

UNIVERSITÉ DU QUÉBEC À MONTRÉAL

RAPPORT FINAL
INTERFACE WEB VIGENO

TRAVAIL PRÉSENTÉ À
VIRGINIE CALDERON

DANS LE CADRE DU COURS
BIOINFORMATIQUE ET SCIENCES DE LA SANTÉ
BIF7104

PAR
LOGAN SCHWARTZ
SCHL13068709

2 MAI 2016

SOMMAIRE

1. Introduction
2. Besoins et technologies utilisées
3. Interface web Vigeno
4. Les défis/difficultés du projet
5. Conclusion

INTRODUCTION

1.1 Contexte biologique:

La protéogénomique est le champ d'étude qui s'intéresse à l'annotation et la caractérisation de séquences génomiques potentiellement codantes à partir de données de protéomiques. La plupart du temps, ces données sont obtenues à partir de séquençages "shotguns" en spectrométrie de masse en phase gazeuse (MSG) couplées à la chromatographie en phase liquide. Cette méthode ne permet toutefois que de déterminer les séquences issues d'un pool de peptides clivés enzymatiquement dont la longueur n'excède pas 20 acides aminés. Les peptides en phase gazeuse qui pénètrent dans le MSG entrent en collision de faible énergie avec un gaz inerte, généralement de l'hélium ou de l'argon, permettant de séparer les peptides au niveau des liens peptidiques uniquement et d'analyser chaque acide aminé séparément. Les séquences partielles obtenues à partir des spectres MSG sont par la suite comparées à une banque de séquences protéiques connues et dont l'identification repose sur la reproductibilité de l'alignement. Bien entendu, il s'agit essentiellement d'une identification hypothétique car seulement certains fragments de protéines seront utilisés afin de déterminer la composition totale de l'échantillon.

Le séquençage protéique réalisé à l'aide de cette méthode a permis de cartographier plus de 80% du protéome humain et de la souris. Toutefois, cette façon de faire ne permet que d'identifier les protéines déjà répertoriées faisant parties des régions codantes et du bon cadre de lecture. En effet, il arrive que les protéines exprimées comportent des mutations, polymorphismes, ou encore, originent de régions intergéniques, de transcrits initiés par un codon non-AUG et/ou de mauvais cadre de lecture. Conséquemment, ces protéines étant pour la plupart non répertoriées dans les bases de données, la comparaison des résultats obtenus à partir du spectre MSG ne permet pas d'identification directe. Il devient dès lors nécessaire de réaliser des alignements manuels sur les séquences génomiques afin de retrouver les gènes d'origines.

Initialement, Vigeno a vu le jour dans le cadre d'un projet de recherche visant à caractériser la diversité des peptides de reconnaissance du soi par le complexe d'immuno-histocompatibilité de classe 1 (MAPs) des cellules B. Ces peptides qui proviennent généralement de débris protéiques, forment l'immuno-peptidome et n'excèdent généralement pas 10 acides aminés. Au cours de cette étude il a été démontré qu'environ 10% des MAPs étaient formés à partir de transcrits de régions non-génomiques ou en dehors du cadre de lecture conventionnel. Il s'est avéré que ces peptides "cryptiques" provenaient de protéines très courtes aux régions C-terminale atypiques et qu'ils arboraient davantage de polymorphismes dont certains avaient un potentiel immunogénique important. Les données générées ont nécessité plus de 3000 alignements dont environ 300 n'ont pu être identifiés directement en raison d'un mauvais cadre de lecture d'origine. Bien qu'il puissent être utilisés à d'autres fins, le projet Vigeno ayant vu le jour au cours de l'élaboration de cette étude s'intégrait parfaitement à la problématique des nombreux alignements répétitifs puisqu'il permet l'alignement de petits peptides cryptiques sur la séquence protéique issues du cadre de lecture d'origine en plus de mettre en évidence les différents polymorphismes qu'ils arborent.

1.2 ViGeno:

Dans le cadre d'un projet de stage à la plateforme de bio-informatique de l'IRIC, j'ai travaillé sur la mise au point d'une application offrant aux biologistes d'extraire l'information d'un ensemble de données SNPs d'une façon rapide et visuel. Si d'autres interfaces graphiques existent dans le cadre de projets similaires tels que IGV, bioedit ou sniplay, aucune n'est conçue dans l'idée de travailler sur de courtes séquences telles que des k-mers ou des peptides. ViGeno (*Visual Genomic*) est une application web (open source) encapsulant certaines fonctionnalités de pyGeno (<http://pygeno.irc.ca/>). Cette application propose une interface graphique présentant une forme augmentée des résultats soumis en input, ainsi que des outils visuels permettant à l'utilisateur de filtrer les informations pertinentes plus facilement à partir d'un ensemble de données.

