



**Rapport de projet
M1 BIM-info 2020-2021**

Discovery of protein specificity signatures in evolutionary splicing graphs

Antoine Szatkownik

Responsables de projet :
Elodie Laine
Hughes Richard

Remerciements

Mes plus vifs remerciements à Elodie Laine, de par sa disponibilité et qui en dépit de cette période insipide m'a chaleureusement convié à ces réunions de projet au laboratoire LCQB où les échanges empreints d'enthousiasme et de bonne humeur ont été très féconds. Ainsi qu'à Hugues Richard et Diego Javier Zea qui malgré leurs distances ont su m'éclairer par leurs commentaires et conseils.

Contents

1	Contexte	3
2	Enoncé du problème	4
3	Discussion sur l'approche résolutive	5
4	Description du jeu de données	6
5	Description des algorithmes	6
6	Résultats	11
7	Analyse statistique sur les ASRU détectées	13
8	Visualisations des detections sur les ESG	18
9	Analyse de séquences similaires entre les paralogues d'une même famille	23
10	Difficultés rencontrées	27
11	Conclusions	27
	Appendix A Fonction qui calcule les marges	28
	Appendix B ESG	29
	References	30

1. Contexte

Le jeu de données traité a été produit par ThorAxe [1], il s'agit d'un algorithme de détection automatique d'exons orthologues que nous appellerons s-exon (spliced-exon) et qui se retrouvent dans un ensemble d'espèces (chez l'homme, le gorille, la souris, la xénope...). Dans le contexte de l'épissage alternatif, l'identification de relations d'orthologies entre exons est nécessaire pour inférer des scénarios évolutifs qui expliquent la diversité des transcrits observés.

Rappelons brièvement ce qu'est l'épissage alternatif, il s'agit d'un processus biologique existant uniquement chez les eucaryotes. La machinerie de transcription des eucaryotes peut produire plusieurs transcrits d'ARNm à partir d'un seul gène grâce à un mécanisme combinatoire de sélection et d'agencement d'exons constitutants le gène. En résulte différentes versions de la protéine, appelées les isoformes de la protéine, qui peuvent adopter différentes conformations tridimensionnelles, interagir avec différents partenaires, et donc avoir des fonctions biologiques différentes dans la cellule.

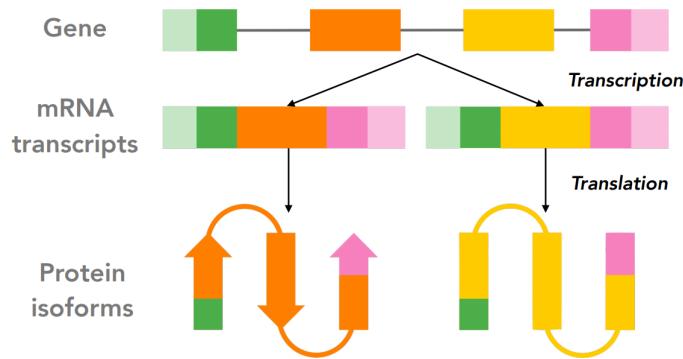


Figure 1. Epissage alternatif d'un gène

Un **Evolutionary Splicing Graph** (ESG) est une notion introduite dans [1], servant à représenter pour un gène commun dans un ensemble d'espèces la variabilité des transcrits générée par l'épissage alternatif. A un gène on associe un graphe orienté, dans un tel graphe chaque noeud est un s-exon, c'est-à-dire un alignement de séquences exoniques appartenant à différentes espèces, et une arête entre deux noeuds représente le fait que les séquences associées à chaque noeud co-occurrent dans un transcrit. Autrement dit un transcrit est un chemin dans ce graphe (et non l'inverse car certains chemins ne correspondent pas à des transcrits). Le transcrit traduit est une protéine isoforme donc l'ensemble des protéines isoformes d'un gène est un sous-ensemble de l'ensemble des chemins dans un ESG. Dans [1] et dans le reste de ce rapport les séquences sur lesquelles nous travaillons sont des séquences d'acides aminés traduites des transcrits. Antérieurement aux ESG circulait la notion de **Splicing Graph** introduite dans [2], où un noeud est un exon pour une seule espèce et les arêtes représentent la jonction entre les exons. Le Splicing Graph représente donc la variabilité des transcrits au sein d'une seule espèce. L'ESG étend ce concept à plusieurs espèces.

En analysant ces graphes, les auteurs de [1] ont identifié plusieurs milliers de gènes parmi les gènes codant pour le protéome humain, où des événements d'épissage alternatif évolument conservés viennent moduler le nombre et la composition de pseudo-repeats.

Nous travaillons sous l'hypothèse que l'apparition de ces repeats est dû à des événements de

duplication d'exons. Plusieurs exemples de protéines, tels que des facteurs de transcriptions qui se lient à l'ADN ou impliquées dans la contraction musculaire et bien d'autres encore suggèrent que de telles modulations influencent le nombre de partenaires qui se lient à la protéine et l'affinité de ces liaisons.

2. Enoncé du problème

Soit $\mathcal{G} = (\mathcal{V}, \mathcal{E}^{ev}, \mathcal{E}^{sim})$ un graphe associé à un gène g (il faut en fait voir là deux graphes : $(\mathcal{V}, \mathcal{E}^{ev})$ qui est un graphe orienté, l'ESG et $(\mathcal{V}, \mathcal{E}^{sim})$ le graphe de similarité qui est un graphe non-orienté). Chaque noeud $v_i \in \mathcal{V}$ représente un s-exon et se définit par un ensemble de séquences alignées $\tilde{s}_i = \{s_1^i, s_2^i, \dots, s_k^i, \dots, s_n^i\}$, autrement dit il s'agit d'un MSA.

Il existe une arête orientée $e_{ij}^{ev} \in \mathcal{E}^{ev}$ du noeud v_i vers le noeud v_j si pour au moins une espèce k il existe un transcrit t tel que $s_k^i s_k^j \in t$. Il existe une arête $e_{ij}^{sim} \in \mathcal{E}^{sim}$ si $p-value(v_i, v_j) \leq 0.001$ tel que calculée par l'alignement HMM entre les s-exons correspondants, c'est-à-dire que la séquence consensus de \tilde{s}_i soit similaire à celle de \tilde{s}_j .

On définit un événement au sens de l'épissage alternatif comme une bulle dans l'ESG $(\mathcal{V}, \mathcal{E}^{ev})$. Plus précisément, un événement b est un triplet $((v_s, v_e), (v_1^c, v_2^c, \dots, v_l^c), (v_1^a, v_2^a, \dots, v_m^a))$ où chaque élément est une liste de noeuds. La liste $(v_1^c, v_2^c, \dots, v_l^c)$ respectivement $(v_1^a, v_2^a, \dots, v_m^a)$ définit un **sous-chemin canonique** respectivement **sous-chemin alternatif**, bornés par les ancre v_s et v_e . On a $l \geq 0$ et $m \geq 0$. Le cas $= 0$ occure quand on a une insertion ou une délétion et alors un des deux chemins ne contient aucun noeud.

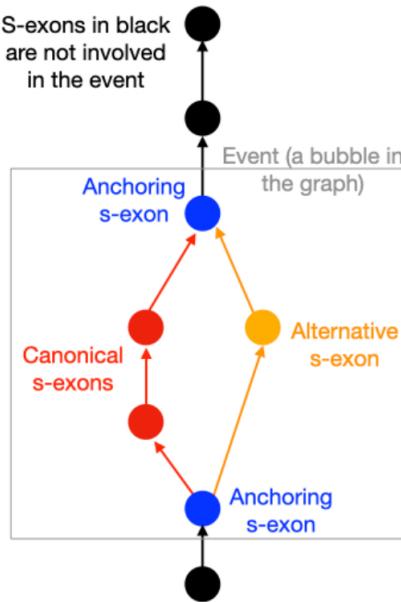


Figure 2. Portion d'un ESG. La paire concernée est de classe MEX. En bleu les ancrès start et end v_s, v_e de l'évènement.

Le transcrit canonique est défini comme une référence au sens où il est bien conservé à travers les espèces. Dans chaque évènement, les ancrès et le sous-chemin canonique sont inclus dans un transcrit canonique.

Chaque paire de s-exons similaires (v_i, v_j) est liée par une arête $e_{ij}^{sim} \in \mathcal{E}^{sim}$. On associe un ensemble d'événements b_1, b_2, \dots, b_p nous informant ainsi sur l'usage alternatif des s-exons correspondants. Si aucun des s-exons de la paire n'est inclus dans le transcript canonique, ou si les deux s-exons co-occurrent toujours dans le même sous-chemin, qu'il soit canonique ou alternatif, alors l'ensemble des événements associé à cette paire est vide. Sinon il est non-vide et chacun de ces événements confère à cette paire de s-exons une relation (aussi appelée classe). Les classes possibles, voir figure 2, sont :

MEX, mutually exclusive, c'est-à-dire que soit l'un des s-exons de la paire est dans le transcript soit c'est l'autre qui y est.

ALT, alternatif, c'est-à-dire non mutually exclusive, donc contient le cas MEX ou possibilité que les deux s-exons co-occurrent dans le même sous-chemin.

REL, l'un des s-exons est dans le sous-chemin canonique ou alternatif tandis que l'autre sert d'ancre à l'évènement (l'ancre apparaît dans les deux sous-chemins).

UNREL, même chose que REL sauf que l'ancre est située en dehors de l'évènement et donc les deux s-exons n'apparaissent jamais dans le même évènement.

A chaque paire est associée une unique classe dépendamment des événements quand il y en a et selon l'ordre de priorité MEX>ALT>REL>UNREL.

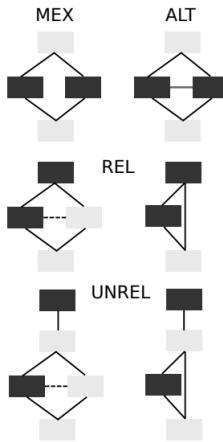


Figure 3. les rectangles noirs sont les s-exons de la paire

Nous voulons déterminer pour un gène donné un ensemble d'unités répétées $\{U_1, U_2, \dots, U_r\}$, que nous appellerons alternatively spliced repetitive units (ASRU). Chaque unité répétée U est un ensemble d'instances où chaque instance peut-être composée d'un ou plusieurs s-exons.

Une ASRU doit être une entité significative dans le contexte de l'épissage alternatif, pour ce faire une propriété désirable d'une ASRU est que les instances soient à peu près de même longueur en terme d'acide aminés. Pour atteindre cette homogénéité au sein de l'unité se pose le problème de l'extension des instances : si possible étendre l'instance d'une paire avec un autre s-exon sans jamais sortir de l'évènement dans laquelle cette paire est impliquée.

3. Discussion sur l'approche résolutive

Afin de trouver les unités répétées pour un gène donné, nous allons dans un premier temps détecter les composantes connexes dans le graphe de similarité. Pour ce faire on part d'un ensemble de paires détectées par ThorAxe. Sachant qu'un s-exon peut apparaître dans plusieurs paires, il va falloir par transitivité regrouper dans un même ensemble les paires qui ont un élément en commun. Nous appelerons les éléments de cet ensemble nouvellement construit des

instances-seeds. Seed car l'ensemble construit représente une forme incomplète de la composante connexe. Dans un second temps nous allons dans la mesure du possible étendre les instances-seeds avec d'autres s-exons sous la contrainte que la jonction instance-seed+extension ne soit jamais brisée par un évènement.

4. Description du jeu de données

Nous travaillerons sur un ensemble de 2190 gènes. C'est effectivement dans cet ensemble de gènes que ThorAxe a détecté la mise à l'oeuvre du phénomène d'usage alternatif évolutionnairement conservé de séquences exoniques similaires (= pseudo-repeats).

L'élément clé du jeu de données est une table (nous l'appellerons "eventsDup") qui résume les propriétés des 82 496 évènements d'épissage alternatif impliquant un ensemble de 41 472 paires de s-exon retenus pour l'analyse de duplications. Chaque ligne est une paire de s-exons de p-value inférieure à un seuil (0.001), il s'agit donc des arêtes du graphe de similarité $(\mathcal{V}, \mathcal{E}^{sim})$.

L'autre partie du jeu de données traitée est un ensemble de dossiers (contenant 2 765 238 fichiers), un dossier par gène contenant toutes les combinaisons possibles d'alignements (au format .hhr et généré par hhalign) entre les séquences des s-exons. Et enfin un ensemble de table (appelée "ases") pour chaque gène qui recense tout les évènements détectés par ThorAxe, à chaque évènement est associé un couple de sous-chemins, sous-chemin canonique et alternatif. Ces évènements sont listés dans un ordre allant du plus conservé/fréquent vers le moins conservé/fréquent.

