultats des algorithmes de clustering des séquences de protéines sont en général les suivantes : « *Sensitivité* » [134], « *Précision* » [134], et « *F-measure* » [101, 140, 150]. Initialement, ces trois méthodes ont été utilisées pour évaluer les algorithmes de classifications [33, 36, 41, 52, 85, 94, 149].

D'une manière formelle ces trois méthodes peuvent être définies comme suit. Soit un ensemble d'objets dont $C = \{C_1, \dots, C_K\}$ est le résultat du clustering, et dont $C^* = \{C_1^*, \dots, C_l^*\}$ est le clustering référence. La sensitivité ainsi que la précision utilisent le clustering référence pour évaluer la qualité du clustering résultat. La sensitivité du cluster résultat j par rapport au cluster référence i est définie par $|C_j \cap C_i^*|/|C_i^*|$, la précision du cluster résultat j par rapport au cluster référence i est définie par $|C_j \cap C_i^*|/|C_j|$. À partir de ces deux définitions, nous pouvons définir la sensitivité ainsi que la précision du clustering résultat C comme suit (pour plus de détails veuillez consultez Stein *et al.* [129]) :

$$\text{Sensitivité} = \sum_i^l \max_{j=1,\dots,K} \frac{|C_j \cap C_i^*|}{|C_i^*|}$$

$$\text{Précision} = \sum_i^l \max_{j=1,\dots,K} \frac{|C_j \cap C_i^*|}{|C_j|}$$

La F-measure est simplement une combinaison de la sensitivité et de la précision [129], elle est définie comme suit :

$$F - \text{measure} = 2 \times \frac{\text{Sensitivité} \times \text{Précision}}{\text{Sensitivité} + \text{Précision}}$$

D'une manière moins formelle, la sensibilité et la précision peuvent être définies suit :

- La sensibilité est la proportion des cas positifs qui ont été correctement identifiés par rapport à tous les cas positifs existants, elle est définie par : $VP / (VP + FN)$
- La précision est la proportion des cas positifs qui ont été correctement identifiés par rapport à tous les cas identifiés, elle est définie ainsi : $VP / (VP + FP)$

Tel que : $VP = \text{Vrais Positifs}$; $FN = \text{Faux Négatifs}$; $FP = \text{Faux Positifs}$.

En bioinformatique, la sensibilité est plus utilisée que la précision et la F-measure [15, 27, 43, 50, 59, 111, 115, 134, 140], car elle offre une évaluation des résultats sur l'exactitude de la classification (ou clustering), et elle est moins complexe à implémenter que la F-measure. Pour cela nous avons utilisé dans nos travaux de recherches la sensibilité pour le développement d'une nouvelle méthode pour l'évaluation de la qualité des résultats de clustering des séquences de protéines.

L'inconvénient de la sensibilité (même chose pour la précision et la F-measure) est qu'elle ne tient pas vraiment compte des séquences qui sont classées orphelines dans l'évaluation du clustering des séquences de protéines. Ce sont aussi des cas triviaux où la sensibilité donne les meilleurs résultats car ces clusters orphelins sont « purs ». Alors que, pour un biologiste c'est très important qu'une protéine soit classée dans un cluster avec les autres protéines de la même famille. Car, une protéine orpheline n'apporte pas d'information utile pour la prédiction de sa fonction.

Pour remédier à ce problème, nous avons introduit une nouvelle mesure de la qualité du clustering appelé Q-measure. Cette mesure vise à évaluer la capacité des algorithmes de clustering à classifier toutes les séquences de protéines d'une même famille dans le même cluster et cela sans laisser de séquences orphelines appartenant à la même famille.

La *Q-measure* est définie comme étant la « *Sensibilité* » mais pénalisée quand il y a des séquences orphelines dans le résultat du clustering. Elle est définie ainsi (OR est le nombre de séquences orphelines obtenues) :

$$Qmeasure = \frac{VP - OR}{VP + FN}$$

Notre mesure *Q-measure* a été déjà utilisée au moins à deux reprises dans des publications par d'autres chercheurs:

- Evolutionary Bioinformatics; 5:137–146; 2009
- BMC Bioinformatics; 9:394; 2008

ANNEXE 2 : COMPLEXITÉ

Actuellement, l'objectif principal de nos travaux de recherche n'est pas de développer les algorithmes les plus rapides, et ce n'est pas non plus de développer les algorithmes qui ont la meilleure complexité, vu que nous traitons seulement des ensembles de données relativement petit. Nos travaux se focalisent plutôt sur un certain nombre de problèmes et défis qui sont liés à la nature même et aux propriétés intrinsèques des données que nous traitons. Nos travaux ont pour objectif premier de développer des méthodes capables de mieux s'adapter aux particularités des données dans le but de produire des résultats biologiquement valables. Toutefois, nous présentons dans cette annexe les résultats de l'analyse de complexité des algorithmes que nous avons développés, et nous les comparons aux complexités des algorithmes existants dans la littérature.

Lors du développement de SMS et CLUSS, il ne nous a pas été possible d'établir la complexité moyenne de SMS, car cette complexité dépend essentiellement de la nature des données à traiter ainsi que le nombre et la longueur des motifs importants qu'elles contiennent, qui ne peuvent pas être déterminés à l'avance. Néanmoins, la complexité maximale de SMS est quadratique, car dans le pire cas qui arrive quand les séquences ne partagent aucun motif important, SMS va visiter toutes les paires de positions possibles entre les séquences (c.à.d. parcourir toute la « dot-plot »), ce qui a une complexité quadratique par rapport aux longueurs des séquences. CLUSS a aussi une complexité quadratique, car il utilise la méthode de Batagelj [11] pour la représentation hiérarchique des relations entre les séquences de protéines, et aussi la méthode de Thompson *et al.* [137] pour évaluer l'importance de chaque protéine dans la structure hiérarchique. Ces deux méthodes ont des complexités quadratiques par rapport au nombre des séquences. Dans le premier chapitre qui présente CLUSS et SMS, nous avons fourni pour tous les résultats obtenus les temps de

calcul de chaque algorithme utilisé. Ainsi, le lecteur peut se faire une idée générale sur le temps de calcul et la complexité moyenne de chaque algorithme.

Concernant tSMS et CLUSS2, nous avons établi mathématiquement, dans Kelil et al. [66] qui est le papier présenté dans le deuxième chapitre, que la complexité moyenne de tSMS est linéaire par rapport aux longueurs des séquences de protéines. Aussi, CLUSS2 a une complexité moyenne linéaire, car pour la représentation des séquences de protéines dans un espace multidimensionnel il utilise la décomposition spectrale qui a une complexité linéaire. Ceci est grâce à l'utilisation de l'algorithme de décompositions rapides développé par Brand [17]. Dans le deuxième chapitre qui présente CLUSS2 et tSMS, nous avons fourni pour tous les tests expérimentaux les temps de calcul de chaque algorithme utilisé.

Dans le troisième chapitre qui présente SCS, nous avons donné une estimation approximative de sa complexité moyenne, et pour appuyer cela, nous avons établi sa complexité empiriquement sur un grands nombre de séquences de différentes longueurs, et nous l'avons comparé aux autres algorithmes testés.