Le but initial de ce travail était de mettre au point et d'implémenter de nouveaux scripts permettant la transformation de l'output des résultats augmentés en un input interprétable par une/des animations disponibles via la librairie D3.js. Les animations produites devaient être complémentaires à celles déjà existantes dans ViGeno, et permettre de faire ressortir de nouvelles informations pertinentes de l'ensemble de données. Finalement, ces objectifs ont été revu à la baisse par manque de temps.

BESOINS ET TECHNOLOGIES UTILISÉES

Dans le cadre de ce projet, nous avons eu besoin de disposer d'une interface adaptative (*'responsive'*) afin que l'utilisateur puisse interagir directement avec l'application afin d'explorer ses données plus facilement. Pour ce faire, AngularJS s'est avéré un choix logique. Tout d'abord parce que cette technologie englobe un ensemble de mécaniques que l'on retrouve séparé dans d'autre technologie. AngularJS permet entre autre de faire du *'data binding'* qui lie une variable de façon transparente à travers toutes les vues de l'utilisateur. AngularJS permet également de gérer les vues comme Flask et propose en plus un langage de *'templating'* relativement poussé. AngularJS est pensé pour être modulaire, c'est à dire que l'on peut gérer de façon indépendante des panneaux présents au sein de la même page, on obtient donc un plus grand contrôle sur le rendu de la page.

L'arbre interactif permettant d'explorer le jeux de données représenté dans le tableau du client est codé avec la librairie Javascript D3. Cette librairie est reconnue comme la référence concernant les animations côté client. C'est pour cette raison et pour l'accessibilité à sa documentation sur internet que nous avons utilisé cette librairie. Finalement, les *'templates'* de notre applications utilisent la librairie HTML/CSS Bootstrap afin d'offrir un support multi-plateforme à nos utilisateurs.

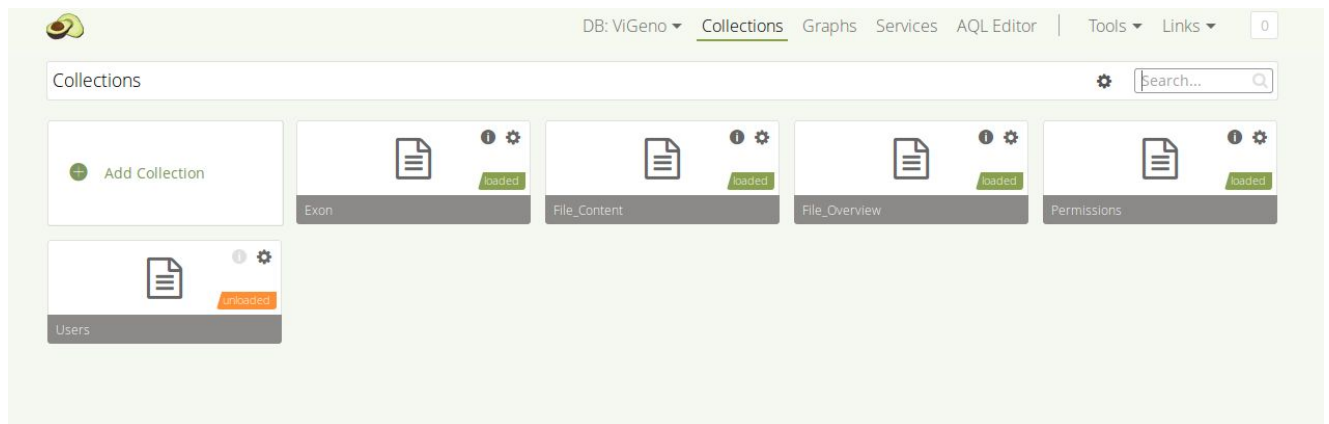
Dans le cadre de ce projet nous avons eu besoins d'un serveur codé en python servant d'intermédiaire entre le client et pyGeno et l' *"Object Relational Mapping"* (ORM) pyArango permettant de communiquer avec notre banque de données NoSQL ArangoDB. Par soucis de simplicité nous avons utilisé le cadre de travail Flask en back end. Nous avions besoins initialement d'une banque de

données pour gérer nos comptes utilisateurs et les fichiers hébergés sur le serveur afin de faciliter la collaboration entre les utilisateurs. ArangoDB a été choisie car nous ignorions au départ qu'elles seraient nos besoins futurs et cette solution apportait d'avantage de flexibilité que ses homologues SQL en nous permettant justement de modifier éventuellement les champs dans nos collections par la suite si le besoin était.