5. Description des algorithmes

Listons les principales fonctions et scripts construits :

Des fonctions qui permettent de parser des csv, elles prennent en input le nom d'un gène cible et extraient les évènements, les paires correspondantes et les informations relatives à ces objets (taille, coverage ...)

Une fonction permettant de passer des évènements aux paires, qui prend en entrée le nom d'un gène cible et en sortie produit un dictionnaire de la forme suivante

$\{ b_i : [(paire, classe), \dots, \text{sous-chemin canonique et alternatif associé à l'évènement}], \dots \}$, où les b_i sont les évènements, et une fonction permettant de passer des paires aux évènements, sa sortie est de la forme suivante $\{ paire : [b_1, \dots, b_p], \dots \}$

Une fonction qui étant donné en entrée une liste de listes vient par transitivité fusionner les listes possédant au moins un élément en commun. Nous appliquerons cette fonction sur la liste des paires pour un gène donné, en résultera les composantes connexes du graphe de similarité. Ces composantes peuvent être incomplètes, il faudra vérifier s'il est possible d'étendre ses éléments.

Une fonction qui étant donné en entrée une paire, calcule d'après les informations sur l'alignement de la paire les marges pour étendre. Il y a quatres entiers naturels à calculer, marge pour A ou/et B en N-terminal ou/et C-terminal. Renvoie une liste $[marge_B^{Nter}, marge_B^{Cter}, marge_A^{Nter}, marge_A^{Cter}]$. Si $marge_B$ est positive alors on c'est A qu'on étend sinon c'est B.

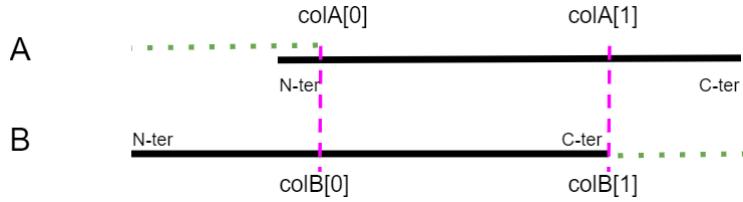


Figure 4. Cas général où pour une paire (A,B) alignée entre les pointillés roses on a la possibilité d'étendre A en N-ter et B en C-ter, les marges sont représentées en vert. Dans ce cas de figure sera détecté une $marge_B^{Nter}$ et $marge_A^{Cter}$ non nulle. Les barres noires représentent les séquences consensus des s-exons (=MSA).

Plus précisément, cette fonction va récupérer des informations essentielles dans l'alignement de la paire (A,B) soit la position dans A et B de la première et dernière colonne de l'alignement, c'est-à-dire les positions des pointillés roses dans A et dans B, ainsi que le taille des séquences consensus des s-exons A et B. Elle calcule ensuite le coverage pour chacune des séquences :

$$coverage_A = \frac{|colA[1]-colA[0]+1| \times 100}{taille_A}$$

$$coverage_B = \frac{|colB[1]-colB[0]+1| \times 100}{taille_B}$$

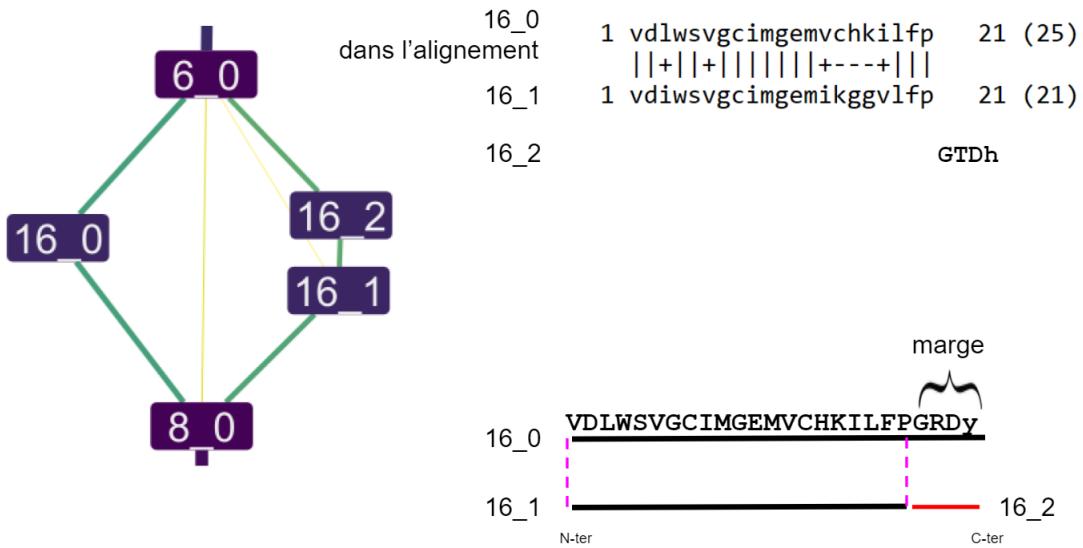


Figure 5. Dans le splice graph de MAPK8, 16_0 et 16_1 ont une p-value ≤ 0.001 . On a $marge_{16_0}^{Cter} = 4$ et 16_2 étant de taille inférieur à 5 on peut étendre 16_1 en C-ter avec 16_2.

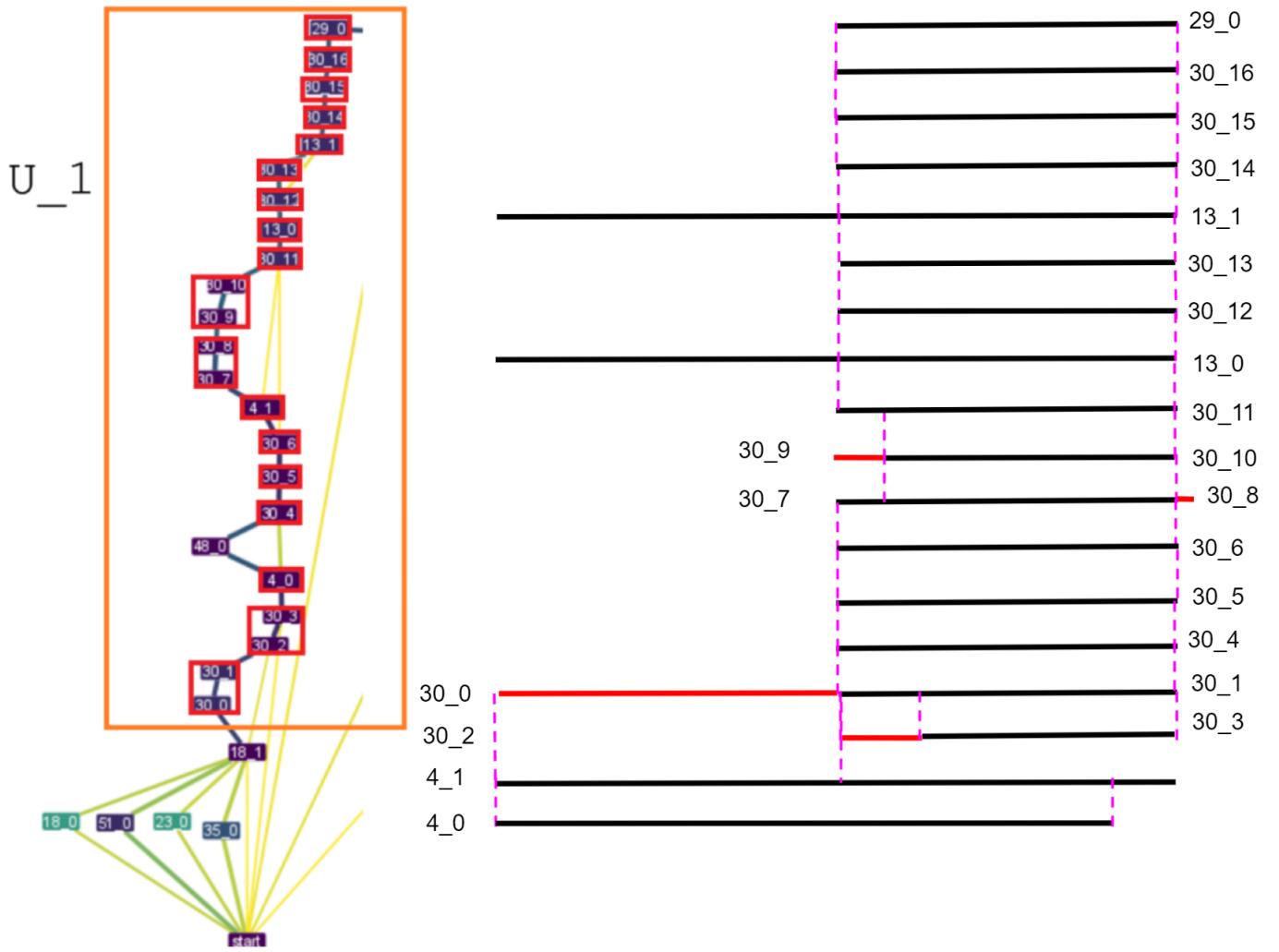


Figure 6. U_1 , une unité répétée dans ANK2. Les alignements ne suivent pas forcément l'ordre du graphe, nous avons cherché à mettre en évidence comment s'alignaient les instances quitte à perdre cet ordre. L'alignement entre 30_7 et la première moitié de 4_1 a servi à faire l'extension 30_7+30_8.

Une fonction chargée d'étendre les instances d'une unité répétée pour un gène cible.

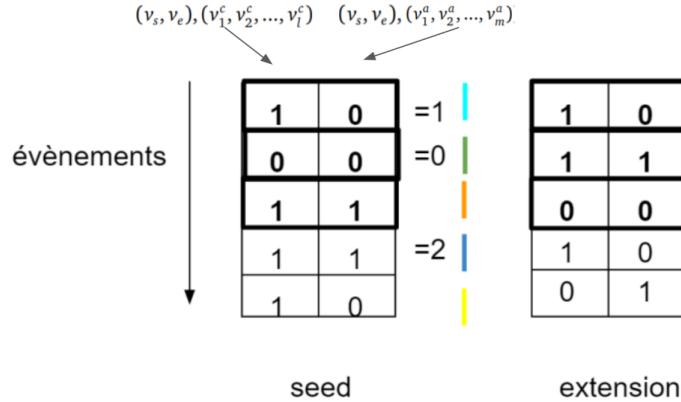


Figure 7. Codage binaire des différentes situations du problème des extensions. En gras les situations qui sont acceptées. Chaque ligne est un évènement. Chaque colonne représente la présence ou non de la seed/extension dans le sous-chemin canonique/alternatif de l'évènement avec les ancre associées.

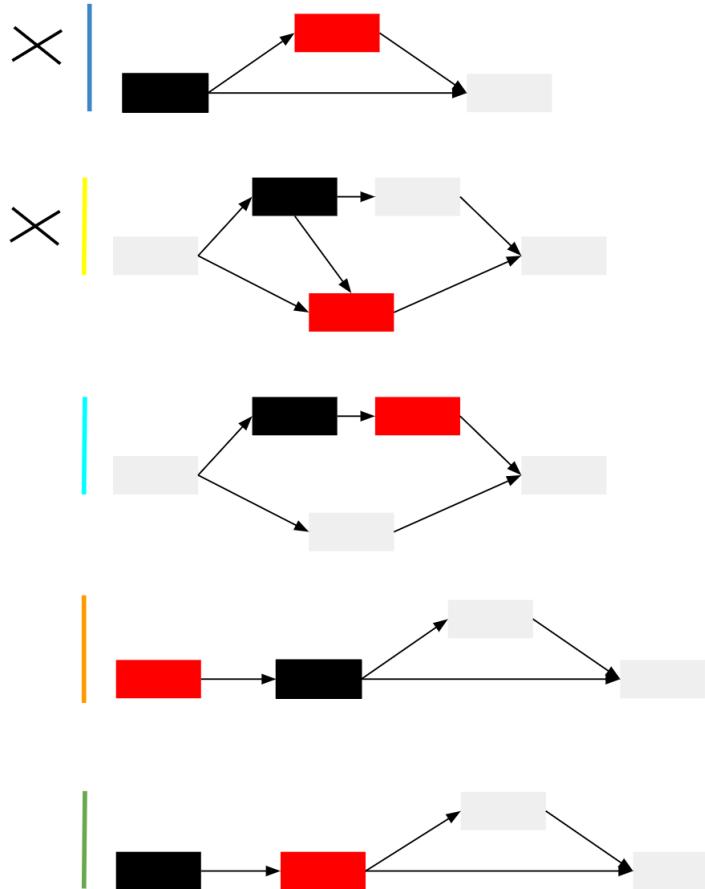


Figure 8. L'instance-seed est en noir, l'extension en rouge. En bleu : la seed est une ancre mais on n'étend pas l'ancre depuis l'intérieur de l'évènement car l'évènement coupe cette jonction. En jaune : la seed est sur le sous-chemin canonique ou alternatif et l'extension est sur complémentaire.

