Université de Montréal

Évolution à fine échelle des sites d'épissage des introns dans les gènes des oomycètes

par
Steven Sêton Bocco

Département de biochimie et médecine moléculaire
Faculté de médecine

Mémoire présenté à la Faculté des études supérieures
en vue de l'obtention du grade de Maître ès sciences (M.Sc.)
en bio-informatique

août, 2015

Université de Montréal
Faculté des études supérieures

Ce mémoire intitulé:

Évolution à fine échelle des sites d'épissage des introns dans les gènes des oomycètes

présenté par:

Steven Sêton Bocco

a été évalué par un jury composé des personnes suivantes:

Sylvie Hamel,	président-rapporteur
Miklós Csűrös,	directeur de recherche
Simon Joly,	membre du jury

Mémoire accepté le: 29 septembre 2015

RÉSUMÉ

Les introns sont des portions de gènes transcrites dans l'ARN messager, mais retirées pendant l'épissage avant la synthèse des produits du gène. Chez les eucaryotes, on rencontre les introns splicéosomaux, qui sont retirés de l'ARN messager par des splicéosomes.

Les introns permettent plusieurs processus importants, tels que l'épissage alternatif, la dégradation des ARNs messagers non-sens, et l'encodage d'ARNs fonctionnels. Leurs rôles nous interrogent sur l'influence de la sélection naturelle sur leur évolution. Nous nous intéressons aux mutations qui peuvent modifier les produits d'un gène en changeant les sites d'épissage des introns. Ces mutations peuvent influencer le fonctionnement d'un organisme, et constituent donc un sujet d'étude intéressant, mais il n'existe actuellement pas de logiciels permettant de les étudier convenablement. Le but de notre projet était donc de concevoir une méthode pour détecter et analyser les changements des sites d'épissage des introns splicéosomaux.

Nous avons finalement développé une méthode qui repère les évènements évolutifs qui affectent les introns splicéosomaux dans un jeu d'espèces données. La méthode a été exécutée sur un ensemble d'espèces d'oomycètes. Plusieurs évènements détectés ont changé les sites d'épissage et les protéines, mais de nombreux évènements trouvés ont modifié les introns sans affecter les produits des gènes.

Il manque à notre méthode une étape finale d'analyse approfondie des données récoltées. Cependant, la méthode actuelle est facilement reproductible et automatise l'analyse des génomes pour la détection des évènements. Les fichiers produits peuvent ensuite être analysés dans chaque étude pour répondre à des questions spécifiques.

Mots clés: intron, évolution, eucaryote, épissage, gène.

ABSTRACT

Introns are portions of genes transcribed into messenger RNA, but removed during RNA splicing. In eukaryotes, they are called spliceosomal introns as they are removed by spliceosomes.

Introns allow many important processes such as alternative splicing, nonsense-mediated decay and functional-RNA coding. These roles leads to the question of the influence of natural selection on evolution of introns. We focus on mutations that are able to change gene products by modifing introns splice sites. These mutations seems to be an interesting topic as they can affect proteins, but there is currently no software to study them properly. The aim of our project was to design a method to detect and analyze changes in splice sites of spliceosomal introns.

We finally developed a method that locates the evolutionary events on splice sites of spliceosomal introns in a given species set. The method was performed on a set of oomycetes. Several detected events change splice sites and proteins, but there is also many events that seems to modify introns without affecting gene products.

Our method lacks a final step for thorough analysis of the collected events. However, the current method is easily reusable and automates genome analysis for the detection of events. The resulting files can then be analyzed in each study to answer specific questions.

Keywords: intron, evolution, eukaryote, splicing, gene.

TABLE DES MATIÈRES

LISTE DES TABLEAUX

LISTE DES FIGURES

LISTE DES SIGLES

albu	*Albugo laibachii*
AMF	Aligned Marked Fasta
EJC	Exon-exon Junction Complex
hyal	*Hyaloperonospora arabidopsidis*
ILS	Incomplete Lineage Sorting
LECA	Last Eukaryotic Common Ancestor
PCA	Parent-Children Alignments
phca	*Phytophthora capsici*
phci	*Phytophthora cinnamomi*
phin	*Phytophthora infestans*
phpa	*Phytophthora parasitica*
phra	*Phytophthora ramorum*
phso	*Phytophthora sojae*
pyul	*Pythium ultimum*

REMERCIEMENTS

Un grand merci à mon directeur de recherche M Miklós Csűrös, mon parrain M Simon Joly, mes parents, ma famille, mes amis, et toutes les personnes qui m'ont de près ou de loin aidé à finir cette maîtrise !

CHAPITRE 1

INTRODUCTION

1.1 Définition et propriétés des introns

1.1.1 Définition

À la lumière des connaissances actuelles, on peut définir un gène comme étant une suite de séquences localisée sur un ADN et contenant les codes de fabrication pour une ou plusieurs molécules nécessaires au fonctionnement d'un organisme [20, 21]. Les molécules encodées dans les gènes sont des protéines ou des ARNs, à la base du fonctionnement des êtres vivants.

Il arrive cependant qu'un gène ne contienne pas exclusivement les codes de fabrication de ses produits. Dans de nombreux gènes, la séquence codante est en réalité découpée en plusieurs pièces, appelées exons, entrecoupées de séquences qui ne codent pas pour les produits de ces gènes. Ces séquences dites non-codantes sont appelées introns. Pour synthétiser les produits d'un gène, ce dernier est transcrit en ARN messager sans perte d'information. Les introns sont ensuite retirés de cet ARN au cours d'une phase appelée épissage. L'ARN messager mature finalement obtenu ne contient que les exons, et est utilisé soit pour fabriquer des protéines, soit en tant qu'ARN fonctionnel dans la cellule.

Les introns sont observés dans de nombreux gènes chez tous les eucaryotes, mais on en trouve aussi chez les procaryotes [53].

1.1.2 Phases d'introns

Dans les gènes qui codent pour des protéines, la séquence codante est une suite de triplets de nucléotides, appelés codons, qui codent chacun pour un acide aminé précis des protéines encodées dans ce gène. Comme les introns fragmentent la séquence codante en plusieurs exons, on peut les retrouver entre deux codons, ou à l'intérieur d'un codon. La phase d'un intron désigne

ainsi la situation relative de l'intron par rapport aux codons du gène [17]. On peut classer les introns dans 3 phases :

- Les introns en phase 0 (ou en phase 3, selon l'appellation choisie) sont les introns situés entre deux codons (donc après le 3e nucléotide d'un codon).
- Les introns en phase 1 sont les introns situés après le 1er nucléotide d'un codon.
- Les introns en phase 2 sont les introns situés après le 2ème nucléotide d'un codon.

1.1.3 Classes d'introns

Il existe différentes classes d'introns, en fonction du processus d'épissage qui les élimine. Certains, comme les introns du groupe I et II, sont capables de s'auto-épisser grâce à un repliement dépendant de leur structure. D'autres, appelés « introns splicéosomaux » (« *spliceosomal introns* »), sont identifiés par des signaux particuliers qui permettent leur épissage au moyen d'un complexe ribonucléoprotéique appelé splicéosome [44, 53]. Les introns splicéosomaux sont rencontrés spécifiquement chez les eucaryotes, et constituent le sujet d'étude de notre projet.

1.1.4 Structure des introns splicéosomaux

Les introns splicéosomaux sont reconnus par le splicéosome grâce à leur structure caractéristique [44] présentée dans la figure 1.1. La structure d'un intron splicéosomal est composée de 4 sites particuliers :

- Le site donneur : il représente la frontière entre l'extrémité 5' de l'intron (en amont) et l'exon qui le précède.
- Le site accepteur : il s'agit de la frontière entre l'extrémité 3' de l'intron (en aval) et l'exon qui le suit.
- Le point de branchement : c'est une molécule d'adénine située dans l'intron, plus proche du site accepteur que du site donneur, et qui interagit avec le site donneur lorsque l'intron est retiré de l'ARN messager.

- La chaîne de polypyrimidines : c'est une suite de pyrimidines (cytosine et thymine dans l'ADN, cytosine et uracile dans l'ARN) située avant le site accepteur et rencontrée dans la majorité des introns splicéosomaux.

Les sites donneur et accepteur sont appelés sites d'épissage, et représentent les frontières de l'intron. Un site d'épissage est composé de 2 parties : une partie exonique située dans l'exon voisin, et une partie intronique située dans l'intron. Dans le site donneur, la partie exonique précède la partie intronique, tandis que dans le site accepteur, c'est la partie intronique qu'on rencontre avant la partie exonique. Lorsqu'un intron est enlevé de l'ARN messager au cours de l'épissage, il subsiste dans l'ARN la partie exonique du site donneur de l'intron et la partie exonique du site accepteur de l'intron. Ces deux parties mises bout à bout forment une séquence reconnaissable appelée site de proto-épissage.

En fonction de la séquence de nucléotides des sites d'épissage, on distingue deux principales structures d'introns, chacune reconnue par un splicéosome particulier. La structure la plus souvent rencontrée est représentée dans la figure 1.2. Dans cette structure, le site donneur a pour séquence-consensus « AG|GT », avec « AG » comme partie exonique et « GT » comme partie intronique. Dans le site accepteur, la partie intronique a pour séquence-consensus « AG », mais la partie exonique est plus variable [44].

D'autres variations de structure sont parfois rencontrées chez certains eucaryotes. Par exemples, certains eucaryotes unicellulaires n'ont pas de chaînes de polypyrimidines dans leurs introns [44].

1.2 Rôles et fonctions des introns

Comme les introns sont retirés de l'ARN messager pendant son épissage, on pourrait penser qu'ils sont inutiles pour les cellules. Cependant, de nombreuses recherches montrent que les introns peuvent jouer plusieurs rôles dans le fonctionnement des cellules et dans la diversification des produits des gènes. Ils permettent notamment l'épissage alternatif et la dégradation des ARNs messagers non-sens, et peuvent même coder pour certains ARNs actifs dans les cellules.

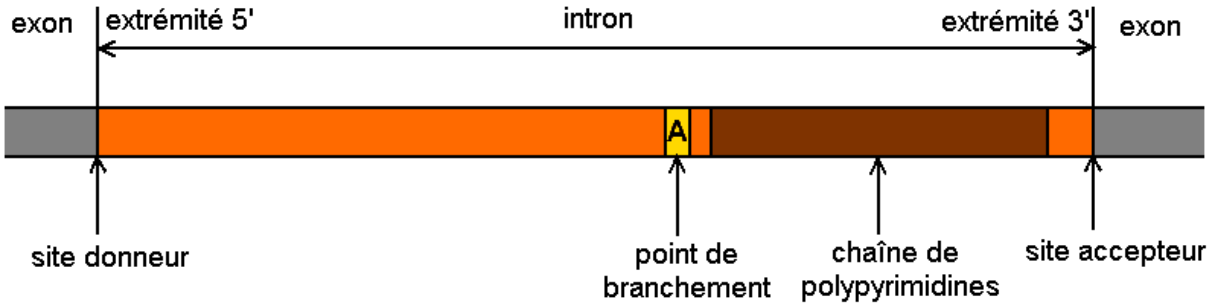


Figure 1.1 – Structure générique d'un intron splicéosomal.

(a) Donor splicing motif consensus

(b) Acceptor splicing motif consensus

Figure 1.2 – Structure consensus des sites d'épissage donneur et accepteur de la majorité des introns splicéosomaux observés chez les eucaryotes. Les lignes verticales rouges représentent les frontières entre l'intron et les exons voisins. La figure provient de la référence [44].

1.2.1 Épissage alternatif

Lorsqu'un gène est exprimé, il est transcrit en ARN messager qui subit un épissage au cours duquel les introns sont retirés, et les exons restants sont alors assemblés dans l'ARN messager mature. Cependant, pour certains gènes, et selon les besoins de la cellule, tous les exons en provenance du gène ne sont pas forcément insérés dans l'ARN messager mature, et les exons retenus ne sont pas obligatoirement assemblés dans l'ordre dans lequel ils apparaissent sur l'ADN. Ce processus de sélection et de mise en ordre des exons d'un gène dans l'ARN messager est appelé épissage alternatif.

L'épissage alternatif rend donc possible la création de plusieurs ARNs messagers matures différents à partir d'un même gène de départ, qui seront convertis en molécules (protéines ou ARNs fonctionnels selon le gène) également différentes. Grâce à ce processus, un seul gène peut ainsi coder pour plusieurs molécules différentes, ce qui augmente la diversité des molécules synthétisées par un organisme sans que le nombre ou la longueur des gènes varient

significativement. Un exemple d'épissage alternatif est donné par la drosophile, dont le gène « dsx » code pour deux protéines impliquées dans la différenciation sexuelle de la mouche. Une des deux protéines est exprimée à un stade précoce du développement de la mouche, et, selon la protéine, l'organisme deviendra soit un mâle, soit une femelle [27].

Il existe plusieurs modes d'épissage alternatif [4, 10], parmi lesquels on peut citer :

- La sélection d'un exon. Un exon du gène peut être retenu ou ignoré dans l'ARN épissé.
- La sélection mutuellement exclusive entre deux exons. L'un ou l'autre est présent dans l'ARN après épissage, mais jamais les deux simultanément.
- La sélection alternative d'un site d'épissage en 5' ou en 3'. La frontière de l'intron enlevé est modifiée, et la longueur de l'exon retenu change donc elle aussi, selon la portion d'exon qui a été enlevée avec l'intron retiré.
- La rétention d'intron. Une portion génique ayant les caractéristiques typiques d'un intron peut être retenue dans l'ARN messager épissé au lieu d'être enlevée.

L'épissage alternatif joue donc un rôle important dans la diversification des produits des gènes, et notamment des protéines. Certaines études estiment que, chez l'Homme, 60 % de tous les gènes [4], ou 95 % des gènes qui contiennent des introns [33], subissent l'épissage alternatif.

Lorsque des mutations se produisent au niveau des sites d'épissage, elles peuvent affecter les possibilités d'épissage alternatif d'un gène, et même engendrer des maladies génétiques. Ainsi, d'après López-Bigas et al. [26], 62 % des mutations causant des maladies chez l'Homme affecteraient les sites d'épissage plutôt que les séquences codantes elles-mêmes. Lim KH et al. [25] soutiennent quant à eux qu'un tiers des maladies causées par des mutations génétiques seraient liées à des problèmes d'épissage.

Les introns, sans lesquels l'épissage alternatif n'existerait pas, sont ainsi impliqués dans des maladies d'ordre génétique et dans la diversité des protéines fabriquées par un organisme.

1.2.2 Dégradation des ARNs messagers non-sens

Les gènes qui codent pour des protéines sont convertis en ARNs messagers matures qui contiennent une suite de codons, chaque codon correspondant à un acide aminé de la protéine à synthétiser. Pour fabriquer la protéine, des ribosomes lisent l'ARN messager mature à partir d'un codon de départ qui indique le premier acide aminé de la protéine, et s'arrêtent lorsqu'ils rencontrent un codon qui indique la fin de la protéine, et qui ne correspond lui-même à aucun acide aminé. Le codon d'arrêt est appelé codon STOP, et se trouve donc dans un exon. On peut rencontrer 3 codons STOP dans l'ADN des êtres vivants : UAA, UAG et UGA.

Si le gène contient des introns, on s'attend donc à ce que le codon STOP soit situé dans le tout dernier exon présent dans l'ARN messager mature. Cependant, à la suite par exemple de mutations ponctuelles, il peut arriver que des codons STOP apparaissent dans d'autres exons du gène en amont du dernier codon STOP. Il s'agit de codons STOP prématurés. Si les ribosomes rencontrent un tel codon, ils produiront une protéine tronquée, inutilisable par la cellule ou susceptible de perturber gravement son fonctionnement. Des ARNs messagers contenant des codons STOP prématurés doivent donc être rapidement repérés et détruits, afin qu'ils ne soient pas réutilisés pour fabriquer d'autres protéines dysfonctionnelles. Ces ARNs sont appelés ARNs messagers non-sens, et leur dégradation est un mécanisme de régulation important observé chez tous les eucaryotes [9].

On sait maintenant que le découpage des gènes en exons, du fait de la présence des introns, est impliqué dans un mode de détection des ARNs messagers non-sens chez les eucaryotes, notamment chez les mammifères [3, 9]. En effet, la présence d'introns permet l'existence de jonctions exon-exon repérables dans l'ARN messager mature. Un complexe de jonction exon-exon appelé EJC (« *exon-exon junction complex* »), se fixe en amont de chaque jonction exon-exon à une distance d'environ 20 à 24 nucléotides de la jonction. L'ARN messager mature et les EJCs qui se sont associés à lui forment un complexe qui est ensuite lu par les ribosomes. Au cours de la traduction, les ribosomes enlèvent les EJCs qu'ils rencontrent, et arrêtent la traduction au premier codon STOP lu.

Puisque le premier codon STOP rencontré devrait normalement se trouver dans le dernier exon, il devrait être situé après la dernière frontière exon-exon dans l'ARN messager, donc après l'EJC le plus proche de la région terminale de l'ARN messager. Ainsi, si les ribosomes ont lu l'ARN jusqu'à rencontrer ce codon STOP, tous les EJCs auront été retirés. Mais si un codon STOP prématuré est rencontré plus tôt par les ribosomes, avant qu'ils n'arrivent au dernier exon, alors il restera au moins un EJC rattaché à l'ARN messager. Or un complexe ARN messager - EJC peut être repéré par la machinerie de dégradation des ARNs messagers non-sens. Celle-ci est alors mobilisée autour de l'ARN non-sens ainsi détecté, et se charge de supprimer rapidement l'ARN afin qu'il ne soit pas réutilisé pour synthétiser de nouvelles protéines tronquées.

Dans les gènes qui ne contiennent pas d'introns, mais qui comporteraient des codons STOP prématurés, un tel mode de détection des ARNs messagers non-sens est impossible, car aucune jonction exon-exon n'est disponible pour la liaison des EJCs. Les introns peuvent ainsi jouer un rôle crucial dans les divers mécanismes qui assurent le bon fonctionnement d'une cellule.

1.2.3 Encodage d'ARNs fonctionnels

On appelle ARN non-codant un ARN qui ne code pas pour une protéine. Il existe une grande variété d'ARNs qui sont non-codants mais qui jouent des rôles importants dans le fonctionnement des cellules. On connaît notamment les petits ARNs du nucléole (snoRNA), les micro-ARNs (miRNA), et les petits ARNs interférents (siRNA), qui sont connus pour réguler l'expression des gènes [5].

On sait désormais que de nombreux introns du génome humain contiennent les codes d'une grande variété d'ARNs non-codants régulateurs utilisés dans les cellules [5, 42]. Les introns qui codent pour ces ARNs sont donc directement impliqués dans le contrôle de la production de nombreuses protéines.

1.3 Évolution des introns

La présence d'introns dans de nombreuses espèces nous incite à nous demander s'ils étaient déjà présents dans des espèces ancestrales, voire dans des organismes situés très hauts dans l'arbre du vivant. En outre, puisqu'ils peuvent influencer le fonctionnement des organismes, on peut supposer que les introns constituent une source intéressante de variation pour la sélection naturelle. Les études menées sur l'évolution des introns cherchent donc aussi à déterminer s'ils sont effectivement soumis à des pressions sélectives pendant l'évolution des diverses lignées du vivant. Plusieurs études ont déjà été menées pour tenter d'apporter des réponses à ces interrogations.

1.3.1 Origine des introns splicéosomaux

La présence d'introns splicéosomaux dans toutes les lignées d'eucaryotes suggère que leur apparition est ancienne, et pose la question de leurs origines. Deux théories ont été proposées pour expliquer l'origine des introns splicéosomaux : la théorie des introns tôt et la théorie des introns tard [44].

La théorie des « introns tôt » propose que tous les introns des eucaryotes proviennent d'un ancêtre procaryote, commun aussi bien aux eucaryotes qu'aux procaryotes actuels. Les différences observées au niveau des introns chez les espèces eucaryotes actuelles proviendraient donc essentiellement de pertes d'introns spécifiques dans chaque lignée. Cette théorie implique que les procaryotes actuels, qui partageaient le même ancêtre, ont progressivement perdu presque tous leurs introns au cours de l'évolution de leurs génomes. Des versions plus récentes de la théorie admettent la possibilité que de nouveaux introns soient apparus chez certains eucaryotes au fil du temps, s'ajoutant aux introns ancestraux.

La théorie des « introns tard » quant à elle propose plutôt que les introns splicéosomaux sont apparus exclusivement chez les eucaryotes, via des gains d'introns qui se sont produits pendant leur évolution. Selon cette théorie, les procaryotes n'auraient donc jamais eu d'introns splicéosomaux ni de splicéosomes.

Les études récentes tendent à montrer que les ancêtres des eucaryotes, incluant le dernier ancêtre commun à tous les eucaryotes (« LECA »), possédaient des gènes riches en introns [44]. Les lignées d'eucaryotes auraient donc évolué en combinant des évènements de pertes et de gains d'introns, avec des pertes se produisant plus fréquemment que des gains.

Ces études posent alors la question de l'origine des introns de LECA. Des comparaisons des régions terminales des introns splicéosomaux et de certains introns auto-épissables du groupe II suggèrent que ces deux classes d'introns sont apparentées. Les introns splicéosomaux auraient ainsi évolué à partir des introns auto-épissables du groupe II, qu'on retrouve principalement chez certains procaryotes [44]. Une théorie, associant cette observation et les connaissances sur l'origine des eucaryotes, propose que les introns splicéosomaux dérivent d'introns du groupe II qui se seraient trouvés chez un procaryote absorbé au cours d'une endosymbiose par un autre procaryote, au cours de la formation des premières cellules eucaryotes [44]. Les introns du groupe II auraient alors migré vers le génome de l'hôte, tout en subissant des modifications qui ont rendu leur épissage dépendant d'un splicéosome.

À l'heure actuelle, la théorie des introns « tôt » est donc la plus soutenue pour expliquer l'origine des introns des eucaryotes.

1.3.2 Mécanismes d'évolution des introns

Les théories relatives à l'origine des introns considèrent que des gains et des pertes d'introns dans les espèces peuvent se produire au fil du temps. Ces hypothèses suggèrent que les introns changent au cours de l'évolution des espèces, ce qui est soutenu par les observations faites dans les génomes connus, car le nombre et la longueur des introns varient significativement d'une espèce à une autre [44]. On peut donc se demander si les introns sont soumis à des pressions sélectives qui guident leur évolution, ou s'ils évoluent de façon totalement aléatoire.

Pour répondre à cette question, il est nécessaire d'identifier les différents mécanismes d'évolution des introns, afin de vérifier si certains mécanismes sont privilégiés par rapport à d'autres en fonction de conditions observées, et donc déterminer si l'évolution des introns n'est pas aléatoire.

Plusieurs modèles d'évolution ont été proposés pour expliquer les changements observés au niveau des introns dans les gènes. Certains modèles, tels que l'évolution par mutations ponctuelles de nucléotides, sont évidents et faciles à soutenir. D'autres modèles, tels que l'évolution par transposition d'introns, manquent d'études approfondies et ne sont donc pas suffisamment soutenus [53].

Les modèles proposés nous permettent cependant de définir plusieurs mutations introniques potentielles. Nous appelons mutation intronique une mutation qui insère, supprime ou modifie un intron dans un gène. À partir des modèles d'évolution proposés, et en se servant du gène comme repère, nous pouvons distinguer 2 catégories de mutations introniques :

- Les mutations introniques génomiques, qui sont dues à l'interaction du gène avec d'autres séquences issues du génome et situées à l'extérieur de ce gène. On peut les qualifier de mutations introniques globales, car elles se produisent à l'échelle du génome.
- Les mutations introniques intragéniques, qui sont dues exclusivement à des mutations ponctuelles (changement de nucléotides) à l'intérieur du gène. On peut les qualifier de mutations introniques locales, car elles se produisent à l'échelle du gène seul.