ALIGNER a une complexité quadratique par rapport au nombre et aux longueurs des séquences à aligner. ALIGNER est basé sur la méthode d'alignement progressif qui a une complexité quadratique par rapport au nombre de séquences à aligner. ALIGNER utilise l'algorithme de recherche de motifs utilisé dans SMS [68], et aussi l'algorithme de programmation dynamique [100], qui ont tous les deux une complexité quadratique par rapport aux longueurs des séquences. Ce qui implique qu'ALIGNER a une complexité quadratique. Cependant, nous avons montré par des tests expérimentaux présentés dans le quatrième chapitre que la complexité d'ALIGNER reste tout de même dans la moyenne des complexités des algorithmes existants.

Nous récapitulons dans les tableaux ci-dessous les complexités des algorithmes que nous avons développés, ainsi que les algorithmes que nous avons cités dans nos différents travaux. La complexité de certains algorithmes dépend des paramètres d'entrées utilisés, nous présentons leurs complexités en utilisant leurs paramètres d'entrées par défaut.

Dans les tableaux ci-dessous, N est le nombre de séquences, L est la longueur moyenne des séquences, M est la longueur maximale des séquences, et K est le nombre de modèles de Markov cachés.

Tableau 6. Complexités des algorithmes de mesure de similarité

Algorithme	Complexité moyenne	Complexité maximale	Auteurs
SMS	inconnue	$O(N^2)$	Kelil et al. [69]
tSMS	$O(N)$	$O(N^2)$	Kelil et al. [66]
SCS	$O(N)$	$O(N^2)$	Kelil et al. [65]
SAF	$O(N)$	$O(N^2)$	Kelil et al. [67]
BLAST	$O(N)$	$O(N)$	Altschul et al. [2]
Gapped-BLAST	$O(N)$	$O(N)$	Altschul et al. [3]
PSI-BLAST	$O(N)$	$O(N)$	Altschul et al. [3]
Distances de transformation	non évaluée	non évaluée	Varré et al. [142]
Scoredist	non évaluée	non évaluée	Sonnhammer et al. [128]
Alignement	$O(N^2)$	$O(N^2)$	NW [100] SW [126]
N-Grams	$O(N)$	$O(N)$	Suen [132]
K-Mers	$O(N)$	$O(N)$	Edgar [29]
Distance de Levenshtein	$O(N^2)$	$O(N^2)$	Levenshtein [81, 82]

Tableau 7. Complexités des algorithmes de clustering des séquences de protéines

Algorithme	Complexité moyenne	Complexité maximale	Observation
CLUSS	inconnue	$O(N^2)$	Kelil et al. [69]
CLUSS2	$O(N)$	$O(N^2)$	Kelil et al. [66]
BlastClust	$O(N)$	$O(N)$	Altschul et al. [2]
CD-HIT	$O(N)$	$O(N)$	Huang et al. [55]
Tribe-MCL	non évaluée	non évaluée	Enright et al. [30]
gSPC	non évaluée	non évaluée	Tetko et al. [134]
FORCE	$O(N^2)$	$O(N^2)$	Wittkop et al. [150]
TransClust	$O(N^2)$	$O(N^2)$	Baumbach et al. [140]
SCPS	$O(N^2)$	$O(N^2)$	Nepusz et al. [101]

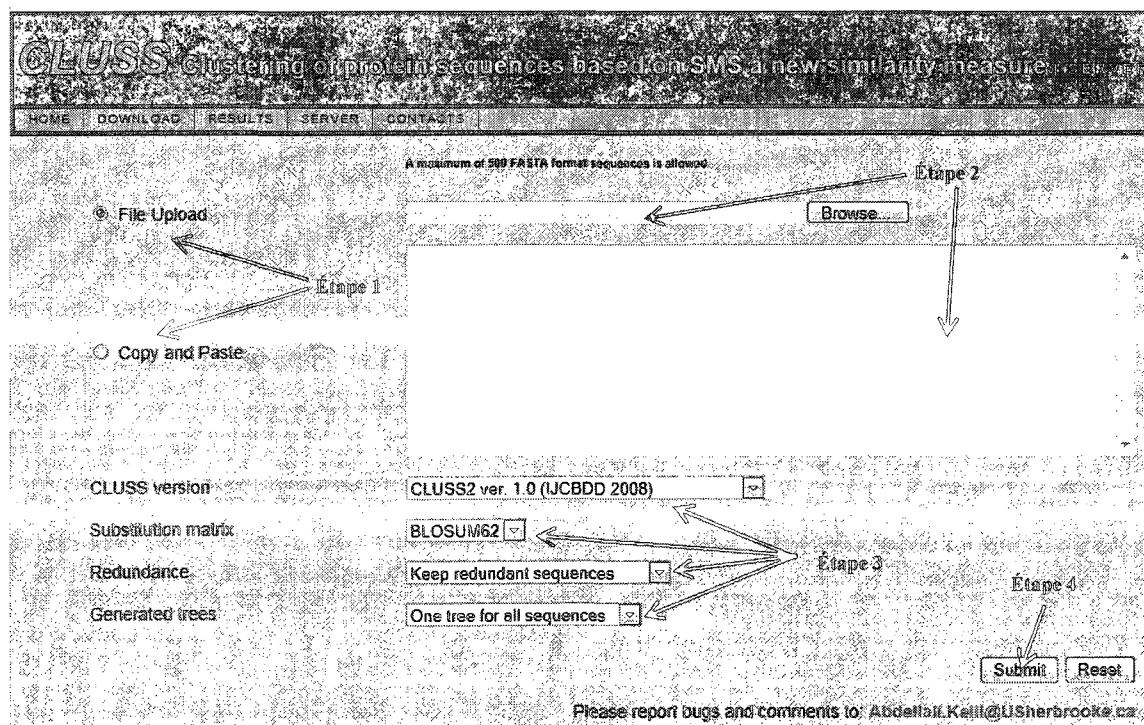
Tableau 8. Complexités des algorithmes d'alignement multiple des séquences de protéines

Algorithme	Complexité moyenne	Complexité maximale	Observation
ALIGNER	inconnue	$O(N^2L^2)$	Kelil et al.
MUSCLE	$O(N^4+NL^2)$	non évaluée	Edgar [29]
TCOFFEE	$O(N^2L^2)+O(NL^2)$	non évaluée	Notredame et al. [104]
MAFFT	$O(N^3)$	non évaluée	Katoh et al. [64]
PRRP	non évaluée	non évaluée	Gotoh [39]
ClustalW2	non évaluée	non évaluée	Larkin et al. [77]
DIALIGN	non évaluée	non évaluée	Morgenstern [97][98]
DIALIGN-T	non évaluée	non évaluée	Subramanian et al. [131]
DIALIGN-TX	non évaluée	non évaluée	Subramanian et al. [130]
MULTALIN	$(N(N-1)L^2)$	non évaluée	Corpet [20]
PROBCONS	$O(N^2L^2)$	non évaluée	Do et al. [24]
POA	$O(\alpha N^\beta)$ α et β des variables	non évaluée	Grasso et al. [40] Lee et al. [80]
PSALIGN + PROBCONS	$O(NM)+O(N^2L^2)$	non évaluée	Sze et al. [133]
PSALIGN + TCOFFEE	$O(NM)+O(N^2L^2)+O(NL^2)$	non évaluée	Sze et al. [133]
KALIGN	non évaluée	non évaluée	Lassmann et al. [78]
PROBALIGN	non évaluée	non évaluée	Roshan et al. [118]
PCMA	non évaluée	non évaluée	Pei et al. [110]
ALIGN-M	$O(N^3L)$	non évaluée	Van Walle et al. [141]
PRANK	non évaluée	non évaluée	Löytynoja et al. [89, 90]
MUMMALS	non évaluée	non évaluée	Pei et al. [109]
SAM	non évaluée	non évaluée	Hughey et al. [56]
AMAP	$N^2L^2K^2$	non évaluée	Schwartz et al. [5]

ANNEXE 3 : LE SERVEUR WEB CLUSS

Le serveur web CLUSS est situé à l'adresse <http://prospectus.usherbrooke.ca/CLUSS>. Le serveur fonctionne de la manière suivante (voir illustration en bas) :

- Étape 1 : Choisir la méthode d'entrée des données;
- Étape 2 : Introduire les données d'entrées;
- Étape 3 : Choisir l'algorithme de clustering et aussi les paramètres d'entrées;
- Étape 4 : Soumettre la requête.