INTERFACE WEB VIGENO

3.1 ArangoDB:

La gestion des comptes utilisateurs impliquait l'utilisation d'une banque de donnée, nous avons choisi ArangoDB pour les raisons mentionnées précédemment. Notre banque de données est constituée de plusieurs collections, soit Exon contenant l'ensemble des exons du génome issu de Grch 37.75. Cette collection permet d'ajouter de l'information à notre input en renvoyant pour chaque ligne l'information qu'elle contient correspondant au "transcriptID" de la ligne. Cette version augmentée de l'input est enregistrée dans la collection "File_content" afin de sauvegarder le résultat de l'analyse et de pouvoir le visualiser plus tard. Au moment de l'upload d'un fichier, ses colonnes sont enregistrées dans la collection 'File_Overview' avec pour chaque colonne du fichier le typage lui étant associé. Cette étape est nécessaire puisqu'en NoSQL le typage de nos variables est perdu lorsqu'elles sont enregistrées dans la banque de données. On pourra par la suite manipuler/filtrer les données de chaque colonne en fonction du type. La collection Permissions et Users conserve les informations personnelles des comptes utilisateurs et des permissions de chaque utilisateurs pour accéder aux fichiers.



3.2 ViGeno:

Page d'accueil :

La page d'accueil de Vigeno offre un résumé de son rôle qu'il joue dans l'analyse des SNPs de l'immuno-peptidome, informe l'utilisateur du format de fichier accepté pour les analyses (csv) et montre une image récapitulative des diverses fonctionnalités de l'application. Dans le haut de la page, des liens ont été créés dans l'intention de diriger l'utilisateur vers une page contenant un formulaire de connexion qui lui même en contient un qui dirige vers une page de formulaire d'enregistrement pour les utilisateurs n'ayant pas de compte. Tous les comptes enregistrés sur cette page seront ultimement enregistré sur un document de la base de données ArangoDB.

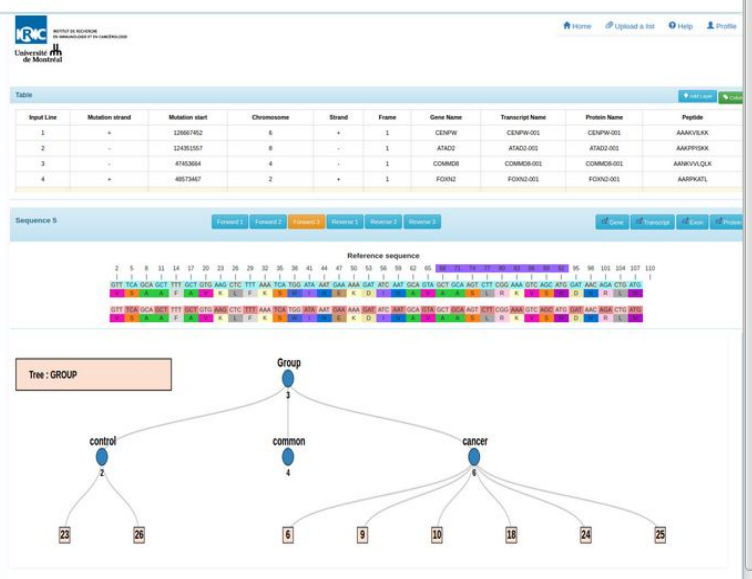
Welcome in ViGeno !

ViGeno is a web application based on the well known pyGeno software. It offer features that analyze your SNPs/immunopeptide data for you and retrieve data of interest out of data's jungle quickly and efficiently.

ViGeno takes csv data in input and return an augmented version of them after looking out in databanks for you. It also let you work with custom columns so you could filter your data with our interactive tree based on those parameter.

ViGeno give you the opportunity to share your analysis with colleagues or people around the world in one click. You may choose to attribute different permissions to any other user such as read, write, give.

ViGeno is open source and downloadable on github so you may also choose to privately host your own version on your server.