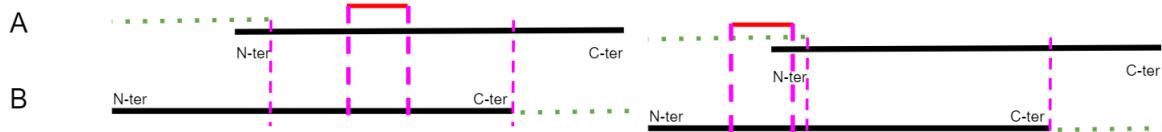


Figure 9. L’alignement de l’extension en rouge avec B s’overlap avec l’alignement entre A et B donc on rejette l’extension. Les alignements ont lieu entre les pointillés rose.

Figure 10. L’alignement de l’extension en rouge avec B ne s’overlap pas avec l’alignement entre A et B donc l’extension est un candidat potentiel.

Dans cette dernière fonction, on vérifie que le pourcentage d’identité fournit par halign dans l’alignement de la paire soit inférieur à un certain seuil (1%) afin d’éliminer les faux-positifs au niveau des paires trouvées par ThorAxe. Un premier run a été fait sans cette condition sur les 2190 gènes, 3024 unités répétées ont été déterminées, lors du second run avec cette condition, 2410 unités répétées ont été déterminées, c’est donc 614 unités répétées qui ont été détectées comme faux-positifs, soit 20.3%. Pour comprendre la nature de ces artefacts il faut aller voir les MSA des s-exons de la paire. Nous détaillerons les cas rencontrés dans les résultats.

Par ailleurs aurait-on pu imaginer récupérer plus d’ASRU en choisissant une p-value moins significative tout en gardant le critère sur le pourcentage d’identité ?

Faisons remarquer qu’il y a eu une ambiguïté sur la définition de ce critère, faut-il retenir l’identité fournit par halign qui en fait choisit le (meilleur) alignement entre séquences chez une espèce , ou l’identité entre les consensus ? L’identité halign malgré sa spécificité (séquences dans une espèce) a été retenue car l’identité entre les consensus ne permet pas de détecter ces artefacts.

Cette dernière fonction qui met à jour les paires en faisant les extensions (par exemple paire (A,B) devient (A+C,B)) est censée ré-itérer sur les paires mises à jour. Cependant ceci n’a pas pu être implémenté, la difficulté étant sur l’accès aux données d’alignement d’une paire mise à jour. Par contre la version courante de l’algorithme permet de créer des instances contenant trois s-exons, du type A+C+D. En effet la première itération va produire des instances overlapantes du type A+C et C+D, or la fusion par transitivité de ces deux instances est légitime car aucun évènement ne vient briser la jonction A+C ni C+D, de plus A,C et D sont forcément sur le même sous-chemin.

Un script qui regroupe les 2190 gènes par famille de paralogues, selon un jeu de données récupéré avec Ensembl-Biomart contenant initialement 19 946 gènes humains. Ainsi, 342 familles ont été recensées.

6. Résultats

A partir des algorithmes et jeu de données décrit précédemment, deux tables ont été produites, l'une contenant les unités répétées pour chaque gène parmi les 2190 gènes, ainsi 2410 unités répétées ont été déterminées, l'autre contenant toutes les instances.

```
gene,uniteRepetee,#instances,max,min,moy,median,ecartType,evenements
TPM1,"{'7_0', '13_0'}",2,26,26,26.0,26.0,0.0,[1]
TPM1,"{'1_2', '6_0'}",2,44,22,33.0,33.0,11.0,[2]
TPM1,"{'6_1', '2_2', '2_0'}",3,45,42,43.0,42.0,1.4142135623730951,[2 4]
TPM2,"{'2_1', '2_0'}",2,45,42,43.5,43.5,1.5,[3]
TPM2,"{'5_2', '5_0'}",2,26,26,26.0,26.0,0.0,[1]
TPM3,"{'10_1', '10_0'}",2,26,26,26.0,26.0,0.0,[2]
TPM3,"{'2_1', '2_0'}",2,45,42,43.5,43.5,1.5,[1]
TPM3,"{'4_0', '4_1'}",2,44,39,41.5,41.5,2.5,[1]
TPM4,"{'2_0', '1_0/1_1'}",2,44,38,41.0,41.0,3.0,[1]
```

Figure 11. extrait de la table des unités répétées, on y voit une instance mise à jour 1_0/1_1

La colonne des événements dans la table des unités répétées a été construite comme suit : pour chaque paire ayant servi à construire l'unité répétée on récupère l'ensemble des événements dans lesquels cette paire est impliquée. Cela nous renseigne sur la complexité de l'unité. Par exemple quand on a une unité avec beaucoup d'instances mais reliées par un seul événement, dans le transcript sur un sous-chemin toutes les instances seront présentes tandis que dans l'autre elles seront tronquées.

```
instance,taille,Nombre,UniteRepetee,gene
13_0,26,1,"{'13_0', '7_0'}",TPM1
7_0,26,1,"{'13_0', '7_0'}",TPM1
1_2,22,1,"{'1_2', '6_0'}",TPM1
6_0,44,1,"{'1_2', '6_0'}",TPM1
2_0,42,1,"{'2_0', '2_2', '6_1'}",TPM1
2_2,42,1,"{'2_0', '2_2', '6_1'}",TPM1
6_1,45,1,"{'2_0', '2_2', '6_1'}",TPM1
2_1,45,1,"{'2_1', '2_0'}",TPM2
2_0,42,1,"{'2_1', '2_0'}",TPM2
5_0,26,1,"{'5_0', '5_2'}",TPM2
5_2,26,1,"{'5_0', '5_2'}",TPM2
10_0,26,1,"{'10_0', '10_1'}",TPM3
10_1,26,1,"{'10_0', '10_1'}",TPM3
2_1,45,1,"{'2_1', '2_0'}",TPM3
2_0,42,1,"{'2_1', '2_0'}",TPM3
4_0,39,1,"{'4_0', '4_1'}",TPM3
4_1,44,1,"{'4_0', '4_1'}",TPM3
1_0/1_1,38,2,"{'1_0/1_1', '2_0'}",TPM4
2_0,44,1,"{'1_0/1_1', '2_0'}",TPM4
```

Figure 12. extrait de la table des instances

Listons des cas d'erreurs de ThorAxe que nous arrivons à détecter, ce sont des cas mal définis tel que le split d'un repeat en plusieurs s-exons ou des séquences mal assignées dans un s-exon etc.

```
antoine@ubuntu:/data/h_sapiens_species_set/thoraxe_0.6.3/CIC/thoraxe/msa$ more msa_s_exon_9_0.fasta
>ENSG00000079432
-REKDHIRPMNAFMIFSKRHALVLHQRHPNQDNRTVSKILGEWWYALGPKEKQKYHDLAFQ-----
>ENSGG0000000192
-REKDHIRPMNAFMIFSKRHALVLHQRHPNQDNRTVSKILGEWWYALGPKEKQKYHDLAFQ-----
>ENSMUJG00000012174
-REKDHIRPMNAFMIFSKRHALVLHQRHPNQDNRTVSKILGEWWYALGPKEKQKYHDLAFQ-----
>ENSMODG000000028836
-REKDHIRPMNAFMIFSKRHALVLHQRHPNQDNRTVSKILGEWWYALGPKEKQKYHDLAFQ-----
>ENSRNOG00000056118
-REKDHIRPMNAFMIFSKRHALVLHQRHPNQDNRTVSKILGEWWYALGPKEKQKYHDLAFQ-----
>ENSMUSG00000005442
-REKDHIRPMNAFMIFSKRHALVLHQRHPNQDNRTVSKILGEWWYALGPKEKQKYHDLAFQ-----
>ENSBTAG00000019785
-REKDHIRPMNAFMIFSKRHALVLHQRHPNQDNRTVSKILGEWWYALGPKEKQKYHDLAFQ-----
>ENSSCG00000003030
-REKDHIRPMNAFMIFSKRHALVLHQRHPNQDNRTVSKILGEWWYALGPKEKQKYHDLAFQ-----
>ENSXETG00000001602
OREKDHIRPMNAFMIFSKRHALVLHQRHPNQDNRTVSKILGEWWYALGPKEKQKYHDLAFQ-----
>WBGene00001560
--NEPHVRPMNAFMIFSKRHRPLVHQYQPNKDNRTVSKILGEWWYSLAADQKAEHYKLATQVKEAHFKAHPDWKSTKEKKIKSESINTTPVALTPLKNK
antoine@ubuntu:/data/h_sapiens_species_set/thoraxe_0.6.3/CIC/thoraxe/msa$ more msa_s_exon_24_0.fasta
>ENSG00000079432
VKEAHFKAHPDWKWCNKDRKSSSEAKPTSLGLAGGHKETRERSMSETGTAAAPG-
>ENSGG00000000192
VKEAHFKAHPDWKWCNKDRKSSSEAKPTSLGLAGGHKETRERSMSETGTAAAPG-
>ENSMUJG00000012174
VKEAHFKAHPDWKWCNKDRKSSSEAKPTSLGLAGGHKETRERSMSETGTAAAPG-
>ENSMODG00000028836
VKEAHFKAHPDWKWCNKDRKSSSDTKPVALGLTGGPKEMRERSMSETGTAAAPG-
>ENSRNOG00000056118
VKEAHFKAHPDWKWCNKDRKSSSEAKPASLGLAGGHKETRERSMSETGTAAAPG-
>ENSMUSG00000005442
VKEAHFKAHPDWKWCNKDRKSSSEAKPASLGLAGGHKETRERSMSETGTAAAPG-
>ENSBTAG00000019785
VKEAHFKAHPDWKWCNKDRKSSSEAKPTSLGLAGGHKEPRERSMSETGTAAAPGV
>ENSSCG00000003030
VKEAHFKAHPDWKWCNKDRKSSSEAKPTSLGLAGGHKETRERSMSETGTAAAPGV
>ENSXETG00000001602
VKEAHFKAHPDWKWCNKDRKSSSDVKVQMPGPWGVPEMRERSMSETGTMAAG-
```

Figure 13. Dans le msa du s-exon 9_0, pour la majorité des espèces les séquences possèdent une délétion, la séquence détectée comme étant un pseudo-repeat existe dans une espèce qui n'apparaît pas dans le msa de l'autre s-exon de la paire. Il y a donc une mauvaise assignation de cette sous-séquence dans 9_0, elle devrait être dans 24_0. L'événement détecté est donc un faux-positif.

```
antoine@ubuntu:/data/h_sapiens_species_set/thoraxe_0.6.3/CIRBP/thoraxe/msa$ more msa_s_exon_11_0.fasta
>ENSG00000099622
PLRPSVLCAPQ
antoine@ubuntu:/data/h_sapiens_species_set/thoraxe_0.6.3/CIRBP/thoraxe/msa$ more msa_s_exon_1_3.fasta
>ENSG00000099622
GGDGRYGG-NRFESRGYY--GGSRDYY-S-
>ENSGG00000002338
GGDGRYGG-NRFESRGYY--GGSRDYY-S-
>ENSMODG00000005290
-GGDGRYGG-SRFESRGYY--GSSRDYY-S
>ENSRNOG00000015999
-GGDGRYGG-GRFESRGYY--GGSRDYYA-S
>ENSMUSG00000045193
GGDGRYGG-GRFESRGYY--GGSRDYY-A-
>ENSBTAG00000007480
GGDGRYGG-SRFESRGYY--GGSRDYY-S-
>ENSOANG00000012230
GGDGRYGG-SRFESRGYY--GGSRDYY-S
>ENSXETG00000006372
GGDGRYGGSSRFENRGYYQSSGSRDYYG-R
```

Figure 14. Ici deux situations co-existent, l'une est que l'identité est nulle, l'autre que la séquence est présente chez une seule espèce donc non-conservée.