Le but de notre projet est d'étudier les mutations introniques locales des introns splicéosomaux.

1.3.2.1 Évolution des introns par mutations introniques globales

Les mutations introniques globales se produisent lorsqu'un gène interagit avec une autre séquence nucléotidique issue du génome. Ces interactions génomiques peuvent conduire à des insertions ou des suppressions complètes d'introns dans le gène [53]. Les mutations introniques globales se manifestent donc par des pertes ou des gains d'introns.

Les pertes d'introns peuvent se produire via une recombinaison génomique [53]. Un ARN messager mature, ne contenant donc aucun intron, peut être converti en une séquence d'ADN complémentaire via une transcription inverse. Cette séquence peut alors être impliquée dans une

recombinaison génomique avec le gène initial, au cours de laquelle les introns situés dans le gène seront perdus. Des introns entiers peuvent ainsi être supprimés sans affecter les exons voisins.

Les gains d'introns peuvent se produire via diverses interactions [53] :

- La transposition d'intron. Un intron retiré d'un ARN messager au cours de l'épissage peut s'insérer dans un autre ARN messager, du même gène ou d'un gène différent. Cet ARN messager peut être converti par transcription inverse en un ADN complémentaire, qui contient donc un nouvel intron. Une recombinaison génomique peut ensuite avoir lieu avec le gène de l'ARN messager rétro-transcrit, au cours de laquelle le nouvel intron est inséré dans le gène.

- Le transfert d'intron entre deux gènes paralogues. Deux gènes sont paralogues s'ils proviennent d'une duplication d'un gène ancestral dans un même génome. Au cours de l'évolution, des introns peuvent apparaître dans un des gènes sans apparaître dans l'autre, ou apparaître dans les deux gènes mais pas forcément aux mêmes positions. Une recombinaison entre deux gènes paralogues après une certaine période d'évolution peut alors permettre le passage de nouveaux introns d'un gène à un autre.

- L'insertion d'un transposon. Un transposon (séquence d'ADN mobile dans le génome) s'insère dans un gène et est progressivement converti en intron au fil de l'évolution.

- L'auto-épissage d'introns du groupe II. Comme nous l'avons vu, ce modèle est suggéré pour expliquer l'origine des introns splicéosomaux. Il propose que des introns auto-épissables du groupe II, libérés par l'épissage de gènes d'organelles acquises par endosymbiose avec des procaryotes, s'insèrent dans le génome de la cellule hôte puis se transforment progressivement en introns splicéosomaux, c'est-à-dire en introns reconnus et épissés par un splicéosome de la cellule eucaryote hôte.

1.3.2.2 Évolution des introns par mutations introniques locales

Les mutations introniques locales se produisent lorsqu'un gène est modifié par des mutations ponctuelles. Les mutations ponctuelles désignent les mutations des nucléotides du gène, telles

que les insertions, les suppressions, les substitutions ou les duplications de nucléotides ou de petits blocs de nucléotides à l'intérieur du gène. Les mutations ponctuelles peuvent avoir divers effets sur les introns, selon l'endroit où elles se produisent.

Par exemple, dans un exon, l'apparition de 2 nouveaux sites d'épissage donneur et accepteur peut délimiter une portion de l'exon qui évoluera ensuite vers un intron. Une duplication génomique en tandem sur un site de proto-épissage situé dans l'exon peut aussi conduire à la naissance d'un nouvel intron. En effet, chez les eucaryotes, la séquence AGGT est un site de proto-épissage potentiel. L'intron se trouve entre AG et GT, mais une majorité d'introns contient également GT à leur extrémité 5' et AG à leur extrémité 3'. Si un tel site est dupliqué dans un exon, on obtient une sous-séquence de la forme AGGT...AGGT. Un nouvel intron peut alors apparaître à partir de la séquence GT...AG, flanqué d'AG en amont et GT en aval, ce qui redonne le site de proto-épissage AGGT [53].

Si les mutations ponctuelles se produisent dans un intron, elles peuvent modifier sa longueur (par insertion ou suppression de nucléotides à l'intérieur de l'intron), ou rendre son épissage impossible (par destruction de sa structure en dénaturant ses sites d'épissage, son point de branchement, ou sa chaîne de polypyrimidines) et le convertir ainsi en exon. Les mutations qui affectent les sites d'épissage peuvent avoir des conséquences plus variées, telles que la modification des frontières de l'intron. La suppression de tout ou une partie de l'intron est également possible suite à divers phénomènes évolutifs (par exemple des suppressions ponctuelles accumulées, ou une suppression segmentale). Dans le cas d'une suppression partielle du début ou de la fin d'un intron, qui supprime donc un de ses sites d'épissage, la partie restante de l'intron peut subsister et modifier les exons voisins en fusionnant avec eux [53].

Cas particulier des glissements d'introns

Certaines observations faites dans des alignements de gènes orthologues montrent des introns, qui semblent apparentés, mais dont les positions varient de quelques nucléotides d'un gène à un autre. Ces observations suggèrent un autre type de mutation intronique potentielle appelée glissement d'introns, qui désigne le déplacement d'un intron dans son gène.

Plusieurs glissements d'introns pourraient n'être qu'apparents, et pourraient représenter, par exemple, des insertions parallèles d'introns dans les gènes d'une même famille à des positions proches. Cependant, Tarrio et al. [48] ont proposé une théorie basée sur l'évolution d'un même intron grâce à l'épissage alternatif. Selon cette théorie, de nouveaux sites d'épissage peuvent apparaître près des sites d'épissage initiaux d'un intron, ce qui permet un épissage alternatif. Cependant, les anciens sites d'épissage peuvent disparaître suite à des mutations, si bien que les nouveaux sites les remplacent définitivement pour l'épissage de cet intron, ce qui provoque un changement de position de ce dernier, et donc une observation qui laisse supposer un glissement d'intron. Des études plus approfondies sont nécessaires pour déterminer si les glissements d'introns sont de véritables mutations introniques, et quelle est leur nature (globale ou locale).

1.3.2.3 Impact des mutations introniques sur les produits des gènes

La conséquence d'une mutation intronique globale est l'insertion ou la suppression d'un intron complet. Ainsi, plusieurs dizaines de nucléotides peuvent apparaître ou disparaître dans le gène, ce qui modifie significativement sa longueur et son apparence sur l'ADN. Cependant, ce sont des introns entiers qui sont gagnés ou perdus, sans que les exons qui les entourent soient forcément affectés. Ainsi, même si le gène change beaucoup, ses produits peuvent rester identiques à la suite de mutations introniques globales. Les mutations introniques globales peuvent ainsi avoir un effet négligeable, voire nul, sur le fonctionnement de l'organisme, puisqu'elles ne modifient pas forcément les protéines qu'il utilise.

Les mutations introniques locales, quant à elles, affectent des portions de nucléotides dans les gènes. Seuls quelques nucléotides sont changés, supprimés ou insérés dans un gène. Les modifications sont donc mineures sur la longueur et l'apparence du gène. Cependant, si ces mutations affectent les sites d'épissage des introns, elles peuvent convertir tout ou une partie d'un intron en exon, ou tout ou une partie d'un exon en intron. Ainsi, même si le gène change très peu, ses produits peuvent être significativement modifiés. Les mutations introniques locales peuvent donc avoir un effet important, voire majeur, sur un organisme, en modifiant les protéines nécessaires à son fonctionnement. Il serait donc intéressant d'étudier ces mutations afin de

déterminer leur fréquence et leur impact réel sur l'évolution des introns et des espèces.

1.4 Outils bio-informatiques disponibles pour l'étude de l'évolution des introns

Les mutations introniques locales pourraient influencer la production des protéines chez les eucaryotes lorsqu'elles affectent les sites d'épissage des introns. L'étude de l'évolution des sites d'épissage peut donc présenter un intérêt important pour la compréhension de l'évolution des gènes, de leurs produits et des organismes. En outre, l'étude des mutations introniques peut fournir de nombreuses informations sur les pressions sélectives qui s'exercent sur les introns.

Pour faciliter l'étude des mutations introniques, il serait convenable de disposer d'un logiciel adapté. Cependant, les outils bio-informatiques actuellement disponibles qui permettent l'étude des introns ne proposent pas d'options pour analyser efficacement les mutations introniques.

Par exemple, le logiciel « CIWOG » [52] (« *Common Introns Within Orthologous Genes* ») permet d'étudier l'évolution des introns splicéosomaux en comparant leur positionnement et leur longueur dans des gènes homologues. Il peut également associer à chaque intron le splicéosome qui l'épisse parmi les 2 splicéosomes qu'on retrouve chez les eucaryotes, et déterminer ainsi les classes des introns splicéosomaux. Avec ce logiciel, il est donc possible d'analyser l'évolution des changements de longueurs, de positions et de classes des introns splicéosomaux, et récolter des statistiques sur ces évolutions. Mais aucune option ne permet d'identifier les mutations introniques responsables des changements observés. On ne peut donc pas utiliser cet outil pour étudier précisément les mutations introniques locales. On ne peut pas non plus chercher des corrélations entre des critères potentiellement sélectifs et les changements observés au niveau des introns.

Un autre logiciel, appelé « MALIN » [12] (« *MAximum Likelihood analysis of INtron* »), est disponible pour étudier les introns dans les familles de gènes. Il propose diverses fonctionnalités, telles que :

- L'étude comparative et statistique des positions et des phases des introns dans des alignements de protéines.
- La prédiction des positions des introns présents chez le gène ancestral.
- La prédiction et l'analyse statistique des pertes et des gains d'introns.
- L'inférence de l'histoire évolutive de sites d'introns individuels (apparition, disparition ou déplacement d'un intron donné).

Cependant, en dépit des options proposées, MALIN ne se focalise que sur l'étude des pertes et des gains d'introns. On ne peut donc pas étudier efficacement d'autres évènements tels que les glissements d'introns. De plus, MALIN n'identifie pas la cause des pertes et des gains, et ne permet donc pas de déterminer si elles sont dues à des mutations introniques globales ou à des mutations introniques locales. Aussi, les évolutions qui changent les sites d'épissage des introns, sans entraîner leur disparition ou leur apparition complètes, ne peuvent pas être prédites ni analysées. Cet outil ne permet donc pas non plus une étude complète et approfondie de l'évolution des sites d'épissage.

1.5 Notre projet

1.5.1 Objectifs

Le but de notre projet était de proposer une méthode qui permettrait d'étudier les mutations introniques locales chez les eucaryotes, et plus particulièrement l'évolution des sites d'épissage susceptibles de modifier les produits des gènes. La méthode devait être une alternative généralisée et plus efficace que les logiciels actuellement disponibles, capable d'analyser non seulement les gains et les pertes d'introns, mais aussi l'ensemble des mutations introniques locales qui surviennent dans les génomes des eucaryotes. À partir de cette méthode, nous devions pouvoir répondre à plusieurs questions relatives aux mutations introniques locales pour un ensemble d'espèces données. Par exemple :

- Évaluer leur importance relative par rapport aux mutations globales. Parmi les introns étudiés, combien ont évolué via des mutations locales ?

- Évaluer la fréquence et l'importance des mutations introniques locales. Combien d'allongements, de raccourcissements, d'apparitions ou de disparitions complètes, combien dans chaque direction (5' ou 3') ou dans les deux directions en même temps ?

- Chercher des corrélations entre la fréquence des évènements évolutifs et des pressions sélectives (par exemple les fonctions des gènes). Certaines évolutions locales surviennent-elles plus souvent que d'autres selon le rôle de la protéine dans la cellule ?

1.5.2 Hypothèse

Pour développer notre méthode, nous avons émis l'hypothèse selon laquelle l'évolution des sites d'épissage des introns pourrait être inférée à partir d'alignements entre génomes et d'analyses comparatives locales dans les gènes, en considérant des génomes assez proches pour être correctement alignés, mais suffisamment éloignés pour que des changements soient visibles. À partir des alignements des familles de gènes, nous pourrions détecter et quantifier les mutations introniques locales qui affectent les sites d'épissage des génomes étudiés.

1.5.3 Aboutissement

Nous avons développé une méthode bio-informatique qui permet, à l'heure actuelle, de détecter les évènements évolutifs qui changent les sites d'épissage dans un ensemble de génomes données, et de les sauvegarder dans un ensemble de fichiers texte qui constituent la base de données des évènements collectés.

Les évènements sont repérés en cherchant des mutations introniques locales, mais plusieurs mutations mises en évidence, telles que les pertes et les gains d'introns, peuvent être des mutations introniques globales, et notre méthode actuelle ne permet pas encore d'identifier clairement la catégorie (locale ou globale) des mutations introniques détectées. Elle ne propose pas non plus d'outils permettant d'analyser l'ensemble des évènements identifiés pour répondre aux questions

d'intérêt sur les pressions sélectives qui s'exercent sur les introns. La méthode est donc perfectible, et de nombreuses options et améliorations seront prochainement ajoutées pour affiner la détection des mutations et l'extraction d'informations pertinentes à partir de la collection des mutations détectées.

Cependant, la méthode automatise tout le processus d'identification et de collecte des mutations introniques. Les chercheurs peuvent donc s'en servir pour rassembler plus facilement les évènements évolutifs des introns, puis développer plus rapidement des requêtes à exécuter sur la base de données obtenue pour répondre à leurs questions de recherche.

Nous avons développé notre méthode en étudiant une famille de champignons unicellulaires parasites, les oomycètes. Les résultats obtenus montrent plusieurs mutations introniques locales reconnaissables détectées dans de nombreuses familles de protéines orthologues. Cependant, les mutations les plus nombreuses sont les pertes et les gains d'introns, qui peuvent être causés par des mutations introniques globales. L'ensemble des mutations détectées suggère des tendances de gains d'introns et de déplacement des sites d'épissage vers le côté 3' des gènes, au cours de l'évolution des oomycètes étudiés. Ces tendances sont des hypothèses, qui pourront être analysés plus précisément avec les prochaines versions de notre méthode.

Dans les chapitres suivants, nous décrivons les étapes de la méthode, les paramètres, les avantages, les inconvénients et les améliorations possibles pour chaque étape, et nous présentons les résultats obtenus pour la famille des oomycètes. Nous expliquons enfin comment récupérer les codes nécessaires pour réutiliser notre méthode dans d'autres projets.

CHAPITRE 2

COLLECTE DES DONNÉES ET TRAITEMENTS PRÉLIMINAIRES

2.1 Choix des espèces à étudier

2.1.1 Théorie

Notre hypothèse de travail requiert que les données en entrée proviennent d'espèces proches (pour faciliter l'identification des séquences homologues) mais suffisamment différentes pour que des variations soient observables au niveau des gènes et des protéines. Il faut donc choisir un ensemble d'espèces en fonction de ce critère. De plus, l'étude de l'évolution des introns nécessitera la construction d'arbres phylogénétiques, et donc la présence d'un *outgroup* dans l'ensemble des espèces choisies. Un *outgroup* est un ensemble d'espèces témoins qui permettent de repérer la racine de l'arbre phylogénétique reconstruit.

Une manière simple de choisir est de prendre des espèces qui sont considérées comme appartenant à un même taxon, et de former un *outgroup* avec quelques espèces prises à l'extérieur de ce taxon, mais proches de ce taxon.

2.1.2 Application

Pour développer notre méthode, nous avons travaillé avec le taxon des oomycètes. Ce sont des organismes unicellulaires semblables à des champignons, dont plusieurs espèces sont des parasites. Certains oomycètes sont bien connus pour provoquer d'importantes maladies chez les plantes cultivées par l'Homme. L'espèce *Phytophthora infestans*, par exemple, est l'oomycète responsable du mildiou de la pomme de terre, une maladie capable de détruire des récoltes entières [31].

Au travers d'études antérieures menées sur ces parasites, nous savions que les oomycètes constituaient un bon candidat d'étude. Des génomes complets et déjà annotés de plusieurs espèces de ce taxon sont déjà disponibles et facilement accessibles. De plus, les espèces annotées sont

suffisamment proches pour faciliter la détection des homologies entre les gènes. Les oomycètes constituent également un sujet d'étude intéressant, car on peut se demander si leurs introns ont évolué en fonction d'adaptations à leurs hôtes ou aux différents environnements qu'ils colonisent.

Nous avons constitué un ensemble de 9 espèces d'oomycètes, qui comporte 6 espèces du genre Phytophthora, et 1 espèce dans chacun des genres *Pythium*, *Hyaloperonospora* et *Albugo*. Nous avons choisi l'espèce *Albugo laibachii* comme *outgroup* en analysant un arbre phylogénétique d'oomycètes disponible dans la littérature [34]. Les données collectées pour ces espèces incluaient les séquences des gènes, des transcrits et des protéines, et les annotations de leurs génomes, ainsi que les séquences de contigs ou de chromosomes pour certaines espèces. Toutes les données proviennent de 3 bases de données génomiques accessibles sur l'Internet :

- La base de données Ensembl Protists [23].
- La base de données de Broad Institute [6].
- La base de données du Joint Genome Institute (JGI) [30].

Le tableau 2.I présente la liste des espèces, l'origine des données pour chacune d'elles, et la date à laquelle ces données ont été récoltées.

2.2 Utilisation des protéines à la place des gènes dans notre méthode

Le développement de notre méthode nécessite de comparer des séquences entre elles pour identifier des séquences homologues et des composants homologues dans les séquences. Pour effectuer ces comparaisons, nous devrons réaliser des alignements de séquences, à partir desquels nous déduirons des informations indispensables à l'étude de l'évolution des introns. Il faut donc déterminer quel type de séquences nous comptons utiliser parmi les différents types disponibles, à savoir les protéines, les gènes, et les dérivés des gènes tels que les transcrits (qu'on peut considérer comme les gènes avec quelques portions de séquences enlevées, comme les introns).

Tableau 2.I – Liste des espèces étudiées et origine des données collectées pour chaque espèce.

Espèce (nom raccourci)	**Source des données (date des versions des données utilisées)**	**Format du fichier d'annotation utilisé**
Albugo laibachii (albu)	Ensembl Protists [36] (27/01/2014)	GTF
Hyaloperonospora arabidopsidis (hyal)	Ensembl Protists [37] (27/01/2014)	GTF
Phytophthora capsici (phca)	JGI [30] (23/01/2014)	GFF
Phytophthora cinnamomi (phci)	JGI [30] (23/01/2014)	GFF
Phytophthora infestans (phin)	Ensembl Protists [38] (29/01/2014)	GTF
Phytophthora parasitica (phpa)	Broad Institute [7] (23/01/2014)	GTF
Phytophthora ramorum (phra)	Ensembl Protists [39] (29/01/2014)	GTF
Phytophthora sojae (phso)	Ensembl Protists [40] (29/01/2014)	GTF
Pythium ultimum (pyul)	Ensembl Protists [41] (27/01/2014)	GTF

Le code génétique, qui représente la table de correspondance entre les codons et les acides aminés, contient 64 codons, dont 3 sont généralement des codons STOP, et 61 codent effectivement pour une vingtaine d'acides aminés [32]. Puisqu'il y a plus de codons que d'acides aminés, plusieurs codons différents codent pour un même acide aminé. De ce fait, les gènes peuvent accumuler plusieurs mutations sans que les protéines (séquences d'acides aminés) pour lesquelles ils codent soient obligatoirement modifiées. Les gènes peuvent donc évoluer plus rapidement que les protéines. Or une évolution rapide peut rendre trop différentes des séquences géniques qui sont apparentées, et donc fausser les alignements de ces séquences et les déductions à faire sur leurs homologies. Pour cette raison, nous avons utilisé les protéines à la place des gènes pour le développement de notre méthode, car elles évoluent moins vite que les gènes, et conservent donc plus d'homologies.

L'utilisation des protéines présente aussi un autre avantage. En effet, elles sont décrites avec un alphabet de 20 états (20 acides aminés), alors que les gènes et les transcrits sont des séquences nucléotidiques décrites avec un alphabet de seulement 4 états (4 nucléotides). Le risque de considérer comme homologues deux états qui ne le sont pas est donc beaucoup plus élevé si nous comparons les gènes, alors que les protéines offrent une plus grande diversité d'états, donc une probabilité un peu plus faible que deux états supposés homologues ne le soient pas réellement.

Travailler avec des protéines présente quelques complications lorsqu'il s'agit d'étudier des introns, car ces derniers sont des séquences nucléotidiques qui sont plus faciles à localiser dans des gènes. Nous avons cependant géré ces problèmes en créant des programmes et des notations adaptés, présentés dans les prochaines sections, et qui permettent de positionner et de mettre en évidence les introns d'un gène directement sur ses protéines.

2.3 Construction des familles de protéines

2.3.1 Théorie

Notre projet a pour but d'étudier les mutations introniques locales, qui se produisent à l'intérieur d'un gène. Pour le faire, il nous semble convenable d'étudier l'évolution du gène proprement dit,

donc en l'occurrence des protéines synthétisées par ce gène. L'étude de l'évolution des protéines nécessite de repérer les homologies entre les protéines, et d'en déduire les groupes de protéines homologues, qu'on appelle communément familles de protéines. Nous pourrons ensuite étudier l'évolution des introns qui sont présents dans chaque famille de protéines déterminée.

2.3.1.1 OrthoMCL

Pour construire les familles de protéines, nous avons utilisé OrthoMCL [24]. Il s'agit d'un ensemble de scripts écrits dans le langage de programmation Perl et qui doivent être exécutés dans une suite précise d'étapes, conjointement avec d'autres logiciels, pour construire les familles de protéines. Le site de téléchargement d'OrthoMCL fournit un guide d'utilisation détaillé pour l'exécution de ces étapes [18]. Les autres logiciels utilisés sont MySQL [14] (pour gérer une base de données), et BLAST [2] (pour comparer et aligner les protéines).

OrthoMCL analyse d'abord les protéines disponibles et retient uniquement les protéines qu'il juge de bonne qualité. Ses critères pour filtrer les protéines sont basés sur leurs longueurs (les protéines trop courtes sont ignorées) et sur le pourcentage de codons STOP qu'elles contiennent (les protéines contenant trop de codons stop prématurés sont ignorées). Le logiciel BLAST est ensuite exécuté pour comparer toutes les protéines entre elles et récolter les informations sur leurs similarités. OrthoMCL utilise enfin ces informations pour générer les familles de protéines. Il produit un fichier texte dans lequel chaque ligne décrit une famille en lui donnant un identifiant unique et en listant les identifiants des protéines de cette famille.

2.3.2 Application

2.3.2.1 Paramètres utilisés pour l'exécution d'OrthoMCL

Tous les scripts d'OrthoMCL ont été exécutés avec les valeurs par défaut chaque fois que c'était possible. La version de BLAST que nous avons utilisée est la version 2.2.29 qui fournit les programmes « BLAST**P** » (pour comparer les protéines) et « makeblastdb » (pour manipuler des bases de données compréhensibles pour BLAST). BLASTP a également été exécuté avec ses options par défaut, sauf pour le paramètre qui contrôle le format des fichiers de sortie, car

OrthoMCL requiert un format particulier pour bien fonctionner. Le paramètre « e-value » de BLASTP a donc lui aussi été laissé à sa valeur par défaut (10), bien que OrthoMCL recommande une valeur différente (10^{-5}). Puisque notre but est avant tout de développer une méthode, nous considérons que l'e-value est un paramètre global qui peut être ajustée si nécessaire à chaque exécution de la méthode. Nous avons donc travaillé avec les familles de protéines produites selon les valeurs par défaut de BLASTP.

2.3.2.2 Résultats obtenus

Notre jeu de données contenait un total de 162 566 protéines réparties dans nos 9 espèces. 162 564 protéines ont été retenues par OrthoMCL, et 2 protéines ont été ignorées : la protéine « HpaP802526 » de l'espèce *Hyaloperonospora arabidopsidis*, qui ne contient que 6 acides aminés, et la protéine « CCA28415 » de l'espèce *Albugo laibachii*, qui ne contient que 10 acides aminés.