The screenshot shows the CLUSS web interface. At the top, there's a navigation bar with links for HOME, DOWNLOAD, RESULTS, SERVERS, and CONTACTS. Below the navigation bar, a banner reads "CLUSS: clustering of protein sequences based on SMS, a new similarity measure". The main form area has a note: "A maximum of 500 FASTA format sequences is allowed." It includes two input methods: "File Upload" (radio button selected) and "Copy and Paste". A "Browse..." button is also present. The "File Upload" section is labeled "Etape 1". To the right, there's a "Browse..." button and a "Browse" link, labeled "Etape 2". Below these, there are several configuration options:

- CLUSS version: "CLUSS2 ver. 1.0 (JCBDD 2008)" (radio button selected)
- Substitution matrix: "BLOSUM62" (radio button selected)
- Redundancy: "Keep redundant sequences" (checkbox checked)
- Generated trees: "One tree for all sequences" (checkbox checked)

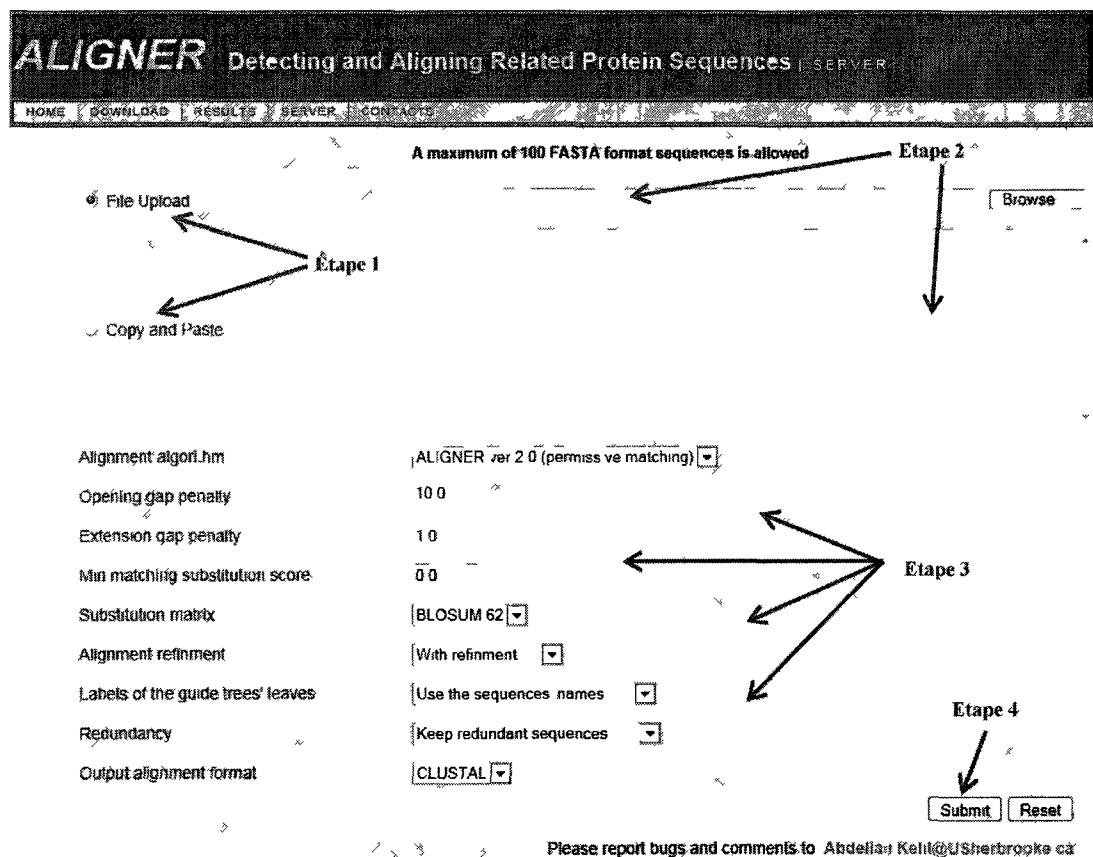
These options are labeled "Etape 3". At the bottom right are "Submit" and "Reset" buttons, labeled "Etape 4". A footer at the bottom says "Please report bugs and comments to: Abdellatif.Kelli@USherbroke.ca".

ANNEXE 4 : LE SERVEUR WEB ALIGNER

Le serveur web ALIGNER est situé à l'adresse <http://prospectus usherbrooke.ca/ALIGNER>.

Le serveur fonctionne de la manière suivante (voir illustration en bas) .

- Étape 1 : Choisir la méthode d'entrée des données;
- Étape 2 : Introduire les données d'entrées;
- Étape 3 : Choisir les paramètres d'entrées;
- Étape 4 : Soumettre la requête.



ANNEXE 5 : Liste des publications

1. Abdellali Kelil, Shengrui Wang, Ryszard Brzezinski, Fleury Alain. ***CLUSS: Clustering of protein sequences based on a new similarity measure.*** *BMC Bioinformatics*, 8(286); 2007.
2. Abdellali Kelil, Shengrui Wang, Ryszard Brzezinski. ***A New Alignment-Independent Algorithm for Clustering Protein Sequences.*** The 7th IEEE International Conference on BioInformatics and BioEngineering, October 14th-17th 2007; Conference Center at Harvard Medical School, Boston; Massachusetts; USA. **(Honorary mention for best paper award)**
3. Abdellali Kelil, Shengrui Wang, Ryszard Brzezinski. ***Clustering of Non-Alignable Protein Sequences.*** The 7th International Workshop on Data Mining in Bioinformatics, August 12th 2007; San Jose CA USA.
4. Abdellali Kelil, Shengrui Wang, Ryszard Brzezinski. ***CLUSS2: An Alignment-Independent Algorithm for Clustering Protein Families with Multiple Biological Functions.*** International Journal of Computational Biology and Drug Design, 1(2); 122-140; 2008.
5. Abdellali Kelil, Shengrui Wang, Ryszard Brzezinski. ***SAF: A Substitution and Alignment Free Similarity Measure for Protein Sequences.*** International Conference on Bioinformatics and Biomedicine, October 29th 2008, Venice, Italy.
6. Abdellali Kelil, Shengrui Wang. ***SCS: A New Similarity Measure for Categorical Sequences.*** The 8th IEEE International Conference on Data Mining, Pisa, Italy, December 15th-19th 2008.

