Caractérisation en séquence et en structure des protéines virales

Maud Jusot

université PARIS DIDEROT PARIS 7

Biologie-Informatique

Rapport de stage de M2

2014/2015

Caractérisation en séquence et en structure des protéines virales

Etude de l'ARN-polymérase ARN-dépendante et des protéines de capside.

Maud JUSOT

Inria-Irisa-équipe Dyliss

Encadré par

François Coste

Résumé

Les virus jouent un rôle important dans l'écologie marine mais leur diversité et leurs rôles demeurent peu connus. Il semble particulièrement intéressant de profiter de la collecte des données métagénomiques du programme *TARA*-Oceans pour commencer à évaluer la diversité et l'impact de ces virus marins. Toutefois, les génomes de virus possèdent un fort taux de mutation qui rend leur identification difficile. Le but de ce stage était d'améliorer l'identification des virus dans les données métagénomiques en explorant deux sources d'informations différentes pour caractériser deux familles de protéines virales : les ARN-polymérases ARN-dépendantes (RdRps) conservées en séquence et présentes chez tous les virus à ARN, et les protéines de capsides, caractéristiques des virus et conservées en structure. La caractérisation en séquences des RdRps a confirmé la difficulté à identifier ces séquences qui mutent beaucoup et à les distinguer des RdRps cellulaires, malgré l'utilisation de différents outils tels que BLAST, HMMER et Protomata ainsi qu'un effort de paramétrisation de ce dernier. L'étude de la spécificité des fragments de contact (paire de segments de squelette carboné en interaction) des protéines de capsides s'est en revanche révélée prometteuse puisqu'elle nous a permis d'identifier des fragments caractéristiques des virus.

Abstract

Viruses play an important role in marine ecology, but their diversity and exact roles remain unclear. It seems particularly interesting to make use of the newly available metagenomics data collected from the *TARA*-Oceans program, to start evaluating the diversity and the impact of marine viruses. Nonetheless, viruses harbor a high mutation rate in their genome, which makes difficult their correct identification. The aim of this internship was to improve viruses identification in metagenomics data exploring two different type of information for characterizing two viral proteins families : the RNA-dependent RNA-polymerases (RdRps), which is conserved in primary sequence and is found in all RNA viruses, and the capsid proteins, which are characteristic of viruses, and are conserved in structure. The characterization of RdRps sequences has confirmed the difficulty to identify them because of the high mutation rate, and the difficulty to discriminate them from eukaryotics RdRps. This was the case for all the different methods we applied, like BLAST, HMMER and Protomata, and even with the parameter optimization done for this last one. The study of the specificity of contact fragments (pair of carbon ribbons segments interacting) of capsid proteins was very promising. Indeed, it allowed us to identify characteristic fragments of viruses.

Remerciements :

Je tiens à remercier en tout premier lieu mon maître de stage, François Coste, CR1 à l'INRIA de Rennes, pour son accueil ainsi que pour le temps qu'il a consacré à me guider dans mon travail.

Je souhaite aussi remercier Clovis Galiez, doctorant à l'INRIA de Rennes, pour sa patience et pour tout le temps qu'il a consacré à répondre à mes innombrables questions sur son travail, mais également tous les autres membres des équipes Dyliss et Genscale pour leur gentillesse et bonne humeur qui m'ont permis d'apprécier mon expérience rennaise.

Je tiens aussi à remercier tout particulièrement Martine Boccara, professeur a l'UPMC, ainsi que Amos Kirilovsky, post-doc à l'ENS, pour leurs précieux conseils et explications qui m'ont permis de mieux appréhender les questions biologiques de ce sujet, ainsi que pour m'avoir donné l'accès aux données métagénomiques de *TARA*-Oceans.

Merci également à Joël Pothier, maitre de conference a l'UPMC, pour le temps qu'il a consacré à m'expliquer le fonctionnement de Yakusa, ainsi qu'à tous les membres de l'Atelier de Bioinformatique pour leur accueil chaleureux lors de mes déplacements à Paris.

Et enfin, un très grand merci à Mathilde Carpentier, maitre de conference à l'UPMC, grâce à qui j'ai pu trouver ce stage, mais surtout pour toute l'aide qu'elle m'a apportée et l'énergie qu'elle a dépensé dans ce projet.

Table des matières

1 Introduction

Contexte Les virus sont extrêmement abondants (nombre estimé à 10^{30}) dans l'océan et varient en fonction de leur environnement (salinité, température, localisation, etc.) [1]. Ils sont principalement de deux types : des phages, ayant pour hôtes des archées ou bactéries, et des virus d'eucaryotes unicellulaires.

De nombreuses études suggèrent que les virus de l'océan jouent un rôle clé dans l'écologie marine [1] [2] : en lysant leurs hôtes, ils régulent et peuvent influencer les communautés microbiennes qui dirigent les cycles biogéochimiques. Ces travaux mettent ainsi en avant leur impact indirect sur la production primaire de carbone, le cycle des nutriments et la séquestration du carbone par les océans. Les virus marins sont donc des candidats clés pour influencer les rétrocontrôles de l'océan sur le changement climatique. De plus, ces virus jouent également un rôle important dans les transferts de gènes et exercent une grande pression évolutionnaire sur leurs hôtes.

Malgré ces études, les communautés de virus marins restent assez peu connues. Leurs rôles dans les océans sont probablement sous estimés, tout comme leur diversité, et leurs effets sur les populations d'hôtes et sur les écosystèmes demeurent incertains. Ainsi, il semble particulièrement intéressant de profiter de l'opportunité offerte par la collecte des données métagénomiques du programme *TARA*-Oceans. Ce projet a en effet permis la récolte d'échantillons marins à travers le monde et le séquençage NGS de ces derniers. L'étude de ces données pourrait permettre de mieux appréhender la diversité ainsi que le contrôle des virus marins sur les communautés microbiennes et de commencer à élucider leurs rôles sur les cycles biogéochimiques.

Les études métagénomiques ont généré un nombre considérable de données, mais celles-ci sont loin d'être totalement exploitées en particulier dans le cas des virus puisqu'il existe peu d'outils dédiés à l'annotation de ces derniers. Les approches classiques de recherche d'homologues (comme BLAST [3]) ne sont pas vraiment appropriées pour les virus à cause du fort taux de mutation de leurs génomes (de l'ordre de 10^{-4}) [4]. De plus, le barcoding, technique de taxonomie moléculaire permettant la caractérisation d'un ensemble d'individus à partir d'un gène particulier, n'est pas non plus approprié puisqu'il n'y a pas de gène universel chez les virus, ce qui rend leur annotation encore plus compliquée. Selon plusieurs études métagénomiques précédemment réalisées, entre 70 et 90% des séquences de la fraction virale (inférieure à 0,8 μm) ne présentaient pas d'annotation [1] [5].

Objectif du projet Le travail présenté ici a justement pour objectif d'améliorer l'identification des virus dans les données métagénomiques. Nous avons souhaité construire des motifs ou profiles nous permettant de mieux les identifier. Il existe déjà des banques de motifs ou de profiles mais ices derniers ne permettent pas d'identifier beaucoup de virus dans les métagénomes. Nous avons donc voulu construire des motifs plus précis et plus petits, et essayer de tirer partie de l'information structurale. Nous avons réalisé nos premiers essais sur deux familles de protéines virales présentant des problématiques différentes :

– Les polymérases, protéines dont les séquences ont été relativement conservées au cours de l'évolution grâce à leur fonction essentielle de réplication. Ces protéines sont-elles suffisamment conservées en séquence pour être identifiées facilement chez les virus ? Et à l'inverse, peut-on facilement différencier les polymérases virales de celles de leurs hôtes ?
– Les protéines de capside, caractéristiques des virus [6] et possédant une conservation en structure mais des séquences très divergentes. Peut-on trouver des motifs structuraux qui soient spécifiques des virus ? Peut-on faire le lien entre la structure conservée et la séquence qui ne l'est pas ?

Polymérases Il existe différents types de polymérases chez les virus (Reverse transcriptase, DNA-polymerase, etc.) possédant différentes propriétés et étant plus ou moins caractéristiques des virus. Notre choix s'est porté sur l'ARN-polymerase ARN-dépendante (RdRp), enzyme présente chez tous les virus à ARN et possédant un certaine conservation de séquence malgré le faible taux d'identité des séquences de virus. En effet, les virus à ARN mutent beaucoup car leur polymérase ne possède pas l'activité de relecture (activité exonuclease 3'-5') qui permet de repérer et corriger les erreurs lors de la réplication. Ils ont également une forte pression sélective due au système immunitaire de l'hôte et à un environnement très compétitif entre les virus. Ceci a entraîné une diversification rapide dans la structure primaire de tous les gènes et protéines de virus à ARN [7]. Pour autant, certaines fonctions de ces protéines sont cruciales pour l'efficacité de la reproduction virale et doivent donc être préservées. C'est justement le cas des RdRps qui, parmi toutes les protéines virales, montrent le plus haut degré de conservation de séquences [7]. On retrouve en effet 8 courts motifs conservés, malgré le fort taux de mutation des séquences virales. Toutefois, même avec cette conservation, les séquences varient fortement et leur similarité entre elles est trop faible pour les études phylogénétiques [7]. Les RdRps existent aussi chez les eucaryotes, mais celles-ci ne sont pas homologues des RdRps virales

malgré leur fonction similaire [8]. Le but de cette partie est donc de caractériser les RdRps en séquences. Pour cela, nous avons utilisé deux logiciels couramment utilisés en recherche d'homologue, BLAST [3] et HMMER [9], ainsi qu'un troisième logiciel développé dans l'équipe, Protomata-Learner [10], qui permet de réaliser une caractérisation plus fine des séquences.

Protéines de capside Le deuxième gène d'intérêt étudié correspond aux gènes des protéines de capsides. Les séquences de cette famille ne sont pas particulièrement conservées en séquence mais le sont en structure. Les protéines de capside sont présentes chez une majorité des virus et s'assemblent entre elles pour former une structure symétrique protectrice englobant le génome du virus. La forme des capsides est à la base des différentes morphologies de virus et est un des principaux critères de taxonomie des virus [11]. L'objectif pour cette partie consiste à se concentrer non pas sur la conservation de la séquence primaire mais sur la conservation des structures. L'originalité de cette approche est de vouloir aller plus loin dans la caractérisation des protéines en faisant le lien entre structure et séquence. En effet, puisqu'il y a des contraintes structurelles dans la capside, les séquences associées, aussi divergentes soient-elles, doivent garder un signal de ces contraintes. Nous voulons ainsi détecter la signature structurale des protéines de capsides pour ensuite faire le lien avec les séquences associées. Les fragments de contact (CFs) [12] correspondant à 2 segments de squelette carboné en interaction (voir définition complète dans Matériel et Méthode) sont de bons candidats pour étudier la conservation de structure. L'étude de la spécificité des CFs, réalisée à l'aide d'une adaptation du logiciel Yakusa [13], a permis d'identifier lesquels sont caractéristiques des capsides de virus, dans l'objectif de rechercher les séquences associées à ces derniers dans les métagénomes.

Ainsi, l'analyse des gènes conservés en sequence et celle des gènes conservés en structure nous permet d'explorer l'identification des virus par deux sources d'informations différentes.