BLASTP, au cours de son exécution, a ignoré 2 autres protéines suite à des erreurs qu'il a trouvées dans leurs séquences : la protéine « 109327 » de l'espèce *Phytophthora capsici* (200 acides aminés), et la protéine « 92918 » de l'espèce *Phytophthora cinnamomi* (109 acides aminés). BLASTP a recommandé de vérifier ces protéines ou les paramètres de filtrage des protéines de bonne qualité. Les protéines concernées ne contenaient aucun caractère atypique, si bien que ces erreurs nous semblent provenir des paramètres de filtrage d'OrthoMCL, ou d'erreurs que nous n'identifions pas encore. Nous avons cependant poursuivi le processus avec les protéines restantes, car les paramètres de filtrage d'OrthoMCL peuvent ici aussi être considérés comme des paramètres de la méthode.

Au total, OrthoMCL a généré 18 955 familles de protéines, contenant 134 154 protéines, soit 82,52 % des protéines de bonne qualité. On constate donc que 28 410 protéines de bonne qualité n'ont pas été classées par OrthoMCL dans des familles. Même si le nombre reste relativement faible par rapport au nombre total de protéines initialement disponibles, nous ne comprenons pas encore pourquoi autant de protéines n'ont pu être rangées dans des familles. Nous pensons que l'utilisation de valeurs appropriées pour les paramètres de filtrage des protéines et pour BLASTP

pourrait modifier la proportion de protéines classées et le nombre de familles déterminées, mais nous devrons exécuter à nouveau l'étape OrthoMCL pour déterminer comment les familles construites varieront en fonction des ajustements de ces paramètres.

2.3.2.3 Distributions des familles selon le nombre d'espèces et le nombre de séquences

Nous avons écrit un programme JAVA qui produit un ensemble de statistiques sur les familles de protéines obtenues, à partir desquelles nous pouvons collecter diverses informations, notamment par rapport au nombre de séquences et au nombre d'espèces dans les familles.

La figure 2.1 présente, pour chaque nombre de séquences de 0 à 40 compté dans les familles, le nombre de familles qui contiennent chacune ce nombre séquences parmi les 18 955 familles déterminées. Les nombres de séquences supérieurs à 40 sont groupés afin d'être tous affichés sur la figure. Ils vont de 41 à 618 séquences. On remarque notamment que 4 163 familles (soit 21,96 % des familles disponibles) contiennent seulement 2 séquences, tandis que 108 familles contiennent plus de 40 séquences, avec 1 famille contenant jusqu'à 618 séquences. Il est très probable que de nombreuses familles soient en fait des faux positifs, notamment celles qui contiennent de si grands nombres de séquences, et plusieurs parmi celles qui en contiennent seulement 2.

La figure 2.2 présente, pour chaque nombre d'espèces de notre jeu de données, le nombre de familles dans lesquelles apparaissent ce nombre d'espèces parmi les 18 955 familles déterminées. La figure décrit donc le nombre de familles dans lesquelles apparaissent un certain nombre d'espèces. On constate notamment qu'une seule espèce apparaît dans 3 948 familles (soit 20,83 % des familles disponibles), tandis que les 9 espèces étudiées apparaissent dans 3 604 familles (soit 19,01 % des familles disponibles). Puisque beaucoup de familles contiennent plus de 9 séquences (comme le montre la figure 2.1), donc plus de séquences que d'espèces étudiées, il semble donc que de nombreuses familles contiennent des protéines issues de gènes paralogues (c'est-à-dire des gènes issus de la duplication d'un gène ancestral). De plus, au moins 1 espèce n'apparaît pas dans 15 351 familles, soit 80,99 % des familles disponibles. Il semble donc y avoir eu beaucoup de disparitions de gènes dans les différentes espèces au cours de leur évolution.

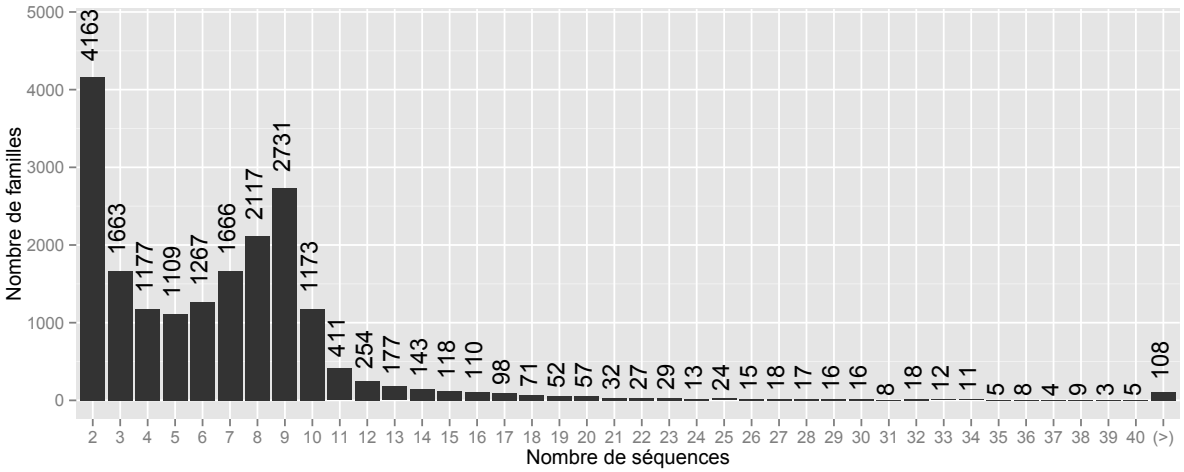


Figure 2.1 – Nombre de familles de protéines en fonction du nombre de séquences contenues dans les familles. « (>) » désigne les nombres de séquences trouvés supérieurs à 40, qui vont de 41 à 618 séquences.

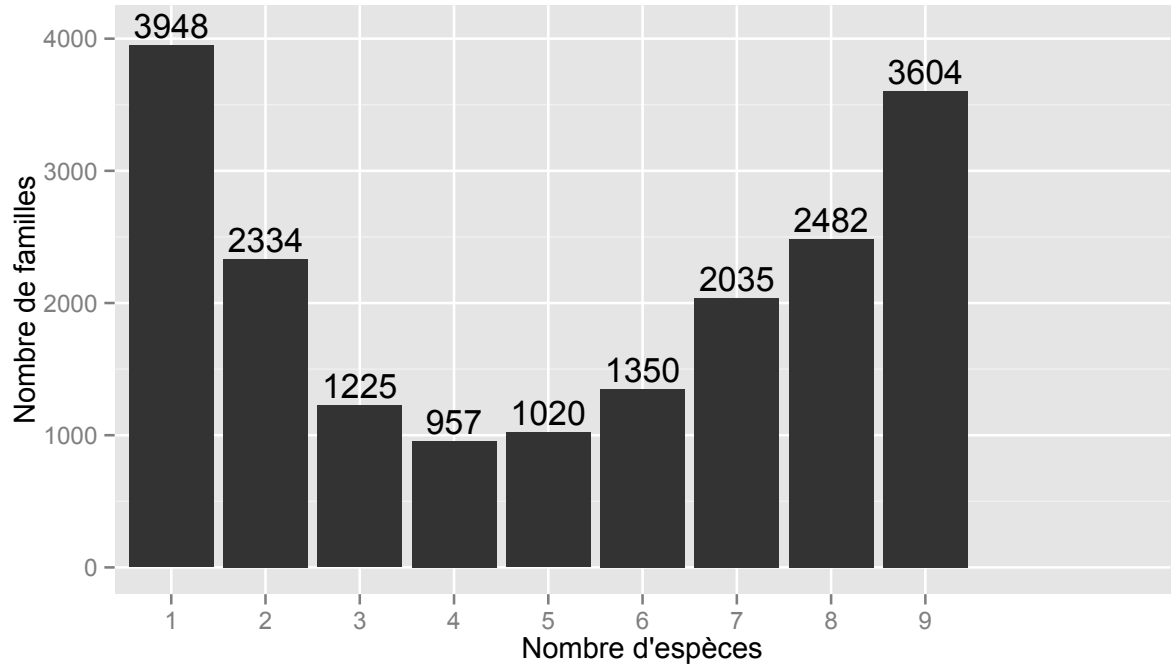


Figure 2.2 – Nombre de familles de protéines en fonction du nombre d'espèces qui apparaissent dans les familles.

2.3.2.4 Génération des fichiers FASTA des familles de protéines

OrthoMCL fournit une description des familles de protéines déterminées, en donnant pour chacune d'elles un identifiant et la liste des noms des protéines qu'elle contient. Nous n'avons donc pas directement les séquences des protéines, qui sont nécessaires à la poursuite de nos analyses.

Nous devons donc générer nous-même pour chaque famille un fichier qui rassemble les séquences de ses protéines. Pour le faire, nous avons écrit un programme JAVA capable de générer ces fichiers au format FASTA. Le programme prend en entrée le fichier de familles fourni par OrthoMCL, un dossier de fichiers contenant les séquences des protéines des espèces étudiées, et le nom d'un dossier de sortie qui contiendra les fichiers FASTA représentatifs des familles. Les fichiers FASTA des 18 955 familles de protéines ont ainsi été générés.

2.4 Alignement des familles de protéines

À partir des familles de protéines déterminées, nous devrons construire des arbres phylogénétiques, et identifier les homologies entre les introns qui apparaissent dans chaque famille. Pour réaliser ces étapes, nous devons d'abord aligner les familles de protéines, pour déterminer les homologies entre les acides aminés à l'intérieur de chaque famille.

Pour aligner les familles, nous avons utilisé le logiciel « MUSCLE » [15] avec l'option «`-maxiters 1000`» qui assure que le logiciel analyse suffisamment chaque famille afin de maximiser la qualité de l'alignement généré. Comme le nombre de familles est grand, et que certaines contiennent beaucoup de séquences, nous avons écrit un script PHP exécutable en ligne de commande, qui permet de lancer parallèlement plusieurs lots d'alignements. On peut donc exploiter plusieurs machines en même temps pour aligner tous les groupes en peu de temps, au lieu d'aligner les 18 955 groupes sur un seul ordinateur, ce qui prendrait beaucoup de temps.

2.5 Utilisation des familles de protéines strictement orthologues dans la suite de l'étude

2.5.1 Théorie

Pour étudier l'évolution des introns dans chaque famille de gènes, il faut disposer d'un arbre phylogénétique pour guider l'ordre d'analyse des changements observables. Deux types d'arbres phylogénétiques peuvent être obtenus à partir des données disponibles :

- Un unique arbre des espèces, qui décrit l'évolution des 9 espèces.
- Un arbre des protéines pour chaque famille, qui décrit l'évolution des protéines spécifiques de cette famille.

Pour une étude efficace et précise, il faudrait idéalement inférer les arbres des protéines des familles. Cependant, comme notre objectif est de développer et de tester une méthode, nous avons décidé de générer uniquement l'arbre des espèces, et de l'utiliser pour étudier spécifiquement les familles de protéines strictement orthologues, en faisant l'hypothèse que l'arbre des espèces serait identique à l'arbre des protéines pour ces familles. Nous appelons famille strictement orthologue une famille de protéines qui contient exactement 1 protéine pour chaque espèce du jeu de données, donc 9 protéines pour les 9 espèces que nous étudions. Dans une telle famille, toutes les espèces sont représentées et chaque espèce est associée à une seule protéine, ce qui permet d'utiliser l'arbre des espèces pour analyser les protéines exactement comme si on analysait les espèces.

L'évolution des protéines dans une famille strictement orthologue ne suit pas forcément le même schéma que l'évolution des espèces. En effet, certains phénomènes évolutifs, tels que l' « *incomplete lineage sorting* » (ILS) et le flux de gènes, peuvent influencer des familles de gènes spécifiques (et donc leurs protéines) et les faire évoluer différemment des espèces. Ces phénomènes sont notamment identifiés dans les génomes de certains primates, dont l'Homme, le Chimpanzé et le Gorille [43].

- L'ILS est un mécanisme de ségrégation de polymorphismes ancestraux, à la suite duquel l'arbre d'une famille de gènes peut être différent de l'arbre des espèces [43]. Lorsqu'une

espèce ancestrale possède plusieurs allèles d'un même gène, chacun de ses descendants peut retenir un seul allèle et éliminer les autres sous l'effet de la dérive génétique ou de la sélection naturelle. Si deux descendants normalement éloignés sélectionnent le même allèle ancestral, alors elles seront considérées comme proches du point de vue de cette famille de gènes. L'arbre des gènes de cette famille ne correspondrait donc pas à l'arbre des espèces auxquelles appartiennent ces gènes.

- Le flux de gènes est le transfert d'allèles ou de gènes entre populations [43]. Si la population d'une espèce reçoit un nouvel allèle pour un gène donné suite à une hybridation avec une population d'une autre espèce éloignée, alors la première population dispose de son allèle initial et du nouvel allèle obtenu. Au cours de l'évolution, le nouvel allèle peut devenir le plus fréquent et s'imposer jusqu'à la disparition de l'ancien allèle. Du point de vue de cette famille de gènes, les deux espèces seront considérées comme très proches (parce qu'elles ont le même allèle), alors qu'elles sont éloignées en réalité. À nouveau, l'arbre des gènes de cette famille ne correspondrait pas à l'arbre des espèces.

L'utilisation de l'arbre des espèces pour étudier les familles strictement orthologues n'est donc pas optimale et ne peut être définitive, mais elle est suffisante pour poursuivre la recherche et le développement des programmes et des algorithmes nécessaires à la méthode, que nous pourrons ensuite réutiliser facilement lorsque nous aurons les vrais arbres des protéines.

2.5.2 Application

Nous avons écrit un programme JAVA qui produit un rapport de statistiques sur les familles de protéines, à partir duquel on peut facilement identifier les familles strictement orthologues. 1 924 familles de protéines sont strictement orthologues sur les 18 955 familles disponibles.

2.6 Construction de l'arbre phylogénétique des espèces

2.6.1 Théorie

2.6.1.1 Sélection des familles à utiliser pour construire l'arbre

Pour construire l'arbre des espèces, il faut sélectionner des familles de protéines strictement orthologues adéquates et les utiliser avec un logiciel de phylogénie.

L'utilisation simultanée de toutes les familles strictement orthologues disponibles est peu recommandable car elles sont trop nombreuses, ce qui ralentirait énormément le logiciel de phylogénie. De plus, puisque les familles sont générées par OrthoMCL, elles ne sont pas encore toutes validées. Il est donc préférable d'opérer une sélection sur les familles, en retenant celles dont les alignements contiennent le plus possible de sites conservés ou similaires et le moins possible de sites instables ou de trous. Pour sélectionner les familles, nous utilisons le logiciel « GBlocks » [8] qui analyse les alignements de séquences.

GBlocks permet de filtrer un alignement pour n'en garder que les meilleures régions à utiliser dans les logiciels de phylogénie. En général il sélectionne des régions bien conservées mais avec quelques sites variables nécessaires pour différencier les espèces. Il prend en entrée des fichiers FASTA et génère en sortie des fichiers FASTA qui ont l'extension « fasta-gb » et qui ne contiennent que les régions qu'il a retenues dans les alignements.

Pendant l'exécution de GBlocks, deux informations ont été collectées pour chaque famille, afin d'aider à les sélectionner :

- Le pourcentage de positions retenues par GBlocks après filtrage.
- La longueur de l'alignement après filtrage.

En analysant ces informations, les familles sont sélectionnées selon les critères suivants :

- Les familles qui ont un pourcentage élevé de positions retenues après filtrage par GBlocks. Il s'agit donc des familles les mieux conservées du point de vue de GBlocks (celles dans lesquelles le filtrage a éliminé relativement peu de positions de l'alignement initial).
- Suffisamment de familles bien conservées pour que l'ensemble des alignements filtrés des familles retenues contienne assez de sites variables, donc informatifs pour les logiciels de phylogénie.

La sélection des familles à utiliser est automatisée grâce à un script PHP qui prend en entrée 3 paramètres :

- Le dossier contenant les alignements des familles.
- Le pourcentage minimal de positions retenues pour une famille qu'on considère comme bien conservée.
- La longueur minimale de l'ensemble des alignements filtrés des familles à sélectionner pour construire l'arbre.

Le script filtre tous les alignements avec GBlocks, puis produit un rapport dans lequel les familles sont triées par ordre décroissant du pourcentage de positions retenues puis par ordre décroissant de la longueur des alignements après filtrage, tout en comptabilisant, d'une famille à l'autre dans le rapport, le nombre total de sites dans les alignements filtrés de ces familles. On peut donc regarder le début du rapport et retenir uniquement les premières familles qui apparaissent, tant que leur pourcentage de position retenues est assez élevé (information indiquée dans le rapport) et/ou tant que la longueur totale de leurs alignements filtrés ne dépasse pas la longueur minimale demandée (information aussi indiquée dans le rapport).

GBlocks a été utilisé avec le paramètre « `Allowed Gap Positions: None` » qui lui indique de ne pas retenir les positions qui contiennent des trous, car les logiciels de phylogénie accordent généralement de l'importance aux changements entre acides aminés, mais ne savent pas toujours comment gérer les changements entre acides aminés et trous [49].

2.6.1.2 Concaténation des familles sélectionnées

Les alignements des familles sélectionnées contiennent chacune 1 séquence par espèce, donc 9 espèces en tout. Pour pouvoir travailler avec toutes ces familles en même temps, il faut obtenir pour chaque espèce toutes ses séquences provenant de ces alignements et mises bout à bout l'une à la suite de l'autre, en suivant toujours le même ordre pour toutes les espèces (par exemple l'ordre alphabétique des noms des familles). On obtient alors 9 séquences d'espèces qui, réunies dans un même fichier, donnent un alignement unique. C'est cet alignement qui est utilisé par les logiciels de phylogénie.

Cette étape est appelée concaténation des familles de protéines. Pour la réaliser, nous avons écrit un programme JAVA qui prend comme paramètres le dossier d'entrée contenant les familles sélectionnées et filtrées par GBlocks, et le dossier de sortie qui contiendra 9 fichiers FASTA.

Le programme génère donc finalement 9 fichiers FASTA, 1 pour chaque espèce. Nous les mettons ensuite ensemble dans un fichier FASTA unique pour obtenir l'alignement final.

2.6.1.3 Conversion du fichier concaténé au format PHYLIP

L'alignement concaténé obtenu précédemment doit être converti en un fichier au format PHYLIP pour qu'il soit utilisable par les logiciels de phylogénie. Nous convertissons le fichier à l'aide du logiciel « seaview » [22].

2.6.1.4 Construction proprement dite de l'arbre phylogénétique des espèces

La méthode que nous développons nécessite uniquement la topologie de l'arbre des espèces, et n'utilise aucune autre information à l'heure actuelle (donc pas besoin des longueurs des branches, par exemple). De ce fait, aucune méthode particulière de reconstruction d'arbre n'est imposée, seul un fichier décrivant l'arbre au format NEWICK [16] est requis pour que la méthode fonctionne. Cependant, nous recommandons d'utiliser les méthodes de reconstruction probabilistes telles que la méthode du maximum de vraisemblance, ou une méthode Bayésienne, car elles sont

considérées comme plus fiables [51] que les autres méthodes déterministes qui existent (telle que la méthode du maximum de parcimonie).

2.6.2 Application

Pour filtrer les familles de notre jeu de données, nous avons choisi comme paramètres un pourcentage de 80 % de positions retenues après filtrage par GBlocks, et 100 000 sites minimum au total dans l'ensemble des alignements filtrés des familles à sélectionner. Nous avons choisi 80 % afin d'avoir suffisamment de familles, et 100 000 sites (un nombre assez grand) en espérant que la concaténation finale contienne suffisamment de sites informatifs pour les logiciels de phylogénie. Avec ces paramètres, nous avons sélectionné 254 protéines qui contiennent au total 99 962 sites après filtrage.

Pour construire l'arbre des espèces, nous avons utilisé le logiciel « RAxML » [46] auquel nous sommes habitués, et qui se base sur la méthode du maximum de vraisemblance.

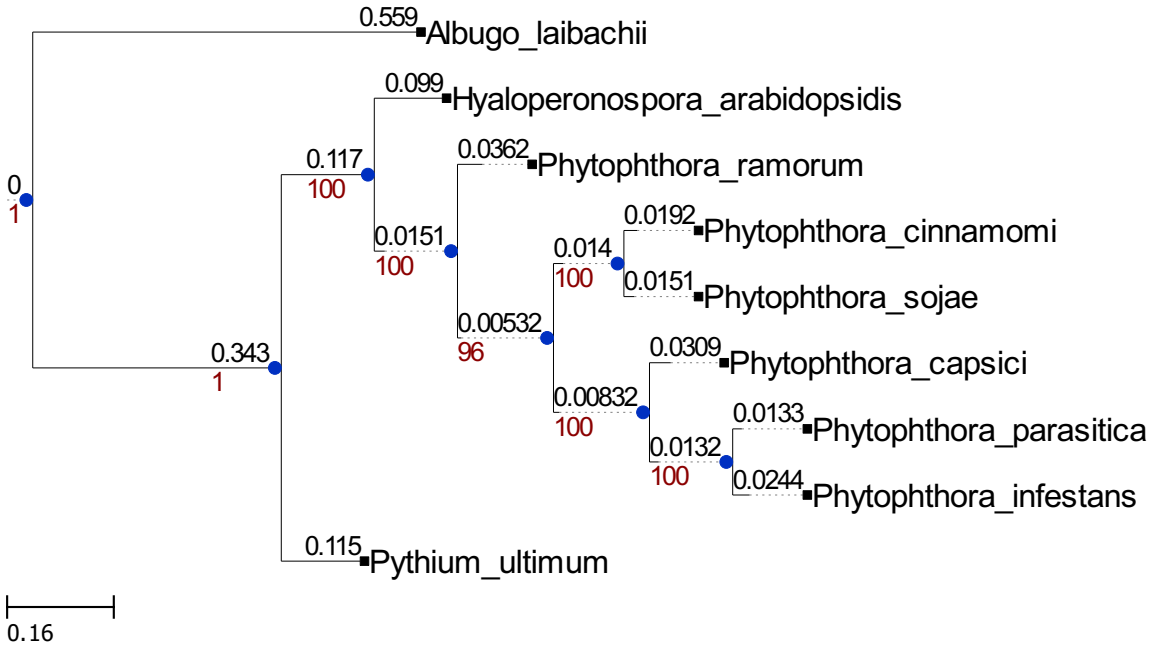
La construction d'un arbre phylogénétique par RAxML nécessite de choisir une matrice de substitution adaptée aux données, et censée décrire de la meilleure façon possible les modalités de mutation des acides aminés des protéines étudiées. Ce choix n'est cependant pas toujours évident à faire, car il est difficile de savoir à l'avance, pour un ensemble d'espèces données, comment les acides aminés changent au fil du temps. Afin de proposer dans notre méthode une utilisation automatisée de RAxML comme option par défaut d'inférence de l'arbre phylogénétique, nous avons exécuté RAxML une première fois avec comme consigne de chercher automatiquement la meilleure matrice de substitution à utiliser pour maximiser la vraisemblance de l'arbre généré. À des fins de comparaison, nous avons également exécuté une deuxième fois RAxML avec une matrice précise, la matrice WAG, comme spécifié dans un article publié en 2012 et qui étudiait lui aussi les oomycètes [45]. Dans les deux cas, nous avons indiqué à RAxML d'effectuer le test de robustesse des branches (« *bootstrap* ») aussi longtemps que nécessaire, jusqu'à convergence des résultats du test. Le tableau 2.II présente les paramètres utilisés pour les deux exécutions de RAxML.