7. Analyse statistique sur les ASRU détectées

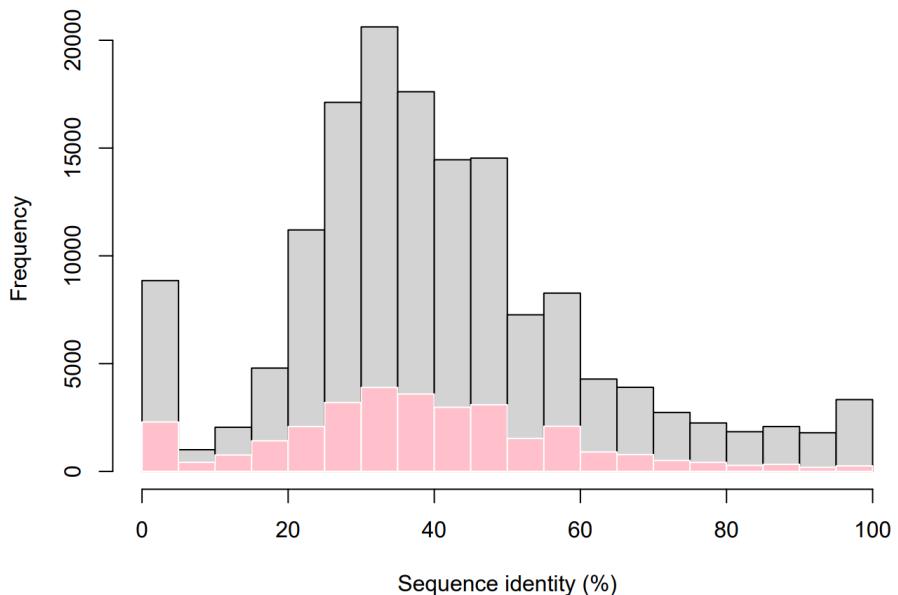


Figure 15. Identité de séquence pour des paires de s-exons similaires. Pour chaque paire a été compté le nombre de positions alignées où la séquence consensus pour les deux s-exons présente le même résidu. Les positions variables ou hautement gappées n'ont pas été considérées (symbole “~” dans l'output hhalign). La distribution grise représente les 150 020 paires de s-exon dont la p-value est inférieure à 0.001. La distribution rose représente les 31 031 paires retenues. A noter que notre algorithme filtre les paires avec un pourcentage d'identité inférieur à 1%, ce qui fait réduire la première barre verticale, celle dans l'intervalle 0-5%

La figure 15 nous permet d'estimer l'homogénéité des unités répétées, on se demande si les instances constitutants l'unité répétée sont de mêmes tailles ? On fera remarquer qu'il y a moins de 100 unités répétées avec un écart-type sur les longueurs supérieur à 130. Le cas extrême est une unité répétée dans le gène *TEX15* avec un écart-type valant 1059.5, contenant deux instances non-étendues l'une de longueur 2340, l'autre de longueur 221, ces deux s-exons s'alignent sur l'extrémité C-terminale de la plus longue séquence.

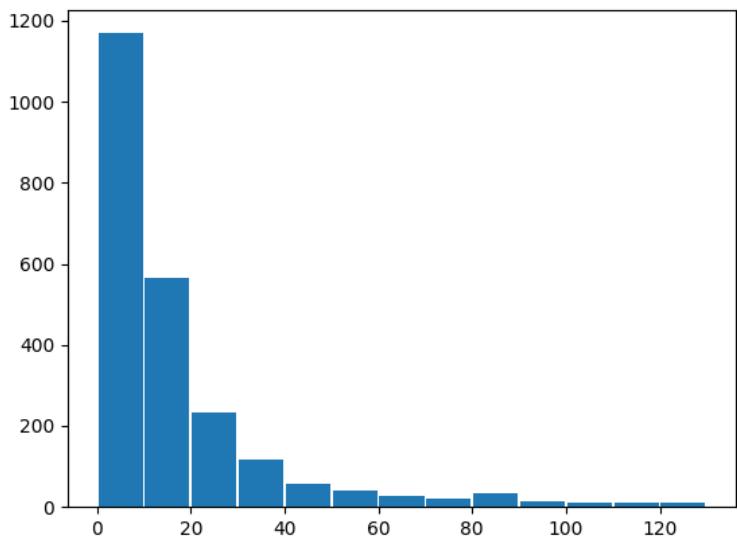


Figure 16. Histogramme des écarts-types sur les longueurs des unités répétées

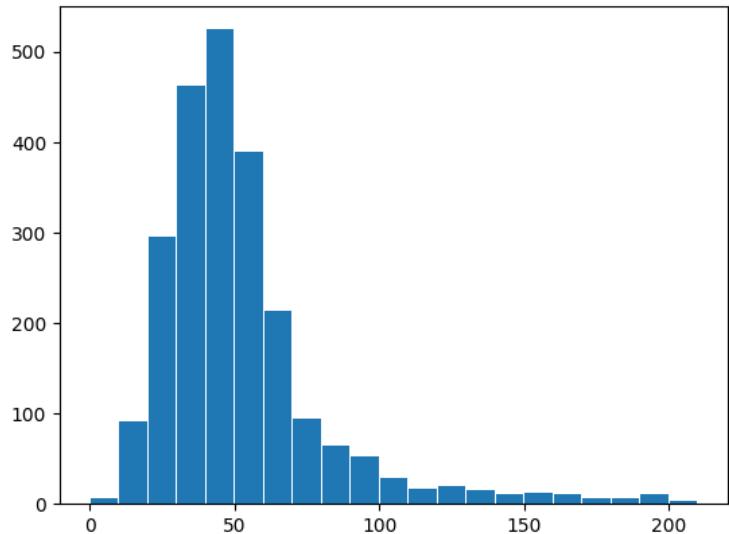


Figure 17. Histogramme des médianes sur les unités répétées

La médiane est représentative de la taille des instances d'une unité répétée. Regardons par exemple le cas de *NEB*, la nébuline, possédant aussi l'unité répétée de plus grande taille, 179 instances. Avec un max à 140, un min à 8 et un écart-type de 25 on s'attendrait à avoir une unité répétée hétérogène, or en regardant la taille des instances, plus de la moitié d'entre elles ont une taille autour de 35 et il s'avère que la médiane est égale à 35.

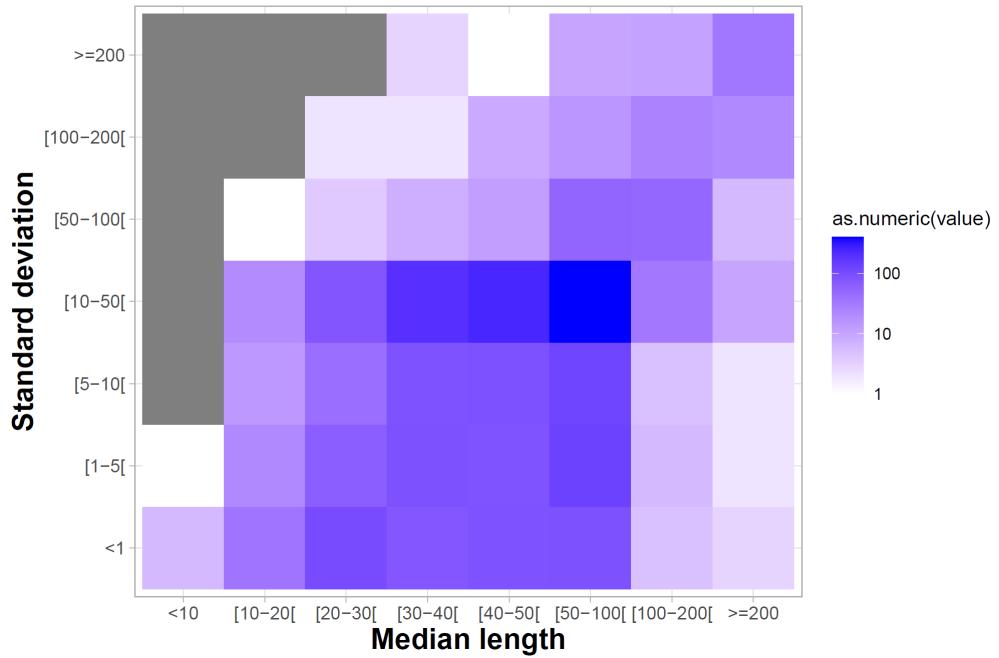


Figure 18. Heatmap de l'écart-type sur les longueurs contre la taille médiane des instances d'une ASRU, qui reflète une variabilité sur la longueur des instances, laissant à penser que la structure des repeats n'est pas préservée par l'épissage.

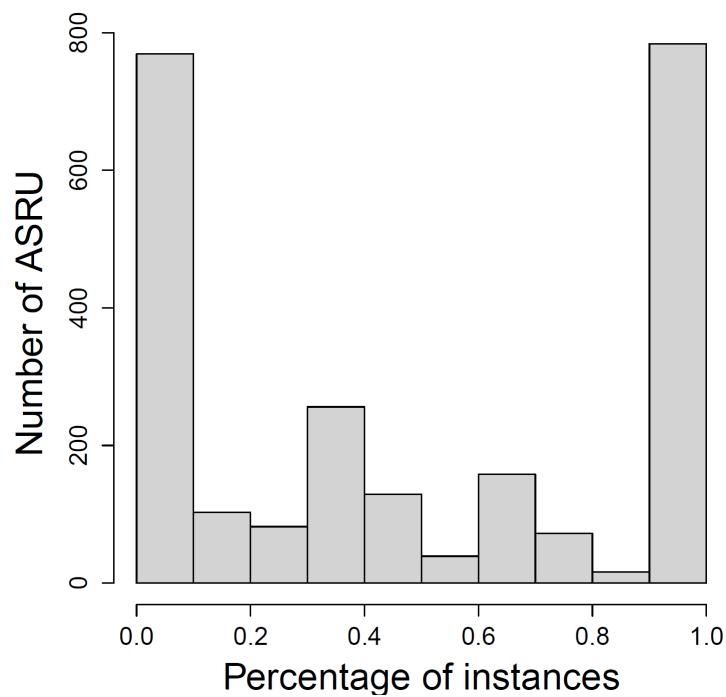


Figure 19. Taux d'instances d'une ASRU dont la distance à la médiane est au plus 5 acide aminés.

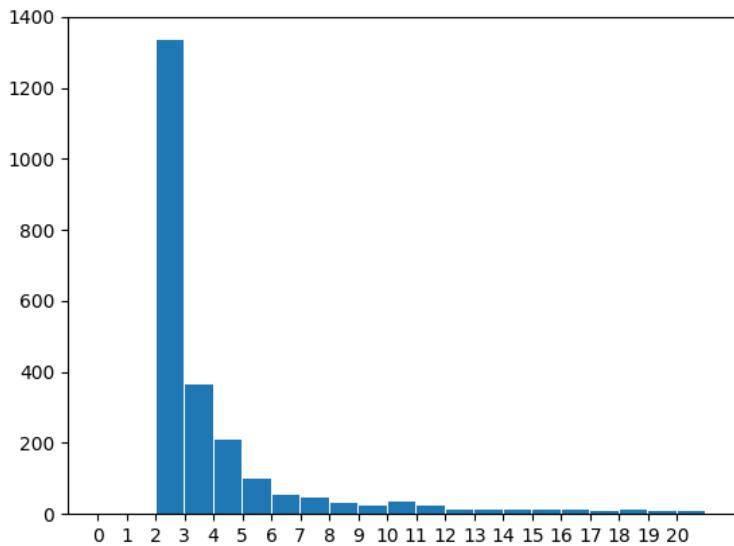


Figure 20. Histogramme des tailles des unités répétées

On remarque qu'il n'y a aucune unité répétée avec une seule instance, ce qui montre que l'algorithme n'a pas généré d'erreurs de ce type. La plupart des unités répétées ont deux instances.

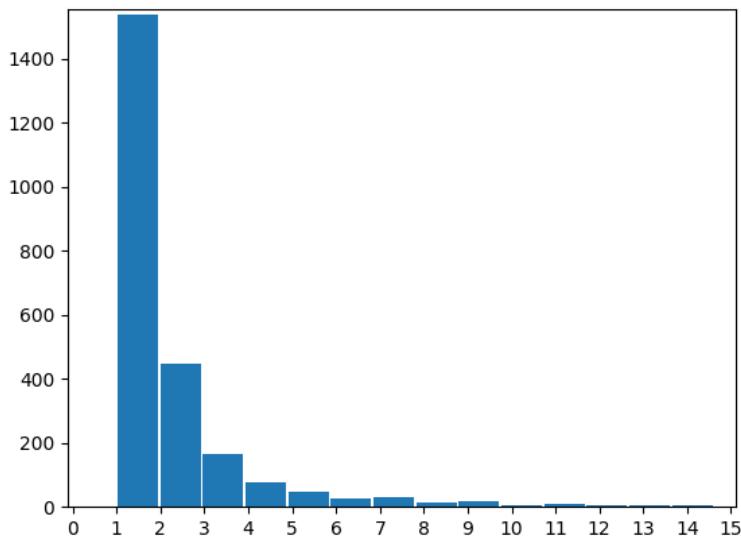


Figure 21. Histogramme du nombre d'évènements dans lesquels sont impliqués les instances d'une unité répétée. On notera qu'il y a 10 unités répétées avec un nombre d'évènements supérieur ou égale à 16, la NEB étant le cas extrême avec un nombre d'évènements égale à 69.