7. Abdellali Kelil, Shengrui Wang, Qingshan Jiang, Ryszard Brzezinski. *A general measure of similarity for categorical sequences*. Springer Knowledge and Information Systems; 24(2); 197-220; 2009.
8. Abdellali Kelil, Alexei Nordell-Markovits, Shengrui Wang. *Classification of Categorical Sequences*. IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, August 23th-26th, 2009, University of Victoria, Victoria, BC, Canada.
9. Aimé Ntwari, Abdellali Kelil, Régen Drouin, Ernest Monga, Shengrui Wang, Ryszard Brzezinski, Marc Bronsard, Ju Yan. *DNAc: A clustering method for identifying kinship relations between DNA profiles using a novel similarity measure*. Journal of Forensic Sciences, 2010. doi: 10.1111/j.1556-4029.2010.01614.x
10. Abdellali Kelil, Alexei Nordell-Markovits, Parakh Ousman Yassine Z., Shengrui Wang. *CLASS: A general approach for classification categorical sequences*. IEEE Canadian Journal of Electrical and Computer Engineering, 34(4):158-166, 2009
11. Abdellali Kelil, Ryszard Brzezinski, Shengrui Wang. *ALIGNER: Detecting and aligning related protein sequences*. BMC Bioinformatics, 2010. (submitted)

ANNEXE 6 : Aperçu des travaux publiés en lien avec nos recherches

Les travaux réalisés pour cette thèse ont déjà un certain impact. Ci-dessous, je fournis une liste des travaux de recherches publiés qui ont utilisé ou cité nos travaux.

Articles :

- Q. Dai, T. Wang. *Comparison study on k-word statistical measures for protein: From sequence to 'sequence space'*. *BMC Bioinformatics* 2008.
- X. Yang, S. Jawdy, T. J. Tschaplinski, G. A. Tuskan. *Genome-wide identification of lineage-specific genes in *Arabidopsis*, *Oryza* and *Populus**. *Genomics* 2009.
- T. David, A. Stephane, L. Celine, B. Frederique. *Computer-Assisted Automatic Classifications, Storage, Queries and Functional Assignments of Orthologs and In-Paralogs Proteins*. *Current Bioinformatics* 2009.
- J. Cai, E. Borenstein, R. Chen, D. Petrov. *Similarly Strong Purifying Selection Acts on Human Disease Genes of All Evolutionary Ages*. *Genome Biol. Evol.* 2009.
- F. Yang, Q. Zhu, D. Tang, M. Zhao. *Using Affinity Propagation Combined Post-processing to Cluster Protein Sequences*. *Protein & Peptide Letters* 2009.
- F. L. Emediato, F. A. C. Nunes, C. de Camargo Teixeira, M. A. N. Passos, D. J. Bertioli, G. J. Pappas Jr., R. N. G. Miller. *Characterization of Resistance Gene Analogs in *Musa acuminata* Cultivars Contrasting in Resistance to Biotic Stresses*.

Q.Y. Shu (ed.), Induced Plant Mutations in the Genomics Era. Food and Agriculture Organization of the United Nations, Rome 2009, 443-445.

- B. Georgi, J. Schultz, A. Schliep. *Partially-supervised protein subclass discovery with simultaneous annotation of functional residues*. *BMC Structural Biology* 2009.
- F. Yang, Q. X. Zhu, D. M. Tang, M. Y. Zhao. *Clustering Protein Sequences Using Affinity Propagation Based on an Improved Similarity Measure*. *Evolutionary Bioinformatics* 2009.
- C. J. Noël, N. Diaz, T. Sicheritz-Ponten, L. Safarikova, J. Tachezy, P. Tang, P. L. Fiori, R. P. Hirt. *Trichomonas vaginalis vast BspA-like gene family: evidence for functional diversity from structural organisation and transcriptomics*. *BMC Genomics* 2010.
- J. Martin, K. Anamika, N. Srinivasan. *Classification of Protein Kinases on the Basis of Both Kinase and Non-Kinase Regions*. *PLoS ONE* 2010.
- A. Albayrak, H. H. Otu, U. O. Sezerman. *Clustering of protein families into functional subtypes using Relative Complexity Measure with reduced amino acid alphabets*. *BMC Bioinformatics* 2010.
- C. Frech, N. Chen. *Genome-Wide Comparative Gene Family Classification*. *PLoS ONE* 2010.

Conférence internationales :

- N. Abu Osman, F. Ibrahim, W. Wan Abas, H. Abdul Rahman, H. Ting. *A Review on Protein Sequence Clustering Research*. BIOMED 2008, Kuala Lumpur, Malaysia.
- M. Zhou, D. Theunissen, M. Wels, R. Siezen. *Genome-scale comparative analysis of the predicted secretomes of 19 sequenced lactic acid bacteria*. ISMB 2008, Toronto, Canada.

- A. Sakhinah, J. Taheri, A. Y. Zomaya. *Fuzzy systems modeling for protein-protein interaction prediction in *Saccharomyces cerevisie**. IMACS / MODSIM 2009.
- D. Tang, Q. Zhu, Y. Zhang, J. Zhang. *An online cluster analysis method for large-scale protein sequences*. BioMedical Information Engineering 2009.

Livres :

- Y. Khudyakov. *Medicinal Protein Engineering*. CRC Press, 2008, ISBN 0849373689.
- J. Bujnicki. *Prediction of Protein Structures, Functions, and Interactions*. Published Online: 18 Dec 2008, Copyright © 2009 John Wiley & Sons, Ltd.
- A. Daskalaki. *Handbook of Research on Systems Biology Applications in Medicine*, Volume 2, November 2008.

REFERENCES

- [1] Abdul Rahman, S., Bakar, A. A. and Hussein, Z. A. M. : A Review on Protein Sequence Clustering Researc. In Anonymous : *4th Kuala Lumpur International Conference on Biomedical Engineering 2008*. Springer Berlin Heidelberg, 2008, 275-278.
- [2] Altschul, S. F., Gish, W., Miller, W., Myers, E. W. and Lipman, D. J. Basic local alignment search tool. *J. Mol. Biol.*, 215(3), 1990, 403-410.
- [3] Altschul, S. F., Madden, T. L., Schäffer, A. A., Zhang, J., Zhang, Z., Miller, W. and Lipman, D. J. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucl. Acids Res.*, 25(17), 1997, 3389-3402.
- [4] Anfinsen, C. B. Principles that govern the folding of protein chains. *Science*, 181(4096), 1973, 223-230.
- [5] Ariel, S. S. and Lior, P. Multiple alignment by sequence annealing. *Bioinformatics*, 23(2), 2007, e24-29.
- [6] Bahr, A., Thompson, J. D., Thierry, J. C. and Poch, O. BALiBASE (Benchmark Alignment dataBASE): enhancements for repeats, transmembrane sequences and circular permutations. *Nucl. Acids Res.*, 29(1), 2001, 323-326.
- [7] Bailey, T. L. and Gribkov, M. Combining evidence using p-values: application to sequence homology searches. *Bioinformatics*, 14(1), 1998, 48-54.
- [8] Bailey, T. and Elkan, C. Fitting a Mixture Model by Expectation Maximization to Discover Motifs in Biopolymers. In *Proceedings of the Second International Conference on Intelligent Systems for Molecular Biology*, , 28-36.
- [9] Bandyopadhyay, S., Sharan, R. and Ideker, T. Systematic identification of functional orthologs based on protein network comparison. *Genome Res.*, 16(3), 2006, 428-435.
- [10] Barabasi, A. and Oltvai, Z. N. Network biology: understanding the cell's functional organization. *Nat. Rev. Genet.*, 5(2), 2004, 101-113.