2 Matériel et méthodes

2.1 Matériel et méthodes pour les RdRps

2.1.1 Méthodes pour caractériser les RdRps

BLAST [3] Cet outil est un logiciel de référence pour rechercher des protéines homologues, basé sur une recherche heuristique de similarité entre séquence. La version utilisée du logiciel

est la version BLAST 2.2.30+. Les paramètres utilisés sont ceux par défaut. Pour la *e-value*, nous avons utilisé deux seuils classiques : un premier assez lâche à 10^{-1} et un second plus stringent à 10^{-4}.

HMMER Ce logiciel est également un outil classique de recherche de protéines homologues basé cette fois sur des modèles probabilistes appelés profils de chaînes de Markov cachées (profils HMMs) [9]. Les profils HMMs ont la particularité d'être suffisamment robustes face aux mutations (insertions et délétions en particulier) ce qui en fait un très bon outil prédicteur, plus précis que BLAST tout en étant aussi rapide. L'utilisation de HMMER a pour objectif de faire émerger les zones les plus similaires et les plus partagées par les séquences. Les profils HMMs permettent un apprentissage très sensible et sont par conséquent très efficaces en reconnaissance des membres d'une famille protéique même lorsqu'il existe des divergences de similarité entre les séquences membres. Cet outil semble par conséquent plus adapté à l'étude des séquences virales que BLAST ne l'est. HMMER est utilisé ici afin de créer un profil HMM à partir de l'alignement multiple (obtenu grâce à l'outil MUSCLE version 3.8.31 [14] utilisé avec ses paramètres par défaut) du jeu d'apprentissage. Ce profil HMM sert ensuite à scanner le jeux de validation pour rechercher des séquences homologues. La version utilisée est la version 3.1b2. Les paramètres utilisés sont également ceux par défaut sauf pour la *e-value* pour laquelle les même seuils que pour BLAST ont été choisis.

Protomata Protomata [10] est un outil permettant la caractérisation d'un ensemble de séquences de façon plus fine que BLAST et HMMER en recherchant les signatures de familles de protéines. En effet, il est capable de caractériser des courts motifs au sein des séquences et leurs enchaînements. Il dispose d'un avantage par rapport aux profils HMMs auxquels il ressemble : ces derniers ne permettant pas de visualiser les zones importantes marquant l'apparition de sous-familles tandis que Protomata le fait. Protomata est basé sur un alignement partiel local multiple des séquences du jeu d'apprentissage qui permet d'obtenir des blocs de paires de fragments locaux significativement similaires alignés et leurs enchaînements dans les séquences (voir figure 1). Il peut alors calculer les PSSMs associées à ces blocs et construire des automates non-déterministes (voir figure 2) modélisant l'enchaînement de ces PSSMs. L'automate sert alors à scanner de nouvelles séquences et leur affecter un score. Le fait de se concentrer sur des blocs conservés permet d'éliminer la variabilité créée par la divergence des séquences de virus.

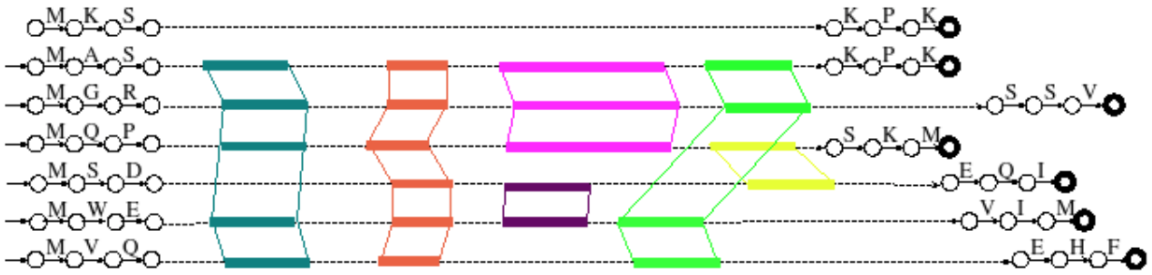


FIGURE 1 – Alignement partiel local multiple réalisé par Protomata [10]. Chaque couleur représente un bloc différent qui sera créé et qui correspond à une zone caractéristique.

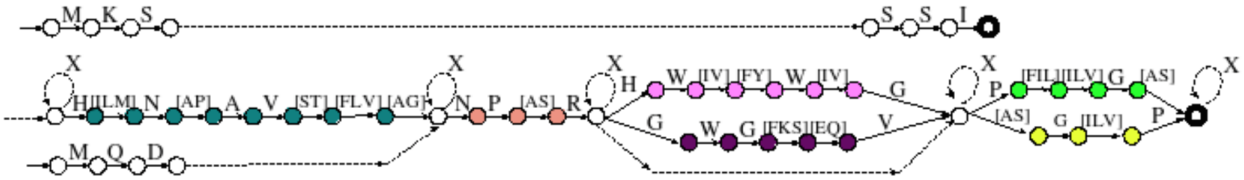


FIGURE 2 – Automate créé à partir de l'alignement en figure 1 exhibant les zones caractéristiques, les chemins exceptions à ces zones, ainsi que les zones de gaps reliant les zones caractéristiques [10].

Les paramètres fixés pour l'ensemble des expériences sont les suivants :

– Utilisation de l'option -c pour chercher des blocs de faible consensus, c'est à dire que si un fragment de séquence A est rattaché à un fragment de séquence B pour former un bloc et que B est rattaché à une séquence C alors A et C feront partis du même bloc par simple lien sans avoir besoin de vérifier si A et C sont rattachés ensemble. Ceci permet de ne pas se concentrer uniquement sur une homologie forte mais plutôt d'autoriser la dérive d'un motif, c'est à dire des mutations progressives au sein de celui-ci, ce qui est donc plus adapté à la forte divergence des séquences virales.

– Affectation d'un poids aux séquences en fonction de leur redondance dans le jeu d'apprentissage.

– Utilisation d'une mixture de Dirichlet permettant d'ajouter des pseudocounts pour chaque position. La mixture de Dirichlet choisie pour l'ensemble des tests est dist.20comp. En effet, elle convient pour les séquences plus éloignées, ayant une homologie lointaine, ce qui est le cas des séquences virales étudiées.

– Le scan des séquences est utilisé avec l'option -allow-partial-model-match qui autorise la correspondance avec des séquences qui ne s'alignent pas avec l'ensemble de l'automate modèle mais seulement une sous partie. Ce point est important pour scanner le données métagénomiques car les séquences NGS sont courtes et ne représentent donc pas forcément les protéines entières.

Les autres paramètres ont été déterminés par validation croisée.

Validation croisée : recherche des meilleurs paramètres Pour déterminer les paramètres optimaux de Protomata pour le jeu de séquences modèles, c'est à dire les paramètres permettant une meilleure caractérisation des séquences, une validation croisée k-fold a été réalisée. En effet, la force et la difficulté de Protomata est de permettre une caractérisation à différent niveaux mais ceci nécessite un effort sur la paramétrisation. Trois paramètres ont ainsi été testés : t le seuil correspondant à la significativité minimum du score accordé à une paire de fragments pour construire un bloc, ms la taille maximale d'un fragment, et p le pourcentage minimum du poids total des séquences pour former un bloc. Pour un k-fold de k, la cross-validation s'effectue de la manière suivante : pour chaque ensemble de paramètres testé, le jeu d'apprentissage correspond aux N séquences du jeu modèle auquel $m = N/k$ séquences ont été retirées. Ces m

séquences sont alors ajoutées à un jeu de séquences négatives pour former un jeu de validation. Les séquences négatives correspondent à des séquences différentes de ce que l'on recherche (en l'occurence des séquences n'étant pas des RdRps virales). L'automate est alors appris sur les $N - m$ séquences du jeu d'apprentissage puis le jeu de validation est scanné et cette opération est répétée k fois de sorte à ce que chaque séquence soit scannée une fois. Pour chaque étape (à savoir pour chaque jeu de paramètres et pour chaque k), on calcule un ensemble de valeurs statistiques de sorte à pouvoir estimer quel jeu de paramètres permet d'obtenir les meilleures prédictions sur l'ensemble des *k-folds*. Le but étant de trouver les paramètres permettant de prédire au mieux l'ensemble des m séquences et de les discriminer du jeu négatif.

Automatisation de la procédure La procédure d'étude des RdRps a été automatisée grâce à l'implémentation d'un script Bash de sorte à ce qu'en fournissant les jeux d'apprentissage, de validation et les paramètres voulus, toutes les étapes en partant des données brutes et en appliquant les 3 méthodes ainsi que le parsing des fichiers de sortie, soient réalisées.

2.1.2 Sélection de séquences modèles.

Nous avons constitué un jeu de données modèles de RdRps qui nous servira pour les jeux d'apprentissage. Pour ce jeu de données, les polymérases ont été obtenues sur Uniprot en ne gardant que les séquences virales de Swissprot ayant un domaine RdRp. Cette sélection correspond à 599 séquences. Puis, les séquences ont été filtrées en fonction de la taxonomie dans le but d'éliminer les groupes de virus qui infectent uniquement des mammifères par exemple (comme le groupe des virus de la grippe). En effet, les virus d'intérêt pour cette étude, sont des virus infectants plutôt les micro-organismes du plancton (micro-algues, bactéries, etc.). Après ce premier filtrage, réalisé à l'aide du ICVT [11], le nombre de séquences est descendu à 410. La liste des séquences sélectionnées en fonction de leur taxonomie est accessible sur `http://www.irisa.fr/dyliss/public/Virus/mjusot/selection_taxonomie.xls`. Etant donné que ces séquences modèles vont être la base de cette recherche, il est important de ne garder uniquement que les données les plus fiables, préférentiellement disposant de preuves expérimentales. Ainsi, un second filtrage a été réalisé en fonction du score d'annotation Uniprot. Celui-ci repose sur une mesure heuristique de la qualité d'annotation et correspond à un chiffre entre 0 et 5 (5 étant le score maximal). Seules les séquences possédant au minimum un score d'annotation de 4 ont été gardées pour la suite soit un total de 288 sé-

quences. Enfin, la redondance des séquences a été réduite à 70% par CD-hit [15] ce qui nous a permis d'obtenir au final 124 séquences de RdRps de divers virus à ARN.

Un nettoyage a également été effectué sur ces données en enlevant les séquences contenant des acides aminés inconnus 'X'. En effet, Protomata-Learner n'accepte pas les séquences contenant ce type d'acides aminés pour apprendre les modèles. Cette étape a été inclue dans l'automatisation de la procédure. Nous avons finalement obtenu 108 séquences pour notre jeu de données modèles.

2.1.3 Jeux de validation

Pour tester les modèles, il est nécessaires d'avoir des jeux de validation. Pour cela, différents jeux de validation ont été constitués :

– un premier utilisé pour la validation croisée contenant 27 séquences de polymérases sélectionnées sur SwissProt et ayant un score d'annotation d'au moins 4 mais n'étant pas des RdRps virales (6 RdRps eucaryotes, 13 polymérases d'archées et 8 polymérases de virus à ADN). Ces séquences dites "négatives" mais relativement proches de ce que l'on recherche ont été sélectionnées pour aider à trouver les paramètres les plus discriminants entre les séquences positives (RdRps virales) et celles-ci. On rajoutera en effet, à chaque étape du k-fold, m séquences positives du jeu modèle de RdRps pour former le jeu de validation.