Un lien qui dirige vers une page membre permettra à chacun des utilisateurs enregistrés de pouvoir consulter, elles aussi, à partir d'un document administré par la base de donnée Arango les différentes analyses lancées préalablement. Sur cette page, une option permet le partage de fichiers entre utilisateurs de manière à pouvoir faciliter les collaborations. Des informations relatives au titre de l'analyse, la date de création, le nom de l'auteur et les permissions de lecture, édition et partage sont aussi accessible pour le moment.

Data tab						
File name	Upload date	Owner	Permissions	See results	Shared	Share file
Out1.csv	2016-04-25	You	Read,Write,Give		No	

Input:

Les fichiers d'entrées sont au format CSV disposant au minimum des paramètres ensT(ensembl transcript id), start, end, strand et le chromosome. L'utilisateur peut à sa guise ajouter d'autre colonnes afin de pouvoir appliquer des filtres basés sur ces paramètres par la suite.

```
peptide,start,end,chromosome,strand,ensG,ensT,group,sex
AAAKRQVL,3405592,3405616,2,+,ENSG00000171853,ENST00000324266,common,Male
AAAKVILKK,126667452,126669626,6,+,ENSG00000203760,ENST00000368328,common,Female
AAKPPISKK,124351557,124351584,8,-,ENSG00000156802,ENST00000287394,common,Male
AANKVVLQK,47453664,47455087,4,-,ENSG00000169019,ENST00000381571,common,Male
AARPKATL,48573467,48573491,2,+,ENSG00000170802,ENST00000340553,common,Female
AASLRKVS,201682764,201682791,2,+,ENSG00000082153,ENST00000409600,common,Female
AAYNVLPK,86406561,86406588,2,-,ENSG00000132305,ENST00000449247,cancer,Male
AEAAEKLKNRY,35815898,35815931,22,+,ENSG00000100297,ENST00000382011,common,Female
AEAEASVRM,30698339,30698485,6,-,ENSG00000137312,ENST00000376389,common,Female
AEAEAVREY,20023073,20023103,20,-,ENSG00000101343,ENST00000536226,cancer,Female
AEAEKLGGSY,41355078,41355111,8,+,ENSG00000147533,ENST00000520817,cancer,Male
```



Interface utilisateur:

L'interface utilisateur est constituée d'un tableau contenant pour chacune des lignes une version augmenté de notre fichier d'input. d'un panneau de polymorphisme présentant les séquences dans le génome de référence et dans dbSNPs. On upload un fichier en cliquant sur le bouton '*upload a list*' en haut à droite de la page.

Table										Add Layer Column	
Input Line	Mutation strand	Mutation start	Chromosome	Strand	Frame	Gene Name	Transcript Name	Protein Name	Peptide		
1	+	126667452	6	+	1	CENPW	CENPW-001	CENPW-001	AAAKVILKK		
2	-	124351557	8	-	1	ATAD2	ATAD2-001	ATAD2-001	AAKPPISKK		
3	-	47453664	4	-	1	COMMD8	COMMD8-001	COMMD8-001	AANKVVLQLK		
4	+	48573467	2	+	1	FOXN2	FOXN2-001	FOXN2-001	AARPKATL		

Modal des colonnes:

Puisque l'application renvoie une version qui génère un nombre de colonne supérieur au fichier d'input, la quantité de colonne s'il devient trop grand peut rapidement diminuer la visibilité des éléments dans le tableau. le bouton 'column' permet donc de masquer/afficher les colonnes selon les besoins de l'utilisateur.

 INSTITUT DE RECHERCHE
EN IMMUNOLOGIE ET EN CANCÉROLOGIE
Université de Montréal

Table

Input Line	Mutation strand	Mutation start	Chromosome
1	+	126667452	6
2	-	124351557	8
3	-	47453664	4
4	+	48573467	2
-	-	-	-

Sequence 1

Forward 1

2 5 8 11 14 17 20 23 26 29 32 35 38 41 44

GTC CAT CTG AAC TGT TTA CTG TTT GTT CAT CGA TTA GCA GAA GAG

V H L N C L L F V H R L A E E

GTC CAT CTG AAC TKT TTA CTG TTT GTT CAT CGA TTA GCA GAA GAG

V H L N C / F L L F V H R L A

Table Columns

Update shown columns

☒strand_mutation
☒start_mutation
☐end_mutation
☐chromosome
☒strand
☐start
☐end
☐length
☒frame
☐CDS_start
☐CDS_end
☐CDS_length
☒gene_name
☐gene_id
☒transcript_name
☐transcript_id
☐id
☐number
☒protein_name
☐protein_id
☒peptide
☐group