Tableau 2.II – Paramètre d'exécution de RAxML pour l'inférence de l'arbre phylogénétique des espèces étudiées.

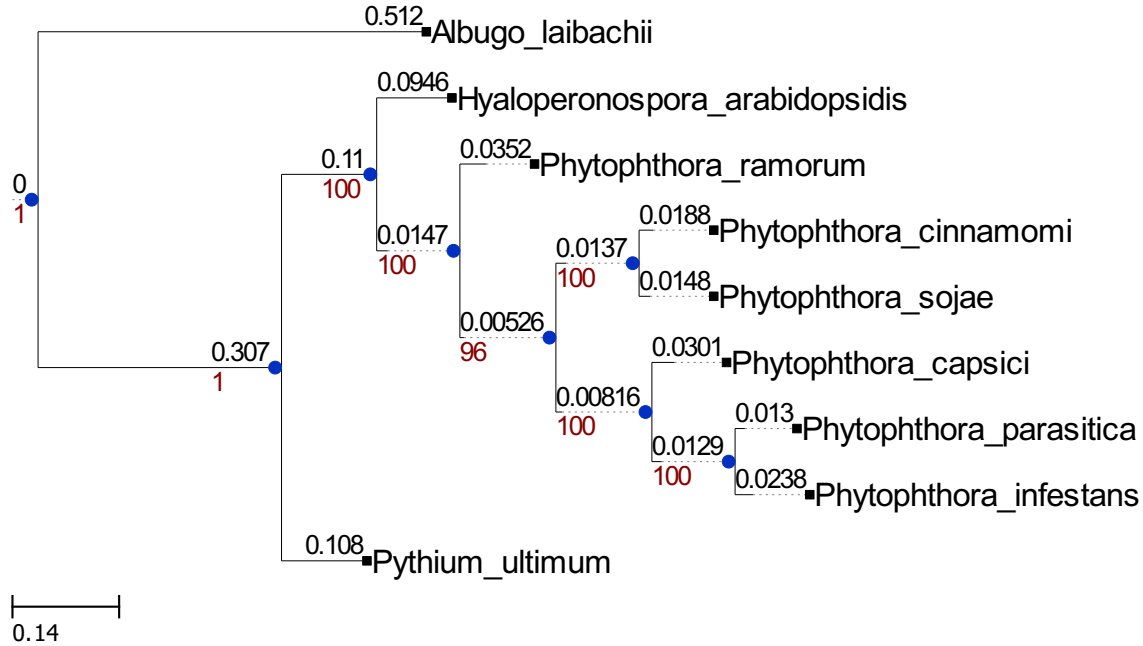
Exécution	Première	Deuxième
Matrice utilisée	Automatique	WAG
Nombre de tests de robustesse	Automatique (jusqu'à convergence)	Automatique (jusqu'à convergence)
Syntaxe complète d'exécution de RAxML	`raxmlHPC-SSE3 -f a -x 12 -k -m PROTGAMMAAUTO -p 181 -N autoMR -o albu -s alignement.phy -n alignement.tre -O`	`raxmlHPC-SSE3 -f a -x 12 -k -m PROTGAMMAWAGF -p 181 -N autoMR -o albu -s alignement.phy -n alignement.tre -O`

RAxML a détecté 15 164 « *distinct patterns* » dans la concaténation finale des 254 protéines utilisées. Les 99 962 sites disponibles contenaient donc un peu moins de 15 164 sites informatifs. La première exécution de RAxML a choisi la matrice LG comme meilleure matrice pour maximiser la vraisemblance de l'arbre des espèces. Les deux arbres inférés sont donc basés sur deux matrices différentes (LG et WAG). Pourtant, bien qu'ayant des longueurs de branches différentes, ils ont exactement la même topologie, ce qui nous rassure sur la validité de cette dernière. La figure 2.3 présente les deux arbres obtenus. Dans les deux cas, RAxML a exécuté seulement 50 tests de robustesse.

La figure 2.4 présente l'arbre fourni par notre article de référence [45], qui a utilisé 189 familles de protéines et a spécifié 1000 tests de robustesse. Nous constatons que sa topologie est différente de celle de nos arbres. En effet, dans l'arbre de l'article, les espèces *Phytophthora sojae* et *Phytophthora ramorum* ont un ancêtre commun plus récent que l'ancêtre commun à ces deux espèces et à *Phytophthora infestans*. Mais dans nos arbres, *Phytophthora sojae* et *Phytophthora infestans* ont un ancêtre commun plus récent que l'ancêtre commun à ces deux espèces et à *Phytophthora ramorum*.



(a) Arbre obtenu avec la première exécution de RAxML (matrice choisie automatiquement).



(b) Arbre obtenu avec la deuxième exécution de RAxML (matrice WAG imposée).

Figure 2.3 – Arbres phylogénétiques obtenus pour la première et la seconde exécution de RAxML. La longueur et la robustesse de chaque branche est affichée respectivement au dessus et en dessous de la branche. La topologie des deux arbres est identique, de même que les robustesses des branches, mais des longueurs de branches sont différentes.

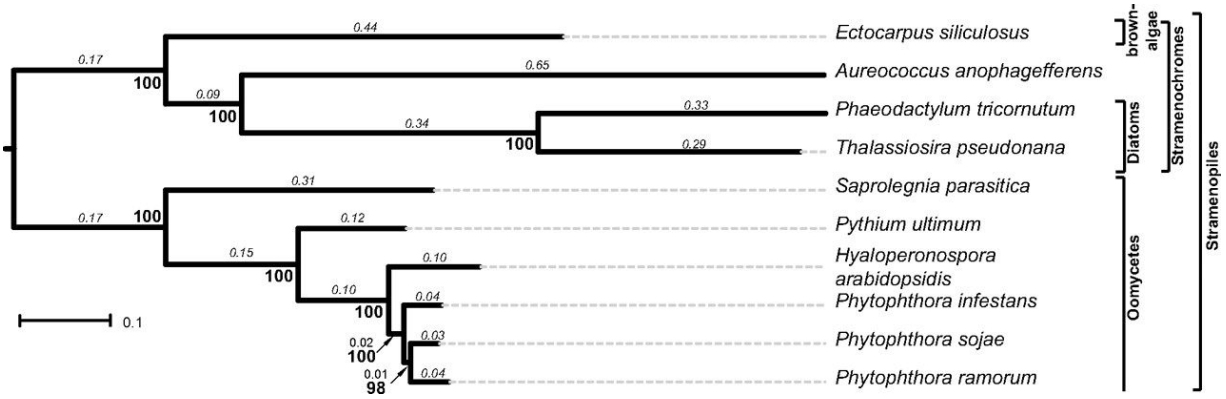


Figure 2.4 – Arbre phylogénétiue d'oomycètes publié en 2012 dans la littérature [45].

Nous ne comprenons pas encore pourquoi nous observons cette différence. Cependant, plusieurs remarques peuvent être faites :

- Les paramètres d'inférence des arbres sont différents entre les deux études. En effet, nous avons utilisé plus de familles de protéines (254 contre 189), mais nos exécutions de RAxML ont effectué beaucoup moins de tests de robustesse (50 contre 1000).

- Les branches sont très courtes dans le sous-arbre du genre *Phytophthora*, aussi bien dans nos arbres que dans l'arbre de la littérature, ce qui suggère que les espèces du genre *Phytophthora* sont très proches entre elles, et difficiles à distinguer lorsqu'on observe leurs protéines. Les relations de parenté entre les espèces du genre *Phytophthora* peuvent donc légèrement varier, comme c'est le cas ici, en fonction des protéines utilisées pour construire l'arbre des espèces.

- Comme nous l'avons vu, des phénomènes évolutifs tels que *l'incomplete lineage sorting* et le flux de gènes peuvent faire en sorte que certaines familles de protéines n'évoluent pas de la même façon que les espèces. Si de telles familles sont utilisées pendant la reconstruction de l'arbre des espèces, elles risquent donc de fausser l'inférence de l'arbre.

Nous avons tout de même poursuivi l'étude avec la topologie que nous avons inférée, car c'était la seule à notre disposition pour l'ensemble des 9 espèces que nous étudions.

Nous avons donc les alignements des familles de protéines strictement orthologues et une topologie pour l'arbre des espèces. Pour poursuivre l'étude, nous devons maintenant réunir les informations relatives aux introns eux-mêmes.

2.7 Extraction des positions des introns à partir des annotations des génomes

2.7.1 Théorie

2.7.1.1 Utilisation des fichiers d'annotations des génomes

Pour pouvoir étudier l'évolution des introns, nous devons connaître leurs positions dans les séquences, et être en mesure de les repérer dans les gènes pour pouvoir si nécessaire analyser leurs propres séquences. L'ensemble des informations relatives aux introns peut être obtenue à partir des fichiers d'annotations des génomes.

Les fichiers d'annotation sont des fichiers texte, dans un format particulier, qui décrivent la nature de plusieurs éléments constitutifs du gène et leurs localisations sur l'ADN. Ces fichiers ne décrivent généralement pas les introns, mais ils mettent notamment en évidence 2 catégories de séquences géniques : les séquences qui sont maintenues dans l'ARN messager mature après transcription et épissage, et les séquences qui codent effectivement pour les produits du gène. Ce sont les premières séquences qui sont appelées « exons » dans les fichiers d'annotation, tandis que les secondes sont simplement appelées « séquences codantes ». L'ARN messager mature peut en effet contenir des séquences non-codantes (telles que les régions 5'UTR et 3'UTR du gène), mais qui ne sont pas enlevées pendant l'épissage, et qui ne sont donc pas des introns. Ces séquences sont considérées comme des « exons » dans les fichiers d'annotation.

Si un gène code pour plusieurs produits, le fichier d'annotation décrit les exons de l'ARN messager mature et les séquences codantes pour chacun des produits. Il est donc toujours possible d'étudier chaque produit du gène individuellement.

Les exons de l'ARN messager mature et les séquences codantes d'un produit d'un gène sont séparés par les introns (si le gène en contient) sur l'ADN. En connaissant les positions et les

longueurs de ces séquences, qui sont fournies dans les fichiers d'annotation, nous pouvons donc en déduire les positions et les longueurs des introns qui séparent ces séquences sur l'ADN. Ce sont ces informations que nous recherchons sur les introns afin de les étudier.

Comme nous travaillons avec des protéines, ce sont les séquences codantes qui nous intéressent, car elles seules ont une correspondance exacte avec les protéines (3 nucléotides de la séquence codante pour 1 acide aminé de la protéine). Nous récupérons tout de même les coordonnées des introns relatives aux exons, car elles peuvent permettre d'obtenir rapidement si nécessaire les séquences proprement dites des introns à partir des séquences complètes des gènes.

2.7.1.2 Gestion des annotations des introns dans les familles strictement orthologues en cas d'épissage alternatif

Nous comptons extraire les positions et les longueurs des introns à partir des séquences codantes des produits des gènes, en considérant que les introns sont les parties de l'ADN qui séparent les séquences codantes consécutives pour un produit donné d'un gène. Il faut cependant noter que l'épissage alternatif peut compliquer cette stratégie. Par exemple, considérons un gène qui comporte 2 introns, et 3 séquences codantes nommées « CDS 1 », « CDS 2 » et « CDS 3 ». On peut décrire le gène comme une succession de 5 séquences comme suit :

```
(CDS 1)(intron 1)(CDS 2)(intron 2)(CDS 3)
```

Supposons que ce gène code pour 2 protéines : la protéine 1 dont la séquence codante complète est (CDS 1)(CDS 2)(CDS 3), et la protéine 2 dont la séquence codante complète est (CDS 1)(CDS 3). Dans cet exemple, la protéine 2 est synthétisée à partir d'un ARN messager qui subit un épissage alternatif au cours duquel l'exon correspondant à CDS 2 est ignoré. Donc, du point de vue de la protéine 2, la « partie » de l'ADN qui sépare ses séquences codantes (CDS 1 et CDS 3) est :

```
(intron 1)(CDS 2)(intron 2)
```

C'est cette partie qui sera considérée comme un « intron » pour la protéine 2. Or nous remarquons que cette portion de séquence inclut « CDS 2 », qui est une séquence codante de la protéine 1, qui n'en est pas une pour la protéine 2, mais qui n'a pas forcément la structure typique d'un intron splicéosomal. Faut-il donc considérer CDS 2 en tant que séquence intronique pour la protéine 2 ?

À l'heure actuelle, nous ne prenons pas en compte ce cas, car nous travaillons exclusivement avec des familles de protéines strictement orthologues, et qu'il est peu probable que ce problème apparaisse dans ce type de famille. En effet, si un gène code pour plusieurs protéines, alors ces protéines auraient plusieurs domaines homologues, et seraient donc probablement rangées dans une même famille. Cette famille aurait donc plusieurs protéines pour un même gène et donc pour une même espèce, et ne serait donc pas strictement orthologue puisqu'il n'y aurait pas exactement 1 protéine pour chaque espèce. Dans l'exemple donné ci-dessus, les protéines 1 et 2 ont deux parties communes : les parties correspondants à CDS 1 et à CDS 3. L'alignement des deux protéines contiendrait donc un trou au milieu (correspondant à CDS 2), mais le début et la fin de l'alignement seraient très conservés, et les deux protéines seraient donc probablement dans la même famille, qui ne serait pas strictement orthologue.

Il est donc peu probable que le problème posé par l'épissage alternatif affecte les familles strictement orthologues que nous allons étudier. Cependant, nous devrons certainement trouver une stratégie pour gérer cette situation lorsque notre méthode devra prendre en compte les familles contenant plusieurs séquences par espèce.

2.7.1.3 Programmes d'extraction et format de sortie

Il existe plusieurs formats de fichiers d'annotations, tels que le format GTF [28] et le format GFF3 [35], et la syntaxe d'un fichier peut varier d'une base de données génomique à une autre. Pour pouvoir analyser convenablement tous les fichiers d'annotations fournis par les bases de données que nous utilisons, nous avons écrit 5 programmes JAVA qui se répartissent comme suit :

- 1 programme pour la base de données Broad Institute, dont les fichiers d'annotations sont

au format GTF.

- 1 programme pour le format GTF et 1 programme pour le format GFF3 de la base de données Ensembl Protists.
- 1 programme pour le format GTF et 1 programme pour le format GFF de JGI.

Chaque programme prend en entrée un fichier d'annotation pour un génome et génère en sortie un fichier d'annotations pour les introns. Le fichier de sortie a pour extension « .introns » et décrit sur chaque ligne l'annotation des introns d'un produit de gènes. Chaque produit de gènes est représenté par deux lignes : une qui décrit les introns dans sa séquence codante (par rapport aux portions codantes), et une autre qui décrit les introns par rapport aux exons. Chaque ligne est composée de plusieurs colonnes séparées par des tabulations. Les colonnes présentent les informations suivantes, dans le même ordre :

- Le type d'annotation : soit « transcript » (par rapport aux exons), soit « CDS » (par rapport à la séquence codante). Le type d'annotation indique si les coordonnées des introns sont données par rapport aux exons ou par rapport à la séquence codante. Les coordonnées ne sont pas les mêmes dans les deux cas, car les séquences des exons peuvent inclure des régions situées en amont et en aval de la séquence codante (régions 5'UTR et régions 3'UTR).
- Le type de séquence génomique : « gene », « pseudogene » ou « pseudogenic_region ». Cette information est disponible seulement dans les fichiers GFF3 d'Ensembl Protists, donc la colonne est vide pour les autres fichiers.
- L'ID de la séquence génomique.
- L'ID du transcrit.
- L'ID de la protéine (si disponible).
- Le brin d'ADN (« + » ou « -») sur lequel se trouve le gène.

- Les positions des introns, séparées par des virgules. La position d'un intron est le nombre de nucléotides situés avant lui dans la séquence dans laquelle il est localisé (donc dans l'ARN messager mature, ou dans la séquence codante).

- Les longueurs des introns (en nucléotides), séparées par des virgules.

2.7.2 Application

Certaines bases de données fournissent des annotations dans plusieurs formats (par exemple, Ensembl Protists fournit des annotations au format GFF3 et au format GTF), d'autres fournissent des annotations dans un format similaire au format GTF, mais avec une syntaxe légèrement variable (par exemple, les fichiers d'annotations fournis par JGI pour nos données ont pour extension « GFF »). Le tableau 2.I indique pour chaque génome quel fichier d'annotation a été utilisé. Les informations relatives aux introns ont été extraites pour toutes les espèces.

2.7.2.1 Analyse statistique des longueurs des introns

Pour avoir une vue d'ensemble sur les annotations d'introns extraites, nous avons écrit un programme JAVA qui peut faire une analyse statistique des fichiers d'annotations générés. Le programme prend en entrée les fichiers « .introns » et génère en sortie un tableau contenant plusieurs informations sur les longueurs et la distribution des introns dans les espèces étudiées. Le tableau 2.III fournit les informations rassemblées. On remarque que la longueur moyenne des introns est comprise entre 84 et 189 nucléotides. Cependant, les longueurs des introns varient de seulement 4 nucléotides jusqu'à 48 310 nucléotides pour le plus long intron de l'espèce *Phytophthora capsici*. Les introns les plus longs contiennent donc tellement de nucléotides qu'on peut considérer que leur influence sur les moyennes calculées est significative. Cela est notamment suggéré par les longueurs des introns les plus rencontrés dans ces espèces, qui oscillent entre 56 et 82 nucléotides, et qui sont donc plus petites que la plus petite longueur moyenne calculée. En nous appuyant sur ces observations, nous pensons que la longueur typique d'un intron dans les oomycètes que nous étudions se rapproche surtout des longueurs les plus rencontrées, donc entre 50 et 100 nucléotides environ. Les introns les plus courts (tels que ceux de 4 à 10 nucléotides)

et les introns les plus longs (plus de 1000 nucléotides) pourraient être des faux positifs détectés pendant l'annotation des génomes.

Notre méthode dans sa version actuelle ne se sert pas des informations relatives aux longueurs des introns, mais il se peut que ces observations soient utiles dans de futures améliorations de notre méthode, afin par exemple de repérer plus facilement des introns mal annotés ou suspects car trop longs ou trop courts.

2.8 Génération de fichiers FASTA personnalisés rassemblant les alignements des familles et les positions des introns

Pour faciliter la suite de l'étude, nous réunissons les données sur les protéines et les introns dans de nouveaux fichiers. Pour chaque famille de protéines, nous générons un fichier FASTA personnalisé qui contient l'alignement de la famille, et dans lequel l'entête de chaque protéine contient les positions et les longueurs des introns associés à cette protéine. Les positions des introns sont définies par rapport à la séquence codante de la protéine, comme dans les fichiers « .introns » précédemment générés.

Ces fichiers ont l'extension « *.aligned.marked.fasta* », on les appelle fichiers AMF. Pour les générer, nous avons écrit un programme JAVA qui prend en entrée le dossier contenant les annotations des introns, le dossier contenant les alignements des familles, et le dossier de sortie dans lequel seront sauvegardés les fichiers AMF.

Les fichiers AMF ainsi générés et l'arbre phylogénétique des espèces constituent les données finales formatées que nous utilisons pour étudier l'évolution des introns.

Tableau 2.III – Statistiques sur les longueurs des introns.

Espèce	**Nombre d'introns**	**Plus petite longueur**	**Plus grande longueur**	**Longueur moyenne**	**Longueur majoritaire (nombre d'occurrences)**
albu	29 204	4	29 413	189,1699	56 (1 780)
hyal	13 844	4	1 797	137,5905	77 (240)
phca	22 865	10	48 310	172,8915	63 (765)
phci	27 930	1	26 680	166,2571	72 (792)
phin	31 219	10	1 074	123,3611	66 (926)
phpa	24 335	21	4 193	84,27088	65 (1 089)
phra	24 735	11	28 977	131,1243	67 (647)
phso	34 141	11	31 818	128,8582	71 (713)
pyul	24 609	4	9 888	114,6851	82 (669)

CHAPITRE 3

DÉTECTION DES MUTATIONS INTRONIQUES LOCALES

Notre but est d'étudier l'évolution des introns par mutations introniques locales dans les familles de protéines strictement orthologues. Pour ce faire, nous avons à notre disposition :

- Les alignements des familles de protéines strictement orthologues.
- Les positions des introns sur les séquences codantes des protéines.
- Une topologie pour l'arbre des espèces, qu'on utilisera donc comme topologie pour les protéines dans chaque famille strictement orthologue.

Lorsqu'on souhaite étudier l'évolution d'un caractère donné, on cherche à déterminer comment ce caractère a changé au cours du temps. Une manière simple de le faire est de comparer une situation initiale à une situation finale impliquant toutes les deux ce caractère. La situation initiale est habituellement présente chez une espèce ancestrale, et la situation finale est observable chez une espèce actuelle.

En l'occurrence, nous étudions les introns. Nous pouvons donc faire les analogies suivantes :

- Les introns sont des caractères particuliers ayant une longueur et dont les frontières (sites d'épissage) peuvent changer au fil du temps.
- Les séquences dans lesquelles se trouvent les introns sont les situations dans lesquelles ils sont impliqués.

Notre stratégie consiste donc à étudier l'évolution des introns en comparant les séquences ancestrales aux séquences descendantes dans lesquelles ils se trouvent. Nous faisons l'hypothèse que l'alignement d'une séquence descendante avec sa séquence ancestrale, dans lesquelles les introns sont mis en évidence, permettrait de déterminer les changements qui se sont produits au niveau des introns, ainsi que la direction (apparition ou disparition, insertion ou substitution, augmentation ou diminution) de ces changements.

Notre approche impose certaines contraintes :

- Nous travaillons avec des protéines, alors que les introns existent dans des séquences nucléotidiques. Il faut donc projeter les introns sur les protéines afin de pouvoir les étudier dans ce type de séquences.
- Nous disposons des séquences des espèces actuelles, mais nous n'avons aucune information sur les séquences des espèces ancestrales. Il faut donc les déterminer.
- Nous voulons comparer les séquences descendantes à leurs séquences ancestrales pour en déduire les changements au niveau des introns. Il faut donc que les introns soient mis en évidence pas seulement dans les séquences descendantes, mais aussi dans les séquences ancestrales.

Pour mettre en œuvre notre approche, nous avons développé un algorithme de reconstruction des séquences ancestrales, qui tient compte de tous ces problèmes à résoudre.

3.1 Reconstruction et alignement des séquences ancestrales

La poursuite de l'étude nécessite la reconstruction des séquences ancestrales dans chaque famille de protéines strictement orthologue. Pour ce faire, nous avons développé un algorithme de reconstruction qui tient compte des contraintes précédemment identifiées. Les introns sont d'abord projetés sur les protéines des espèces actuelles, comme l'impose la première contrainte. Nous obtenons ainsi des séquences protéiques dans lesquelles les introns sont directement positionnés. Ces séquences sont ensuite utilisées pour reconstruire les séquences ancestrales dans chaque famille de protéines. Les ancêtres désignent ici la racine et les nœuds internes de l'arbre, tandis que ses feuilles correspondent aux espèces actuelles. Nous reconstruisons les séquences en suivant une approche simple qui passe par 3 étapes :

- Déterminer les homologies entre les caractères des séquences actuelles, et en déduire les groupes de caractères homologues à travers les séquences actuelles. Dès cette étape, nous tenons compte à la fois des acides aminés et des introns dans les caractères analysés, ce qui nous permet de gérer les deuxième et troisième contraintes avec une procédure commune.

- Déterminer les caractères ancestraux dans chaque groupe de caractères homologues en étant guidé par la topologie de l'arbre des espèces.

- Assembler les caractères ancestraux reconstruits pour chaque ancêtre pour en déduire les séquences ancestrales recherchées, et vérifier les séquences reconstruites.

La projection des introns sur les protéines et les 3 étapes de reconstruction sont décrites dans les sections suivantes.