– Un second jeu de données de validation a été obtenu en prenant toutes les séquences de Swissprot et en ajoutant également les séquences Uniprot étant annotées RdRp. Ce jeu a été nettoyé pour retirer toutes les séquences contenues dans le jeu d'apprentissage. Il a également été filtré par Uniref50. Ce jeu de validation nous servira pour une comparaison des résultats entre les 3 méthodes de caractérisation des RdRps.

2.2 Matériel et méthodes pour les protéines de capside

2.2.1 Sélection des structures de capsides

Pour générer une banque de capsides virales, toutes les séquences de la Protein Data Bank (PDB) contenant le terme Gene Ontology "Viral Capsid" ont été récupérées, ainsi que toutes les structures annotées capsides dans la Viral Protein Structural Database [16]. Les deux listes de structures ont été fusionnées puis nettoyées de sorte à enlever toutes les chaînes qui n'étaient pas des protéines. Une banque non redondante a alors été créée en utilisant Uclust (version

8.0.1623) [17]. A partir de ces résultats, pour chaque cluster, la protéine de meilleure résolution et préférentiellement résolue par cristallographie a été choisie comme représentant. La banque finale contient un total de 327 chaînes PDB.

2.2.2 Extraction des fragments de contacts

Un fragment de contact (CF) est défini comme une paire de segments de squelette carboné d'une protéine qui sont proches dans l'espace (structure 3D) ou plus précisément qui partagent un contact (distance entre Cα inférieure à σ =7 Angstroms), et sont suffisamment proches pour interagir (chaque Cα est au plus loin à τ =13 Angstroms d'un Cα de l'autre segment) (voir figure 3) [12]. Pour extraire ces CFs, nous avons utilisé un script, fourni par C. Galiez, auquel nous avons fourni la liste des chaînes PDB correspondant aux structures de capsides précédemment sélectionnées. Ce script nous permet ainsi d'obtenir pour chaque chaîne les CFs lui correspondant sous format PDB.

2.2.3 Adaptation de Yakusa

Beaucoup de CFs sont générés et ils ne sont pas forcément spécifiques des capsides tant en structure qu'en séquence. Nous avons voulu dans un premier temps essayer de déterminer quels sont les CFs ou sous ensembles de CFs qui sont spécifiques des capsides virales. Pour cela, nous avons voulu rechercher les CFs extraits de chaque capside dans la PDB. Notre choix s'est porté sur Yakusa [13] qui est un programme permettant de rechercher dans une banque toutes structures protéiques similaires à une structure requête car la recherche est rapide, locale et sans gap ce qui est particulièrement adapté à notre cas. De plus comme le code est disponible, il a été facile de l'adapter à nos besoins.

Yakusa transforme toutes les structures (requête et protéines de la banque) en enchaînement d'angles internes (angle α soit l'angle dihédral entre 4 carbones alpha consécutifs). La similarité entre séquences d'angles est établie selon les étapes suivantes :

1. Construction d'un automate déterministe fini décrivant tous les mots structuraux d'une taille donnée (souvent 4) de la protéine requête, c'est à dire toutes les combinaisons de taille fixée d'angles alpha existants.
2. Recherche de tous les mots identiques ou proches de ceux de la protéine requête dans chaque structure de la banque.

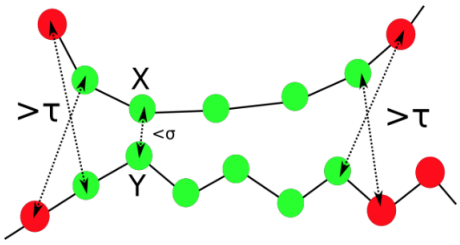


FIGURE 3 – Illustration de la définition d'un fragment de contact (en vert) respectant les seuils σ et τ [12]. Le contact est créé entre X et Y où la distance est inférieure à σ et est étendue tant que la distance reste inférieure à τ.

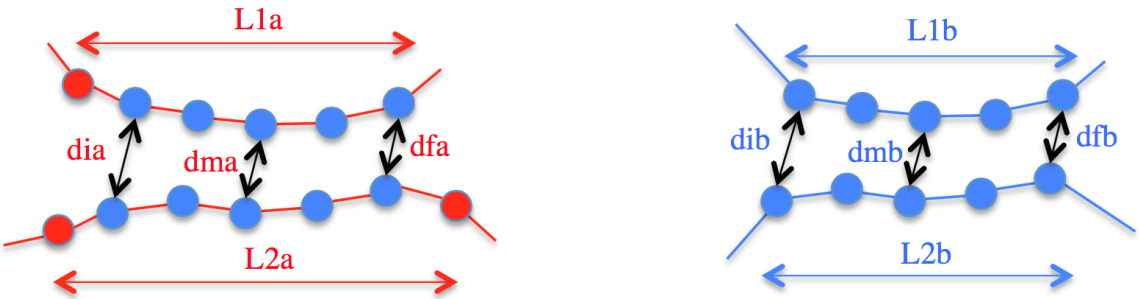


FIGURE 4 – Définition des filtres sur les longueurs et sur les distances ajoutés dans Yakusa. A gauche en rouge, CF requête et à droite en bleu, couple de SHSPs trouvé par Yakusa correspondant au CF requête. La partie colorée en bleue du CF est la partie correspondant aux SHSPs trouvés. Les longueurs $L1a$ et $L2a$ correspondent aux longueurs des 2 segments du CF requête, tandis que les longueurs $L1b$ et $L2b$ correspondent aux longueurs des 2 SHSPs associés. Pour être considérés comme compatible la longueur $L1b$ (respectivement $L2b$) doit être supérieure ou égale à n % de $L1a$ (respectivement $L2a$), n étant fixé par défaut à 80%. Les distances dia, dma et dfa sont comparées avec les distances dib, dmb et dfb. La distance dib doit entre égale à dia à plus ou moins m % (m étant fixé par défaut à 10%). Il en est de même entre dmb et dma ainsi qu'entre dfb et dfa.

3. Filtres des mots répétés et extension des mots trouvés vers des sous-structures (SHSPs pour structural high-scoring pairs) plus longues.
4. Sélection de SHSPs compatibles pour chaque protéine de la banque.
5. Classement des structures basé sur 3 scores : similarité, compatibilité et probabilité.

Ce programme a été adapté ici à partir du code existant de Yakusa en C. La banque utilisée est la PDB non redondante 50% du site de la PDB datant de mai 2015. Pour chaque capside, un fichier PDB contenant tous les CFs identifiés a été généré. Chaque CF est considéré comme une chaîne différente qui est en deux fragments. Chacun de ces fichiers PDB de CFs est utilisé comme requête. On obtient alors un ensemble de SHSPs correspondant chacun à un fragment de CFs. Les deux dernières étapes de Yakusa ont été adaptées afin de filtrer les SHSPs trouvés non compatibles avec la définition des CFs. Nous avons donc mis en place deux filtres qui remplacent la sélection des SHSPs compatibles :

– Le premier filtre repose sur la longueur du SHSP trouvé par rapport à la longueur du segment de CF requête. Le SHSP trouvé doit être de longueur supérieure ou égale à un pourcentage de la longueur de la requête. Ce pourcentage seuil a été fixé par défaut à 80% mais a été rajouté dans les paramètres de Yakusa est peut ainsi être modifié éventuellement dans l'optique de trouver les sous-fragments conservés qui pourraient être plus caractéristiques des structures.
– Le second filtre repose sur la recherche de paires de SHSPs qui doivent être compatibles en distances pour être considérés comme des CFs. Pour cela, il faut déjà s'assurer que les 2 SHSPs proviennent chacun d'un fragment différent du même CF de la protéine requête puis calculer les distances entre les SHSPs et s'assurer qu'elles sont à un pourcentage près les mêmes que celles du CF requête. Ce seuil a été fixé à 10% par défaut mais a également été rajouté en paramètre. Pour être rapide, les distances testées correspondent au début du SHSP, au milieu et à sa fin (voir figure 4).

2.3 Données métagénomiques de *Tara* Oceans

Nous disposons des données métagénomiques de *Tara* Oceans des stations 122 à 125 (stations des îles Marquises). Les données utilisées sont un assemblage des gènes (issus d'un séquençage NGS) correspondant au mélange des métatranscriptomes de toutes les stations (soit 53 959 998 séquences toutes fractions confondues entre 0,8 μm et 2000μm).

3 Résultats et Discussion

3.1 Caractérisation et recherche des RdRps

3.1.1 Etude des séquences modèles de RdRps

Après toutes les étapes de nettoyage, le jeu final d'apprentissage pour les RdRps contient 108 séquences réparties de la manière suivante dans la taxonomie des virus : 2 séquences de virus à ARN double brins, 12 séquences de virus à ARN simple brin à polarité négative et 94 séquences de virus à ARN simple brin à polarité positive. La longueur de ces séquences est très variable (entre 665 et 3945 résidus) avec une longueur moyenne de 2340 acides aminés par séquences et un écart type de 704,6 (distribution présentée figure figA1 en annexe). Cette distribution n'est pas surprenante car la majeure partie de ces séquences virales sont des polyprotéines, c'est à dire le produit d'expression d'un seul gène en une unique protéine généralement non fonctionnelle, permettant ainsi aux virus d'avoir un génome très compact. Cette unique protéine est ensuite clivée en plusieurs protéines plus petites qui sont alors fonctionnelles. Ce point est important à souligner car le fait de caractériser ces polyprotéines, et non uniquement la partie de séquences correspondant aux RdRps, peut biaiser l'analyse. En effet, une séquence d'hélicase peut être reconnue par BLAST par exemple parce qu'elle sera proche de la partie hélicase d'une polyprotéine. Toutefois, si la conservation est très faible dans les parties non RdRps, les méthodes vont avoir tendance à se focaliser sur la partie plus conservée. Notamment en ce qui concerne Protomata, le programme devrait s'affranchir de cette barrière en se concentrant en particulier sur les blocs conservés sur l'ensemble ou sur une majorité de séquences et non sur les parties conservées dans les sous groupes de virus.

3.1.2 Sélection des paramètres optimaux pour Protomata par validation croisée

Des tests effectués préalablement à la validation croisée ont été réalisés sur des sous ensembles de données (petits jeux issus des alignements PFams des différentes familles de RdRps). Ces tests ont permis non seulement de guider le choix des paramètres à tester vers certaines valeurs (et ainsi éviter d'en tester trop) mais également de confirmer le choix de certains paramètres fixés comme par exemple la mixture de Dirichlet (voir Figure A3).

Pour la validation croisée, le k-fold choisi a été de 3. Prendre un k plus petit aurait rendu la procédure plus longue. De plus, ceci permet d'ajouter m=36 séquences positives au jeu de

validation, ce qui est relativement d'un même ordre de grandeur que le nombre de séquences du jeu négatif, et permet ainsi de ne pas trop biaiser les résultats statistiques de la validation croisée. Les différentes valeurs des paramètres testés sont présentées dans la table 1.

Paramètres	Valeurs testées
t	0,01 - 0,1 - 1 - 2 - 3
ms	10 - 12 - 15 - 17 - 20 - 25
p	5 - 10 - 15 - 20 - 25 - 30 - 40 - 50

TABLE 1 – Paramètres testés lors de la validation croisée.