- [11] Batagelj, V. Generalized Ward and Related Clustering Problems. In *H.H. Bock (Ed.), Classification and Related Methods of Data Analysis*, Amsterdam, 67-74.
- [12] Berry, M. W. and Fierro, R. D. Low-Rank Orthogonal Decompositions for Information Retrieval Applications. *Numerical Linear Algebra Applications*, 1, 1996, 1-27.
- [13] Blake, C. C. F., Koenig, D. F., Mair, G. A., North, A. C. T., Phillips, D. C. and Sarma, V. R. Structure of Hen Egg-White Lysozyme: A Three-dimensional Fourier Synthesis at 2 [angstrom] Resolution. *Nature*, 206(4986), 1965, 757-761.
- [14] Bogan-Marta, A., Laskaris, N., Gavrielides, M. A., Pitas, I. and Lyroudia, K. A Novel Efficient Protein Similarity Measure Based On N-Gram Modeling. In Costa da Caparica, Lisbon, Portugal, 29th June - 1st July.
- [15] Bolten, E., Schliep, A., Schneckener, S., Schomburg, D. and Schrader, R. Clustering protein sequences-structure prediction by transitive homology. *Bioinformatics*, 17(10), 2001, 935-941.
- [16] Boucher, I., Fukamizo, T., Honda, Y., Willick, G. E., Neugebauer, W. A. and Brzezinski, R. Site-directed Mutagenesis of Evolutionary Conserved Carboxylic Amino Acids in the Chitosanase from Streptomyces sp. N174 Reveals Two Residues Essential for Catalysis. *J. Biol. Chem.*, 270(52), 1995, 31077-31082.
- [17] Brand, M. Fast low-rank modifications of the thin singular value decomposition. *Linear Algebra and its Applications*, 415(1), 2006, 20.
- [18] Cantarel, B. L., Coutinho, P. M., Rancurel, C., Bernard, T., Lombard, V. and Henrissat, B. The Carbohydrate-Active EnZymes database (CAZy): an expert resource for Glycogenomics. *Nucl. Acids Res.*, 37(D233–D238), 2009.
- [19] Chen, Y., Reilly, K., Sprague, A. and Guan, Z. SEQOPTICS: a protein sequence clustering system. *BMC Bioinformatics*, 7(Suppl 4), 2006, S10.
- [20] Corpet, F. Multiple sequence alignment with hierarchical clustering. *Nucl. Acids Res.*, 16(22), 1988, 10881-10890.
- [21] Côté, N., Fleury, A., Dumont-blanchette, É., Fukamizo, T., Mitsutomi, M. and Brzezinski, R. Two exo- β -D-glucosaminidases/exochitosanases from actinomycetes

- define a new subfamily within family 2 of glycoside hydrolases. *Biochem. J.*, 394(3), 2006, 675-686.
- [22] Dai, Q. and Wang, T. Comparison study on k-word statistical measures for protein: From sequence to 'sequence space'. *BMC Bioinformatics*, 9(1), 2008, 394.
 - [23] Dayhoff, M. O., Schwartz, R. M. and Orcutt, B. C. A Model of Evolutionary Change in Proteins. *Atlas of Protein Sequence and Structure*, 5(3), 1978, 345-352.
 - [24] Do, C. B., Mahabhashyam, M. S. P., Brudno, M. and Batzoglou, S. ProbCons: Probabilistic consistency-based multiple sequence alignment. *Genome Res.*, 15(2), 2005, 330-340.
 - [25] Dong, Q., Wang, X. and Lin, L. Application of latent semantic analysis to protein remote homology detection. *Bioinformatics*, 22(3), 2006, 285-290.
 - [26] Drenth, J., Jansonius, J. N., Koekoek, R., Swen, H. M. and Wolthers, B. G. Structure of Papain. *Nature*, 218(5145), 1968, 929-932.
 - [27] Edgar, R. C. Search and clustering orders of magnitude faster than BLAST. *Bioinformatics*, 26(19), 2010, 2460-2461.
 - [28] Edgar, R. C. Local homology recognition and distance measures in linear time using compressed amino acid alphabets. *Nucl. Acids Res.*, 32(1), 2004, 380-385.
 - [29] Edgar, R. C. MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucl. Acids Res.*, 32(5), 2004, 1792-1797.
 - [30] Enright, A. J., Van Dongen, S. and Ouzounis, C. A. An efficient algorithm for large-scale detection of protein families. *Nucl. Acids Res.*, 30(7), 2002, 1575-1584.
 - [31] Everitt, B. S., Landau, S. and Leese, M. eds. : *Cluster Analysis*. , 2001.
 - [32] Felsenstein, J. An Alternating Least Squares Approach to Inferring Phylogenies from Pairwise Distances. *Syst. Biol.*, 46(1), 1997, 101-111.
 - [33] Forbes, A. Classification-algorithm evaluation: Five performance measures based on confusion matrices. *J. Clin. Monit. Comput.*, 11(3), 1995, 189-206.
 - [34] Forney, G. D., Jr. The viterbi algorithm. *Proceedings of the IEEE*, 61(3), 1973, 268.
 - [35] Friedman, Y. *Building Biotechnology: Business, Regulations, Patents, Law, Politics, Science*. ThinkBiotech LLC, Washington DC, 2008.

- [36] Frunza, O., Inkpen, D. and Matwin, S. Building Systematic Reviews Using Automatic Text Classification Techniques. In *Coling 2010: Posters*, AugustBeijing, China, 303-311.
- [37] Fukamizo, T., Fleury, A., Côté, N., Mitsutomi, M. and Brzezinski, R. Exo-beta-D-glucosaminidase from Amycolatopsis orientalis: catalytic residues, sugar recognition specificity, kinetics, and synergism. *Glycobiology*, 16(11), 2006, 1064-1072.
- [38] Ganapathiraju, M., Klein-Seetharaman, J., Balakrishnan, N. and Reddy, R. Characterization of Protein Secondary Structure Using Latent Semantic Analysis. 2004, .
- [39] Gotoh, O. Significant improvement in accuracy of multiple protein sequence alignments by iterative refinement as assessed by reference to structural alignments. *J. Mol. Biol.*, 264(4), 1996, 823-838.
- [40] Grasso, C. and Lee, C. Combining partial order alignment and progressive multiple sequence alignment increases alignment speed and scalability to very large alignment problems. *Bioinformatics*, 20(10), 2004, 1546-1556.
- [41] Grega, M., Leszczuk, M., Dupлага, M. and Fraczek, R. : Algorithms for Automatic Recognition of Non-informative Frames in Video Recordings of Bronchoscopic Procedures. In Pie'tka, E. and Kawa, J. eds.: *Information Technologies in Biomedicine*. Springer Berlin / Heidelberg, 2010, 535-545.
- [42] Guralnik, V. and Karypis, G. A Scalable Algorithm for Clustering Protein Sequences. In *In Workshop on Data Mining in Bioinformatics*, , 73-80.
- [43] Hannenhalli, S. and Levy, S. Promoter prediction in the human genome. *Bioinformatics*, 17(suppl 1), 2001, S90-S96.
- [44] Hanselmann, M., Kirchner, M., Renard, B. Y., Amstalden, E. R., Glunde, K., Heeren, R. M. A. and Hamprecht, F. A. Concise Representation of Mass Spectrometry Images by Probabilistic Latent Semantic Analysis. *Anal. Chem.*, 80(24), 2008, 9649-9658.
- [45] Harlow, T., Gogarten, J. P. and Ragan, M. A hybrid clustering approach to recognition of protein families in 114 microbial genomes. *BMC Bioinformatics*, 5(1), 2004, 45.