3.1.1 Projection des introns sur les protéines

Nous connaissons pour chaque protéine les positions des introns dans sa séquence codante. La séquence codante d'une protéine est une succession de codons qui contiennent chacun 3 nucléotides et qui correspondent aux acides aminés de la protéine. La longueur de la séquence codante est donc égale au triple de la longueur de la protéine. En sachant cela, on peut projeter les introns de la séquence codante sur la protéine en respectant des règles précises. En effet, pour tout nombre entier naturel non nul n :

- Un intron situé après 3n-2 nucléotides dans la séquence codante est situé après le 1er nucléotide du codon du n-ième acide aminé de la protéine. Il s'agit d'un intron en phase 1.

- Un intron situé après 3n-1 nucléotides dans la séquence codante est situé après le 2ème nucléotide du codon du n-ième acide aminé de la protéine. Il s'agit d'un intron en phase 2.

- Un intron situé après 3n nucléotides dans la séquence codante est situé après n acides aminés dans la protéine. Il s'agit d'un intron en phase 3 (ou en phase 0, selon l'appellation choisie).

Nous pouvons donc ainsi positionner les introns sur une protéine, en considérant si nécessaire les acides aminés comme des codons s'il faut placer des introns en phase 1 ou en phase 2.

3.1.2 Détermination des homologies entre caractères

3.1.2.1 Stratégie utilisée

Pour déterminer les homologies, il faut aligner les séquences. Par hypothèse, tous les caractères dans une colonne d'un alignement sont considérés comme homologues, et forment donc un groupe de caractères homologues.

Nous disposons déjà des alignements des acides aminés des familles que nous étudions, qui nous fournissent donc les homologies entre les acides aminés des protéines. Cependant, ces alignements ne tiennent pas compte des introns, qui sont les caractères qui constituent le sujet d'étude de notre projet. Pour trouver les homologies entre les introns, nous procédons en plusieurs étapes.

Premièrement, nous projetons les introns sur leurs protéines, selon les règles indiquées précédemment.

Ensuite, nous recalculons les positions des introns sur leurs protéines alignées, afin de déterminer où se trouvent ces introns dans les alignements d'acides aminés déjà disponibles. La position d'un intron dans un alignement désigne le nombre de colonnes qui le précèdent dans cet alignement.

Enfin, nous mettons en évidence les introns dans les alignements. Pour ce faire, chaque intron est placé dans une nouvelle colonne dans son alignement. Tous les introns qui ont la même position partagent la même colonne. Pour positionner les introns en phase 1 ou en phase 2, nous utilisons une représentation spécifique. D'abord, tous les acides aminés sont par défaut écrits en majuscule. Ensuite, si un intron est situé dans un acide aminé « X » (donc en phase 1 ou en phase 2), alors cet acide aminé, ainsi que tous les autres acides aminés qui sont dans la même colonne que lui, sont convertis en codons représentés par 3 lettres minuscules « xxx », et la colonne contenant l'intron est alors insérée entre les nucléotides du codon. Ainsi, tous les acides aminés « X » dans une colonne contenant des introns en phase 1 ou en phase 2 peuvent être convertis en :

- « `x?xx` » si cette colonne d'acides aminés contient seulement des introns en phase 1.
- « `xx?x` » si cette colonne d'acides aminés contient seulement des introns en phase 2.
- « `x?x?x` » si cette colonne d'acides aminés contient des introns en phase 1 et en phase 2.

« ? » désignant la position à laquelle les introns sont insérés. Les introns sont concrètement représentés par le symbole « + » (présence d'un intron) dans leurs colonnes. Si une colonne d'introns n'en contient pas dans certaines séquences, alors nous mettons le symbole « . » (absence d'un intron) dans la colonne pour les séquences concernées. La figure 3.1 présente un extrait d'un alignement dans lequel des introns en phase 2 sont mis en évidence, sans que des introns soient présents dans toutes les espèces étudiées.

Dans ces alignements modifiés, nous considérons que tous les introns placés dans une même colonne sont homologues. Nous étendons l'hypothèse en considérant également que des introns situés dans des colonnes proches peuvent aussi être homologues, même s'ils ne sont pas dans la même colonne. Cette extension n'est pas importante dans notre stratégie de reconstruction, mais elle sera prise en compte pendant la détection proprement dite des changements des sites d'épissage.

La stratégie ainsi déployée nous offre certaines garanties :

```
1 albu    EHrr+rLEQE
2 phra    ERrr.rLEQE
3 phso    ERrr.rLEQE
4 phci    ERrr.rLEQE
5 phin    ERrr.rLEQE
6 phpa    ERrr.rLEQE
7 phca    ERrr.rLEQE
8 hyal    ERrr.rLEQE
9 pyul    ERrr+rLEQE
```

Figure 3.1 – Exemple d'alignement de protéines avec mise en évidence des introns en phase 2 dans un codon.

- Les séquences ancestrales obtenues seront déjà alignées entre elles et avec les séquences descendantes. En effet, chaque colonne d'un alignement fournit 1 caractère pour chaque séquence actuelle et fournira 1 caractère pour chaque séquence ancestrale, et tous ces caractères sont dans la même colonne, donc déjà alignés.

- Les séquences ancestrales contiendront déjà des introns positionnés. Notons cependant que seules des positions d'introns sont ainsi repérées dans les séquences ancestrales, sans la moindre information ni sur les longueurs ni sur les séquences de leurs introns.

3.1.2.2 Pertinence de la stratégie

Cette approche pour déterminer les homologies entre les introns n'est pas parfaite. En effet, les introns sont seulement positionnés sur les alignements des protéines, sans être concrètement alignés (ni entre eux ni avec les acides aminés autour d'eux) par un algorithme spécifique. De plus, les séquences des introns ne sont jamais évaluées dans ce processus, donc il est possible que deux introns qui se retrouvent dans une même colonne soient en réalité très différents (par exemple, si l'un d'en entre eux a évolué via une mutation intronique globale).

Nous avons cependant choisi cette stratégie pour des raisons techniques. En effet, nous n'avons pas trouvé dans la littérature un algorithme capable d'aligner simultanément les protéines et les introns tout en répondant à nos besoins. Une étude antérieure a déjà proposé un algorithme d'alignement qui tient compte des introns [13], mais seuls les acides aminés des protéines sont réalignés par cette méthode, sans afficher directement les introns dans les séquences, et sans mettre en évidence les codons. Ainsi, les éventuels trous à placer autour d'introns en phase 1 ou en phase 2 ne sont pas montrés. Il n'y a donc pas d'algorithme convenable disponible, et il s'est avéré très difficile de développer un tel algorithme dans le temps imparti.

De plus, la comparaison directe des séquences des introns n'aurait pas forcément apporté beaucoup plus d'informations. Les éléments de structure des introns (tels que le point de branchement et les sites d'épissage) sont communs à tous les introns et sont pour la plupart très courts, donc les comparer ne permettrait pas de différencier les introns correctement. Les séquences

autour de ces éléments de structure ne sont quant à elles pas soumises à priori à des contraintes évolutives majeures, en dehors de l'éventuel besoin de ne pas interférer avec les éléments de structure de l'intron pour qu'il puisse continuer à être épissé facilement. Ce n'est donc pas forcément une bonne idée d'intégrer directement la comparaison des séquences introniques dans notre méthode, car cette dernière se veut globale et réutilisable, ce qui serait difficile dans des situations où les introns n'ont effectivement aucune contrainte d'évolution. Il est préférable de comparer les séquences introniques lorsqu'on fait l'hypothèse qu'elles peuvent être soumises à la sélection. Enfin, nous faisons l'hypothèse que les introns homologues seraient ceux que nous positionnons dans la même colonne, ou ceux qui sont positionnés dans des colonnes proches. Cette hypothèse, combinée à l'hypothèse globale de notre méthode selon laquelle les séquences comparées sont proches, devrait nous permettre de capturer suffisamment d'homologies d'introns, même si certains faux positifs ou vrais négatifs sont encore possibles.

La recherche des homologies entre les introns constitue dans tous les cas un point crucial de notre méthode, et un point encore certainement perfectible, que nous nous attèlerons à améliorer dans les prochaines versions de notre méthode.

3.1.3 Détermination des caractères ancestraux dans chaque groupe de caractères homologues

Pour reconstituer les séquences ancestrales, il suffit désormais de déterminer les caractères ancestraux dans chaque colonne des alignements formatés dans lesquels les introns sont mis en évidence, puis d'assembler les caractères déduits pour chaque séquence ancestrale en suivant les ordres des colonnes dans chaque alignement.

Pour déterminer les caractères ancestraux, nous utilisons l'algorithme de Fitch [19] avec une modification qui permet de toujours choisir un caractère ancestral de façon précise, en évitant le choix par hasard lorsqu'il n'est pas possible de le choisir via l'algorithme.

Nous travaillons avec des alignements, donc des acides aminés ou des introns ne sont pas forcément présents dans toutes les séquences dans toutes les colonnes. L'algorithme de Fitch

fonctionne cependant en dépit de ces absences en les considérant comme des caractères. Ainsi, on peut tout à fait choisir un caractère « . » (intron absent) ou « - » (acide aminé absent) dans une colonne pour une séquence ancestrale.

3.1.3.1 Algorithme de Fitch adapté

L'algorithme de Fitch est exécuté pour chaque colonne de l'alignement, en plaçant les caractères de la colonne sur les feuilles de l'arbre des espèces. Le caractère de la séquence d'une espèce est positionné à la feuille correspondant à cette espèce dans l'arbre.

L'algorithme de Fitch s'exécute en 2 étapes : une étape dite ascendante, qui permet de déterminer l'ensemble des caractères possibles pour chaque ancêtre, et une étape dite descendante, qui permet d'attribuer un caractère unique à chaque ancêtre. C'est la deuxième étape que nous avons modifiée.

3.1.3.2 L'étape ascendante de l'algorithme de Fitch

Dans cette étape, on détermine l'ensemble des caractères possibles pour chaque nœud de l'arbre en analysant les ensembles de caractères de ses nœuds enfants.

Soit I et U respectivement l'intersection et l'union des ensembles de caractères possibles des enfants du nœud analysé. Si I n'est pas vide, alors l'ensemble des caractères possibles pour ce nœud est I, sinon l'ensemble des caractères possibles pour ce nœud est U. Puisqu'il faut connaître les ensembles des nœuds enfants pour pouvoir déterminer ceux des nœuds parents, l'algorithme calcule d'abord les ensembles des nœuds enfants, puis remonte l'arbre et calcule en dernier l'ensemble de la racine, d'où l'appellation « ascendante » de cette étape. L'ensemble de chaque feuille de l'arbre est automatiquement généré sous la forme d'un singleton qui contient le caractère associé à la feuille et provenant de la colonne étudiée, donc les ensembles des feuilles ne sont pas recalculés. Ainsi, seuls les ensembles de caractères des nœuds ancestraux sont déterminés.

L'algorithme de cette étape est résumé dans le pseudocode « deduireEnsembleCaracteres » présentée ci-après.

3.1.3.3 L'étape descendante de l'algorithme de Fitch

Dans cette étape, on attribue un caractère à chaque nœud de l'arbre en analysant l'ensemble de caractères du nœud parent et l'ensemble de caractères du nœud concerné.

Si le nœud a un nœud parent et que son ensemble de caractères possibles contient le caractère attribué au nœud parent, alors le caractère attribué à ce nœud est le même que celui attribué à son nœud parent. Dans le cas contraire (donc, si le nœud n'a pas de parent, ou s'il en a mais que son ensemble de caractères possibles ne contient pas le caractère associé au nœud parent), il faut choisir un caractère dans son propre ensemble de caractères. Par défaut, l'algorithme de Fitch choisit un caractère au hasard dans l'ensemble. C'est à ce niveau que nous avons modifié l'algorithme, car nous souhaitons minimiser l'influence du hasard sur notre méthode. En effet, choisir un caractère au hasard n'est pas forcément une bonne idée dans une méthode d'étude de l'évolution, car cela signifie que notre méthode peut produire des résultats différents si elle est exécutée plusieurs fois sur les mêmes données. De plus, avec des choix faits par hasard, on ne peut pas interpréter ni contrôler les éventuels biais qui pourraient apparaître dans la détection des évènements évolutifs autour des introns. Nous préférons donc choisir un caractère de façon précise, afin de savoir notamment d'où pourraient provenir les éventuels biais que nous pourrions observer plus tard.

Puisqu'il faut idéalement connaître le caractère attribué au nœud parent pour pouvoir déterminer celui du nœud courant, l'algorithme détermine d'abord le caractère d'un nœud, puis descend dans l'arbre pour déterminer les caractères des nœuds enfants, jusqu'à atteindre les feuilles, d'où l'appellation « descendante » de cette étape. Les caractères des feuilles ne sont pas recalculés, puisqu'elles ont déjà des caractères attribués.

L'algorithme de cette étape est résumé dans le pseudocode « deduireCaractere » décrit ci-après. Le pseudocode « deduireCaractere » utilise la procédure « choisirCaractere » qui représente l'étape

que nous avons modifiée, dans laquelle on choisit un caractère dans l'ensemble de caractères disponibles pour le nœud courant.

3.1.3.4 Modification de l'algorithme de Fitch pour le choix d'un caractère

La procédure « choisirCaractere » choisit un caractère parmi ceux disponibles à un nœud donné au lieu de faire un choix au hasard comme le propose par défaut l'algorithme de Fitch. La stratégie de choix suit les étapes suivantes :

- Ranger les caractères dans des classes de caractères particulières.
- Sélectionner la classe la plus représentée, ou bien la classe identique à celle du caractère du nœud parent.
- Choisir un caractère dans la classe sélectionnée, ou bien, si ce n'est pas possible, utiliser un caractère générique qui représente la classe elle-même.

2 classes de caractères sont utilisées :

- La classe « absence » contenant les caractères qui décrivent des absences, à savoir le point (« . ») qui représente un intron absent, et le tiret (« - ») qui représente une molécule absente (acide aminé ou nucléotide dans un codon).
- La classe « présence » contenant les états effectivement présents, à savoir les introns et les molécules (acides aminés ou nucléotides des codons).

Ainsi, si l'algorithme de Fitch ne peut pas choisir un caractère précis à un nœud donné, notre stratégie consiste à déterminer s'il y avait quelque chose ou s'il n'y avait rien à ce nœud. S'il n'y avait rien, alors on peut facilement placer le caractère d'absence correspondant selon le type de colonne (intron ou molécule). S'il y avait quelque chose, alors un choix doit être fait parmi les caractères disponibles. Dans ce cas, on cherche à sélectionner le caractère le plus représenté dans les feuilles du sous-arbre ayant pour racine ce nœud. Si aucun caractère n'est majoritaire, alors on utilise pour ce nœud un caractère générique qui indique la présence de quelque chose même si

on ne sait pas quoi. Il s'agit du caractère « X » pour les colonnes d'acides aminés, et du caractère « x » pour les colonnes de codons.

Cette stratégie de choix n'est pas parfaite, notamment parce que le fait qu'une classe « absence » ou « présence » soit majoritaire ne signifie pas forcément qu'il n'y avait rien ou qu'il y avait quelque chose à ce nœud. Par exemple, il est possible qu'à un nœud donné aucun caractère n'ait été présent, mais que plusieurs insertions indépendantes se soient produites chez les descendants de ce nœud au cours de l'évolution. Dans un tel cas, l'ensemble des caractères possibles pour ce nœud pourrait contenir tous les caractères qui se sont insérés chez les descendants, si bien que la classe « présence » serait sélectionnée par notre stratégie, alors qu'il faudrait idéalement considérer qu'il y avait « absence » d'éléments à ce nœud.

En outre, l'algorithme de Fitch est tel que la classe « absence » à un nœud ne peut contenir qu'un seul caractère : le caractère d'absence d'intron dans une colonne d'introns, ou le caractère d'absence de molécules dans une colonne de molécules. Il est donc probable que la classe « présence » soit très souvent surreprésentée dans les colonnes de molécules au niveau des nœuds où il aura fallu faire un choix. Il s'agit ici d'un biais qui va inciter notre stratégie de choix à souvent considérer qu'il y avait une molécule à un nœud ancestral donné, quand bien même on ne pourrait pas savoir exactement de quelle molécule il s'agissait.

En dépit de ces problèmes, nous utilisons tout de même cette stratégie, car elle est totalement déterministe, et nous permet donc de savoir d'où proviennent les biais, et de s'assurer que notre méthode produit les mêmes résultats pour les mêmes données d'entrée à chaque exécution. Il nous faudra cependant trouver une manière plus sûre de choisir les caractères ancestraux lorsque l'algorithme de Fitch ne peut pas le faire, voire trouver une toute autre méthode de détermination des caractères ancestraux non basée sur l'algorithme de Fitch.

La procédure choisirCaractere est totalement décrite ci-après. Elle utilise une autre procédure, « choisirEtatMoleculaire », appelée lorsqu'il faut choisir spécifiquement une molécule parmi un ensemble de molécules possibles.

3.1.3.5 Pseudocode général utilisé pour l'algorithme de Fitch

Définitions préliminaires

Un nœud résolu est un nœud auquel un caractère a été attribué. Un nœud irrésolu est un nœud auquel un caractère n'a pas été attribué (c'est-à-dire un nœud pour lequel il faut choisir un caractère dans son ensemble de caractères possibles).

Choix d'un caractère ou d'une classe majoritaire dans les feuilles d'un sous-arbre

Les procédures suivantes utilisent régulièrement une stratégie de choix d'un caractère ou d'une classe de caractères à partir des feuilles dans un sous-arbre. La stratégie consiste à déterminer combien de fois chaque caractère (ou classe de caractère) d'une liste de choix apparaît au niveau des feuilles d'un sous-arbre, puis à sélectionner si possible le caractère (ou la classe de caractère) le plus représenté.

Actuellement, l'algorithme de choix est une fonction récursive qui prend en entrée une liste de choix et un nœud qui représente la racine du sous-arbre à analyser. À chaque exécution de la fonction, le nœud indiqué et ses nœuds descendants sont complètement réanalysés, car la fonction ne se souvient pas des analyses antérieures qu'elle a faites.

Comme le plus grand arbre de notre jeu de données ne contient que 17 nœuds (8 nœuds ancestraux et 9 feuilles correspondant aux espèces que nous étudions), cette fonction est rapide, mais elle devrait idéalement être capable de ne pas parcourir à chaque exécution un nœud dont elle a déjà analysé les feuilles. Dans les prochaines versions de notre méthode, cette fonction sera optimisée.

procédure deduireEnsembleCaracteres(nœudCourant)

Pour chaque nœudEnfant du nœudCourant :

- deduireEnsembleCaracteres(nœudEnfant)

Si l'ensemble des caractères du nœudCourant est vide :

- Soit I l'intersection des ensembles de caractères des nœuds enfants du nœudCourant
- Soit U l'union des ensembles de caractères des nœuds enfants du nœudCourant
- Si I n'est pas vide :
  - L'ensemble de caractères du nœudCourant devient I
- Sinon :
  - L'ensemble de caractères du nœudCourant devient U

procédure deduireCaractere(nœudCourant)

Si le nœudCourant n'a pas encore de caractère attribué :

- Si le nœudCourant a un nœudParent et si l'ensemble du nœudCourant contient le caractère du nœudParent :
  - Le caractère du nœudCourant devient le caractère de son nœudParent
- Sinon :
  - choisirCaractere(nœudCourant)

Pour chaque nœudEnfant du nœudCourant :

- deduireCaractere(nœudEnfant)

Procédure choisirEtatMoleculaire (nœud, ensemble de molécules)

Déterminer le type des molécules (acide aminé ou nucléotide d'un codon).

Vérifier si une des molécules est majoritaire dans les feuilles du sous-arbre enraciné à ce nœud.

Si on trouve une molécule majoritaire, la retenir comme caractère de ce nœud.

Sinon, retenir le type de caractère comme caractère de ce nœud : « X » pour les acides aminés, « x » pour les nucléotides d'un codon.

Procédure choisirCaractere(nœud)

Rappel : le nœud est considéré comme irrésolu.

Si on est dans une colonne d'introns :

- Pour un nœud dans une colonnes d'introns, l'ensemble des caractères possibles peut contenir soit « + », soit « . », soit les deux. Si l'ensemble contient un seul caractère, alors le nœud serait résolu, donc cette procédure ne serait pas appelée. L'ensemble des caractères possibles pour un nœud irrésolu dans une colonne d'introns contient donc forcément les deux caractères « + » et « . ».
- Le nœud ne peut pas non plus avoir de parent, car sinon son ensemble de caractères possibles contiendrait forcément le caractère du nœud parent (soit « + » soit « . ») qui est résolu (car traité avant le nœud courant dans l'étape descendante de l'algorithme de Fitch), donc le nœud serait aussi résolu.
- Seule la racine (qui n'a pas de parent) avec les deux caractères possibles (« + » et « . ») peut donc être un nœud irrésolu dans une colonne d'introns.
- On détermine si un des deux caractères est majoritaire dans les feuilles de l'arbre.
- Si un des deux caractères est majoritaire, on le choisit.
- Sinon, on ne peut toujours pas faire un choix précis avec notre stratégie actuelle. Pour le moment, nous choisissons arbitrairement le caractère d'absence « . ». Ce choix par défaut n'est pas encore paramétrable dans la méthode actuelle, mais le sera dans les prochaines versions.

Sinon on est dans une colonne de molécules :

- Ranger les caractères du nœud dans les deux classes « absence » et « présence ».

- Si une seule classe est trouvée parmi les caractères possibles pour ce nœud :
  - Alors il ne peut s'agir de la classe absence. En effet, elle ne contient qu'un seul caractère « - », donc ce caractère serait le seul possible pour ce nœud, qui serait donc résolu. L'unique classe trouvée est donc la classe « présence », qui contient des molécules.
  - choisirEtatMoleculaire(nœud, ensemble des caractères possibles du nœud)
- Sinon (les deux classes sont trouvées parmi les caractères possibles pour ce nœud) :
  - Si le nœud a un parent :
    - On choisit la classe commune au nœud parent et au nœud enfant. Il s'agit de la classe du caractère du nœud parent, qui est soit « absence » soit « présence », et est donc parmi les deux classes trouvées pour le nœud courant.
    - La classe commune ne peut être la classe « absence ». Sinon, le nœud parent aurait le caractère « - » attribué, qui apparaîtrait aussi chez le nœud courant, qui serait donc résolu. La classe commune est donc la classe « présence ».
    - choisirEtatMoleculaire(nœud, molécules dans la classe « présence » de ce nœud)
  - Sinon, c'est la racine :
    - Si la classe « présence » contient plus de caractères que la classe « absence », alors choisir la classe « présence » :
      - choisirEtatMoleculaire(nœud, molécules dans la classe « présence » de ce nœud)
    - Sinon :
      - Il y a autant de caractères dans les deux classes, or il y en a 1 seul dans la classe « absence », donc il y en a 1 seul dans la classe « présence », donc il y a 2 caractères possibles pour la racine : le caractère « - » et une molécule.
      - Vérifier si une des classes est majoritaire dans les feuilles du sous-arbre enraciné à ce nœud.

· Si une classe est majoritaire, retenir le caractère correspondant à cette classe parmi les deux caractères disponibles.

· Sinon, on ne peut toujours pas faire un choix précis avec notre stratégie actuelle. Ce problème de choix entre l'absence et la présence d'un caractère est également rencontré dans la littérature [1]. Pour le moment, nous choisissons arbitrairement le caractère d'absence « - ». Ce choix par défaut n'est pas encore paramétrable dans la méthode actuelle, mais le sera dans les prochaines versions.