Un nombre total de 360 combinaisons de paramètres ont finalement été testées. Ces expériences ont été parallélisées par groupe de 12 à 16 combinaisons pour optimiser le temps de calcul. Pour chaque combinaison de paramètres, les valeurs suivantes ont été calculées en fonctions de 2 seuils différents pour la prédiction (voir ci dessous) :

– la distribution des scores pour chaque k-fold, en séparant les séquences positives des négatives, permettant ainsi de voir quels paramètres les discriminent le mieux pour l'ensemble des k-folds.

– la marge correspondant à la différence entre la moyenne des scores des séquences positives et la moyenne des scores des séquences négatives ramenée à la somme de leur écarts-types. Ceci permet également de voir quels paramètres les discriminent le mieux pour l'ensemble des k-folds. $Marge = \frac{(moyenne(scores\_positifs) - moyenne(scores\_negatifs))}{ecart-type(scores\_positifs) + ecart-type(scores\_negatifs)}$

– le rappel mesurant le nombre de prédictions positives ($Vrai\_Positifs$) parmi les séquences de RdRps ($Vrai\_Positifs + Faux\_Negatifs$). $Rappel = \frac{Vrai\_Positifs}{(Vrai\_Positifs + Faux\_Negatifs)}$

– la précision mesurant le nombre de séquences de RdRps ($Vrai\_Positifs$) parmi les séquences prédites positives ($Vrai\_Positifs + Faux\_Positifs$).

$$Precision = \frac{Vrai\_Positifs}{(Vrai\_Positifs + Faux\_Positifs)}$$

– la F-mesure qui combine et pondère la précision et le rappel selon la formule suivante :

$$F = \frac{2.(precision \times rappel)}{(precision + rappel)}$$

– le taux d'erreurs de prédiction correspondant au nombre de RdRps prédites négatives ($Faux\_Negatifs$) ainsi que le nombre de séquences non RdRps prédites positives ($Faux\_Positifs$) parmi l'ensemble des séquences.

$$Taux\_d'erreurs = \frac{Faux\_Positifs + Faux\_Negatifs}{Nombre\_total\_de\_sequences}$$

Les deux seuils de scores choisis pour prédire si les séquences sont des RdRps ou non sont les suivants :

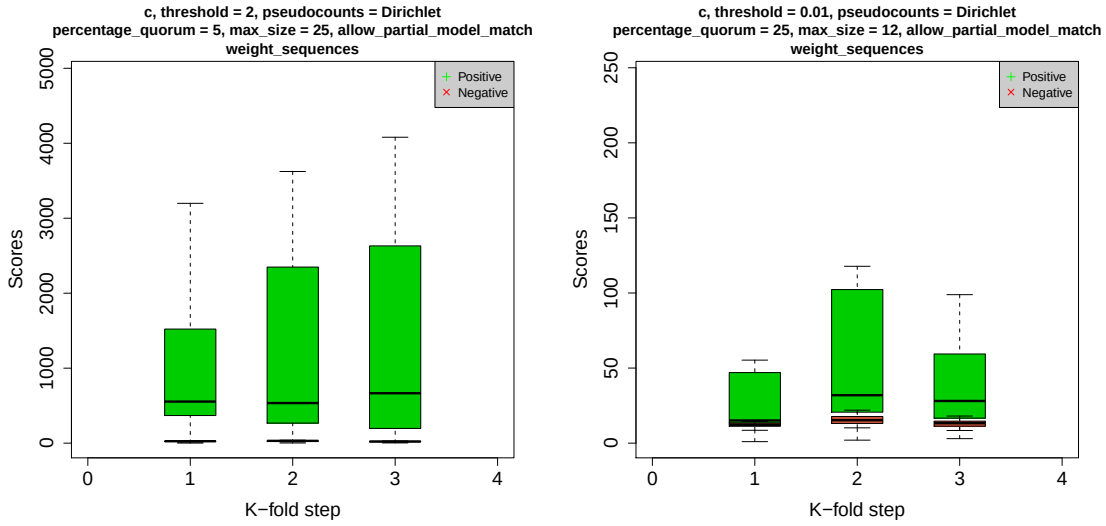
– Mean-Half (MH) calculé en prenant l'addition des moyennes des scores des séquences positives et négatives, divisés par 2. $MH = \frac{(moyenne(scores\_positifs) + moyenne(scores\_negatifs))}{2}$

– Min-pos (mp) correspondant au score minimum des séquences positives.

Ces seuils sont arbitraires et ne sont pas optimaux. Le seuil idéal se situe probablement entre les deux mais ne peut être déterminé qu'en fonction de la distribution. Une perspective serait de trouver un score plus approprié que l'on pourrait automatiser. Toutefois, ces seuils arbitraires nous permettent d'obtenir une vision globale suffisante des résultats, notamment en comparant les résultats selon les 2 seuils. Un seuil plus optimal sera appliqué une fois les paramètres sélectionnés pour la validation et comparaison avec les autres méthodes. Pour chaque groupe de paramètres testés ensembles, le ou les quelque(s) jeu(x) de paramètres présentant les meilleurs résultats ont été sélectionnés. Cette sélection avec les différentes valeurs statistiques associées est présentée en annexe (table A1). Il y a principalement deux jeux de paramètres parmi ceux sélectionnés qui ont retenu notre attention :

– Le premier jeu numéroté 8.1.4 (t = 2, ms = 25, p = 5) présente de bonnes valeurs statistiques. En effet, parmi tous les jeux de paramètres testés, c'est celui qui présente le plus faible taux d'erreur (0,14) avec comme seuil mp, il présente la plus grande marge (149) et la meilleure F-mesure (0.9) avec le seuil MH. C'est en effet le jeu de paramètres qui optimise le plus la précision (0,83) pour un très bon rappel (0,98). De plus, il faut noter que l'on a de bonnes valeurs alors que le seuil t est élevé, et donc plus stringent dans la construction du modèle. La distribution des scores présentée figure 6 (a) est très large puisque les scores s'étendent entre près de 0 et plus de 4000 pour les séquences positives. Pour les séquences négatives, la distribution est très écrasée puisqu'on distingue juste la médiane de celle-ci sur la figure. Les scores sont assez bas pour ces séquences puisque la plus haute valeur des scores négatifs s'élève à 37. On peut voir que la distribution des séquences positives chevauche celle des négatives, ce qui confirme bien la difficulté à trouver un seuil pertinent pour les discriminer.

– Le second jeu numéroté 1.2.1 (t = 0,01, ms = 10, p = 25) présente des valeurs un peu moins bonnes avec le seuil MH, mais présente en revanche les meilleures valeurs avec le seuil mp.



(a) (b)

FIGURE 6 – Distributions des scores du jeu de paramètres 8.1.4 en (a) et 1.2.1 en (b) représentées par des diagrammes en boîte en fonction des différentes étapes du k-fold. La partie verte correspond aux m = 36 séquences positives et la partie rouge correspond aux séquences négatives. Sur la figure (a), la partie rouge n'est pas visible car la distribution des séquences négatives est très écrasée. Toutefois on distingue un trait noir épais représentant la médiane de cette distribution.

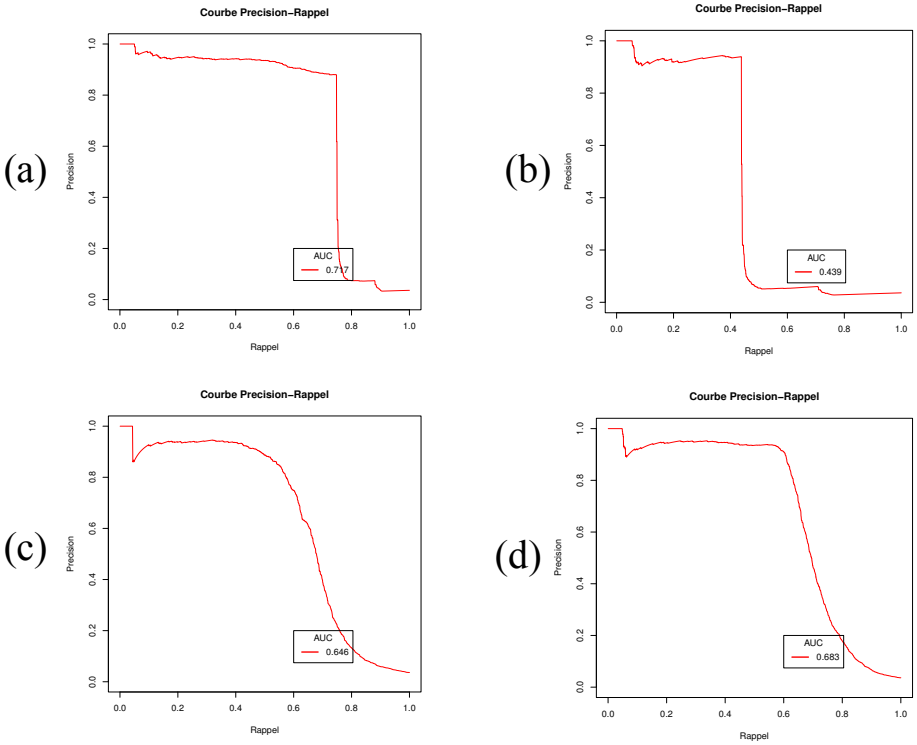


FIGURE 7 – Courbes précision-rappel (a) de BLAST, (b) de HMMER, (c) de Protomata avec les paramètres 1.2.1 et un p = 10 et (d) de Protomata avec les paramètres 8.1.4 et un p = 5. L'encadré AUC indique la valeur de l'air sous la courbe.

En effet, avec le seuil mp on obtient un très bon compromis puisque la précision est à 0,81 et le rappel à 0,83. C'est le seul jeu de paramètre pour lequel on trouve un aussi bon compromis entre les valeurs statistiques. La distribution des scores présentée figure 6 (b) est beaucoup moins large que pour le jeu 8.1.4 puisqu'elle ne s'étend pas au delà de 150. Toutefois, comme pour le jeu 8.1.4 les distributions des scores positifs et négatifs se chevauchent.

Globalement, ces résultats ne permettent pas de séparer complètement les positifs des négatifs lors de la prédiction sur les séquences non utilisées lors de l'apprentissage mais ceci s'explique déjà puisque les seuils ne sont pas optimaux. On peut espérer pouvoir améliorer ces résultats en trouvant un seuil plus approprié. Mais dans tous les cas, ces résultats permettent de choisir des jeux de paramètres les mieux adaptés aux séquences de RdRps pour les appliquer aux tests plus complets qui permettront alors une comparaison avec Blast et HMMER. De plus, il faut également dire que le jeu de séquences négatives a été volontairement choisi de manière a rendre les résultats le plus difficile possible en prenant des séquences proches de celles que l'on recherche.

Pour la validation et comparaison des résultats avec BLAST et HMMER, les jeux de paramètres 1.2.1 et 8.1.4 ont donc été sélectionnés. Pour les appliquer sur le jeu complet de séquences modèles, il est nécessaire de réadapter le paramètre p qui est dépendant du nombre de séquences.