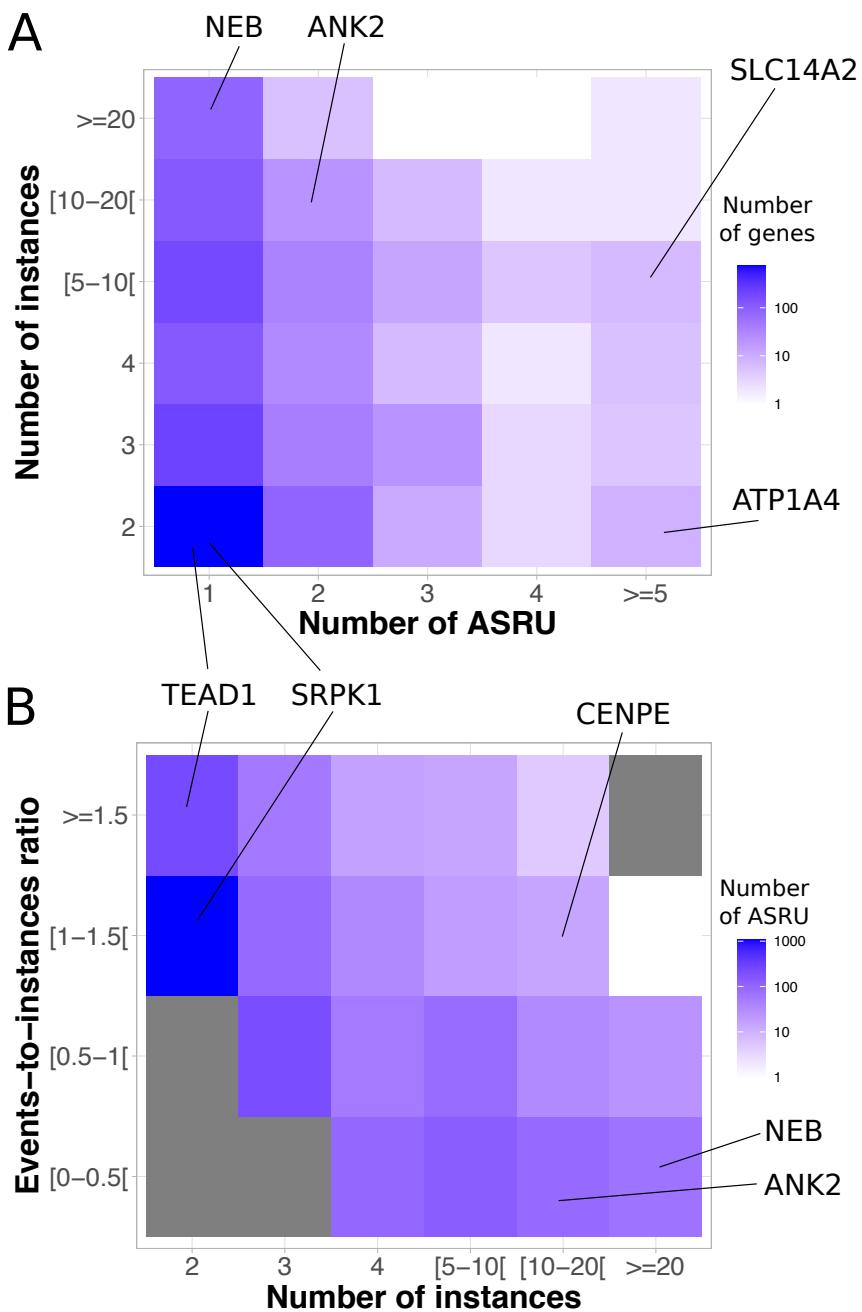


Figure 22. A. Heatmap du nombre d'instances contre le nombre d'ASRU par gène. Le nombre d'instances d'une ASRU d'un gène est maximal. B. Heatmap du ratio évènements-instances contre le nombre d'instances d'une ASRU. Ce ratio est calculé en divisant le nombre d'évènements impliquant une ASRU par le nombre d'instances de cette ASRU moins 1. (Par exemple, 2 instances liées par 1 évènement résulte en un ratio de 1). Les cellules grises correspondent à des valeurs nulles. En A et B l'échelle de couleur est logarithmique.

Quelques gènes ont été mis en évidence sur la figure 21 :

- NEB: 1 seule ASRU mais beaucoup beaucoup d'instances (cas extrême)
- ANK2: beaucoup d'instances, liées par quelques événements.
- SRPK1: une ASRU avec 2 instances liées par un événement.
- TEAD1: plus d'évenements que d'instances, en effet une paire peut-être impliquée dans plusieurs événements.
- CENPE: beaucoup d'instances liées par autant d'événements.
- ATP1A4 : beaucoup d'ASRU (22) avec chacune 2 instances - laisse à penser qu'il s'agit d'un "domaine" dupliqué. (A remarquer qu'il est n'est pas annoté dans Pfam comme domaine dupliqué, cette détection laisse-t-elle présager que l'on détecte des domaines dupliqués ?)
- SLC14A2 : beaucoup d'ASRU (7) avec plus ou moins d'instances (jusqu'à 5) - même conclusion que celui d'avant et on a bien le domaine répété dans Pfam.

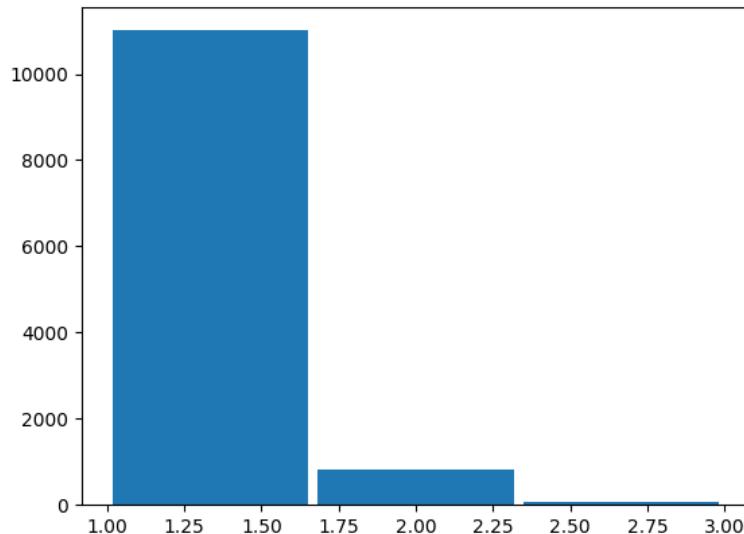


Figure 23. Histogramme du nombre de s-exons par instance. Sur 11928 instances, 832 ont été étendues une fois, soit 7% du nombre total d'instances et 84 ont été étendues deux fois, c'est-à-dire que l'instance contient trois s-exons, soit 0.7%. 572 ASRU contiennent un s-exon étendu, soit 23.7% du nombre total d'ASRU. Justifiant ainsi le travail de programmation effectué pour étendre les instances.

Pourrions-nous nous attendre à avoir plus d'extensions si l'on enlevait les événements les moins conservés ?

8. Visualisations des détections sur les ESG

Nous choisissons ANK2 et NEB qui sont des cas connus dans la littérature de structures possédant de nombreux repeats [3], [4].

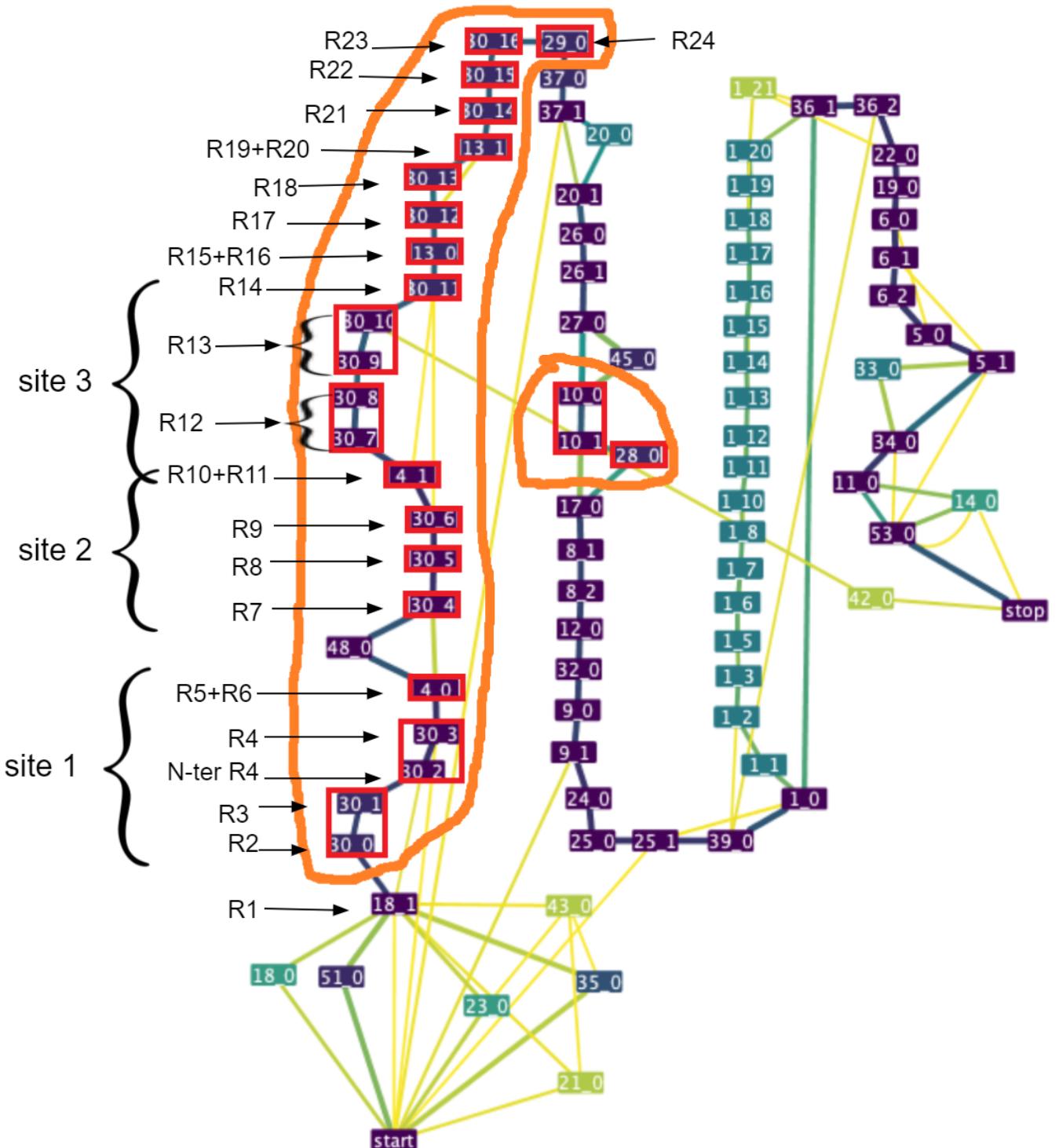


Figure 24. E-Splicing graph de ANK2 avec ses deux unités répétées en orange et ses instances/pseudo-repeats. Nous avons annoté nos instances en faisant un mapping avec les repeats connus dans [3]. La liaison du segment C-terminal auto-inhibitoire de ANK2 avec ses repeats se divise en 3 sites.