3.1.4 Assemblage et vérification des séquences reconstruites

L'algorithme de Fitch ainsi modifié est exécuté sur chaque colonne d'un alignement. Les caractères déduits sont ensuite assemblés dans l'ordre de successions des colonnes de l'alignement, ce qui fournit les séquences ancestrales reconstruites et alignées, dans lesquelles sont mises en évidence des positions d'introns.

3.1.4.1 Réajustement des extrémités des séquences reconstruites

Dans les séquences ancestrales reconstruites, il se peut qu'on trouve des introns positionnés avant le premier acide aminé ou après le dernier acide aminé de la séquence.

Un exemple concret de cette situation est présenté dans la figure 3.2, et provient d'une des familles de protéines strictement orthologues étudiées, la famille « oomycetes4897 ». La figure 3.2a présente la fin de l'alignement de la famille avec les séquences reconstruites. Cette extrémité de l'alignement contient 1 colonne d'introns. Les acides aminés reconstruits et les trous reconstruits sont affichés, mais les introns à reconstruire dans les séquences ancestrales sont remplacées par « ? ». Pour la colonne d'introns, la phase ascendante de l'algorithme de Fitch se déroule normalement comme suit :

- ancestor7 : ses descendants ont les deux caractères possibles (« + » et « . »). Son ensemble de caractères est donc {« + » ; « . » }.

- ancestor6 : son descendant phca a le caractère « . » présent dans l'ensemble de ancestor7. Son caractère attribué est donc « . ».
- ancestor5 : ses descendants ont le caractère « + » en commun. Son caractère attribué est donc « + ».
- ancestor4 : ses descendants (ancestor5 et ancestor6) ont les deux caractères possibles. Son ensemble de caractères est donc {« + » ; « . » }.
- ancestor3 : ses descendants (phra et ancestor4) ont le caractère « + » en commun. Son caractère attribué est donc « + ».
- ancestor2 : ses descendants ont les deux caractères en commun. Son ensemble de caractères est donc {« + » ; « . » }.
- ancestor1 : ses descendants (pyul et ancestor1) ont le caractère « + » en commun. Son caractère attribué est donc « + ».
- root : ses descendants (albu et ancestor1) ont le même caractère « + ». Son caractère attribué est donc « + ».

Nous constatons que la colonne d'intron contient « + » pour ancestor3, alors qu'il n'y a plus d'acides aminés après cette colonne dans la séquence de cet ancêtre. La présence d'un intron a ainsi été prédite après le dernier acide aminé de la séquence d'ancestor3. Comme cette prédiction est déjà fixée dans l'étape ascendante de Fitch (car le caractère « + » est considéré comme le seul caractère possible dans cette colonne pour ancestor3), elle ne sera plus modifiée pendant l'étape descendante.

Puisque nous n'étudions que les introns situés dans les séquences codantes des protéines, nous ne nous attendons donc pas à trouver des introns avant ou après la protéine elle-même. La présence de tels introns dans les séquences reconstruites est donc considérée comme une erreur de reconstruction, et à l'heure actuelle, ces introns sont enlevés avant de poursuivre l'étude. La suppression de ces introns se déroule comme suit :

- Repérer les colonnes d'introns dans lesquelles un intron apparaît avant le premier acide aminé ou après le dernier acide aminés dans au moins une séquence ancestrale.

- Prendre l'arbre des espèces et positionner les caractères de la colonne sur les feuilles mais aussi dans tous les autres nœuds de l'arbre (puisque des reconstructions sont disponibles pour tous les nœuds à ce stade).

- Pour chaque nœud d'ancêtre qui possède un intron dans cette colonne (donc un intron situé avant ou après la protéine reconstruite pour cet ancêtre), remplacer l'intron indiqué (caractère « + ») par le caractère d'absence d'intron (« . »).

- Exécuter intégralement l'algorithme de Fitch modifié sur la colonne, puis mettre à jour la colonne avec les nouveaux caractères reconstruits.

  – L'algorithme ne modifie pas les nœuds déjà résolus, donc les nœuds internes qui possédaient avant des introns, mais auxquels on a imposé de ne pas en avoir, resteront sans introns. L'algorithme peut cependant mettre à jour les autres nœuds internes.

La figure 3.2b présente l'alignement de la figure 3.2a après réajustement et exécution intégrale de l'algorithme de Fitch modifié. On constate qu'il n'y a effectivement plus de présence d'intron prédite après la fin d'une protéine.

Une fois cette dernière vérification réalisée, les séquences ancestrales reconstituées, alignées et utilisables pour la suite de l'étude, sont enfin disponibles.

3.1.5 Implémentation de l'algorithme de reconstruction

Nous avons implémentés l'algorithme complet de reconstruction des séquences ancestrales sous la forme d'un programme JAVA qui prend en entrée l'arbre des espèces au format NEWICK, et un fichier AMF unique ou un dossier contenant tous les fichiers AMF à analyser. Le programme suppose qu'il traite des fichiers AMF de familles strictement orthologues. Le comportement du programme est indéfini pour les autres types de familles.

```
root                YYNXrr?rHVTLLTNLIQSPELNSSNPSPVXKDXEQRILRCYFRXLCRXYLX-
  albu              RYNQrr+rAVKLLTNLVQTPELNTCNSETVFETLKDRMLRCYFRHLCRRYLV-
  ancestor1         YYNSss?sHVTLLTHLIQSPELNSSNPSPVLKDVEQRILRCYFRELCRVHLR-
    ancestor2       YLN---?-------------------------------------------------
      hyal          ------.-------------------------------------------------
      ancestor3     YLN---?-------------------------------------------------
        phra        YLNDss+sLPS---------------------------------------------
        ancestor4   YLN---?-------------------------------------------------
          ancestor5 YLNErr?rHVTLLTXLIQSPELNSSNPAPVVKDAEQRILRCYFRDLCRIYLGN
            phci    YLNErr+rHVTLLTSLIQSPELNSSNPAPVVKDAEQRILRCYFRDLCRIYLGN
            phso    YLNErr+rHVTLLTNLIQSPELNSSNPAPVVKDAEQRILRCYFRDLCRIYLGN
          ancestor6 ------?-------------------------------------------------
            phca    ------.-------------------------------------------------
            ancestor7 ------?-----------------------------------------------
              phpa  YLNDrr+rHVTLLTNLIQSPELNSSNPSPVVKDAEQRILRCYFRDLCRIYLSD
              phin  ------.-------------------------------------------------
    pyul            HYNSss+sHVTLLTHLIQSPELNSSNPSPVLKDVEQRILRCYFRELCRVHLR-
```

(a) Fin de l'alignement de la famille de protéines nommée oomycetes4897, contenant une colonne d'introns. Les introns des espèces actuelles sont affichés, les introns à prédire dans les séquences ancestrales sont indiquées par un « ? ». La phase ascendante de l'algorithme de Fitch prédit un intron pour l'ancêtre ancestor3, alors que la colonne d'introns se trouve après le dernier acide aminé de la séquence de cet ancêtre.

```
root                YYNXrr+rHVTLLTNLIQSPELNSSNPSPVXKDXEQRILRCYFRXLCRXYLX-
  albu              RYNQrr+rAVKLLTNLVQTPELNTCNSETVFETLKDRMLRCYFRHLCRRYLV-
  ancestor1         YYNSss+sHVTLLTHLIQSPELNSSNPSPVLKDVEQRILRCYFRELCRVHLR-
    ancestor2       YLN---.-------------------------------------------------
      hyal          ------.-------------------------------------------------
      ancestor3     YLN---.-------------------------------------------------
        phra        YLNDss+sLPS---------------------------------------------
        ancestor4   YLN---.-------------------------------------------------
          ancestor5 YLNErr+rHVTLLTXLIQSPELNSSNPAPVVKDAEQRILRCYFRDLCRIYLGN
            phci    YLNErr+rHVTLLTSLIQSPELNSSNPAPVVKDAEQRILRCYFRDLCRIYLGN
            phso    YLNErr+rHVTLLTNLIQSPELNSSNPAPVVKDAEQRILRCYFRDLCRIYLGN
          ancestor6 ------.-------------------------------------------------
            phca    ------.-------------------------------------------------
            ancestor7 ------.-----------------------------------------------
              phpa  YLNDrr+rHVTLLTNLIQSPELNSSNPSPVVKDAEQRILRCYFRDLCRIYLSD
              phin  ------.-------------------------------------------------
    pyul            HYNSss+sHVTLLTHLIQSPELNSSNPSPVLKDVEQRILRCYFRELCRVHLR-
```

(b) Fin de l'alignement de la famille de protéines oomycetes4897, montrant la reconstruction dans la colonne d'introns après réajustement et exécution intégrale de l'algorithme de Fitch. Il n'y a plus d'introns prédits après la fin d'une séquence.

Figure 3.2 – Exemple de réajustement des extrémités des séquences ancestrales de la famille oomycetes4897 pour empêcher la prédiction d'introns à l'extérieur des séquences.

Le programme génère en sortie 3 fichiers aux formats différents pour chaque alignement en entrée. Tous les fichiers contiennent les séquences ancestrales reconstruites et les séquences des espèces actuelles, toutes mettant en évidence les introns qu'elles contiennent. Les 3 fichiers générés pour un alignement sont :

- Un fichier au format FASTA. Il a pour extension « withAncestors.fasta ».
- Un fichier qui affiche l'arbre des espèces, en plaçant les séquences devant chaque nœud. Il a pour extension « withAncestors.seqtree ».
- Un fichier qui affiche chaque triplet de séquences (ancêtre ; descendant gauche ; descendant droit) pour cette famille de protéines. Il a pour extension « withAncestors.pca ». L'extension « pca » est un acronyme pour « *parent-children alignments* ».

Ce sont les fichiers « withAncestors.pca » qui seront ensuite utilisés, car ils permettent de récupérer facilement toutes les paires de séquences (ancêtre ; descendant) , et donc de les analyser plus facilement.

3.1.6 Application

Nous avons reconstruit les séquences ancestrales pour les 1 924 familles de protéines strictement orthologues. L'arbre des espèces contient 8 espèces ancestrales, auxquelles nous avons attribué des noms. L'espèce à la racine est appelée « root » et les autres sont désignées par les noms allant de « ancestor1 » à « ancestor7 ». La figure 3.3 présente la topologie de l'arbre avec les noms associés aux espèces ancestrales.

Pendant la vérification des séquences reconstruites, 8 erreurs ont été repérées dans 8 familles de protéines différentes (1 erreur par famille). Dans 4 familles nous avons trouvé un intron après la fin d'une protéine ancestrale reconstruite, et dans 4 autres nous avons trouvé un intron avant le début d'une protéine ancestrale reconstruite. Nous avons tout de même poursuivi l'étude avec ces 8 familles.

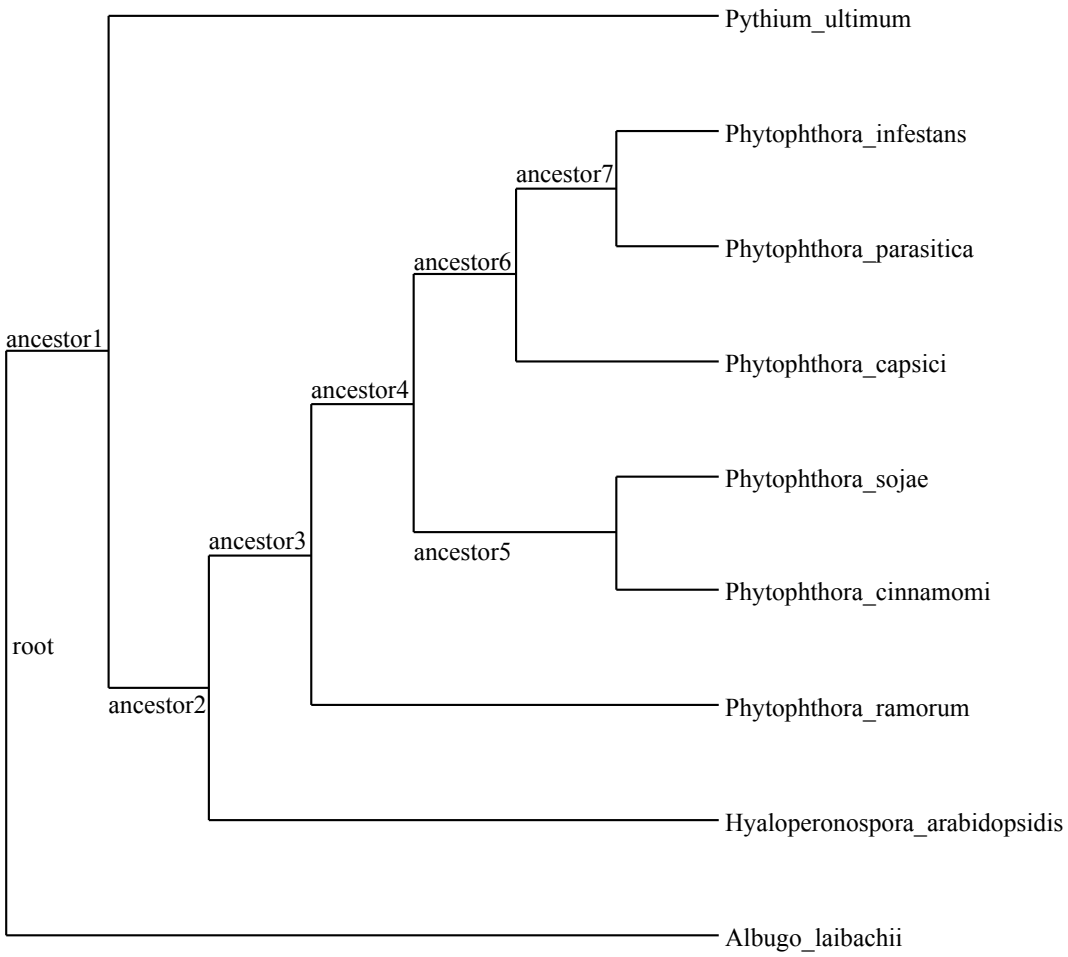


Figure 3.3 – Topologie de l'arbre des espèces montrant les noms attribués aux noeuds internes.

Une fois les séquences ancestrales reconstruites et alignées, nous pouvons désormais étudier concrètement les mutations introniques locales.

3.2 Déduction des évènements évolutifs impliquant les introns

La dernière étape dans la version actuelle de notre méthode consiste à détecter les mutations qui se produisent au niveau des introns dans les familles de protéines strictement orthologues des espèces étudiées. Cependant, plusieurs hypothèses et simplifications ont été utilisées dans les étapes précédentes, et certaines erreurs sont probablement présentes dans nos données sans avoir été clairement mises en évidence (par exemple, des erreurs d'annotations pour les introns trop courts ou trop longs). Pour ces raisons, l'algorithme que nous avons développé pour détecter les mutations est une heuristique, qui détecte plusieurs mutations potentielles, mais dont plusieurs peuvent être des faux positifs. Cependant, l'algorithme garantit que tous les introns disponibles et prédits dans les alignements seront analysés, et même les « non-évènements » potentiels (c'est-à-dire des introns dont les positions et les sites d'épissage n'ont pas changé) sont pris en compte. Le risque d'ignorer une mutation est ainsi réduit, et donc les vrais négatifs devraient être rares.

3.2.1 Hypothèse de travail

Nous avons défini 2 catégories de mutations relatives aux introns : les mutations introniques locales, dues à des changements de nucléotides à l'intérieur d'un gène, et les mutations introniques globales, dues à des interactions entre un gène et une autre séquence issue du génome. Notre projet se focalise sur les mutations introniques locales, mais il est tout à fait possible que des mutations introniques globales soient présentes dans nos jeux de données. Pour le moment, notre méthode n'a pas de stratégie particulière permettant de différencier les mutations globales des mutations locales. Seule notre hypothèse de départ, selon laquelle les génomes étudiés par notre méthode seraient proches, nous permet de supposer que les mutations globales seraient plus rares que si les espèces étudiées étaient plus éloignées, sans toutefois le garantir de manière certaine. Une future évolution de notre méthode devra permettre d'identifier les mutations globales afin de ne pas les confondre avec les mutations locales, ce qui permettrait aussi d'avoir des informations supplémentaires sur l'évolution des introns à l'échelle de l'ADN dans son ensemble, et non plus

seulement à l'intérieur des gènes.

Pour détecter les mutations introniques locales, une stratégie possible aurait été de prédéterminer tous les types de mutations introniques locales possibles, de caractériser chaque type pour savoir comment le repérer dans un alignement de séquences, puis de chercher les occurrences des types définis dans les familles de protéines étudiées, en se servant des caractéristiques identifiées pour les types à chercher. Cependant, comme nous l'avons vu, les mutations introniques locales sont causées par les mutations des nucléotides à l'intérieur d'un gène. La diversité des types possibles de mutations introniques locales pourrait être assez élevée, et n'être limitée que par le nombre de combinaisons de mutations ponctuelles nucléotidiques susceptibles d'affecter un intron ou plusieurs introns en même temps dans un gène. Il semble donc assez compliqué d'imaginer à l'avance tous les types de mutations introniques globales qui peuvent exister. En utilisant cette stratégie, nous risquions de ne pas détecter de nombreuses mutations d'introns si nous n'avions pas prévu certains cas d'évolution.

Nous utilisons donc une approche différente, basée sur les observations que les mutations introniques locales peuvent générer dans les alignements de séquences. En effet, les mutations introniques locales sont dues aux mutations nucléotidiques. Peu importent les mutations ou les séries de mutations de nucléotides qui se produisent, on obtient finalement soit la conversion de tout ou une partie d'un exon en intron, soit la conversion de tout ou une partie d'un d'intron en exon, soit un déplacement des frontières des introns, soit des suppressions complètes ou des insertions complètes d'introns (par exemple, via des insertions ou suppressions successives de nouveaux nucléotides, ou par des duplications ou suppressions de segments du gène contenant des introns complets). On observe alors des trous situés près des introns dans les alignements. De plus, il peut arriver que 2 portions de séquences protéiques situées près d'introns s'alignent par erreur à cause des contraintes des algorithmes d'alignement, alors qu'elles devraient être séparées par des trous. Nous pouvons donc observer des substitutions apparentes autour des introns dans les alignements. Par extension, nous pouvons aussi considérer que de telles substitutions ne sont pas forcément apparentes mais belles et bien réelles, et dues à des mutations que nous n'avons pas identifiées.

Ces réflexions nous permettent de proposer une hypothèse qui définit notre approche : l'évolution des introns par mutations locales pourrait être détectée dans nos séquences en repérant les changements qui se produisent autour des introns et qui nous font observer des trous proches des introns, des substitutions proches des introns, ou des changements de positions des introns (donc des introns qui ne sont pas dans une même colonne mais qui sont proches) dans les alignements de nos séquences. Plus généralement, ces changements se traduiraient par des zones dans les alignements qui contiennent des introns et des colonnes non conservées (qu'on peut qualifier « d'instables ») autour d'eux.

3.2.1.1 Observation en faveur de notre hypothèse

Notre hypothèse est encouragée par certaines observations concrètes qu'on peut faire sur les espèces que nous étudions. En effet, nous avons écrit un programme JAVA qui peut analyser les alignements des familles pour y dénombrer des portions d'alignements aux caractéristiques particulières, que nous appelons fenêtres. 4 types de fenêtres sont cherchés :

- Les fenêtres qui ne contiennent que des trous.
- Les fenêtres qui ne contiennent que des introns.
- Les fenêtres qui contiennent à la fois des trous et des introns.
- Les fenêtres qui ne contiennent ni trous ni introns.

Le programme prend en entrée un ensemble de fichiers AMF et une taille fixe pour les fenêtres à analyser. Nous avons exécuté le programme sur toutes les familles de protéines disponibles (les 18 955 familles détectées) avec une fenêtre de longueur 10. Chaque alignement est donc subdivisé en sous-parties consécutives contenant chacune 10 colonnes, et chaque sous-partie est ensuite analysée pour être classée parmi les 4 types de fenêtres cherchées. Le résultat de l'analyse est présenté dans le tableau 3.I. Nous pouvons remarquer que, sur les 83 476 fenêtres contenant des introns, 71 024 d'entre elles (soit 85,08 %) contiennent aussi des trous. Ce résultat suggère que les introns ont tendance à être proches de trous, ce qui pourrait être dû aux effets des mutations introniques sur les alignements, que notre hypothèse met en évidence.

Tableau 3.I – Dénombrement des fenêtres d'alignements de longueur 10 détectées dans les familles de protéines et définies en fonction de la présence de trous et d'introns.

Nombre de fenêtres sans trous ni introns	282 745
Nombre de fenêtres avec seulement des trous	635 170
Nombre de fenêtres avec seulement des introns	12 452
Nombre de fenêtres avec des trous et des introns	71 024

3.2.2 Algorithme

Nous avons donc développé un algorithme de détection des mutations introniques qui repose sur l'idée que ces mutations laissent des traces autour des introns dans les séquences. Il s'agit donc d'analyser les alignements disponibles et de repérer les traces proches des introns. Comme nous avons reconstruit les séquences ancestrales, nous pouvons travailler sur chaque paire de séquence ancêtre-descendant, ce qui permet aussi d'avoir une idée sur le sens de chaque évolution détectée, c'est-à-dire faire la différence, par exemple, entre une apparition et une disparition de caractères.

L'algorithme prend en entrée une paire de séquences alignées provenant d'une famille de protéines : une séquence considérée comme « descendante », et une séquence considérée comme « ancestrale » dont dérive la séquence descendante. L'algorithme n'impose pas que la séquence descendante soit une séquence actuelle située à la feuille de l'arbre des espèces. On peut donc aussi étudier les changements entre deux ancêtres situés plus haut dans l'arbre.

L'algorithme s'exécute en plusieurs étapes :

- On cherche tous les évènements évolutifs autour des introns. Un évènement évolutif autour d'un intron est une zone (portion) d'un alignement de deux séquences contenant au moins un intron et présentant des différences entre les deux séquences. Pour repérer ces évènements, l'algorithme parcourt l'alignement et s'arrête à chaque colonne contenant un intron. Pour chaque colonne ainsi rencontrée, on définit une zone dont le point de départ est la colonne et que nous allons étendre en amont et en aval de la colonne le plus possible et selon certains critères. Chaque zone étendue est un évènement évolutif potentiel autour

d'un intron.

- Toutes les zones ainsi trouvées sont ensuite analysées pour détecter et fusionner les zones qui se chevauchent. On considère en effet que, si deux zones ont des éléments en commun, alors leurs introns ont peut-être été affectés par une mutation commune, et il est donc préférable de regrouper ces zones.

- Après ce traitement, on obtient une liste de zones finales qui représentent les évènements évolutifs cherchés. On réalise ensuite le typage de chaque zone selon des critères spécifiques, afin d'attribuer un même type à tous les évènements qu'on pourrait considérer comme similaires. Cela permettra de repérer et d'étudier plus facilement les évènements similaires dans une famille et à travers toutes les familles.

- Finalement, les évènements sont sauvegardés dans des fichiers avec les informations disponibles, à savoir le type, la localisation (famille et position dans l'alignement), et l'apparence (portion de l'alignement correspondant à l'évènement, pour pouvoir le visualiser immédiatement) de chaque évènement.

L'extension des zones autour des introns et le typage des évènements évolutifs se font avec des critères spécifiques qui sont détaillés ci-après.

3.2.2.1 Extension des zones autour des introns

En partant d'une colonne contenant des introns, nous voulons déterminer la zone autour de cette colonne dans laquelle des effets de l'évolution de ces introns sont observés. Pour le faire :

- On associe une valeur supérieure à 1 à la zone, qui représente son instabilité basale. L'instabilité basale est calculée en fonction de la nature de la colonne de référence, c'est-à-dire la colonne contenant des introns et autour de laquelle on veut étendre la zone.