- [46] Henikoff, S. and Henikoff, J. G. Amino acid substitution matrices from protein blocks. *Proc. Natl. Acad. Sci. USA*, 89(22), 1992, 10915-10919.
- [47] Henikoff, S. and Henikoff, J. G. Automated assembly of protein blocks for database searching. *Nucl. Acids Res.*, 19(23), 1991, 6565-6572.
- [48] Henikoff, S., Henikoff, J. G. and Hughes, H. Position-based sequence weights. *J. Mol. Biol.*, 243, 1994, 574-578.
- [49] Higgins, D. G. : Multiple alignment. In Salemi, M. and Vandamme, A. eds.: *The Phylogenetic Handbook, A Practical Approach to DNA and Protein Phylogeny*. Cambridge University Press, 2004, 45-71.
- [50] Higham, D. J., Kalna, G. and Kibble, M. Spectral clustering and its use in bioinformatics. *J. Comput. Appl. Math.*, 204(1), 2007, 25.
- [51] Hirosawa, M., Totoki, Y., Hoshida, M. and Ishikawa, M. Comprehensive study on iterative algorithms of multiple sequence alignment. *Comput. Appl. Biosci.*, 11(1), 1995, 13-18.
- [52] Holden, N. P. and Freitas, A. A. A hybrid PSO/ACO algorithm for classification. In *Proceedings of the 2007 GECCO conference companion on Genetic and evolutionary computation*, London, United Kingdom, New York, NY, USA, 2745-2750.
- [53] Holm, L. and Sander, C. Removing near-neighbour redundancy from large protein sequence collections. *Bioinformatics*, 14(5), 1998, 423-429.
- [54] Horst, S. Symbols and Computation A Critique of the Computational Theory of Mind. *Minds Mach.*, 9(3), 1999, 347-381.
- [55] Huang, Y., Niu, B., Gao, Y., Fu, L. and Li, W. CD-HIT Suite: a web server for clustering and comparing biological sequences. *Bioinformatics*, 26(5), 2010, 680-682.
- [56] Hughey, R. and Krogh, A. *SAM: Sequence Alignment and Modeling Software System*. University of California at Santa Cruz, Santa Cruz, CA, USA, 1995.
- [57] Jain, A. K. and Dubes, R. C. *Algorithms for clustering data*. Prentice-Hall, Inc, Upper Saddle River, NJ, USA, 1988.

- [58] Jothi, R., Zotenko, E., Tasneem, A. and Przytycka, T. M. COCO-CL: hierarchical clustering of homology relations based on evolutionary correlations. *Bioinformatics*, 22(7), 2006, 779-788.
- [59] Kapetanovic, I. M., Rosenfeld, S. and Izmirlian, G. Overview of commonly used bioinformatics methods and their applications. *Ann. N. Y. Acad. Sci.*, 1020, 2004, 10-21.
- [60] Karlin, S. and Ghassan, G. Comparative statistics for DNA and protein sequences: multiple sequence analysis. *Proc. Natl. Acad. Sci. U. S. A.*, 82(18), 1985, 6186-6190.
- [61] Karlin, S. and Ghassan, G. Comparative statistics for DNA and protein sequences: single sequence analysis. *Proc. Natl. Acad. Sci. USA*, 82(17), 1985, 5800-5804.
- [62] Kartha, G. Tertiary Structure of Ribonuclease. *Nature*, 214(5085), 1967, 234-330.
- [63] Katoh, K., Kuma, K., Toh, H. and Miyata, T. MAFFT version 5: improvement in accuracy of multiple sequence alignment. *Nucl. Acids Res.*, 33(2), 2005, 511-518.
- [64] Katoh, K. and Toh, H. Recent developments in the MAFFT multiple sequence alignment program. *Brief Bioinform.*, 9(4), 2008, 286-298.
- [65] Kelil, A. and Wang, S. SCS: A New Similarity Measure for Categorical Sequences. In *ICDM '08: Proceedings of IEEE International Conference on Data Mining*, DecemberPisa, Italy, 498-505.
- [66] Kelil, A., Wang, S. and Brzezinski, R. CLUSS2: an alignment-independent algorithm for clustering protein families with multiple biological functions. *IJCBD*, 1(2), 2008, 122-140.
- [67] Kelil, A., Wang, S. and Brzezinski, R. SAF: A Substitution and Alignment Free Similarity Measure for Protein Sequences. In Venice, Italy, OCTOBER 29-31.
- [68] Kelil, A., Wang, S. and Brzezinski, R. Clustering of Non-Alignable Protein Sequences. In San Jose, CA, USA, August 12.
- [69] Kelil, A., Wang, S., Brzezinski, R. and Fleury, A. CLUSS: clustering of protein sequences based on a new similarity measure. *BMC Bioinformatics*, 8, 2007, 286.
- [70] Kim, J., Pramanik, S. and Chung, M. J. Multiple sequence alignment using simulated annealing. *Comput. Appl. Biosci.*, 10(4), 1994, 419-426.

- [71] Kimura, M. Evolutionary rate at the molecular level. *Nature*, 217(5129), 1968, 624-626.
- [72] Kocsor, A., Kertesz-Farkas, A., Kajan, L. and Pongor, S. Application of compression-based distance measures to protein sequence classification: a methodological study. *Bioinformatics*, 22(4), 2006, 407-412.
- [73] Kondrak, G. N-Gram Similarity and Distance. In *SPIRE*, , 115-126.
- [74] Krause, A., Stoye, J. and Vingron, M. Large scale hierarchical clustering of protein sequences. *BMC Bioinformatics*, 6(1), 2005, 15.
- [75] Kriventseva, E. V., Fleischmann, W., Zdobnov, E. M. and Apweiler, R. CluSTR: a database of clusters of SWISS-PROT+TrEMBL proteins. *Nucl. Acids Res.*, 29(1), 2001, 33-36.
- [76] Kulkarni, A., Noronha, A., Roy, S. and Angadi, S. Fuzzy pattern extraction for classification of protein sequences. In *ISB '10: Proceedings of the International Symposium on Biocomputing*, Calicut, Kerala, India, New York, NY, USA, 1-4.
- [77] Larkin, M. A., Blackshields, G., Brown, N. P., Chenna, R., Mcgettigan, P. A., Mcwilliam, H., Valentin, F., Wallace, I. M., Wilm, A., Lopez, R., Thompson, J. D., Gibson, T. J. and Higgins, D. G. Clustal W and Clustal X version 2.0. *Bioinformatics*, 23(21), 2007, 2947-2948.
- [78] Lassmann, T. and Sonnhammer, E. Kalign - an accurate and fast multiple sequence alignment algorithm. *BMC Bioinformatics*, 6(1), 2005, 298.
- [79] Lecompte, O., Thompson, J. D., Plewniak, F., Thierry, J. and Poch, O. Multiple alignment of complete sequences (MACS) in the post-genomic era. *Gene*, 270(1-2), 2001, 17.
- [80] Lee, C., Grasso, C. and Sharlow, M. F. Multiple sequence alignment using partial order graphs. *Bioinformatics*, 18(3), 2002, 452-464.
- [81] Levenshtein, V. I. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, 10, 1966, 707.
- [82] Levenshtein, V. I. *Binary codes capable of correcting deletions, insertions, and reversals*. 10. , 1966.