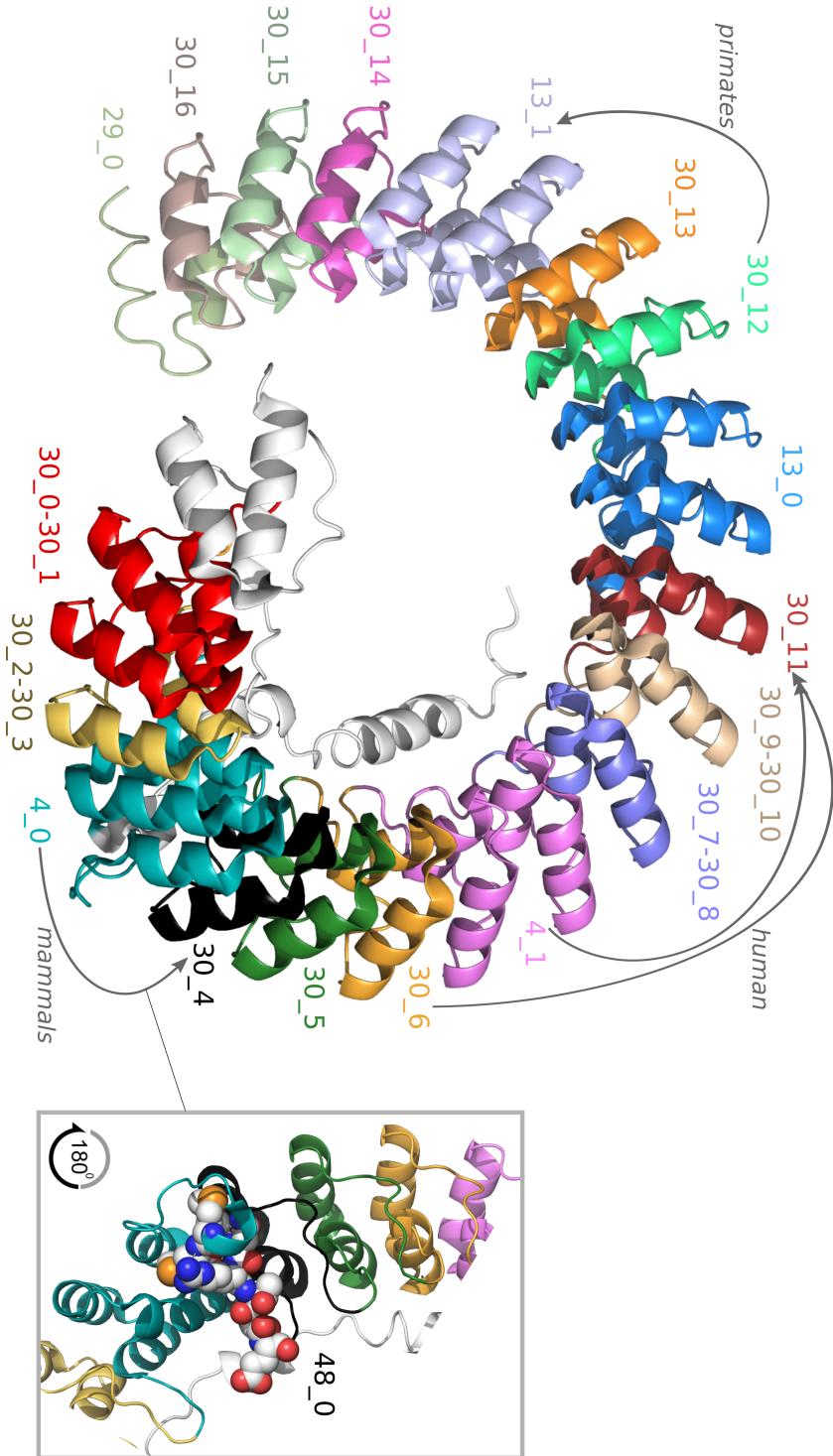


Figure 25. Affichage cartoon de la structure de l’Ankyrin-2 (ANK2, PDB id: 4lrv). Les régions montrées contiennent les 18 instances d’une des ASRU que nous avons détecté dans cette protéine. Les instances sont en couleurs et indexées avec les noms des s-exons correspondants. Les évènements (délétions) détectés à travers les différentes espèces sont indiquées par des flèches (2 évènements ne sont présents que chez l’homme). L’une d’elle est une délétion du s-exon 48_0, montré par des sphères dans la sous-image. Ce s-exon ne fait pas partie de l’ASRU comme on peut le voir sur l’ESG ci-dessus. Le seul morceau qui n’est pas dans les instances forme une espèce de boucle plus étendue que les autres, et c’est exactement là que la délétion conservée est détectée.

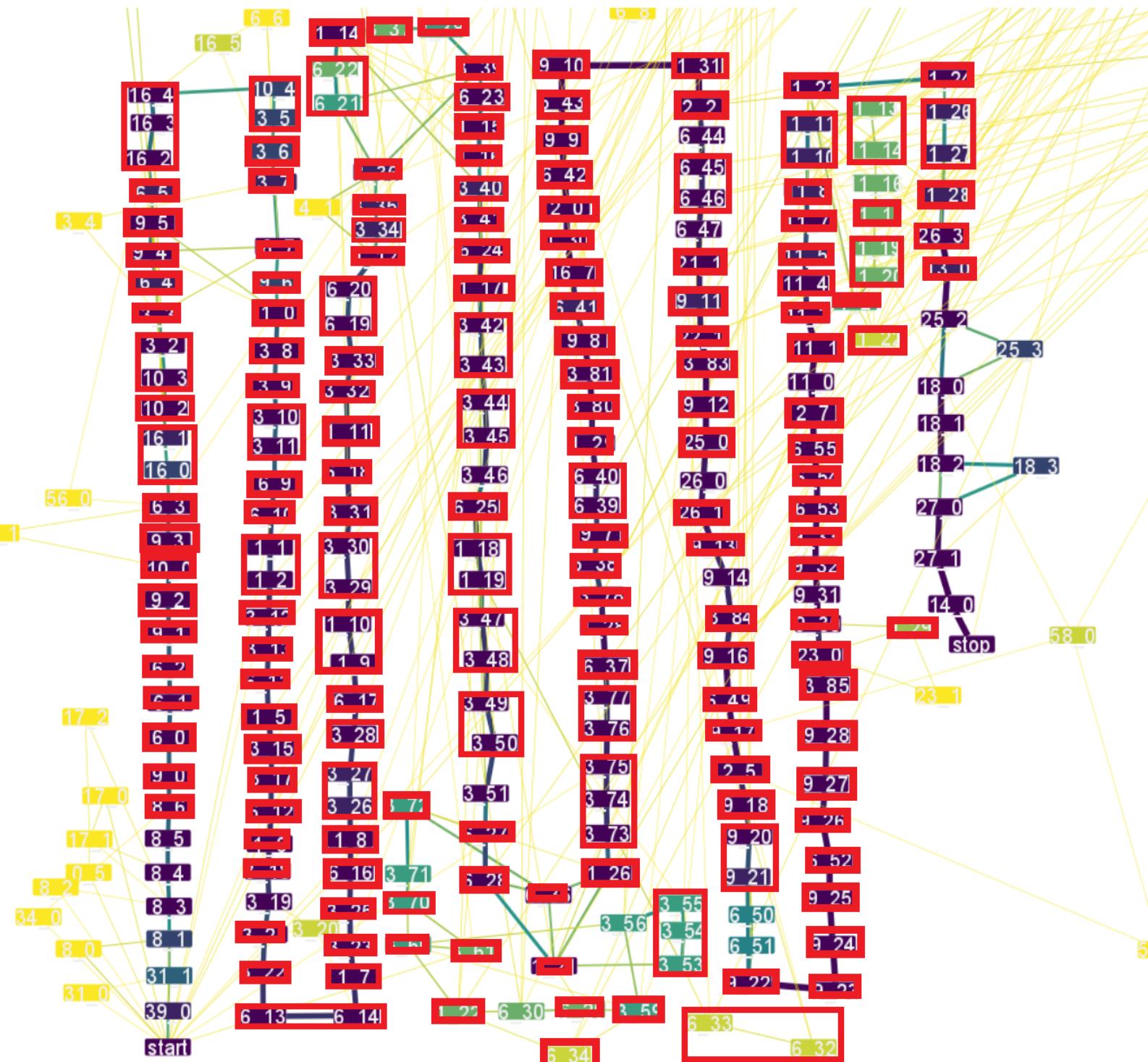


Figure 26. E-Slicing graph de NEB avec les instances/pseudo-repeats de son unité répétée. [4]

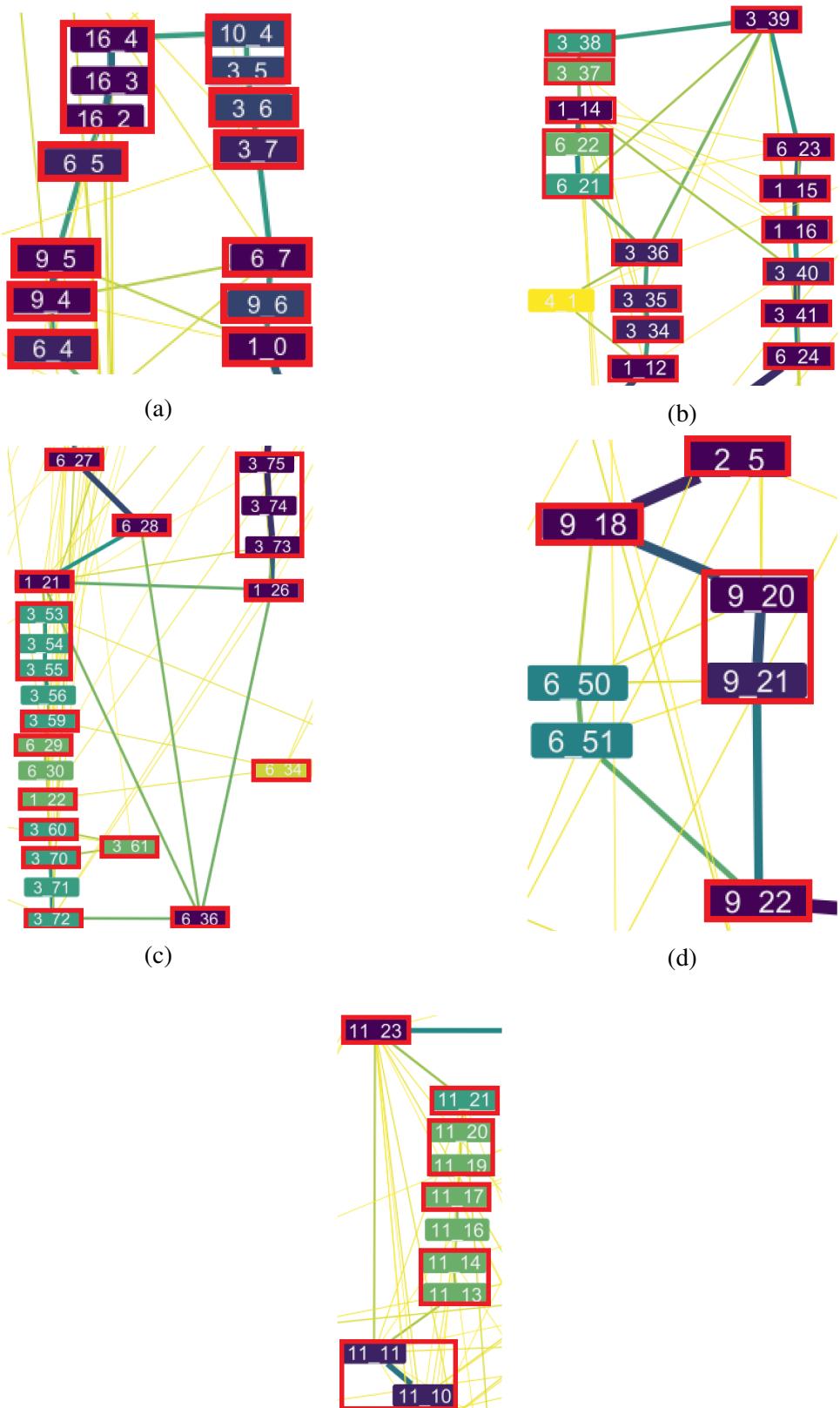


Figure 28. Zoom sur les évènements les plus conservés dans la NEB

9. Analyse de séquences similaires entre les paralogues d'une même famille

L'analyse de l'usage alternatif de séquences similaires entre les paralogues d'une même famille révèle comment surgit la diversification des fonctions de la protéine.

Nous avons cherché à automatiser cette analyse, cependant cela n'a pas pu être effectué dans le temps imparti. Cette étude sera menée post-soutenance avec l'objectif de la faire tourner sur les 342 familles de paralogues recensées. Détaillons néanmoins la démarche que nous avons suivis pour la mettre en oeuvre à la main. En regroupant ensemble les ASRU des paralogues d'une même famille nous cherchons des séquences similaires entre les gènes. Pour ce faire on priorise les séquences inter-gènes de même taille et proches de la taille médiane.

FLNA	{'18_7', '3_31', '10_0', '3_9', '8_4', '12_1/3...', ...}	34	200	21	60.441176	52.0	36.165625	[1 13 15 16 22 24 25 26]
FLNB	{'8_1', '20_0', '9_3', '5_1', '20_2', '4_0', ...}	23	200	28	62.913043	58.0	32.673501	[6]
FLNC	{'11_1', '13_5', '8_1', '8_0', '5_4', '12_2', ...}	33	200	13	57.575758	51.0	31.070819	[6 7 8 9 12]

Figure 29. FLNA/B/C sont les paralogues d'une même famille. La difficulté de cette analyse réside dans le fait que les ASRU ne sont pas forcément homogènes, il y a donc une ambiguïté sur le choix des instances pour chaque gène. En réalité la famille de paralogues contenant FLNA/B/C contient d'autres gènes mais aucunes des instances de ces gènes ne possèdent de similarités avec les séquences que nous avons trouvés ci-dessous, pourtant certaines instances avaient les bonnes tailles.

Pour relever des signatures fonctionnelles dans ces séquences nous allons produire des alignements. Ci-dessous se trouvent des alignements de séquences consensus de taille homogènes et appartenant à des ASRU de gènes paralogues. Nous avons mené cette étude sur trois familles de gènes, TEAD (A), SRPK (B) et FLN (C). Chaque lettre est l'acide aminé conservé dans toutes les séquences du MSA correspondant (les substitutions sont permises). Le jeu de couleur est celui de Clustal X (Thompson et al., 1997). Les groupes "a" et "b" ont été définis selon les similarités entre séquences. Les symboles en bas du MSA représentent les positions hautement conservées et les specificity-determining sites (SDS) [5] à travers la famille de gènes. Disque: position totalement conservée. Carré: position conservée uniquement dans un groupe de s-exon (type II SDS). Triangle vers le haut: Position conservé uniquement dans le groupe *a* (type I SDS). Triangle vers le bas: Position conservé uniquement dans le groupe *b* (type I SDS). Au cours de l'évolution, les SDS jouent un rôle clé dans la diversification des fonctions de la protéine [5].