- On parcourt l'alignement dans deux directions, en avançant et en reculant à partir de la colonne de référence. Pour chaque parcours, on définit un poids qui représente l'instabilité de l'ensemble des colonnes visitées par le parcours. Il s'agit donc du poids en amont (en

reculant) et du poids en aval (en avançant). Au départ, les deux poids ont pour valeur l'instabilité basale.

- On parcourt l'alignement en reculant et en mettant à jour le poids en amont pour chaque colonne rencontrée. Le parcourt s'arrête à la première colonne rencontrée qui fait chuter le poids en amont en dessous de 1. Cette colonne représente la borne inférieure de la zone d'évolution.

- On parcourt l'alignement en avançant et en mettant à jour le poids en aval pour chaque colonne rencontrée. Le parcours s'arrête à la première colonne rencontrée qui fait chuter le poids en aval en dessous de 1. Cette colonne représente la borne supérieure de la zone d'évolution.

La borne inférieure et la borne supérieure délimitent ainsi la zone que nous recherchions autour de la colonne de référence.

Chaque colonne rencontrée dans un parcours a un effet sur son poids, qui dépend du type de la colonne. En combinant les différents caractères qu'on peut rencontrer dans nos alignements, on obtient plusieurs types de colonnes, dont certains sont impossibles, et d'autres ignorés pendant l'extension des zones autour des introns.

Le tableau 3.II résume le traitement des types de colonnes possibles. Dans ce tableau, « . » désigne un intron absent, « + » désigne un intron présent, « E » et « F » désignent des molécules (acides aminés ou nucléotides dans un codon) différentes, « - » désigne un trou qui indique une absence d'une molécule. Les colonnes au traitement impossible sont celles dans lesquelles un caractère intronique s'aligne avec un caractère moléculaire. Ce cas ne peut se produire car les introns ne sont pas comparés aux molécules dans nos alignements. Les colonnes ignorées sont les colonnes vides, qui ne contiennent aucun intron ni aucune molécule.

Seuls 9 types de colonnes peuvent être réellement rencontrés, et seules 7 d'entre eux (en vert et en gras dans le tableau 3.II) sont pris en compte pour étendre les zones d'évolution.

Tableau 3.II – Types de colonnes possibles dans un alignement simple mettant en évidence les introns, et traitement de ces colonnes pendant la détection des zones d'évolutions entourant les introns.

Ancêtre	Descendant	Description	Traitement
.	.	Absence d'introns	Ignoré (colonne vide)
.	**+**	**Intron absent et intron présent**	**Pris en compte**
.	−	Intron absent et molécule absente	Impossible
.	E	Intron absent et molécule présente	Impossible
+	**.**	**Intron présent et intron absent**	**Pris en compte**
+	**+**	**Deux introns présents**	**Pris en compte**
+	−	Intron présent et molécule absente	Impossible
+	E	Intron présent et molécule présente	Impossible
−	.	Molécules absente et intron absent	Impossible
−	+	Molécules absente et intron présent	Impossible
−	−	Deux molécules absentes	Ignoré (colonne vide)
−	**E**	**Insertion de molécule**	**Pris en compte**
E	.	Molécule présente et intron absent	Impossible
E	+	Molécule présente et intron présent	Impossible
E	**−**	**Suppression d'une molécule**	**Pris en compte**
E	**E**	**Molécule conservée**	**Pris en compte**
E	**F**	**Substitution de molécules**	**Pris en compte**

Les types de colonnes considérés peuvent être rangés dans 2 catégories : les colonnes dites « déstabilisantes », qui feront augmenter le poids des parcours, et les colonnes dites « stabilisantes », qui feront diminuer le poids des parcours. Le tableau 3.III définit notre catégorisation des colonnes considérées.

Les colonnes considérées comme déstabilisantes sont celles dans lesquelles les caractères changent entre les deux séquences. Une exception est faite pour les colonnes de substitutions de molécules, car on peut considérer que, même si les molécules changent, la structure de la protéine et la fonction de la partie de la protéine qui contient cette substitution peuvent demeurer identiques et être conservées au fil de l'évolution. Les colonnes de substitutions de molécules sont donc considérées comme stabilisantes, mais beaucoup moins que les colonnes de conservation de molécules (dans lesquelles la molécule est identique dans les deux séquences), qui sont les plus stabilisantes.

Toutes les colonnes contenant des introns sont considérées comme déstabilisantes, ce qui permet, lorsqu'elles sont rencontrées dans les parcours, de les intégrer dans les zones en extension, et donc de rassembler dans une même zone les introns qui sont proches dans les alignements. Il en va de même pour la colonne « +/+ » qui propose une conservation d'intron, car le fait que deux introns partagent la même colonne ne signifie pas forcément qu'ils sont identiques (leurs séquences peuvent changer). Même si les séquences introniques ne sont pas encore analysées par notre méthode, il nous paraît intéressant de collectionner les évènements évolutifs autour des

Tableau 3.III – Catégorisation des colonnes prises en compte pendant la détection des zones d'évolution entourant les introns.

Ancêtre	Descendant	Description	Catégorie
.	+	Intron absent et intron présent	Déstabilisante
+	.	Intron présent et intron absent	Déstabilisante
+	+	Deux introns présents	Déstabilisante
−	E	Insertion de molécule	Déstabilisante
E	−	Suppression d'une molécule	Déstabilisante
E	**E**	**Molécule conservée**	**Très stabilisante**
E	F	Substitution de molécules	Stabilisante

introns conservés, afin de pouvoir les analyser dans de futures améliorations de notre méthode.

À partir des catégories des colonnes, nous définissons enfin leur effet sur les poids des parcours.

Nous attribuons aux colonnes déstabilisantes un poids qui s'additionne aux poids des parcours et donc les augmente. L'effet peut être décrit de la façon suivante :

Poids du parcours après effet = poids du parcours avant effet + poids de la colonne déstabilisante

Nous associons aux colonnes stabilisantes un facteur qui fait chuter les poids des parcours en les multipliant par un nombre positif plus petite que 1. L'effet peut être décrit de la façon suivante :

Poids du parcours après effet = (poids du parcours avant effet) x (1 – facteur de la colonne stabilisante)

Les poids des colonnes déstabilisantes sont contrôlés par un paramètre p, qui est un entier naturel non nul. Les facteurs des colonnes stabilisantes sont contrôlés par un paramètre q, qui est un nombre réel supérieur à 0 et inférieur à 1 (q ne peut pas être un entier). Le tableau 3.IV décrit les poids et les facteurs des colonnes et leurs effets sur les poids des parcours.

Le choix des valeurs d'effet est fait selon l'importance que nous attribuons à la force de chaque colonne. On considère que les colonnes dans lesquelles un intron apparaît (« ./+ ») ou disparaît (« +/. ») sont les plus déstabilisantes, car elles peuvent indiquent une insertion ou une suppression complètes d'un intron, ou des modifications majeures au niveau de ses sites d'épissage. Viennent ensuite les colonnes dans lesquelles la présence d'un intron est conservée (« +/+ »), qui peuvent aussi impliquer des changements des sites d'épissage (notamment du site d'épissage accepteur de l'intron si un trou se trouve après une telle colonne), et des modifications à l'intérieur de l'intron. Les colonnes contenant les introns sont considérées comme étant les plus déstabilisantes, car elles

Tableau 3.IV – Effet des colonnes d'alignement prises en compte sur l'extension des zones d'évolution autour des introns.

« w » désigne le poids d'extension d'une zone dans une direction. Le paramètre p doit être un entier naturel non nul ($p \in \mathbb{N}^*$). Le paramère q doit être un nombre réel compris entre 0 et 1 exclus ($q \in \mathbb{R}$ et $0 < q < 1$).

Ancêtre	Descendant	Catégorie	Valeur d'effet	Mode d'effet
`.`	`+`	Déstabilisante	`4p`	`w <- w + 4p`
`+`	`.`	Déstabilisante	`4p`	`w <- w + 4p`
`+`	`+`	Déstabilisante	`2p`	`w <- w + 2p`
`-`	`E`	Déstabilisante	`P`	`w <- w + p`
`E`	`-`	Déstabilisante	`P`	`w <- w + p`
`E`	**`E`**	**Très stabilisante**	**`q`**	**`w <- w(1-q)`**
`E`	`F`	Stabilisante	$\frac{q}{2}$	`w <-` $w(1-\frac{q}{2})$

constituent notre sujet d'étude central. De plus, il s'agit également des colonnes qui servent de base à l'extension des zones, et nous utilisons directement leurs poids comme instabilité basale de ces zones pour initier les parcours en amont et en aval et étendre les zones. Il faut donc idéalement que leur poids soit assez élevé, pour que l'extension des zones ne s'arrête pas trop vite.

Les colonnes déstabilisantes font augmenter les poids des parcours de façon approximativement linéaire, si plusieurs colonnes du même type sont rencontrées, tandis que les colonnes stabilisantes font chuter le poids via des divisons successives. Si le facteur q est bien choisi, la chute des poids peut donc se faire de façon exponentielle. Nous avons volontairement défini les effets de cette manière afin que la décroissance des poids lorsque des colonnes stabilisantes sont rencontrées soit beaucoup plus rapide que la croissance des poids lorsque des colonnes déstabilisantes sont rencontrées. Ainsi, même si plusieurs colonnes déstabilisantes consécutives sont rencontrées, les zones sont rapidement délimitées dès que quelques colonnes stabilisantes sont rencontrées à la fin des parcours. Cela permet de ne pas avoir des zones beaucoup trop grandes, et un réglage des paramètres p et q permet de ne pas avoir des zones trop petites.

Notre algorithme actuel utilise par défaut les valeurs 1 pour p et $\frac{1}{2}$ pour q. Ainsi, chaque colonne de conservation divise le poids par 2, et chaque colonne de substitution divise le poids par $\frac{4}{3}$.

3.2.2.2 Typage des évènements évolutifs autour des introns

Une fois les zones détectées et fusionnées, on obtient les évènements évolutifs que nous typons afin de les caractériser et de repérer les évènements similaires. Pour typer un évènement, nous le réécrivons sous une version simplifiée. L'idée est de remplacer une suite consécutive de colonnes similaires dans l'évènement par une colonne unique. Chaque colonne de conservation (de type « E/E ») ou de substitution (de type « E/F ») est remplacée par « E/E ». Chaque colonne contenant une insertion est remplacée par « -/E ». Chaque colonne contenant une suppression est remplacée par « E/- ». Les colonnes contenant des introns ne sont pas modifiées. Ensuite, toute suite de colonnes moléculaires identiques est remplacée par une colonne unique. Les colonnes introniques consécutives sont laissées telles quelles, car le type de l'évènement doit mettre en évidence toutes les colonnes d'introns et leur nombre exact, sans les grouper.

Dans la version simplifiée, une colonne d'intron est donc la même que dans l'évènement, tandis qu'une colonne moléculaire représente une région de l'alignement composée de colonnes successives similaires (plusieurs suppressions consécutives, plusieurs insertions consécutives, ou plusieurs colonnes consécutives de conservations et de substitutions).

Par exemple, considérons l'évènement hypothétique suivant :

```
AKLVETKV.LDAGATKRFG+YKASDATSSVNREKADTERY
AKLRETKV+----------.YKASDATSSVNREMADYETY
```

Pour typer cet évènement, on simplifie d'abord chaque colonne :

```
EEEEEEEE.EEEEEEEEEE+EEEEEEEEEEEEEEEEEEEE
EEEEEEEE+----------.EEEEEEEEEEEEEEEEEEEE
```

Puis on compresse les colonnes simplifiées, pour obtenir la version simplifiée de l'évènement :

```
E.E+E
E+-.E
```

La version simplifiée ainsi obtenue représente le type de l'évènement. On peut la réécrire sous une forme « linéaire » en séparant la séquence ancestrale de la séquence descendante par une barre oblique : « E.E+E/E+-.E ». Chaque version simplifiée détectée définit donc un type d'évènement, et tous les évènements qui ont la même version simplifiée sont considérés comme ayant le même type.

Un autre exemple de typage d'évènement est donné dans le tableau 3.V, dans lequel un vrai évènement détecté dans les familles étudiées est présenté.

3.2.3 Implémentation de l'algorithme

L'algorithme de détection des évènements évolutifs autour des introns est implémenté sous la forme d'un programme JAVA. Le programme prend en entrée 3 paramètres :

- Un fichier « .pca » ou un dossier contenant des fichiers .pca. Les fichiers .pca ont été générés à l'étape précédente de reconstruction des séquences ancestrales. Un fichier correspond à une famille et contient les alignements de chaque séquence ancestrale avec ses deux séquences descendantes directes. On peut donc récupérer facilement les paires ancêtre-descendant pour les analyser.
- La valeur pour le paramètre p (qui vaut 1 par défaut).
- La valeur pour le paramètre q (qui vaut $\frac{1}{2}$ par défaut).

Tableau 3.V – Exemple de typage d'un évènement réel détecté dans une famille de protéines orthologues.

L'évènement est détecté dans la famille nommée oomycetes4843, en passant de la séquence de ancestor2 (séquences en haut) à la séquence de son descendant *Hyaloperonospora arabidopsidis* (séquence en bas).

Évènement détecté	`Ar+rrHVLESATLLSILREDMTAFE` `Ah+hh--------------MTAYE`
Simplification	`EE+EEEEEEEEEEEEEEEEEEEE` `EE+EE--------------EEEEE`
Compression pour obtenir le type de l'évènement	`E+EEE` `E+E-E`

Pour chaque fichier .pca analysé, le programme génère deux fichiers d'extensions « .instability » et « .events » dans le dossier de son exécution.

Le fichier .instability présente chaque paire de séquence ancêtre-descendant en y mettant en évidence les évènements évolutifs détectes. Des astérisques sont en dessous de chaque évènement évolutif dans chaque alignement de séquences ancêtre-descendant.

Le fichier .events contient les évènements détectés. Chaque évènement est décrit sur 3 lignes :

- La ligne 1 commence par @ et indique des informations sur l'évènement. Dans l'ordre :
  - Le nom de la famille de protéines.
  - Le nom de l'espèce ancestrale.
  - Le nom de l'espèce descendante.
  - La longueur de l'alignement de la paire de séquences ancêtre-descendant dans laquelle cet évènement a été trouvé.
  - La position de début de l'évènement dans l'alignement (les positions commencent à partir de 1).
  - La position de fin de l'évènement dans l'alignement (les positions commencent à partir de 1).
  - (après 3 tabulations, donc 9$^{\text{ème}}$ colonne) la forme linéaire du type déduit pour cet évènement.
  - (après 3 tabulations, donc 12$^{\text{ème}}$ colonne) la forme linéaire de l'évènement proprement dit (vraies séquences) dans le format « séquenceAncêtre/séquenceDescendant ».
- Les lignes 2 et 3 affichent le type de l'évènement et l'évènement proprement dit sur deux lignes pour une meilleure visualisation.

La figure 3.4 montre un exemple d'évènement mis en évidence dans un fichier .instability et dans un fichier .events .

```
1 oomycetes4738
2 root    ---SASRFAFQLDANNE+ECFMEENAAR
3   albu  ---SSSRIAFQVSANNH+ECFYEENAAR
4                          *****
```

(a) Extrait du fichier .instability de la famille oomycetes4738.

```
1 @oomycetes4738  root  albu  424  40  44  E+E/E+E  NE+EC/NH+EC
2   E+E       NE+EC
3   E+E       NH+EC
```

(b) Extrait du fichier .events de la famille oomycetes4738. La première ligne indique (dans l'ordre) : l'identifiant de la famille, l'espèce ancestrale, l'espèce descendante, la longueur de l'alignement, la colonne de départ et la colonne de fin de l'évènement dans l'alignement, le type de l'évènement et l'évènement proprement dit sur une ligne. Les lignes suivantes présentent l'évènement et son type sur deux lignes pour une meilleure visualisation.

Figure 3.4 – Exemples de contenus des fichiers de sortie pour la détection des évènements évolutifs.

Le programme génère aussi 3 fichiers de statistiques sur l'ensemble des évènements détectés :

- Un fichier « events.types.counted.sortedByCount » qui présente les types d'évènements et le nombre d'évènements dans chaque type. Les types sont triés par ordre décroissant de leurs nombres d'évènements. Les types les plus représentés sont donc visibles dans les premières lignes du fichier.
- Un fichier « events.types.counted.sortedByType » qui présente lui aussi les types d'évènements et le nombre d'évènements dans chaque type, mais en triant les types par ordre alphabétique.
- Un fichier « events.types.counted.sortedByBlood.tsv ». Il s'agit d'un tableau qui présente le nombre d'évènements par branche de l'arbre des espèces (sur les lignes) et par type d'évènement (sur les colonnes).

3.2.4 Application

3.2.4.1 Pertinence des évènements détectés

Nous avons exécuté l'algorithme sur les fichiers PCA obtenus pendant la reconstruction des séquences ancestrales des 1 924 familles de protéines strictement orthologues disponibles. Nous

avons détecté un total de 56 238 évènements évolutifs répartis dans 755 types d'évènements.

Le nombre de types d'évènements nous semble tout de même assez grand, et nos futurs travaux consisteront notamment à affiner la détection des évènements et leur typage, car il est probable que certains types soient en fait des compositions de types plus courts à décrire. Par exemple, nous avons trouvé un type « E+E+E/E+E+E » (26 évènements) qui semble décrire une suite d'évènements du type « E+E/E+E », mais que notre algorithme n'a pas divisé parce que les introns étaient assez proches dans les alignements. Nous devrons donc modifier les valeurs des paramètres p et q, considérer ces types à description longue comme des types vraisemblables et les analyser pour évaluer leur validité, considérer ces types comme composites et trouver une façon de les décomposer, ou choisir une approche appropriée en fonction de chaque évènement trouvé.

Les 755 types d'évènements mis en évidence contiennent aussi plusieurs types qui semblent décrire des insertions ou des suppressions de séquences contenant des introns aux extrémités des protéines. Nous avons par exemple le type « -.-.-.-E/E+E+E+EE » (insertion au début d'une protéine ?), avec 31 évènements, et le type « E+E+EE/-.-.-E » (suppression au début d'une protéine ?) avec 80 évènements. On pourrait considérer que certains de ces évènements qui modifient les extrémités des protéines ne sont pas de réels mutations introniques. En effet, il s'agit souvent de séquences entières qui sont ajoutées ou enlevées, et qui contiennent des introns. Les sites d'épissage des introns contenus dans ces séquences ne sont pas spécifiquement modifiés, et les introns ne sont pas non plus individuellement insérés ou supprimés. Nous devrons donc analyser plus précisément ces types d'évènements, pour éventuellement en ignorer plusieurs, ou pour les considérer seulement comme informations statistiques pour étudier les variations du nombre d'introns dans les lignées. De plus, il faut aussi noter que des erreurs d'assemblage du génome peuvent être à l'origine de ces types d'évènements. Par exemple, le début d'un gène peut être assemblé avec la fin d'un autre gène, ce qui produit des gènes chimériques, dont on déduit donc des protéines chimériques. Lorsque ces protéines sont alignées avec des protéines dont les gènes sont homologues aux deux gènes assemblés par erreur, de grands trous peuvent alors apparaître au début ou à la fin des alignements, et représentent les parties des deux

gènes malencontreusement assemblées. Nous devrons également développer une stratégie pour différencier les évènements déduits de ces erreurs et les véritables évènements.

3.2.4.2 Identification des types d'évènements

Il est plus facile d'étudier les types d'évènements si on peut leur attribuer un nom. Pour le faire, on peut imaginer une séquences de mutations précises susceptibles de générer un type particulier, et nommer ce type en fonction de ces mutations, ou on peut identifier la version simplifiée d'une mutation connue, et nommer le type associé avec le nom de cette mutation. Par exemple, nous imaginons une conversion simultanée d'une portion exonique en amont d'un intron et d'une portion exonique en aval du même intron, de telle manière que l'intron « absorbe » des acides aminés des exons qui lui sont frontaliers. Nous appelons un tel type d'évènement « intronization double 5' - 3' », et nous pouvons déduire facilement la version simplifiée de ce type :

```
E.E+EE
E+-.-E
```

Comme exemple d'identification de type à partir d'une mutation considérée comme connue, nous pouvons simplement déterminer la version simplifiée d'une insertion d'un intron complet dans un exon. La position de l'intron apparaît simplement dans la séquence descendante :

```
E.E
E+E
```

Cependant, il est très difficile de nommer les 755 types dont nous disposons. Tout d'abord, ils sont trop nombreux, et il est donc difficile de trouver un nom pour eux tous. Ensuite, comme nous l'avons expliqué, certains pourraient être idéalement ignorés, tandis que d'autres sont potentiellement composites, et ne devraient donc pas recevoir un nom particulier. De plus, il est tout à fait possible que plusieurs suites de mutations produisent les mêmes versions simplifiées. Il devient alors impossible d'associer à un type un nom qui décrit une suite précise de mutations, car ce n'est pas la seule suite susceptible de générer la version simplifiée d'un tel type. Par exemple,

nous appelons intronization 5' l'absorption par un intron d'acides aminés dans l'exon situé avant lui. La version simplifiée d'un tel évènement est donc :

E.E+E

E+−.E

Cependant, une suppression d'acides aminés exactement avant l'intron peut produire le même résultat dans notre méthode actuelle, étant donné que l'intron est simplement positionné dans la séquence après l'acide aminé ou le nucléotide (dans un codon) qui le précède :

Alignement idéal (suppression avant l'intron qui devrait être positionné après le trou qui indique la suppression)

EE+E

E−+E

Alignement apparent dans nos séquences (intron positioné après l'acide aminé qui le suit)

E.E+E

E+−.E

Nous ne pouvons donc pas nommer correctement tous les types que nous avons détectés. Cependant, nous pouvons déterminer les versions simplifiées reconnaissables de nombreuses mutations connues ou faciles à imaginer, et attribuer les noms de ces mutations en tant qu'interprétations possibles pour les types qui ont ces versions simplifiées. Par exemple, dans le cas du type qui peut désigner soit une intronization 5' soit une insertion d'acides aminés avant l'intron, nous utiliserons le nom « intronization 5' » comme interprétation possible de ce type, et donc comme nom par défaut pour ce type. Chaque évènement du type devra ensuite être analysé pour confirmer qu'il s'agit effectivement d'une intronization 5'. Cette analyse est également une étape à développer dans les futures versions de notre méthode, car nous n'avons pas encore de stratégie pour identifier clairement les réelles mutations associées aux évènements.

Le tableau 3.VI présente quelques types d'évènements aux allures caractéristiques auxquels nous avons attribué des interprétations possibles. Il s'agit notamment des pertes d'introns, gains

d'introns, ou modifications des sites d'épissage d'un seul intron à la fois. Les types sont triés par ordre décroissant de leur nombre d'évènements. Pour chacun de ces types, un évènement concret détecté dans les familles de protéines analysées est donné comme exemple dans le tableau 3.VII. L'évènement est décrit par l'identifiant de la famille de protéines dont il provient (exemple : « oomycetes4764 »), la branche sur laquelle il est détectée, et qui indique l'espèce ancestrale puis l'espèce descendante (exemple : « root -> albu », de la racine vers l'espèce *Albugo laibachii*), puis l'évènement proprement dit dans l'alignement de la séquence ancestrale et de la séquence descendante. La portion de la séquence ancestrale est présentée en haut, et celle de la séquence descendante en bas.

Nous avons ensuite effectué quelques analyses sur les types d'évènements du tableau 3.VII, pour en tirer diverses observations qui sont présentées dans les sections suivantes.