OK

[Home](#) [Upload a list](#) [Help](#) [Profile](#)

[Add Layer](#) [Column](#)

Protein Name		Peptide
001	CENPW-001	AAAKVILKK
001	ATAD2-001	AAKPPISKK
001	COMMD8-001	AANKVVLQLK
001	FOXN2-001	AARPKATL

[Gene](#) [Transcript](#) [Exon](#) [Protein](#)

98 101 104 107 110 113

STA CTG GCC GCA GCA AAG

V L A A A K

STA CTG GCC GCA GCA AAG

K E H V L A A A K

Panneaux polymorphisme:

Le panneau de polymorphisme permet de visualiser les séquences identifiées comme mutées dans notre input. On observe tout d'abord la séquence de référence (acides nucléiques/acides aminés) du génome humain (GRCh37.75) séparé par un espace et également en dessous la séquence de dbSNP. De cette façon, l'utilisateur peut rapidement observer la présence de modifications SNPs connu pour cette position dans la séquence de dbSNP laissant présager que cette modification serait à l'origine de la séquence mutée pour cette ligne du tableau. Ce panneau possède des boutons permettant de changer le cadre de lecture où d'aller consulter la page ensembl associée avec cette ligne du tableau.

Sequence 5

Forward 1Forward 2Forward 3Reverse 1Reverse 2Reverse 3

GeneTranscriptExonProtein

Reference sequence

258111417202326293235384144475053565962656871747780838689929598101104107110

GTT TCA GCA GCT TTT GCT GTG AAG CTC TTT AAA TCA TGG ATA AAT GAA AAA GAT ATC AAT GCA GTA GCT GCA AGT CTT CGG AAA GTC AGC ATG GAT AAC AGA CTG ATG

V S A A F A V K L F K S W I N E K D I N A V A A S L R K V S M D N R L M

GTT TCA GCA GCT TTT GCT GTG AAG CTC TTT AAA TCA TGG ATA AAT GAA AAA GAT ATC AAT GCA GTA GCT GCA AGT CTT CGG AAA GTC AGC ATG GAT AAC AGA CTG ATG

V S A A F A V K L F K S W I N E K D I N A V A A S L R K V S M D N R L M

Lien croisé vers ENSEMBL:

Lorsque l'utilisateur identifie une séquence qui présente un intérêt pour lui où qu'il veut simplement avoir davantage d'information, les boutons de liens croisés permettent de le rediriger vers la page d'ensembl associé à la séquence qu'il est en train de visualiser dans le panneau de polymorphisme. De cette façon, il peut aller extraire de l'information supplémentaire sur cette séquence.

Human (GRCh38.p5) Location: 2:200,811,546-200,827,338 Gene: BZW1 Transcript: BZW1-001

Transcript-based displays

- Summary
- Supporting evidence
- Sequence
 - Exons**
 - cDNA
 - Protein
- External References
 - General identifiers
 - Oligo probes
- Genetic Variation
 - Variant table
 - Variant image
 - Population comparison
 - Comparison image
 - Haplotypes
- Protein Information
 - Protein summary
 - Domains & features
 - Variants
- ID History
 - Transcript history
 - Protein history

Transcript: BZW1-001 ENST00000409600

Description basic leucine zipper and W2 domains 1 [Source:HGNC Symbol;Acc:HGNC:18380]

Synonyms Nbla10236, BZAP45, KIAA0005

Location [Chromosome 2: 200,811,546-200,827,338](#) forward strand.

About this transcript This transcript has [12 exons](#), is annotated with [9 domains and features](#), is associated with [228 variations](#) and maps to [63 oligo probes](#).

Gene This transcript is a product of gene [ENSG00000082153](#) [Show transcript table](#)

Exons

[Download sequence](#)

Exons/ Introns Translated sequence Flanking sequence Intron sequence UTR

Variants 3 prime UTR 5 prime UTR Frameshift Inframe deletion Missense Splice acceptor Splice donor

Splice region Stop gained Synonymous

Markup loaded

Show **All** entries [Show/hide columns](#)

No.	Exon / Intron	Start	End	Start Phase	End Phase	Length	Sequence
	5' upstream sequence					ccccagcctccgccccgcacgacccgagggccccgcctcgggcttc
1	ENSE00001578290	200,811,546	200,811,990	-	-	445	GACTTCGGTGCTGAGGAGGGGGCCGGCGGCAGGGACTCCAAGAGGACGCTCCAAC TCGAGACGGCGGGGGCGGGTCCGCTCTTCCAACCTCCATGTGTGAAGAGGGCGGAC

Sur cette page une redirection vers la page du transcrit est observable. Cependant, les boutons du panneau de polymorphismes permettent une redirection vers l'exon, le gène, le transcrit ou la protéine de la séquence affichée dans le panneau de polymorphisme.