Par suite nous avons mappé les s-exons correspondants aux structures 3D des protéines.

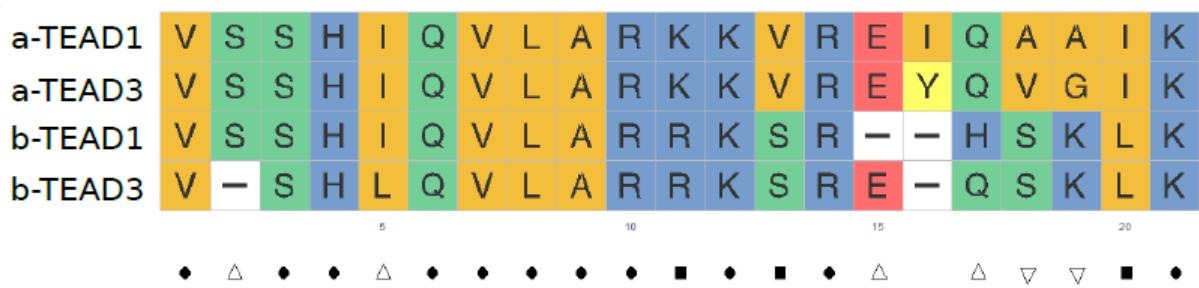
A

Figure 30. La séquence consensus de 9_3 correspond à b-TEAD1. 9 SDS identifiés, 6 SDS de type I et 3 de type II. La paire de TEAD1 est en ALT, celle de TEAD3 en MEX.

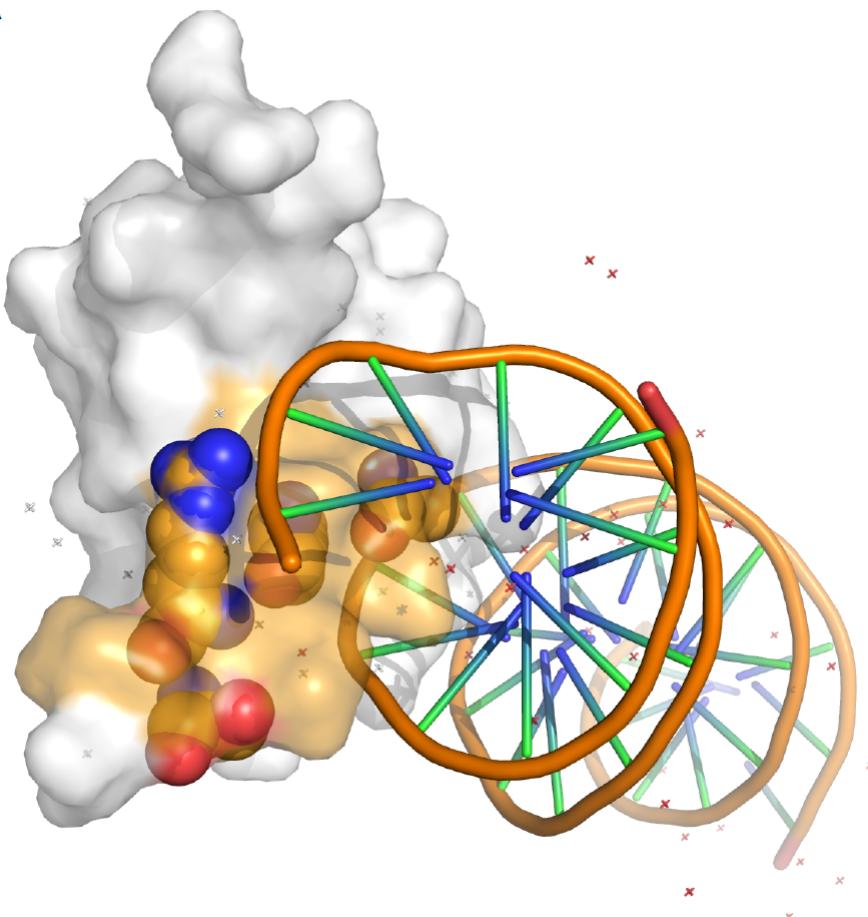
A

Figure 31. Structure du transcriptional enhancer factor TEF-1 (TEAD1, surface transparente) lié à de l'ADN (cartoon) (PDB id: 5nnx). Le s-exon 9_3 est en orange, et les résidus identifiés comme SDS (carrés et triangles dans le MSA) sont représentés par des sphères.

B

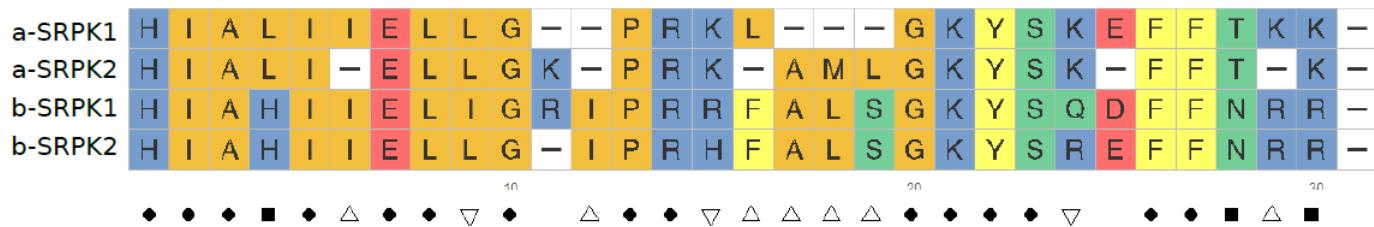


Figure 32. La séquence consensus de 10_2 correspond à a-SRPK1. 13 SDS identifiés, 10 SDS de type I et 3 de type II. La paire de SRPK1 est en ALT, celle de SRPK2 aussi.

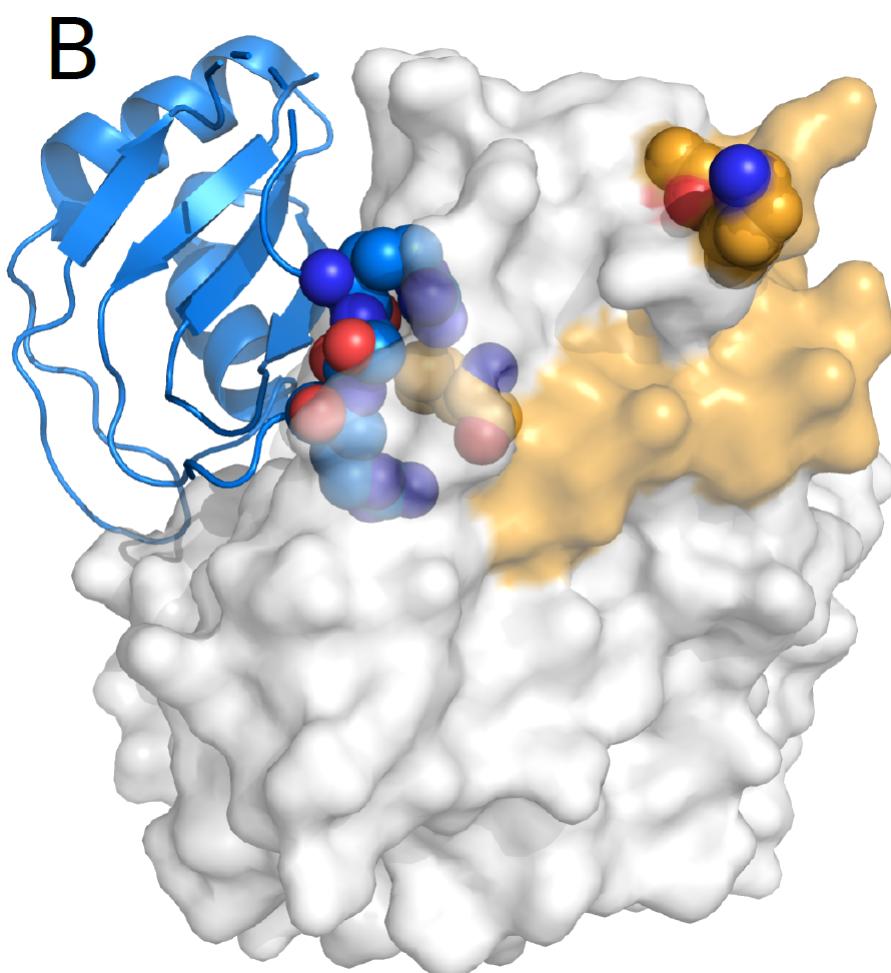


Figure 33. Structure du complexe entre SRSF protein kinase 1 (SRPK1, surface transparente) et son substrat ASF/SF2 (cartoon bleu) (PDB id: 3BEG). Le s-exon 10_2 est en orange, et les trois résidus identifiés comme étant des SDS de type II (carrés dans le MSA) sont représentés par des sphères, ainsi que les résidus du partenaire impliqués dans le contact.

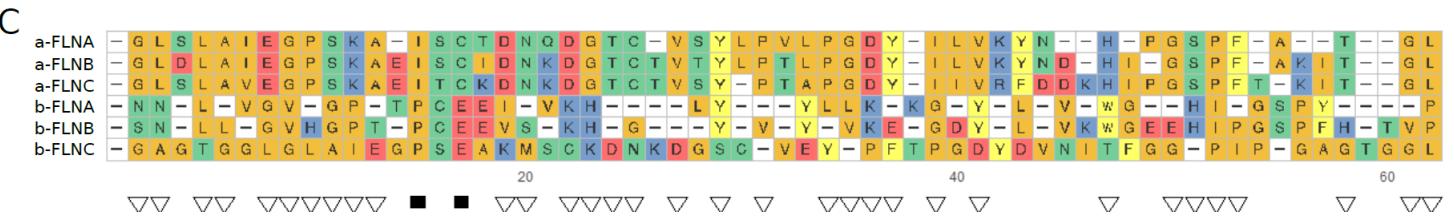


Figure 34. La séquence consensus de 3_26 correspond à a-FLNA. 35 SDS identifiés, 33 SDS de type I et 2 de type II. La paire de FLNA est en UNREL, celle de FLNB en UNREL et celle de FLNC aussi.