- [83] Li, M., Badger, J. H., Chen, X., Kwong, S., Kearney, P. and Zhang, H. An information-based sequence distance and its application to whole mitochondrial genome phylogeny. *Bioinformatics*, 17(2), 2001, 149-154.
- [84] Li, W., Jaroszewski, L. and Godzik, A. Clustering of highly homologous sequences to reduce the size of large protein databases. *Bioinformatics*, 17(3), 2001, 282-283.
- [85] Liu, Z., Tan, M. and Jiang, F. Regularized F-Measure Maximization for Feature Selection and Classification.
- [86] Lodish, H., Berk, A., Kaiser, C. A., Krieger, M., Scott, M. P., Bretscher, A., Ploegh, H. and Matsudaira, P. : Basic Molecular Genetic Mechanisms. In Anonymous : *Molecular Cell Biology*. W.H. Freeman and Co., 2008, .
- [87] Lodish, H., Berk, A., Kaiser, C. A., Krieger, M., Scott, M. P., Bretscher, A., Ploegh, H. and Matsudaira, P. : Protein Structure and Function. In Anonymous : *Molecular Cell Biology*. W.H. Freeman and Co., 2008, .
- [88] Lodish, H., Berk, A., Matsudaira, P. and Kaiser, C. A. *Molecular Cell Biology*. W.H. Freeman and Co., New York and Basingstoke USA, 2005.
- [89] Löytynoja, A. and Goldman, N. Phylogeny-Aware Gap Placement Prevents Errors in Sequence Alignment and Evolutionary Analysis. *Science*, 320(5883), 2008, 1632-1635.
- [90] Löytynoja, A. and Goldman, N. An algorithm for progressive multiple alignment of sequences with insertions. *Proc. Natl. Acad. Sci. U. S. A.*, 102(30), 2005, 10557-10562.
- [91] Manning, C. D., Raghavan, P. and Schütze, H. *Introduction to Information Retrieval*. Cambridge University Press}, , 2008.
- [92] Marcotte, E. M., Monzingo, A. F., Ernst, S. R., Brzezinski, R. and Robertas, J. D. X-ray structure of an anti-fungal chitosanase from *streptomyces* N174. *Nat. Struct. Biol.*, 3, 1996, 155-162.
- [93] McClure, M. A., Vasi, T. K. and Fitch, W. M. Comparative analysis of multiple protein-sequence alignment methods. *Mol. Biol. Evol.*, 11(4), 1994, 571-592.

- [94] Mete, M., Hennings, L., Spencer, H. and Topaloglu, U. Automatic identification of angiogenesis in double stained images of liver tissue. *BMC Bioinformatics*, 10(Suppl 11), 2009, S13.
- [95] Mhamdi, F., Rakotomalala, R. and Elloumi, M. A Hierarchical N-Grams Extraction Approach for Classification Problem. In *IEEE International Conference on Signal-Image Technology and Internet-Based Systems (SITIS 06)*, Hammamet Tunisia, December 17-21, 310-321.
- [96] Mitrophanov, A. Y. and Borodovsky, M. Statistical significance in biological sequence analysis. *Briefings in Bioinformatics*, 7(1), 2006, 2-24.
- [97] Morgenstern, B. DIALIGN 2: improvement of the segment-to-segment approach to multiple sequence alignment. *Bioinformatics*, 15, March 1999, 211-218.
- [98] Morgenstern, B. DIALIGN: multiple DNA and protein sequence alignment at BiBiServ. *Nucl. Acids Res.*, 32(suppl_2), 2004, W33-36.
- [99] Mount, D. W. ed. : *Bioinformatics: Sequence and Genome Analysis (2nded.)*. Cold Spring Harbor Laboratory Press, , 2004.
- [100] Needleman, S. B. and Wunsch, C. D. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.*, 48(3), 1970, 443-453.
- [101] Nepusz, T., Sasidharan, R. and Paccanaro, A. SCPS: a fast implementation of a spectral method for detecting protein families on a genome-wide scale. *BMC Bioinformatics*, 11(1), 2010, 120.
- [102] Norman, R. P., Brian, C. T. and Carl, R. W. : Probing RNA Structure, Function, and History by Comparative Analysis. In Raymond, F. G., Thomas, R. C. and John, F. A. eds.: *The RNA World, Second Edition*. 1999, .
- [103] Notredame, C. and Higgins, D. G. SAGA: sequence alignment by genetic algorithm. *Nucl. Acids Res.*, 24(8), 1996, 1515-1524.
- [104] Notredame, C., Higgins, D. G. and Heringa, J. T-Coffee: A novel method for fast and accurate multiple sequence alignment. *J. Mol. Biol.*, 302(1), 2000, 205-217.

- [105] Notredame, C., O'Brien, E. A. and Higgins, D. G. RAGA: RNA sequence alignment by genetic algorithm. *Nucl. Acids Res.*, 25(22), 1997, 4570-4580.
- [106] Oliver, R. W. *The Coming Biotech Age, The Business of Bio-Materials*. McGraw-Hill Companies, USA, 2000.
- [107] Pandey, G., Kumar, V. and Steinbach, M. *Computational Approaches for Protein Function Prediction A Survey Technical Report*. TR 06-028. , 2006.
- [108] Pearson, W. R. Rapid and sensitive sequence comparison with FASTP and FASTA. *Meth. Enzymol.*, 183, 1990, 63-98.
- [109] Pei, J. and Grishin, N. V. MUMMALS: multiple sequence alignment improved by using hidden Markov models with local structural information. *Nucl. Acids Res.*, 34(16), 2006, 4364-4374.
- [110] Pei, J., Sadreyev, R. and Grishin, N. V. PCMA: fast and accurate multiple sequence alignment based on profile consistency. *Bioinformatics*, 19(3), 2003, 427-428.
- [111] Peterson, E. L., Kondev, J., Theriot, J. A. and Phillips, R. Reduced amino acid alphabets exhibit an improved sensitivity and selectivity in fold assignment. *Bioinformatics*, 25(11), 2009, 1356-1362.
- [112] Phuong, T. M., Do, C. B., Edgar, R. C. and Batzoglou, S. Multiple alignment of protein sequences with repeats and rearrangements. *Nucl. Acids Res.*, 34(20), 2006, 5932-5942.
- [113] Pipenbacher, P., Schliep, A., Schneckener, S., Schonhuth, A., Schomburg, D. and Schrader, R. ProClust: improved clustering of protein sequences with an extended graph-based approach. *Bioinformatics*, 18(suppl_2), 2002, S182-191.
- [114] Randall, F. S. and Temple, F. S. Pattern-Induced Multi-sequence Alignment (PIMA) algorithm employing secondary structure-dependent gap penalties for use in comparative protein modelling. *Protein Eng.*, 5, 1992, 35-41.
- [115] Rao, D. M., Moler, J. C., Ozden, M., Zhang, Y., Liang, C. and Karro, J. E. PEACE: Parallel Environment for Assembly and Clustering of Gene Expression. *Nucleic Acids Res.*, 38(suppl 2), 2010, W737-W742.