3.1.3 Présentation des résultats - comparaison BLAST - Protomata - HMMER sur jeu de validation

Les trois méthodes de caractérisation des RdRps ont été appliquées avec pour jeu d'apprentissage, le jeu complet de séquences modèles et pour jeu de validation le jeu basé sur les séquences de Swissprot. Pour chaque méthode, la courbe précision-rappel a été calculée ainsi que les valeurs des tables de contingences pour différents seuils. Il faut noter que le jeu de validation contient très peu de RdRps. L'utilisation de la courbe précision-rappel est adapté à ce genre de données présentant un biais [18] (contrairement aux analyses de courbes ROC par exemple).

BLAST Cette méthode a été utilisée avec un seuil de *e-value* assez lâche fixé à 0,1. L'aire sous la courbe précision-rappel (figure 7 (a)) est à 0,72. On peut voir 3 pentes sur la courbe, ce qui laisse à penser qu'il y a 3 coupures concernant les e-values. La première coupure est

(a)	RdRps	Non RdRps
+	1345	185
-	453	47753

(b)	RdRps	Non RdRps
+	904	176
-	894	47762

TABLE 2 – Tables de contingence des résultats de BLAST avec une *e-value* fixée à (a) 0,1 et (b) 10^{-4}.

immédiate : on passe d'une précision de 1 à 0,96 dès les premières valeurs car des faux positifs (séquences prédites RdRps mais n'en étant pas) sont immédiatement reconnus par BLAST comme étant des RdRps. En effet, 6 faux positifs sont immédiatement détectés. Cinq de ces séquences correspondent à des polyprotéines de virus à ARN simple brin n'étant pas des RdRps mais contenant des protéines de capside, d'enveloppe, des protéases ou encore des hélicases. Le dernier faux positif correspond à une RdRp eucaryote. La deuxième pente de la courbe (chute de la précision pour un rappel supérieur à 0,75) correspond aux séquences qui n'ont pas été reconnues par BLAST comme étant des RdRps virales. Le logiciel ne leur a ainsi pas attribué de e-value et pour pouvoir réaliser cette courbe nous leur en avons mis une par défaut à 1 (donc supérieur au seuil), ce qui explique cette chute. Et enfin, la troisième petite pente correspond à un artéfact de la méthode de calcul de la courbe précision rappel. Celle-ci va en effet récupérer à nouveau des vrais positifs parmi les séquences qui n'ont pas été reconnues ce qui créé une petite augmentation locale de la précision (pour un rappel entre 0,80 et 0,90) mais qui ne doit pas être prise en compte.

Les tables de contingence sont présentées table 2. On constate que si l'on prend une *e-value* plus stringente à 10^{-4}, on perd de nombreux vrais positifs (441) pour une perte minimale de faux positifs (9). En effet, le rappel passe de 0,74 à 0,50 et même la précision passe de 0,88 à 0,83. Ainsi, ce choix ne semble pas plus approprié. Parmi les faux positifs trouvés avec une *e-value* à 0,1, il y a 127 séquences eucaryotes dont 44 correspondant à des RdRps eucaryotes, ainsi que diverses enzymes. On retrouve également 58 protéines de virus correspondant essentiellement à des hélicases et protéinases.

HMMER Cette methode a également été utilisée avec un seuil de e-value assez lâche fixé a 0,1. L'aire sous la courbe précision-rappel (figure 7 (b)) est à 0,44. On peut voir comme pour BLAST les 3 mêmes pentes sur la courbe. La première pente descend plus vite que BLAST mais arrive un peu plus tard. En effet, on obtient le premier faux positif pour une *e-value* de 10^{-194} alors que l'on a déjà trouvé 70 vrais positifs. La fin de cette première pente se situe autour d'une *e-value* de 10^{-98}. Lors de cette pente, 19 faux positifs sont trouvés : on passe alors

(a)	RdRps	Non RdRps
+	788	51
-	1010	47887

(b)	RdRps	Non RdRps
+	686	44
-	1112	47894

TABLE 3 – Tables de contingence des résultats de HMMER avec une *e-value* fixée à (a) 0,1 et (b) 10^{-4}.

d'une précision de 0,98 à 0,89. Ces 19 faux positifs sont presque tous des protéinases de virus à ARN simple brin à l'exception de 3 polyprotéines de virus à ARN simple brin contenant des séquences de protéines d'enveloppe et de capside ainsi que des protéases. La deuxième pente arrive pour un rappel légèrement supérieur à 0,40 soit beaucoup plus tôt que pour BLAST. Et on retrouve le même artéfact qu'avec blast. Les tables de contingence sont présentées table 3. De même que pour BLAST, on constate que si l'on prend une *e-value* plus stringente à 10^{-4}, on perds 102 vrais positifs pour seulement 7 faux positifs. On passe alors d'un rappel à 0,43 à une valeur de 0,38 pour une même précision à 0,94. Ainsi, ce seuil plus stringent ne semble également pas adapté. Concernant les 51 faux positifs trouvés avec une *e-value* à 0,1, 20 séquences correspondent à des RdRps eucaryotes, mais on a également plusieurs enzymes ou protéines eucaryotes présentant un site de fixation à l'ARN ou l'ADN.

Protomata Pour cette méthode, il a fallu tester différents paramètres p pour trouver celui adapté au nombre de séquences. Ainsi, nous avons gardé ceux présentant les meilleures aires sous la courbe précision-rappel (voir figure A4 et A5 en annexe). Ainsi pour le jeu de paramètre 1.2.1, nous avons choisi p = 10 ayant une aire sous la courbe précision-rappel à 0,65 (figure 7 (c)). Tandis que pour le jeu 8.1.4, nous avons choisi p = 5 qui présente une aire sous la courbe à 0,68 (figure 7 (d)). Pour les deux courbes précision-rappel, on retrouve 2 pentes : la première ressemblant à celle trouvée chez BLAST et HMMER correspond à l'identification de 14 faux positifs ayant de très bons scores et correspondant tous à des protéinases de virus à ARN simple brins. Les deux modèles trouvent les même faux positifs lors de cette pente. La deuxième pente correspond à une augmentation des faux positifs. Elle commence plus tôt pour le jeu 1.2.1 (à savoir pour un plus faible rappel à 0,45 environ contre 0,6 pour le jeu 8.1.4). Il n'y a pas d'artéfact sur ces courbes puisque Protomata a calculé un score pour chaque séquence. Les courbes précision-rappel nous permettent ici de fixer un seuil de score plus adapté que ceux utilisés pour la validation croisée. En effet, il faut trouver les seuils de scores qui optimisent aussi bien le rappel que la précision et pour cela, on peut choisir de se placer sur la courbe juste avant la deuxième pente et trouver les seuils de scores associés à ces valeurs. Ainsi, pour le

(a)	RdRps	Non RdRps
+	946	144
-	852	47794

(b)	RdRps	Non RdRps
+	1045	79
-	753	47859

TABLE 4 – Tables de contingence des résultats de Protomata (a) pour le jeu de paramètres 1.2.1 avec un seuil fixé à 21 et (b) pour le jeu de paramètres 8.1.4 avec un seuil fixé à 35.

jeu de paramètre 1.2.1, on a choisi un seuil à 21 permettant d'obtenir un rappel à 0.53 et une précision à 0,87. Et pour le jeu 8.1.4, on a choisi un seuil à 35 permettant d'obtenir un rappel à 0.58 et une précision à 0,93. Les tableaux de contingence associés sont présentés table 4.

Dans les 2 cas, les faux positifs trouvés sont essentiellement des protéinases et hélicases de virus à ARN, des RdRps eucaryotes, des enzymes eucaryotes présentant un site de fixation à l'ARN ou l'ADN (par exemple la Ribonuclease J). Le modèle 1.2.1 a également trouvé une reverse transcriptase de retrovirus (protéine P11283).

Discussion En comparant les valeurs statistiques obtenues avec les différentes méthodes, on constate que BLAST présente de meilleurs résultats, il trouve en effet beaucoup plus de vrais positifs sans augmenter énormément le nombre de faux positifs. Ces résultats sont surprenants puisqu'il était attendu que HMMER et Protomata soient plus appropriés pour les séquences virales qui présentent un fort taux de mutation. Concernant HMMER, il est possible au vu de sa courbe précision-rappel 7(b), que la *e-value* soit plus stringente que BLAST. Il faudrait envisager de regarder des seuils encore plus lâches que 0,1. Concernant Protomata, malgré tout le travail de paramétrisation effectué, cet outil ne parvient pas à faire mieux que BLAST sur les polymérases. Ainsi, cet outil n'est peut être pas adapté à l'étude des séquences virales divergentes. Concernant les faux positifs, les 3 méthodes retrouvent le même type de séquences : des séquences de virus à ARN non RdRps, des RdRps eucaryotes et d'autres séquences eucaryotes. Ces résultats sont à la fois surprenants et attendus. En effet, dans la littérature les RdRps virales et eucaryotes sont décrites comme partageant peu d'homologie de séquences [19] [8]. Il est donc surprenant de les retrouver aussi facilement. Toutefois, le motif DxDGD, essentiel pour l'activité des RdRps cellulaires, semble être un reste du motif GDD des RdRps virales [19]. Cela met donc en avant une certaine conservation entre les RdRps eucaryotes et celles virales, même si elles appartiennent à des super-familles différentes et cela confirme la difficulté à les distinguer. Egalement, on trouve selon les méthodes, des séquences plus ou moins nombreuses d'enzymes eucaryotes présentant un site de fixation à l'ADN ou l'ARN. Ainsi peut être que la similarité de séquence se situe essentiellement aux niveaux des domaines de fixation à l'ARN

ou l'ADN. A l'inverse, un résultat prévisible est le fait de trouver des séquences de virus à ARN n'étant pas des RdRps. Ceci est très certainement dû au fait de travailler sur des polyprotéines comme expliqué dans la partie sur l'étude des séquences modèles de RdRps. Ainsi, même Protomata n'est pas capable de passer cette barrière et pour améliorer les résultats il serait nécessaire de cliver les séquences du jeu modèle pour ne garder que la partie correspondant aux RdRps. Cela permettrait assurément de diminuer le nombre de faux positifs en éliminant toutes les protéinases, hélicases et autres protéines virales identifiées. Et cela permettrait ainsi de faire disparaître la première pente de la courbe précision-rappel. Il est possible que cela influe également sur la distribution des scores de Protomata et améliore ainsi la discrimination entre les séquences négatives et positives.

3.2 Caractérisation et étude de la spécificité des capsides de virus

3.2.1 Etude des CFs

Après extraction des CFs sur l'ensemble de la banque de structures de capsides, 11395 CFs ont été obtenus. Les chaînes de structures utilisées présentent entre 1 et 168 CFs avec une moyenne de 25 CFs par chaîne et un écart-type de 26,9 (voir distribution en annexe figure A6). Les fragments de CFs présentent une longueur moyenne de 16,9 résidus avec un écart-type de 11 (voir figure A7 en annexe). Il faut comprendre que 2 fragments formant un CF n'ont pas forcément la même longueur et également que ceux-ci peuvent être chevauchants. Ainsi, certains CFs sont très longs (une dizaine de fragments de CFs dépassent les 100 résidus). Ce sont des fragments qui s'alignent avec eux même (auto-fragments). Mais ces cas sont rares puisque 89% des fragments de CFs sont d'une longueur inférieure à 30 résidus. Dans 12 chaînes de capsides, aucun CFs n'a été trouvé.