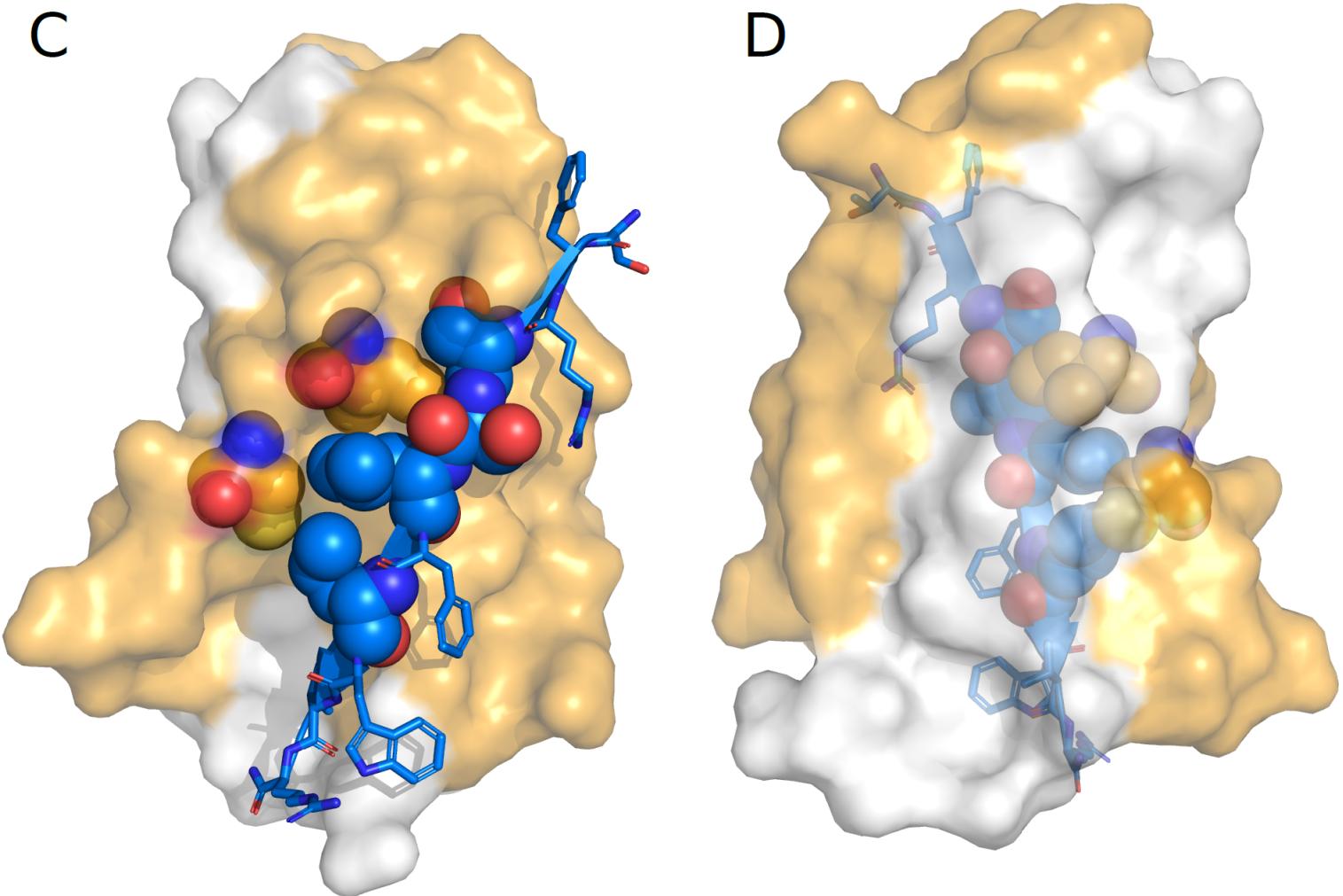


Figure 35. Structure du complexe entre la Filamin A (FLNA, surface transparente) et la platelet glycoprotéine IB alpha chain (cartoon bleu) (PDB id: 2bp3). Les deux protéines sont incomplètes, et la région d'intérêt de FLNA est annotée dans Uniprot comme étant le Filamin repeat 17. Le s-exon 3_26 est en orange, et les deux résidus identifiés comme étant des SDS de type II (carrés dans le MSA) sont représentés par des sphères, ainsi que les résidus du partenaire impliqués dans le contact. C. Face avant. D. Face arrière.

10. Difficultés rencontrées

Au cours du projet, plusieurs difficultés ont été rencontrées, la plupart ont été surmontées, d'autres pas dans le temps imparti.

Parmi les difficultés rencontrées surmontées, il y a eu des problèmes sur le parsing des données du dataset. Certaines données contenaient des bugs au niveau de l'écriture (format) du fichier. Par ailleurs, on s'était accordé de faire boucler la fonction chargée d'effectuer les extensions sur les paires étendues. Pour étendre plus d'une fois dans la mesure du possible. Cela n'a pas pu être fait car la résolution du problème des extensions a complexifié l'accès aux données sur les paires étendues. Nous avons quand même réussi à aller parfois jusqu'à deux extensions.

De plus pour la résolution de ce problème nous étions initialement partis sur un ensemble de conditions devant être vérifiées mais au bout d'un certain temps nous avons remarqué que cela n'était pas suffisant, il fallait vérifier que la liaison instance-seed + extension ne soit jamais coupée par un évènement et que si la paire candidate pour faire l'extension est en UNREL alors le chemin considéré doit être changé, ce qui nous a permis d'aboutir à une solution finale. Les extensions ont parfois générées des instances redondantes, apparaissant ainsi dans deux ASRU. Par exemple si on a les paires (A,B) et (A,D), on a une ASRU contenant A,B et D. Si (A,B) a servi à faire l'update (A+C,B) et que C soit contenue dans une autre ASRU alors C allait apparaître dedans comme A+C. Il a donc fallu reconnaître dans quelle ASRU l'extension devait avoir lieu.

Enfin, le code n'a pas été complètement révisé en terme d'optimisation. Sur le set de 2190 gènes (contenant 82 496 paires), il met environ 5h avant de finir l'analyse.

Pour les difficultés non surmontées à temps, il y a la factorisation du code et l'automatisation de l'analyse des séquences similaires entre paralogues d'une même famille et ce pour toutes les familles, voir figure 28 pour les détails à ce sujet.

11. Conclusions

En conclusion nous avons fourni une analyse détaillée des pseudo-repeats dans le cadre de l'épissage alternatif, de leurs variations de séquences et de leur usage alternatif, en vue d'identifier des signatures spécifiques. Nous avons défini le concept d'Alternatively Spliced Repetitive Units (ASRU) et développé un algorithme identifiant des groupes de noeuds dans un graphe correspondants aux pseudo-repeats, les pseudo-repeats détectés sont en accords avec ceux mentionnés dans la littérature (cas de ANK2, NEB, ...). Les sorties de cet algorithme nous ont permis de détecter des faux-positifs dans les sorties de ThorAxe. Nous avons également mis en évidence des cas d'utilisation complexe de plusieurs instances d'une même unité, au sein d'une protéine (ANK2, NEB). Les ASRU que nous détectons contiennent possiblement des domaines répétés, avec des cas d'insertion/délétion de fragments de tailles variables de domaines, typiquement quand un gène possède plusieurs ASRU avec deux instances ou plus (ATP1A4, SLC14A2, ...). Nous avons mis en évidence des cas d'utilisation alternative de motifs similaires dans des familles de paralogues (spécificité de liaison à des partenaires), pour le cas de TEAD, la substitution détectée n'est pas référencée sur Uniprot. Nous avons également remarqué qu'en faisant tourner le code avec une condition qui exclut les paires UNREL, les ASRU pouvaient s'éclater en plusieurs ASRU, nous enseignant ainsi que la structure d'une ASRU est potentiellement constituée de sous-groupes de repeats (de classe MEX, ALT ou REL) liés entre eux par transitivité. Les sous-groupes d'instances impliqués dans des meta-événements (par exemple des évènements à l'intérieur de l'évènement dans laquelle est impliquée la paire en UNREL) seraient reliés entre eux par une instance "high level" (l'une des instances figurant dans la paire UNREL).

Appendix A. Fonction qui calcule les marges

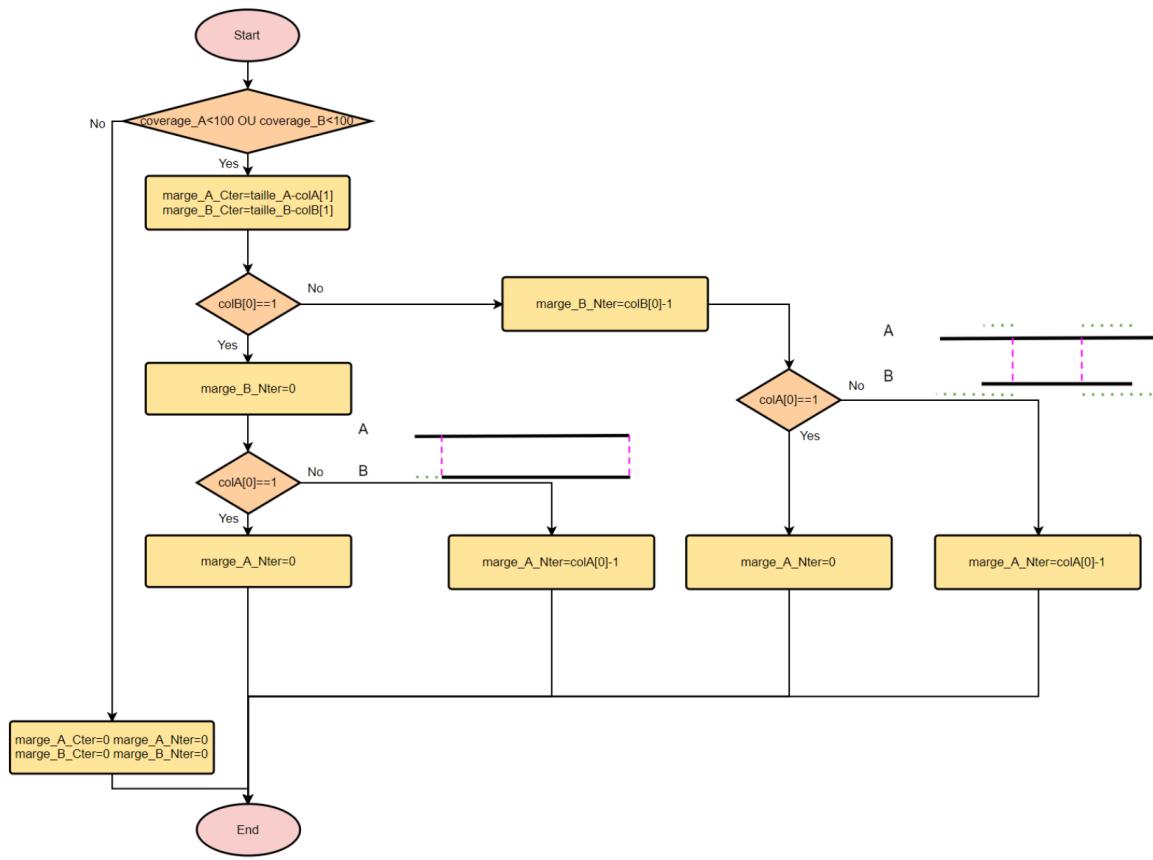
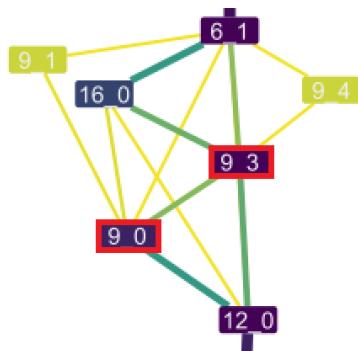
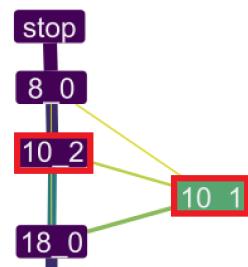


Figure 36. algorigramme de la fonction qui calcule les marges avec illustrations de deux cas.

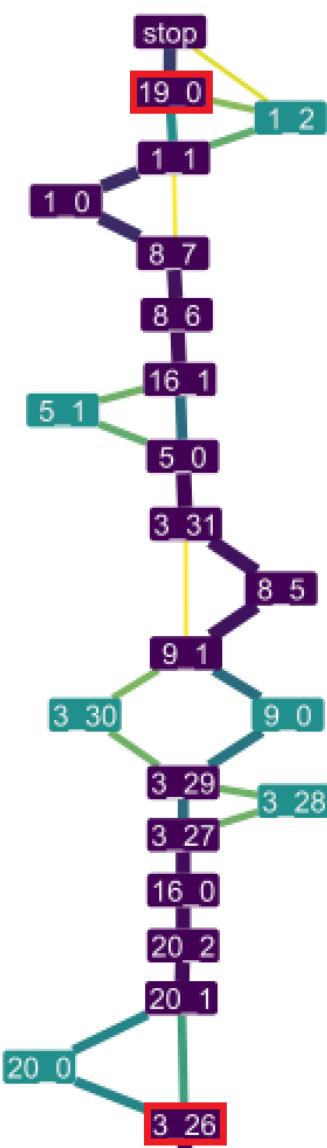
Appendix B. ESG



(a) Portion de l'ESG de TEAD1. Encadrés en rouge les s-exons dans les groupes a et b.
les s-exons dans les groupes a et b.



(b) Portion de l'ESG de SRPK1. Encadrés en rouge



(c) Portion de l'ESG de FLNA. Encadrés en rouge
les s-exons dans les groupes a et b.

References

- [1] Diego Javier Zea; Sofya Laskin; Hugues Richard; Elodie Laine; Assessing Conservation of Alternative Splicing with Evolutionary Splicing Graphs. *Biorxiv* **2020**, doi: 10.1101/2020.11.14.382820.
- [2] Steffen Heber; Max Alekseyev; Sing-Hoi Sze; Haixu Tang; Pavel A. Pevzner; Splicing graphs and EST assembly problem. *Bioinformatics*, Volume 18, Issue suppl_1, **July 2002**, Pages S181–S188, doi: 10.1093/bioinformatics/18.suppl_1.s181.
- [3] Chao Wang; Zhiyi Wei; Keyu Chen; Fei Ye; Cong Yu; Vann Bennett; Mingjie Zhang; Structural basis of diverse membrane target recognitions by ankyrins. *eLife*, vol. 3 e04353, **10 Nov. 2014**, doi: 10.7554/eLife.04353.
- [4] Åsa K. Björklund; Sara Light; Rauan Sagit; Arne Elofsson; Nebulin: A Study of Protein Repeat Evolution. *Journal of Molecular Biology*, Volume 402, Issue 1, **2010**, doi: 10.1016/j.jmb.2010.07.011.
- [5] Abhijit Chakraborty; Saikat Chakrabarti; A survey on prediction of specificity-determining sites in proteins, *Briefings in Bioinformatics*, Volume 16, Issue 1, **January 2015**, doi: 10.1093/bib/bbt092.