Visualisation de peptide cryptique:

L'un des besoins à la base de la création de cette application est de permettre de distinguer les lignes de notre tableau correspondant aux peptides conventionnelles où cryptiques où étant inverse (mutation sur le brin opposée au brin normalement transcrit). Lorsqu'on clique sur une ligne du tableau, la séquence du génome de référence associé au transcrit est affiché dans le panneau de polymorphisme dans son cadre de lecture d'origine.

Table

[Add Layer](#)
[Column](#)

Input Line	Mutation strand	Mutation start	Chromosome	Strand	Frame	Gene Name	Transcript Name	Protein Name	Peptide	Group
1	+	126667452	6	+	1	CENPW	CENPW-001	CENPW-001	AAAKVILKK	common
2	-	124351557	8	-	1	ATAD2	ATAD2-001	ATAD2-001	AAKPPISKK	common
3	-	47453664	4	-	1	COMMD8	COMMD8-001	COMMD8-001	AANKVVLQK	common
4	+	48573467	2	+	1	FOXN2	FOXN2-001	FOXN2-001	AARPKATL	common
-	-	-	-	-	-	-	-	-	-	-

Sequence 4

[Forward 1](#)
[Forward 2](#)
[Forward 3](#)
[Reverse 1](#)
[Reverse 2](#)
[Reverse 3](#)

[Gene](#)
[Transcript](#)
[Exon](#)
[Protein](#)

Reference sequence

2 5 8 11 14 17 20 23 26 29 32 35 38 41 44 47 50 53 56 59 62 65 68 71 74 77 80 83 86 89 92 95 98 101 104 107 110 113 116 119 122 125 128 131 134 137 140 143 146 149 152 155 158 161 164 167 170
 AACTGTAAGAGTAAATGGGTCCAGTAATTGGAATGACTCCAGATAAGAGAGCTGAAACCCAGGAGCTGAAAAGATTGCAGGATTAGCCAGATTACAAAATGGGAAGCTTGCTGAAGCTGTTGATGCTGCCAGGCCGAAGGCCACTCTAGTGGACAGTGAGTCAGCAG
 N C K S K W V Q * L E * L Q I R E L K P Q E L K R L Q D * A R F T K W E A C L K L L M L P G R R P L * W T V S Q Q
 AACTGTAAGAGTAAATGGGTCCAGTAATTGGAATGACTCCAGATAAGAGAGCTGAAACCCAGGAGCTGAAAAGATTGCAGGATTAGCCAGATTACAAAATGGGAAGCTTGCTGAAGCTGTTGATGCTGCCAGGCCGAAGGCCACTCTAGTGGACAGTGAGTCAGCAG
 N C K S K W V Q * L E * L Q I R E L K P Q E L K R L Q D * A R F T K W E A C L K L L M L P G R R P L * W T V S Q Q

On peut alors comparer facilement si le peptide identifié pour cette ligne est celui retrouvé à la position mutée (surlignée en violet ci-dessus). Lorsque la séquence ne correspond pas, on peut modifier le cadre de lecture en cliquant sur les boutons des différents cadre de lecture. De cette façon, on se rend compte que le motif peptidique du panneau de polymorphisme et de la séquence match lorsque l'on utilise le cadre de lecture 'Forward 3' (ci-dessous).

Table

Add Layer Column

Input Line	Mutation strand	Mutation start	Chromosome	Strand	Frame	Gene Name	Transcript Name	Protein Name	Peptide	Group
1	+	126667452	6	+	1	CENPW	CENPW-001	CENPW-001	AAKVILKK	common
2	-	124351557	8	-	1	ATAD2	ATAD2-001	ATAD2-001	AAKPPISKK	common
3	-	47453664	4	-	1	COMMD8	COMMD8-001	COMMD8-001	AANKVVLQK	common
4	+	48573467	2	+	1	FOXN2	FOXN2-001	FOXN2-001	AARPKATL	common
-			-	-	-					