3.2.2 Etude de la spécificité des CFs viraux avec Yakusa modifié

Validation de la méthode : test d'une chaîne contre elle même Pour valider notre méthode et tester ses limites, la version modifiée de Yakusa a été appliquée avec pour banque la même chaîne protéique que celle d'où ont été issus les CFs de la requête. Cela revient à tester la protéine contre elle même, dans le but de vérifier que l'on retrouve bien les CFs. Cette procédure a été appliquée plusieurs fois, un exemple est détaillé ci-dessous : 4 CFs ont été extraits de la chaîne T de la PDB 4CWU. Ces CFs sont présentés figure 8. Sur ces 4 CFs, un seul a été retrouvé

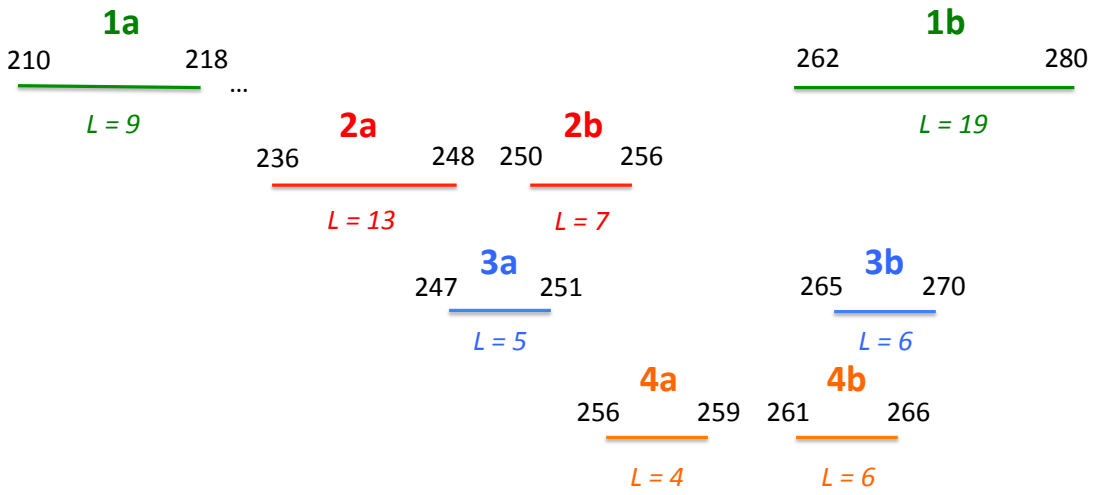


FIGURE 8 – Représentation des 4 CFs extraits de la chaîne T de la PDB 4CWU. Chaque trait représente un fragment. Chaque couleur représente un CF associé à un numéro de 1 à 4 et les fragments correspondants sont numérotés a et b. Les numéros en noir correspondent aux positions de début et de fin de chaque fragment sur la structure entière. L correspond à la longueur du fragment.

par Yakusa, mais lorsque l'on regarde le détail des CFs, ce résultat est tout a fait attendu. En effet, la taille minimal d'un fragment dans Yakusa a été paramétrée à 8 : il faut au moins 4 résidus pour pouvoir calculer un angle α. Un fragment de taille 8 permettra de générer une séquence d'angles alpha de longueur 4. Comme les mots dans l'automate sont de taille 4, un fragment de longueur inférieure à 8 ne pourra pas être mis dans l'automate. Ainsi pour avoir un fragment reconnu par Yakusa, il est nécessaire qu'il fasse au moins 8 résidus soit 5 angles α consécutifs. Les fragments 2b, 3a, 3b, 4a et 4b ne seront donc pas considérés. De plus, le fragment 2b étant éliminé, le fragment 2a ne peut plus former de paire et sera également éliminé. La limite est donc due à la taille des mots dans l'arbre. Ce sont des paramètres modifiables dans la ligne de commande. Toutefois, les fragments trop courts ne sont pas très significatifs. Ce logiciel présente aussi un avantage majeur puisqu'il est très rapide et permet ainsi pour une requête de scanner la PDB non redondante à 50% en moins d'une minute. Celle-ci contient 30780 chaînes appartenant à 28909 PDB dont notamment 1489 PDB virales.

Etude des CFs contre la PDB non redondante Les 11395 CFs obtenus à partir de la banque de structures de capsides ont été donnés en requête à Yakusa contre la PDB non redondante à 50%. Sur l'ensemble de ces structures, 24937 CFs ont été retrouvés par Yakusa. Parmi eux, il y a 1453 CFs uniquement retrouvés chez des virus dans 161 chaînes différentes. La distribution du nombre de CFs viraux par chaîne PDB est présentée en annexe A8. Puisque certains CFs sont retrouvés uniquement chez les virus, on peut en conclure qu'il y a bien des CFs qui semblent spécifiques des virus. Ce résultat est très prometteur puisqu'il confirme qu'il y a probablement de la conservation au sein des structures de virus et en particulier que les CFs sont de bons candidats pour étudier cette conservation. Pour les CFs spécifiques des virus, on pourra envisager de récupérer les séquences associées pour les rechercher ensuite dans les métagénomes avec BLAST par exemple ou encore un programme plus adapté tel que SmartConsAlign qui est basé sur un algorithme de programmation dynamique permettant une coupure quelconque entre 2 fragments ou plus. Egalement, on pourrait s'intéresser aux enchaînements de CFs dans une structure. En effet, il est possible que certains CFs se retrouvent de manière privilégiée en présence d'un second CF. Cette configuration a été trouvée dans les résultats comme par exemple pour les CFs 8, 11 et 13 de la chaîne 1 du PDB 1AYM. Le CF 8 est retrouvé 9 fois dans la PDB non redondante, le CF 11, 7 fois et le CF 13, 4 fois. Mais en particulier, le CF 13 est toujours retrouvé avec le CF 11 et le CF 8, et le CF 11 est toujours retrouvé avec le CF 8. Ainsi ce type de "super-motif" est particulièrement intéressant et devra être étudié plus en détail. Il

est possible que la conservation se situe également dans un enchaînement de motifs structuraux caractéristiques. Cette piste est donc également prometteuse.

3.3 Etude des séquences des données métagénomiques de *TARA*-Oceans

Données Les séquences du jeu de données métagénomiques de *TARA*-Oceans ont une longueur de 50 à 1204 résidus avec une médiane à 159 résidus. Comme attendu, ces séquences sont courtes comparées aux séquences du jeu modèle d'où l'intérêt d'avoir travaillé soit sur des fragments (pour les CFs) soit d'autoriser les "matchs" partiels avec Protomata.

Recherche des séquences de RdRps Les 3 méthodes de caractérisation des RdRps ont été appliquées avec comme jeu test les données métagénomiques de *TARA*-Oceans. Pour HMMER et BLAST, les seuils de *e-value* ont été fixés à 10^{-4}. Pour Protomata, nous avons utilisé le modèle basé sur le jeu de paramètres 8.1.4 avec un seuil de score fixé à 35. Ce jeu étant très grand, nous avons fait un premier test sur un échantillon contenant 523913 séquences. Sur cet échantillon, HMMER et Protomata n'ont identifié aucune séquence de RdRps. Blast n'en a trouvé qu'une. La méthode a été relancée sur l'ensemble du jeu de données métagénomiques. Cette fois HMMER en a trouvé 8 et BLAST en a trouvé 76 (dont 6 en commun avec HMMER). En revanche, les résultats de Protomata sont toujours en cours. En effet, le scan de Protomata est très dépendant du modèle. Pour un modèle très précis comme celui sélectionné, il va prendre beaucoup de temps (plusieurs jours voir semaines, sur un jeu de données aussi gros que celui de *TARA*-Oceans) alors que pour un modèle comme 1.2.1 avec un p à 20 il prend environ 24heures. BLAST a mis un peu plus de 24 heures et HMMER en revanche a tourné en quelques minutes. On retrouve peu de séquences avec nos méthodes mais il faut comprendre ici que le jeu de données correspond à la fraction cellulaire et non virale. En effet, les données de la fraction virale ne sont pas disponibles. Toutefois, la fraction cellulaire contient des organismes infectés. Donc il y a des virus mais surement une faible proportion comparée à l'ensemble des séquences. Il n'est donc pas surprenant de retrouver peu de séquences de RdRps virales et au vu des résultats de validation il est également possible qu'il y ait des RdRps eucaryotes qui soient identifiées parmi nos résultats.

4 Conclusions

Nous avons étudié ici deux familles de protéines virales selon deux approches totalement différentes : la caractérisation des RdRps virales en séquence par 3 méthodes et l'étude de la spécificité des CFs des protéines de capsides.

Les résultats ont permis de cerner les difficultés pour caractériser et identifier les séquences de RdRps virales et les discriminer correctement des RdRps cellulaires. Toutefois, ces résultats pourraient être améliorés en se concentrant uniquement sur la partie RdRp des polyprotéines virales. Nous avons réussi à mettre en avant des fragments de structures spécifiques des virus grâce aux CFs, ce qui semble très prometteur pour améliorer la caractérisation des protéines virales et leur identification dans les données métagénomiques.

Ainsi, il serait très intéressant de continuer la caractérisation des structures de capsides. Une étude du type de structure des CFs spécifiques des virus pourrait être envisagée, mais surtout il faudrait s'assurer que les séquences associées aux CFs permettent d'identifier les séquences virales. Nous pourrions également envisager d'étudier l'influence sur la spécificité des CFs des différents seuils tels que σ et τ utilisés pour leur création. En effet, ces paramètres influent sur le nombre et la longueur moyenne des CFs. Ainsi, certains paramètres pourraient permettre d'obtenir plus de CFs spécifiques des virus, ou moins de non spécifiques. Enfin, certaines améliorations pourraient être envisagées dans l'implémentation de Yakusa, comme par exemple l'ajout du score ASD [20] (score prenant en compte la totalité du fragment contrairement au filtre des distances implémenté) pour comparer de manière plus précise les fragments de la requête et ceux trouvés par le logiciel.

Références

[1] Suttle, C. A. Marine viruses—major players in the global ecosystem. *Nature Reviews Microbiology* **5**, 801–812 (2007).

[2] Danovaro, R. *et al.* Marine viruses and global climate change. *FEMS microbiology reviews* **35**, 993–1034 (2011).

[3] Altschul, S. F., Gish, W., Miller, W., Myers, E. W. & Lipman, D. J. Basic local alignment search tool. *Journal of Molecular Biology* **215**, 403–410 (1990).

[4] Crotty, S., Cameron, C. E. & Andino, R. Rna virus error catastrophe : direct molecular test by using ribavirin. *Proceedings of the National Academy of Sciences* **98**, 6895–6900 (2001).

[5] Hurwitz, B. L. & Sullivan, M. B. The pacific ocean virome (pov) : a marine viral metagenomic dataset and associated protein clusters for quantitative viral ecology. *PLoS One* **8**, e57355 (2013).

[6] Cheng, S. & Brooks III, C. L. Viral capsid proteins are segregated in structural fold space. *PLoS Comput Biol* **9**, e1002905 (2013).

[7] Černỳ, J., Bolfíková, B. Č., Valdes, J. J., Grubhoffer, L. & Ržek, D. Evolution of tertiary structure of viral rna dependent polymerases. *PloS One* **9**, e96070 (2014).

[8] Ahlquist, P. Rna-dependent rna polymerases, viruses, and rna silencing. *Science* **296**, 1270–1273 (2002).