- [116] Reinert, G., Schbath, S. and Waterman, M. S. Probabilistic and statistical properties of words: an overview. *J. Comput. Biol.*, 7(1-2), 2000, 1-46.
- [117] Rocha, J., Rosselló, F. and Segura, J. Compression ratios based on the Universal Similarity Metric still yield protein distances far from CATH distances. *eprint arXiv:q-bio/0603007*, , 2006.
- [118] Roshan, U. and Livesay, D. R. Probalign: multiple sequence alignment using partition function posterior probabilities. *Bioinformatics*, 22(22), 2006, 2715-2721.
- [119] Rui, X. and Wunsch, D., II. Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, 16(3), 2005, 645.
- [120] Saito, J., Kita, A., Higuchi, Y., Nagata, Y., Ando, A. and Miki, K. Crystal Structure of Chitosanase from *Bacillus circulans* MH-K1 at 1.6-A Resolution and Its Substrate Recognition Mechanism. *J. Biol. Chem.*, 274(43), 1999, 30818-30825.
- [121] Schmitt, A. O., Ebeling, W. and Herzl, H. The modular structure of informational sequences. *BioSystems*, 37(3), 1996, 199.
- [122] Sean, E. Multiple alignment using hidden Markov models. In San Diego, California, USA, August 19-23, 114–120.
- [123] Sérgio, D. and Paulo, C. : Efficient Exact Pattern-Matching in Proteomic Sequences. In Anonymous : *Distributed Computing, Artificial Intelligence, Bioinformatics, Soft Computing, and Ambient Assisted Living*. 2009, 1178-1186.
- [124] Sjölander, K. Phylogenomic inference of protein molecular function: advances and challenges. *Bioinformatics*, 20(2), 2004, 170-179.
- [125] Sjölander, K. Phylogenetic inference in protein superfamilies: analysis of SH2 domains. *Proc. Int. Conf. Intell. Syst. Mol. Biol.*, 6, 1998, 165-174.
- [126] Smith, T. F. and Waterman, M. S. Identification of common molecular subsequences. *J. Mol. Biol.*, 147(1), 1981, 195.
- [127] Song, W. and Park, S. Latent semantic analysis for vector space expansion and fuzzy logic-based genetic clustering. *Knowledge and Information Systems*, , 2009.
- [128] Sonnhammer, E. and Hollich, V. Scoredist: A simple and robust protein sequence distance estimator. *BMC Bioinformatics*, 6(1), 2005, 108.

- [129] Stein, B. and Eissen, S. M. Z. Automatic Document Categorization: Interpreting the Performance of Clustering Algorithms. In *In Günter, Kruse, Neumann (Eds.): Advances in Artificial Intelligence. LNAI 2821*, , 254-266.
- [130] Subramanian, A. R., Kaufmann, M. and Morgenstern, B. DIALIGN-TX: greedy and progressive approaches for segment-based multiple sequence alignment. *Algorithms for Molecular Biology*, 3, 2008, 6.
- [131] Subramanian, A. R., Weyer-Menkhoff, J., Kaufmann, M. and Morgenstern, B. DIALIGN-T: An improved algorithm for segment-based multiple sequence alignment. *BMC Bioinformatics*, 6(1), 2005, 66.
- [132] Suen, C. Y. n-Gram Statistics for Natural Language Understanding and Text Processing. *IEEE TPAMI*, PAMI-1(2), 1979, 164-172.
- [133] Sze, S., Lu, Y. and Yang, Q. A Polynomial Time Solvable Formulation of Multiple Sequence Alignment. *Journal of Computational Biology*, 13(2), 2006, 309-319.
- [134] Tetko, I., Facius, A., Ruepp, A. and Mewes, H. Super paramagnetic clustering of protein sequences. *BMC Bioinformatics*, 6(1), 2005, 82.
- [135] Thompson, J. D., Plewniak, F. and Poch, O. BAliBASE: a benchmark alignment database for the evaluation of multiple alignment programs. *Bioinformatics*, 15(1), 1999, 87-88.
- [136] Thompson, J. D., Gibson, T. J., Plewniak, F., Jeanmougin, F. and Higgins, D. G. The CLUSTAL_X windows interface: flexible strategies for multiple sequence alignment aided by quality analysis tools. *Nucl. Acids Res.*, 25(24), 1997, 4876-4882.
- [137] Thompson, J. D., Higgins, D. G. and Gibson, T. J. Improved sensitivity of profile searches through the use of sequence weights and gap excision. *Comput. Appl. Biosci.*, 10(1), 1994, 19-29.
- [138] Thompson, J. D., Koehl, P., Ripp, R. and Poch, O. BAliBASE 3.0: latest developments of the multiple sequence alignment benchmark. *Proteins*, 61(1), 2005, 127-136.
- [139] Thompson, J. D., Plewniak, F. and Poch, O. A comprehensive comparison of multiple sequence alignment programs. *Nucl. Acids Res.*, 27(13), 1999, 2682-2690.

- [140] Tobias, W., Dorothea, E., Sita, L., Sven, R., Mario, A., John, H. M., Sebastian, B., Jens, S. and Jan, B. Partitioning biological data with transitivity clustering. *Nature Methods*, , 2010, 419-420.
- [141] Van Walle, I., Lasters, I. and Wyns, L. Align-m--a new algorithm for multiple alignment of highly divergent sequences. *Bioinformatics*, 20(9), 2004, 1428-1435.
- [142] Varré, J. S., Delahaye, J. P. and Rivals, E. Transformation distances: a family of dissimilarity measures based on movements of segments. *Bioinformatics*, 15(3), 1999, 194-202.
- [143] Vinga, S. and Almeida, J. Alignment-free sequence comparison-a review. *Bioinformatics*, 19(4), 2003, 513-523.
- [144] Viterbi, A. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *Information Theory, IEEE Transactions on*, 13(2), 1967, 260.
- [145] Vogel, C., Bashton, M., Kerrison, N. D., Chothia, C. and Teichmann, S. A. Structure, function and evolution of multidomain proteins. *Curr. Opin. Struct. Biol.*, 14(2), 2004, 208.
- [146] Ward, J. H. Hierarchical Grouping to Optimize an Objective Function. *J. Am. Stat. Assoc.*, 58, 1963, 236-244.
- [147] Wicker, N., Perrin, G. R., Thierry, J. C. and Poch, O. Secator: A Program for Inferring Protein Subfamilies from Phylogenetic Trees. *Mol. Biol. Evol.*, 18(8), 2001, 1435-1441.
- [148] Widera, P., Garibaldi, J. M. and Krasnogor, N. Evolutionary design of the energy function for protein structure prediction. In *CEC'09: Proceedings of the Eleventh conference on Congress on Evolutionary Computation*, Trondheim, Norway, Piscataway, NJ, USA, 1305-1312.
- [149] Wilcox, A., Phil, M. and Hripcsak, G. Classification Algorithms Applied to Narrative Reports. In *Proc AMIA Symp*, , 455-459.
- [150] Wittkop, T., Baumbach, J., Lobo, F. and Rahmann, S. Large scale clustering of protein sequences with FORCE -A layout based heuristic for weighted cluster editing. *BMC Bioinformatics*, 8(1), 2007, 396.

- [151] Wu, K. P., Lin, H. N., Sung, T. Y. and Hsu, W. L. A new similarity measure among protein sequences. *Proc. IEEE Comput. Soc. Bioinform. Conf.*, 2, 2003, 347-352.
- [152] Yona, G., Linial, N. and Linial, M. ProtoMap: automatic classification of protein sequences and hierarchy of protein families. *Nucl. Acids Res.*, 28(1), 2000, 49-55.
- [153] Yoshimasa, T., John, M., Jun'i chi, T. and Sophia, A. Learning string similarity measures for gene/protein name dictionary look-up using logistic regression. *Bioinformatics*, 23(20), 2007, 2768-2774.
- [154] Zeng, Z., Zhang, S., Li, H., Liang, W. and Zheng, H. A novel approach to musical genre classification using probabilistic latent semantic analysis model. In *ICME'09: Proceedings of the 2009 IEEE international conference on Multimedia and Expo*, New York, NY, USA, Piscataway, NJ, USA, 486-489.