Sequence 4

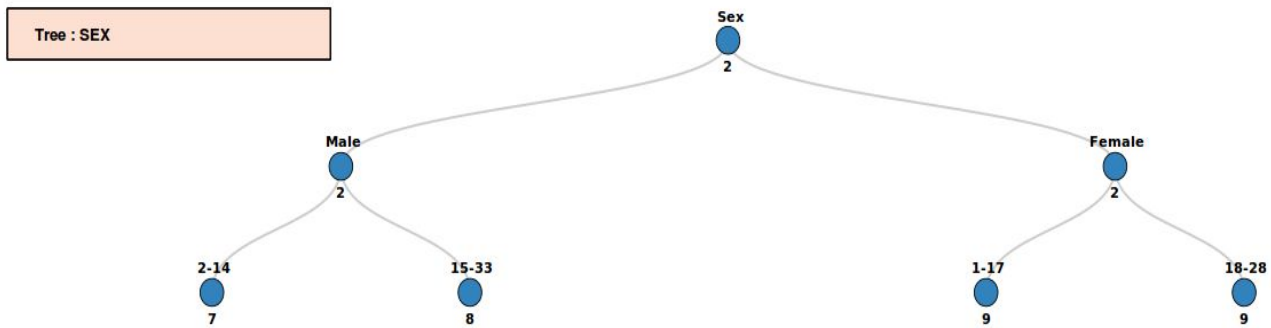
Forward 1 Forward 2 Forward 3 Reverse 1 Reverse 2 Reverse 3

Gene Transcript Exon Protein

Reference sequence																
2	5	8	11	14	17	20	23	26	29	32	35	38	41	44	47	50
53	56	59	62	65	68	71	74	77	80	83	86	89	92	95	98	101
104	107	110	113	116	119	122	125	128	131	134	137	140	143	146	149	152
155	158	161	164	167	170											
CTGTAAAGAGTAAATGGGTCCAGTAATTGGAATGACTCCAGATAAGAGAGCTGAAACCCAGGAGCTGAAAAGATTGCAGGATTAAGCCAGATTTACAAAATGGGAAGCTTGCTGAAGCTGTTGATGCTGCCAGGCCGAAGGCCACTCTAGTGGACAGTGAGTCAGCAGAT																
L	*	E	*	M	G	P	V	I	G	M	T	P	D	K	R	A
E	T	P	G	A	E	K	I	A	G	L	S	Q	I	Y	K	M
G	S	L	P	E	A	V	D	A	A	R	P	K	A	T	L	V
D	S	E	S	A	D											

Arbre interactif:

L'arbre interactif est l'une des *fonctionnalités* principales de ViGeno. L'idée en arrière de son implémentation est de permettre de réorganiser le contenu du tableau à l'utilisateur en fonction d'un paramètre de son choix. Étant donnée que le nombre de colonne dans le fichier d'input n'est pas limité, le paramètre sélectionné peut tout aussi bien être des méta-données ajoutées par l'utilisateurs. De cette façon, on peut très vite savoir comment les éléments de notre tableau se répartisse en fonction d'un paramètre. Ici par exemple, on cherche à savoir combien il y a de femme dans notre tableau. générer un arbre sur ce paramètre permet en 2 cliques d'obtenir la réponse (9+9=>18). Pour générer un arbre, il suffit de cliquer sur le titre de la colonne désirée.



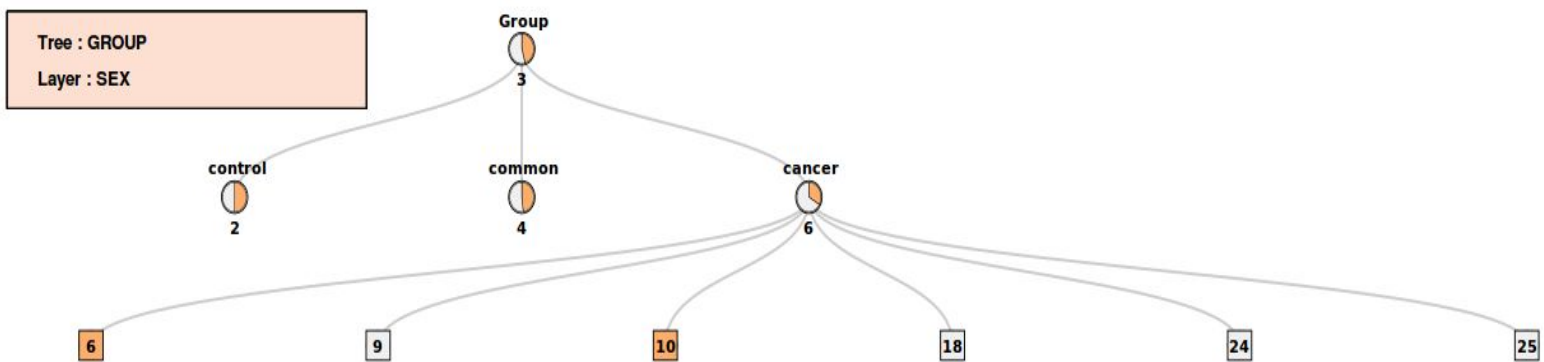
Arbre interactif et filtre de couleur:

Le filtre de couleur permet de répondre rapidement à plusieurs questions biologiques. Si, par exemple, l'utilisateur désirait connaître le niveau de dépendance du sexe d'une personne par rapport à la condition biologique générale dans le cadre d'une étude sur les cancers du sein hormonaux dépendant. Afin d'appliquer un filtre de couleur, il suffit de sélectionner le titre d'une colonne (i.e chromosome, group) afin de produire un arbre sur ce paramètre. Ensuite, il est possible d'activer le filtre de couleur en cliquant sur le bouton 'layer' en haut à droite du tableau et en re cliquant ensuite sur le paramètre que l'on désire utiliser comme calque.

On observe notre arbre avec un graphique en pointe à chaque noeuds représentant la répartition des noeuds sous-jacents en fonction du sexe des patients ('Male/Female'). Avec cette représentation l'utilisateur est en mesure de réorganiser l'information contenu dans le tableau client afin de pouvoir visualiser rapidement s'il existe des liens entre les paramètres correspondant aux colonnes du tableau.

Ci dessous, on observe que la répartition homme/femme est similaire dans les groupes de peptides communs et contrôles mais qu'il existe un déséquilibre pour le groupe de peptide associés au groupe des patients atteints d'un cancer. Le chiffre en dessous du noeud représente le nombre de noeuds sous-jacents. Cette information nous permet de conclure si le déséquilibre est représentatif de la

population totale. Les feuilles de l'arbre (correspondant aux numéro de ligne dans le tableau) sont connectées au tableau, ce faisant, un double clique sur une feuille de l'arbre permet de revenir à la ligne correspondante dans le tableau.



Les défis/difficultés du projet:

D'abord ce projet était complètement démesuré par rapport à mes compétences respectives de l'époque, c'est pour cette raison que cette expérience a autant été enrichissante. Développer une application '*fullstack*' (client/serveur) de A à Z est un défi relativement challengeant à la base et le faire dans un contexte où l'on part de 0 et où l'on doit apprendre à manipuler toutes les technologies en même temps, rehausse le niveau d'un cran. Chaque journée a amené son lot de nouveaux concepts et '*bugs*' à résoudre pouvant prendre parfois jusqu'à 20h de travail acharné. Apprendre à maîtriser les technologies dans un laps de temps suffisamment raisonnable afin de pouvoir en faire une application présentable représentait par conséquent le défi principal.

La seconde difficulté en importance fut de réaliser l'arbre interactif le script générant l'input de l'arbre en Javascript Object Notation (JSON). Le JSON est un format interprétable par Javascript qui est vite devenu un format standard pour

communiquer l'information du serveur vers le client. Le JSON est globalement une suite récursive de dictionnaire dans des listes de dictionnaire dans laquelle il faut structurer l'information selon le format d'input de l'animation D3 que l'on souhaite utiliser. La complexité réside dans le fait d'ajouter de l'information dans le JSON sans nuire à la structure d'origine afin de pouvoir ajouter des fonctionnalités sans briser l'animation D3. Un bon exemple est la présence de la fonctionnalité qui permet de créer des listes de noeuds lorsque le nombre de noeuds sous-jacents excède 10 noeuds. Ajouter cette fonctionnalité a augmenté énormément le niveau de complexité dans la mesure où chacun des niveau hiérarchique peut devenir infini, contrairement à l'animation originale, si elle est gérée de façon récursive. De plus, l'ensemble du code mériterait d'être repensé sur le modèle actuel pour diminuer la complexité et optimiser le code.

Conclusion

En mars 2016, mon stage s'est terminé et ne trouvant pas de repreneur chez les nouveaux stagiaires le projet a été abandonné. Il reste une grande quantité de travail pour envisager de déployer Vigeno. Ce projet aura eu au moins le mérite de montrer qu'il est possible de faire ressortir naturellement de l'information pertinente à partir de tableau pouvant faire des milliers de lignes avec son arbre interactif.