[9] Eddy, S. R. Profile hidden markov models. *Bioinformatics* **14**, 755–763 (1998).

[10] Kerbellec, G. *Apprentissage d'automates modélisant des familles de séquences protéiques*. Ph.D. thesis, Université Rennes 1 (2008).

[11] King, A. M., Adams, M. J. & Lefkowitz, E. J. *Virus taxonomy : ninth report of the International Committee on Taxonomy of Viruses*, vol. 9 (Elsevier, 2012).

[12] Galiez, C. & François, C. Structural conservation for remote homologues : better and further in contact fragments. *3DSIG : Structural Bioinformatics and Computational Biophysics* (2015).

[13] Carpentier, M., Brouillet, S. & Pothier, J. Yakusa : a fast structural database scanning method. *Proteins : Structure, Function, and Bioinformatics* **61**, 137–151 (2005).

[14] Edgar, R. C. Muscle : multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Research* **32**, 1792–1797 (2004).

[15] Li, W., Jaroszewski, L. & Godzik, A. Clustering of highly homologous sequences to reduce the size of large protein databases. *Bioinformatics* **17**, 282–283 (2001).

[16] Sharma, O. P., Jadhav, A., Hussain, A. & Kumar, M. S. Vpdb : Viral protein structural database. *Bioinformation* **6**, 324 (2011).

[17] Edgar, R. C. Search and clustering orders of magnitude faster than blast. *Bioinformatics* **26**, 2460–2461 (2010).

[18] Davis, J. & Goadrich, M. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on Machine learning*, 233–240 (ACM, 2006).

[19] Maida, Y. & Masutomi, K. Rna-dependent rna polymerases in rna silencing. *Biological Chemistry* **392**, 299–304 (2011).

[20] Galiez, C. & Coste, F. Amplitude spectrum distance : measuring the global shape divergence of protein fragments. *Sous presse* (2015).

Annexes

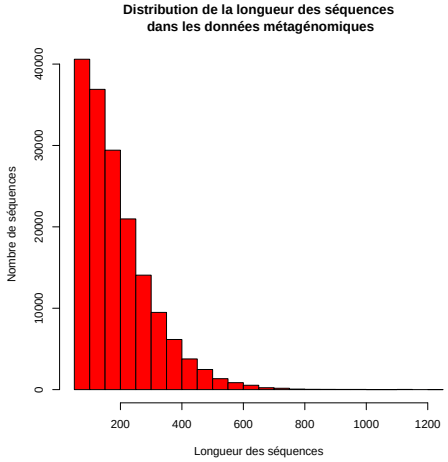


FIGURE A1 – Histogramme de distribution de la longueur des séquences des données métagénomiques de *TARA*-Oceans (en nombre de résidus).

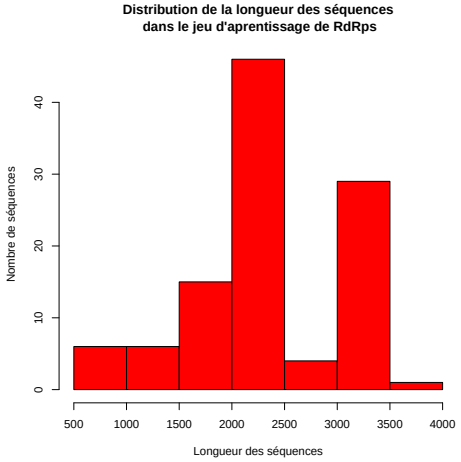


FIGURE A2 – Histogramme de distribution de la longueur des séquences du jeu d'apprentissage de RdRps (en nombre de résidus).

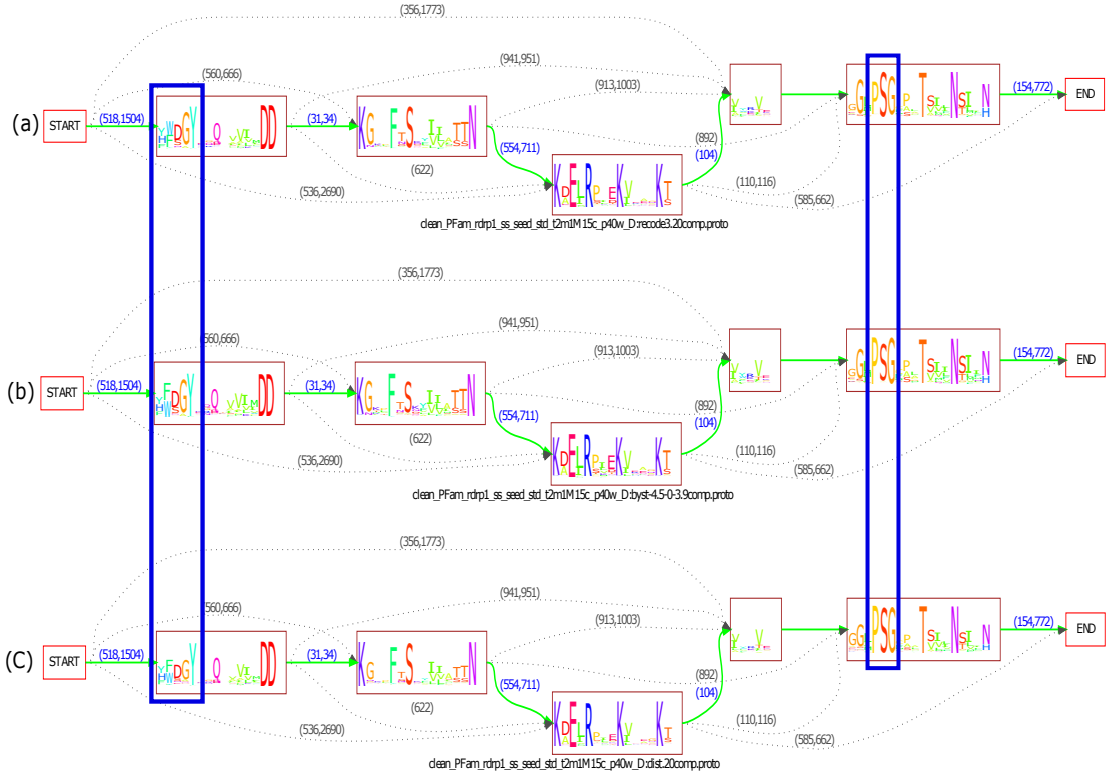


FIGURE A3 – Représentation et comparaison des différents automates produits par Protomata avec différentes mixtures de Dirichlet pour un même jeu d'apprentissage (séquences de la graine de la famille PFam RdRp_1 : PF00680). Les mixtures utilisées ici sont (a) recode3.20comp, (b) byst-4.5-0-3.9comp, et (c) dist.20comp. Les cadres bleus suggèrent des positions où les *pseudocounts* influent les PSSMs. Pour chaque automate, on peut voir l'enchaînement des différents blocs représentés par leur PSSM associée. La mixture utilisée pour l'automate (a) a tendance à favoriser les acides aminés polaires (comme la sérine ou lysine) tandis que celle utilisée pour l'automate (b) a tendance à favoriser les acides aminés apolaires (comme la proline ou la glycine). Celle utilisée pour construire l'automate (c) est plus neutre. Or les motifs conservés chez les RdRps présentent aussi bien des résidus apolaires que polaires, il est donc difficile de pencher vers un modèle favorisant l'une des deux catégories.

Identifiant	t	ms	p	Taux d'erreur MH	Taux d'erreur mp	Marges	Pr MH	R MH	F-m MH	Pr mp	R mp	F-m mp
1.1.3	0.01	15	5	0.4	0.18	15	0.77	0.98	0.86	1	0.3	0.46
1.1.6	0.01	15	10	0.41	0.19	50	0.76	1	0.86	1	0.28	0.44
1.1.10	0.01	10	20	0.29	0.25	4	0.73	0.98	0.84	0.97	0.57	0.72
3.1.10	0.1	10	20	0.30	0.24	4	0.74	0.98	0.84	0.94	0.58	0.72
4.1.4	0.1	25	5	0.38	0.17	57	0.83	0.98	0.90	1	0.32	0.48
4.1.16	0.1	25	20	0.30	0.25	5	0.71	0.99	0.83	1	0.49	0.66
5.1.1	1	10	5	0.39	0.21	14	0.73	0.99	0.84	1	0.31	0.47
5.1.11	1	12	20	0.29	0.29	4	0.67	0.99	0.79	0.84	0.68	0.75
7.1.2	2	12	5	0.38	0.25	28	0.70	0.98	0.82	1	0.32	0.48
8.1.4	2	25	5	0.38	0.14	149	0.83	0.98	0.9	1	0.33	0.50
8.1.16	2	25	20	0.27	0.30	11	0.65	1	0.78	1	0.51	0.68
9.1.2	3	12	5	0.39	0.23	20	0.71	0.99	0.83	1	0.30	0.46
9.1.12	3	15	20	0.32	0.35	5	0.61	1	0.76	0.98	0.44	0.61
10.1.12	3	25	15	0.34	0.20	5	0.74	1	0.85	1	0.38	0.55
10.1.13	3	17	20	0.30	0.29	5	0.66	0.98	0.79	1	0.46	0.63
1.2.1	0.01	10	25	0.25	0.20	7	0.75	0.97	0.85	0.81	0.83	0.82
2.2.6	0.01	20	30	0.23	0.20	15	0.75	1	0.86	1	0.58	0.73
2.2.10	0.01	20	40	0.16	0.20	24	0.73	0.99	0.84	0.97	0.73	0.83
2.2.11	0.01	22	40	0.20	0.20	7	0.75	0.97	0.85	1	0.64	0.78
3.3.6	0.1	15	30	0.2	0.21	12	0.73	0.97	0.83	0.96	0.67	0.79
3.3.1	0.1	10	25	0.31	0.29	24	0.66	0.98	0.78	0.75	0.83	0.79
4.2.6	0.1	20	30	0.21	0.22	25	0.71	0.99	0.83	1	0.62	0.77
4.2.8	0.1	25	30	0.21	0.22	9	0.72	0.98	0.83	1	0.62	0.77
6.2.2	1	20	25	0.26	0.17	27	0.78	0.97	0.86	1	0.52	0.68
6.2.5	1	17	30	0.22	0.32	19	0.64	0.98	0.77	0.97	0.62	0.77
5.2.2	1	12	25	0.25	0.29	4	0.68	0.95	0.79	0.70	0.97	0.81
5.2.3	1	15	25	0.26	0.25	9	0.70	0.98	0.81	0.97	0.53	0.69
7.2.3	2	15	25	0.26	0.31	5	0.65	0.98	0.78	0.84	0.73	0.78
8.2.6	2	20	30	0.22	0.30	9	0.65	0.98	0.78	0.93	0.66	0.77

TABLE A1 – Résultats sélectionnés lors de la validation croisée. L'identifiant fait référence aux différents tests réalisés. Les valeurs t, ms et p correspondent aux différents paramètres testés. MH correspond au seuil mean-half et mp au seuil min-po. Pr correspond à la précision, R correspond au rappel, et F-m correspond à la F-mesure.

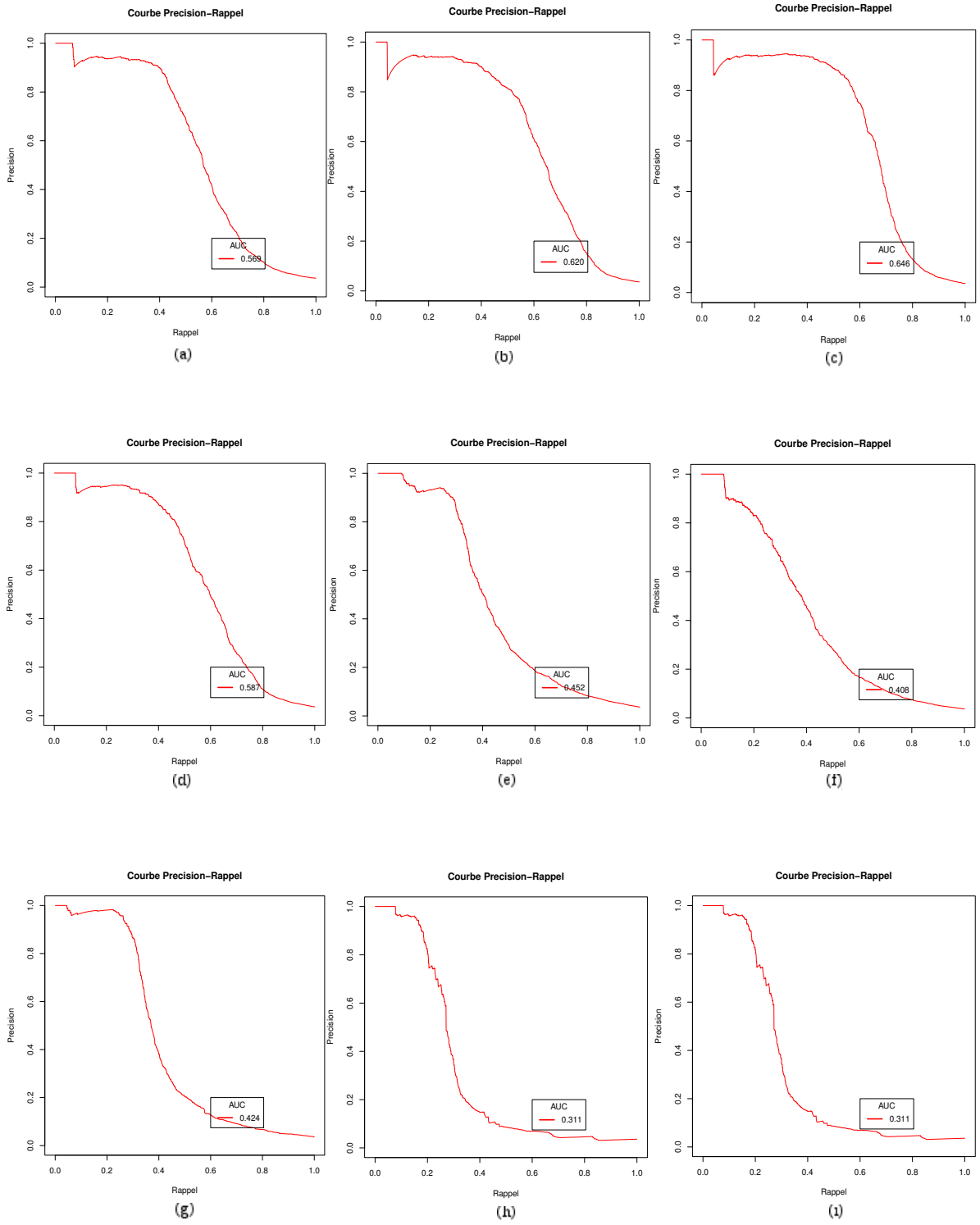


FIGURE A4 – Courbes précision rappel pour le jeu de paramètres 1.2.1 avec un p égal à (a) 5, (b) 7, (c) 10, (d) 12, (e) 15, (f) 17, (g) 20, (h) 22 et (i) 25. L'encadré AUC indique la valeur de l'aire sous la courbe.

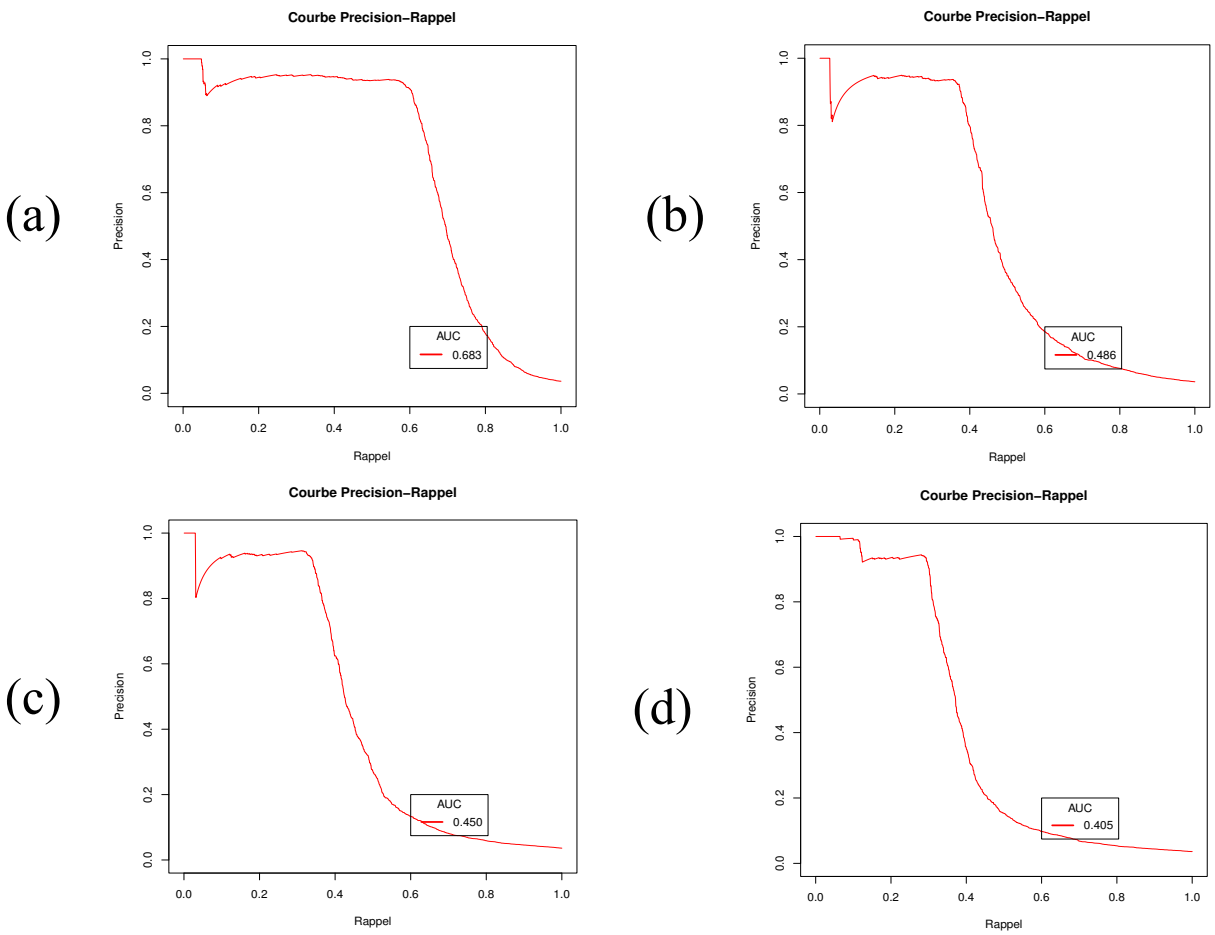


FIGURE A5 – Courbes précision rappel pour le jeu de paramètres 8.1.4 avec un p égal à (a) 5, (b) 7, (c) 10 et (d) 12. L'encadré AUC indique la valeur de l'aire sous la courbe.

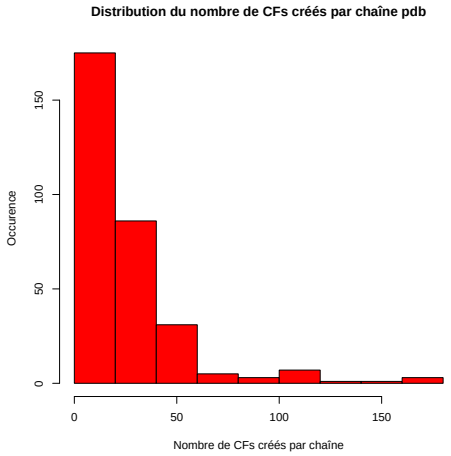


FIGURE A6 – Histogramme de distribution du nombre de CFs extraits par chaîne PDB.

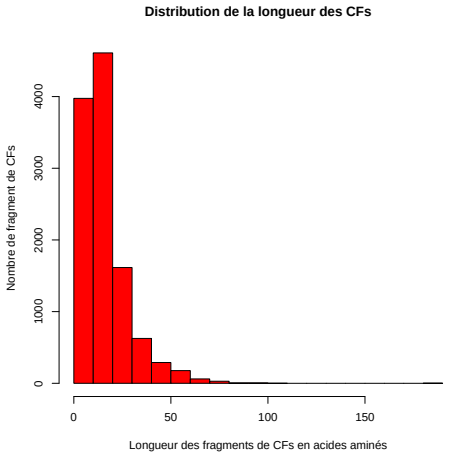


FIGURE A7 – Histogramme de distribution de la longueur des fragments de CFs.

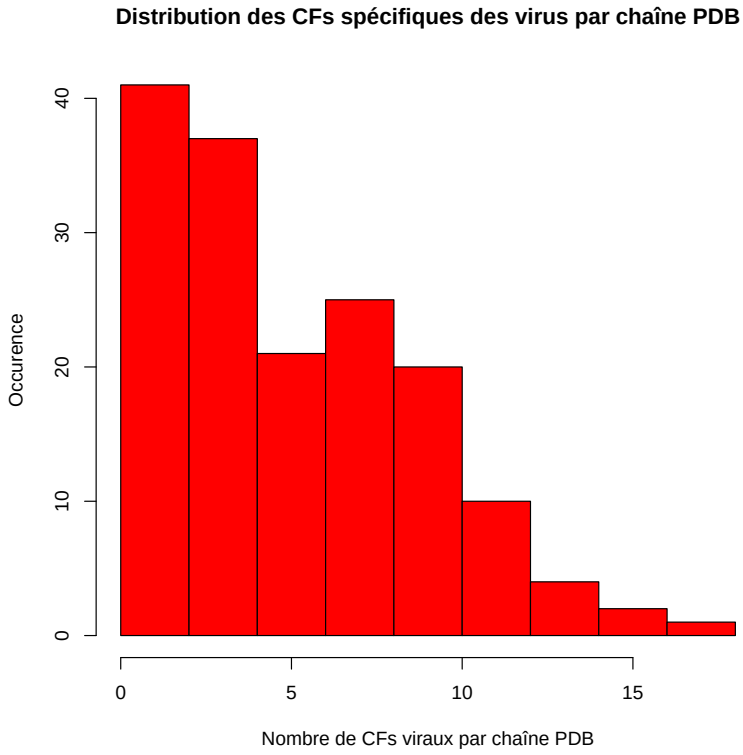


FIGURE A8 – Histogramme de distribution du nombre de CFs spécifiques des virus par chaîne PDB.