3.2.4.3 Évènements de conservation d'introns

Le type le plus représenté dans notre tableau et dans l'ensemble des évènements détectés est le type « E+E/E+E », avec 45 407 évènements (soit 80,74 % des évènements détectés), qui décrit la conservation de la présence d'un intron, sans trous aux alentours. Il est très probable que de nombreux évènements comptés dans ce type soient en fait des introns qui n'ont subi de modifications que dans leurs séquences. Ces évènements n'impliquent pas de changements des sites d'épissage, ni de changements de positions des introns. Ils ne sont donc pas forcément intéressants pour répondre aux questions relatives aux mutations introniques locales qui affectent les protéines. Nous les avons cependant collectés pour nous assurer de traiter tous les introns présents dans les séquences. Nous pourrons aussi analyser plus précisément ces évènements dans les prochaines versions de notre méthode, pour chercher d'éventuelles tendances évolutives relatives aux longueurs ou à la composition interne des introns de ces évènements.

3.2.4.4 Évènements d'insertions et de suppressions complètes d'introns

Après les évènements de conservation des présences d'introns, ce sont les types « E.E/E+E » et « E+E/E.E » qui sont les plus représentés dans notre tableau et dans l'ensemble des évènements

Tableau 3.VI – Information sur quelques types d'évènements reconnaissables.

Type	**Nombre d'évènements**	**Interprétation possible**	**Description**
E+E E+E	45 407	Conservation d'un intron	Les sites d'épissage de l'intron ne changent pas.
E.E E+E	2 862	Gain d'un intron	Un nouvel intron apparait totalement.
E+E E.E	1 717	Perte d'un intron	L'intron disparait totalement.
E.EE E+−E	558	Formation d'un intron dans un exon	Deux sites d'épissage donneur et accepteur apparaissent dans un exon.
E+EE E+−E	206	intronization 3'	Le site d'épissage accepteur de l'intron est déplacé plus en aval dans l'exon après l'intron.
E+−E E+EE	110	exonization 3'	Le site d'épissage accepteur de l'intron est déplacé vers l'intérieur de l'intron.
E+−.E E.E+E	66	exonization 5'	Le site d'épissage donneur de l'intron est déplacé vers l'intérieur de l'intron.
E+−E E.EE	61	Transformation d'un intron en exon	Les deux sites d'épissage de l'intron sont supprimés.
E.E+E E+−.E	46	intronization 5'	Le site d'épissage donneur de l'intron est déplacé plus en amont dans l'exon avant l'intron.
E+E.E E.E+E	45	Intron sliding en aval	Les deux sites d'épissage de l'intron sont déplacés simultanément vers le côté 3' (en aval) du gène.
E.E+E E+E.E	23	Intron sliding en amont	Les deux sites d'épissage de l'intron sont déplacés simultanément vers le côté 5' (en amont) du gène.
E.E+EE E+−.−E	14	intronization double 5' - 3'	Les deux sites d'épissage de l'intron sont déplacés simultanément vers l'extérieur de l'intron.
E+−.−E E.E+EE	3	exonization double 5' - 3'	Les deux sites d'épissage de l'intron sont déplacés simultanément vers l'intérieur de l'intron.

Tableau 3.VII – Exemples d'évènements détectés dans les familles de protéines orthologues pour les types d'évènements décrits dans le tableau 3.VI.

Type	**Interprétation possible**	**Exemple d'évènement**
E+E E+E	Conservation d'un intron	**oomycetes4764 :** root -> albu AK+LV AK+LV
E.E E+E	Gain d'un intron	**oomycetes4764 :** ancestor1 -> pyul VVE.ILE VVE+ILE
E+E E.E	Perte d'un intron	**oomycetes4764 :** root -> albu TEd+ddT LEd.ddT
E.EE E+-E	Formation d'un intron dans un exon	**oomycetes4848 :** ancestor6 -> phca ARA.VFKYALDTLPKEEAP ARA+----------EEAP
E+EE E+-E	intronization 3'	**oomycetes4764 :** ancestor2 -> hyal DE+LDAIGTKRFGGEQSGDR DE+----------SGDR
E+-E E+EE	exonization 3'	**oomycetes5411 :** ancestor5 -> phso ER+-MT ER+VMT
E+-.E E.E+E	exonization 5'	**oomycetes6577 :** ancestor3 -> phra T.VMYK+---------.AASDA T.VMYK.ARCLWLNPR+AASDE
E+-E E.EE	Transformation d'un intron en exon	**oomycetes6651 :** ancestor2 -> hyal PGV+-----------------------HLAYA PGM.VSHSLPRYLKHLTVSNCAPFRTFVQHLAYA
E.E+E E+-.E	intronization 5'	**oomycetes6577 :** ancestor7 -> phin YVST.VMYK+---------.AASD YVST+----.---------.AASD
E+E.E E.E+E	Intron sliding en aval	**oomycetes4843 :** ancestor6 -> phca QLK+Ma.aaLT QLK.Vp+ppLT
E.E+E E+E.E	Intron sliding en amont	**oomycetes5517 :** ancestor5 -> phci M.VIkk+kTE M+VIrr.rTE
E.E+EE E+-.-E	intronization double 5' - 3'	**oomycetes5686 :** ancestor2 -> hyal RLYS.VSNv+vvGTKHVIEEYLDTIVQSMNVV RLYS+----.----------------SMNVV
E+-.-E E.E+EE	exonization double 5' - 3'	**oomycetes6678 :** ancestor7 -> phin IESG+--.----------------IVPVM IESG.Vr+rrIYLWQPSLICGSYLWQIVPVM

détectés. Nous comptons 2 862 évènements pour le type « E.E/E+E » (5,09 % des évènements détectés) qui décrit l'insertion d'un intron, et 1 717 évènements pour le type « E+E/E.E » (3,05 % des évènements détectés) qui décrit la suppression d'un intron. Ces types suggèrent que plusieurs évènements de gains et de pertes d'introns se sont produits parmi les espèces que nous étudions.

En observant plus précisément le type d'évènement « E.E/E+E » qui décrit une insertion d'intron, nous constatons qu'aucun trou n'est présent autour de l'intron. Les exons frontaliers de l'intron ne semblent pas avoir été affectés pendant son insertion, en dehors d'éventuelles substitutions dans les nucléotides proches des sites d'épissage. Cette observation nous paraît possible seulement si l'intron a été entièrement inséré à partir de séquences qui ne se trouvaient pas déjà dans le gène. Les évènements évolutifs de ce type pourraient donc inclure beaucoup de mutations introniques globales, qui sont capables d'insérer des introns complets sans modifier les séquences codantes du gène. Il en est de même pour le type d'évènement « E+E/E.E » qui décrit une suppression d'intron et dans lequel les exons voisins ne sont pas non plus ou très peu affectés, alors qu'un intron entier est supprimé. À partir de ces observations, nous estimons que ces 2 types d'évènements font partie des principaux types à analyser pour repérer les mutations introniques globales, et les différencier ainsi des mutations introniques locales qui nous intéressent réellement. Une amélioration importante de notre méthode consistera donc à analyser les évènements de ces deux types, afin d'y rechercher les évènements qui correspondent à des mutations introniques globales.

3.2.4.5 Fichier-rapport sur les nombres d'évènements par branche de l'arbre des espèces

Les évènements sont également comptés en fonction des paires (ancêtre ; descendant direct), dans le fichier TSV produit par le programme. Nous n'avons pas encore conçu les programmes nécessaires à une analyse complète et approfondie de ce fichier, mais il est clair que nous pouvons en extraire de nombreuses informations statistiques, telles que les variations des quantités d'évènements en fonction des lignées, qui donneraient diverses indications sur les modes d'évolution des introns dans les familles de protéines strictement orthologues étudiées. Nous avons cependant analysé manuellement le fichier TSV pour en déduire quelques observations. Nous en avons

extrait les dénombrements des types d'évènements particuliers du tableau 3.VI pour chaque branche de l'arbre des espèces, et nous avons placé certains décomptes sur l'arbre pour chercher des tendances. Les figures 3.5, 3.6 et 3.7 présentent les valeurs étudiées. Elles sont analysées dans les sections suivantes.

3.2.4.6 Tendances de gains d'introns

La figure 3.5 présente le nombre de pertes et de gains d'introns complets pour chaque branche de l'arbre des espèces. Les évènements énumérés sont ceux des types « E.E/E+E » (gains d'introns) et « E+E/E.E » (pertes d'introns) qui sont susceptibles de contenir beaucoup de mutations globales. Nous pouvons remarquer que, sur les 16 branches de l'arbre, 12 présentent plus de gains que de pertes. Le constat est le même pour toutes les branches qui mènent directement aux espèces actuelles. Les nombres de gains sur les branches qui mènent à *Albugo laibachii* (812 gains) et *Pythium ultimum* (492 gains) sont très élevés lorsqu'on les compare aux valeurs sur les autres branches. Il faut cependant noter que ces espèces sont les seules représentantes de leurs genres (*Albugo* et *Pythium*) dans l'arbre, alors que le genre *Phytophthora* est représenté par 6 espèces. De ce fait, le sous-arbre du genre *Phytophthora* affiche plus d'ancêtres, donc plus de branches, sur lesquelles les évènements peuvent se répartir. Nous pensons que les nombres de gains sur les branches d'*Albugo laibachii* et *Pythium ultimum* seraient répartis en plus petits nombres sur plusieurs branches si nous pouvions ajouter d'autres espèces de ces genres dans l'arbre.

Dans tous les cas, ces observations suggèrent que les protéines des familles strictement orthologues étudiées ont évolué en gagnant plus d'introns qu'elles n'en perdaient. Il sera intéressant, lorsque nous aurons affiné notre méthode, de comprendre pourquoi ces espèces semblent avoir favorisé des gains d'introns dans ces familles de protéines, et notamment pourquoi autant de gains se seraient produits via des évènements qui pourraient être des mutations globales, donc des insertions d'introns libres ou de transposons, ou des recombinaisons génomiques, qui se seraient donc produites à plusieurs reprises et en grande quantité.

Une autre remarque peut être faite concernant les nombres d'introns affichés à droite des nœuds de l'arbre sur la figure. Nous pouvons constater qu'il n'y a pas une correspondance directe entre

le nombre d'intron d'un ancêtre, le nombre d'introns d'un de ses descendants, et le nombre de pertes et de gains sur la branche qui va de cet ancêtre à ce descendant. Par exemple, sur la branche d'ancestor2 vers *Hyaloperonospora arabidopsidis*, ancestor2 a 2 802 introns, 147 gains et 85 pertes. On pourrait en déduire que 2 802 + 147 - 85 = 2 864 introns devraient être comptés chez *Hyaloperonospora arabidopsidis*. Pourtant, ce dernier n'en a que 2 653. Il ne s'agit cependant pas d'une erreur, mais simplement du fait que tous les types d'évènements qui ont ajouté ou supprimé des introns ne sont pas parmi les types d'évènements particuliers que nous avons listés dans le tableau 3.VI. En dehors des pertes et des gains d'introns complets dénombrés sur la figure 3.5, plusieurs autres évènements peuvent avoir modifié la composition en introns des séquences des familles de protéines strictement orthologues analysées, par exemple les évènements décrits sur la figure 3.6.

3.2.4.7 Tendances de créations d'introns

La figure 3.6 présente le nombre de créations et de conversions d'introns complets pour chaque branche de l'arbre des espèces. Contrairement aux évènements de la figure 3.5 qui modifient des introns complets sans altérer les protéines, les évènements énumérés ici sont les évènements de type « E.EE/E+-E » et « E+-E/E.EE », qui peuvent désigner respectivement la conversion d'une portion exonique en intron, et la conversion d'un intron en séquence codante. Ces évènements sont donc potentiellement des mutations introniques locales qui affectent significativement les protéines en créant de nouveaux introns à partir des exons, ou en créant de nouveaux exons à partir des introns déjà présents dans les gènes. Nous remarquons cependant que, sur les 16 branches de l'arbre, 12 présentent plus de créations d'introns que de conversions d'introns. Le constat est le même pour 8 des 9 branches qui mènent directement aux espèces actuelles, seule l'espèce *Phytophthora parasitica* échappe à cette règle. Nous pouvons aussi remarquer, en comparant cette figure à la figure 3.5, que les branches sur lesquelles les gains d'introns sont plus nombreux que les pertes d'introns sur la figure 3.5 sont presque les mêmes que celles sur lesquelles les créations d'introns sont plus nombreuses que les conversions d'introns sur la figure 3.6. 10 branches ont ainsi le même comportement, et 2 échappent à cette règle : la branche de ancestor4 à ancestor6, qui a plus de pertes que de gains, et la branche de ancestor7 à *Phytophthora parasitica*, qui a plus

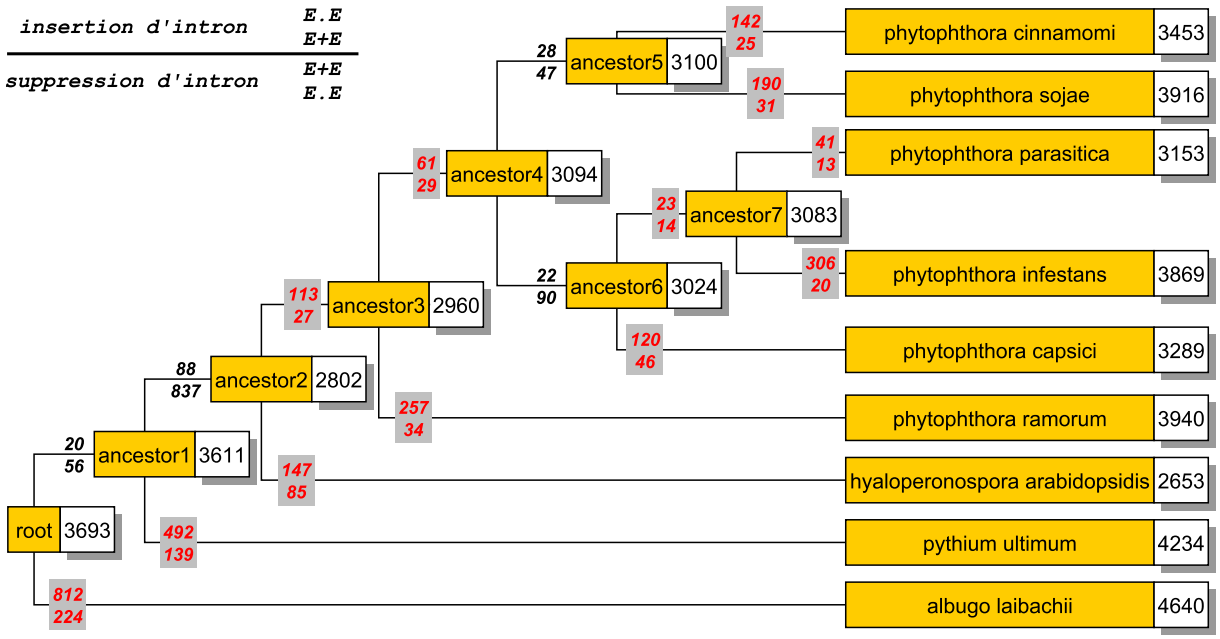


Figure 3.5 – Nombre de gains et de pertes d'introns comptés sur chaque branche de l'arbre des espèces. Le nombre d'introns trouvés dans les familles strictement orthologues pour chaque espèce est affiché à droite du nœud correspondant. Le nombre de gains est affiché en haut du nombre de pertes. Les étiquettes en rouge sur fond gris indiquent les branches sur lesquelles il y a plus de gains que de pertes.

de conversions d'introns que de créations d'introns.

Ces observations suggèrent que les protéines des familles strictement orthologues étudiées ont évolué en créant plus d'introns qu'ils n'en convertissaient en introns. Ces protéines ont donc subi plusieurs modifications au cours de l'évolution, en perdant des acides aminés qui étaient absorbés dans les introns qui étaient créés. De plus, la comparaison des deux figures 3.5 et 3.6 suggère aussi que l'évolution de ces protéines favoriserait l'apparition d'introns, soit par gains d'introns complets, soit par création à partir d'exons. Une telle tendance d'apparition d'introns dans les familles strictement orthologues devra être étudiée plus en profondeur lorsque nous aurons amélioré notre méthode.

3.2.4.8 Tendances de déplacement des sites d'épissage vers le côté 3' des gènes

La figure 3.7 présente le nombre de déplacements des sites d'épissage vers le côté 5' et vers le côté 3' des gènes pour chaque branche de l'arbre des espèces. Nous appelons déplacement d'un site d'épissage un changement qui rapproche un site d'épissage vers le début (du côté 5') ou vers la fin (du côté 3') du gène. Nous avons compté les déplacements susceptibles d'être générés par les types d'évènements particuliers montrés dans le tableau 3.VI. Parmi ces types, ceux qui décrivent les conservations, les gains, les pertes, les créations, et les conversions d'introns ne sont pas pris en compte car ils conservent, créent ou suppriment des sites d'épissage, mais ne changent pas les positions de sites déjà présents. De même, les types d'intronization double et d'exonization double sont ignorés. En effet, même s'ils modifient les positions des sites d'épissage, ils en déplacent toujours deux en même temps et dans des directions opposées, donc sans faire pencher le nombre de déplacements d'un côté ou d'un autre du gène. Ils ne peuvent donc pas fournir d'information statistique sur une éventuelle tendance de déplacement des sites d'épissage.

Les types d'évènements pris en compte sont donc les intronizations d'un seul côté, les exonizations d'un seul côté, et les glissements d'introns. Nous avons réparti ces types comme suit :

- Les déplacements des sites d'épissage vers le côté 5' du gène (en amont) sont provoqués

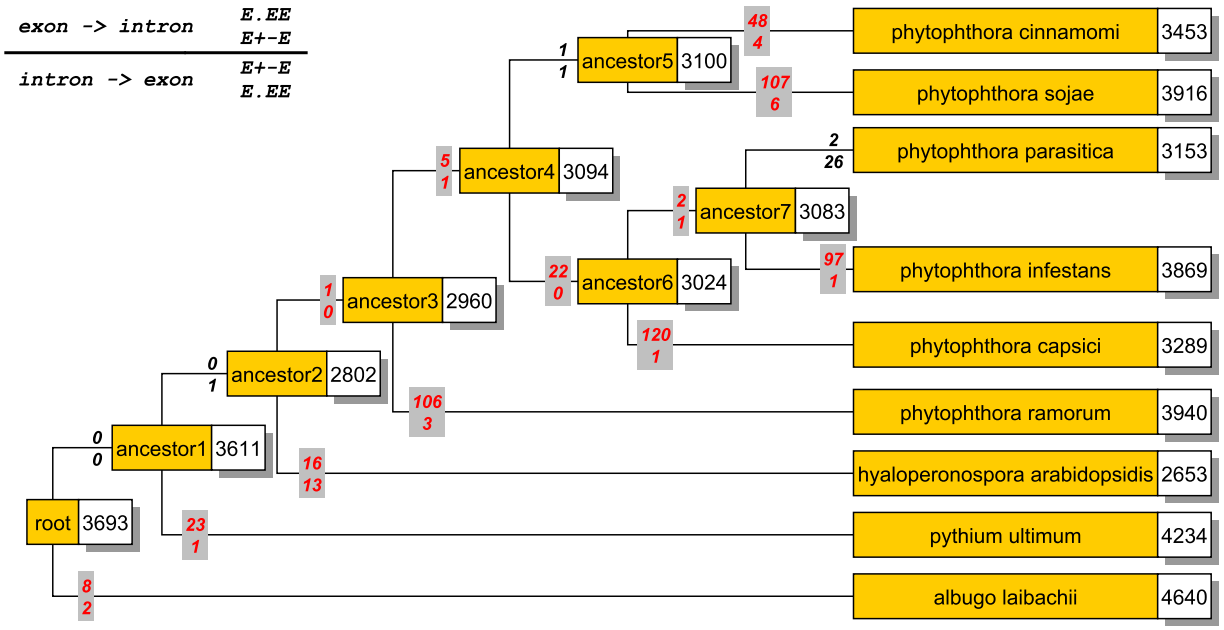


Figure 3.6 – Nombre de créations d'introns, et de conversions d'introns en exons, comptés sur chaque branche de l'arbre des espèces. Le nombre d'introns trouvés dans les familles strictement orthologues pour chaque espèce est affiché à droite du nœud correspondant. Le nombre de créations est affiché en haut du nombre de conversions. Les étiquettes en rouge sur fond gris indiquent les branches sur lesquelles il y a plus d'introns créés à partir d'exons que d'introns convertis en exons.

par les intronizations 5', les exonizations 3' et les glissements d'introns en amont. Une intronization 5' déplace un site d'épissage donneur vers le côté 5'. Une exonization 3' déplace un site d'épissage accepteur vers le côté 5' car un tel évènement rapproche la fin de l'intron vers le début du gène. Un glissement d'intron en amont déplace un intron entier en amont, donc un site d'épissage donneur et un site d'épissage accepteur sont simultanément déplacés vers le côté 5' du gène.

- Les déplacements des sites d'épissage vers le côté 3' du gène (en aval) sont provoqués par les intronizations 3', les exonizations 5' et les glissements d'introns en aval. Une intronization 3' déplace un site d'épissage accepteur vers le côté 3'. Une exonization 5' déplace un site d'épissage donneur vers le côté 3' car un tel évènement rapproche le début de l'intron vers la fin du gène. Un glissement d'intron en aval déplace un intron entier en aval, donc un site d'épissage donneur et un site d'épissage accepteur sont simultanément déplacés vers le côté 3' du gène.

Nous pouvons constater que, sur les 16 branches de l'arbre, 11 présentent plus de déplacements vers le côté 3' que vers le côté 5'. Toutes les branches menant aux espèces actuelles, sauf celle de *Phytophthora parasitica*, suivent la même tendance. Ces observations suggèrent que les gènes des protéines des familles strictement orthologues étudiées ont évolué en déplaçant à plusieurs reprises des sites d'épissage vers leurs extrémités 3'. Là encore, les futures améliorations de notre méthode nous aideront à mieux évaluer cette hypothèse.

Cependant, des études menées sur les mécanismes de pertes d'introns avaient déjà suggéré que les introns étaient plus souvent perdus dans l'extrémité 3' des gènes. Le mécanisme de perte proposé pour expliquer cette tendance est celui de la recombinaison génomique entre un gène et un ADN complémentaire issu de la rétrotranscription d'un ARN messager mature (donc sans introns) du même gène. Certaines transcriptases inverses (enzymes responsables de la rétrotranscription) qui génèrent les ADNs complémentaires peuvent commencer la lectures d'un ARN messager à son extrémité 3' mais arrêter prématurément la transcription inverse, par exemple si leur fonctionnement dépend de la longueur de l'ARN à rétro-transcrire. Elles pourraient donc

synthétiser des ADNs complémentaires incomplets, dont l'extrémité 5' est absente. Ainsi, lors de la recombinaison avec le gène, seule sa partie 3' sera effectivement recombinée, et seuls les introns situés à cet endroit sont susceptibles d'être enlevés [29, 47, 50].

Cette théorie nous permet de proposer une hypothèse quant à l'éventuelle tendance de déplacement des sites d'épissage vers le côté 3' des gènes dans les familles orthologues que nous étudions. En effet, les figures précédentes suggèrent que les évènements d'apparition d'introns ont été plus nombreux que les évènements de disparition dans les familles étudiées. Or, même si on ne sait pas encore expliquer la raison de ces gains et créations d'introns, on peut supposer qu'un excès d'introns dans un gène n'est pas non plus avantageux (car cela pourrait compliquer et ralentir l'épissage, et donc la synthèse des protéines). Il serait donc important pour ces espèces de limiter la quantité d'introns dans les gènes, et donc d'en supprimer. Or les suppressions semblent plus courantes vers le côté 3' du gène. Notre hypothèse est donc que les évènements qui déplacent les sites d'épissage vers le côté 3' seraient favorisés au cours de l'évolution de ces protéines, afin que quelques introns soient relocalisés dans la partie 3' à long terme, et donc que la probabilité qu'ils soient supprimés augmente. Cette hypothèse sera certainement très intéressante à vérifier dans les études ultérieures.

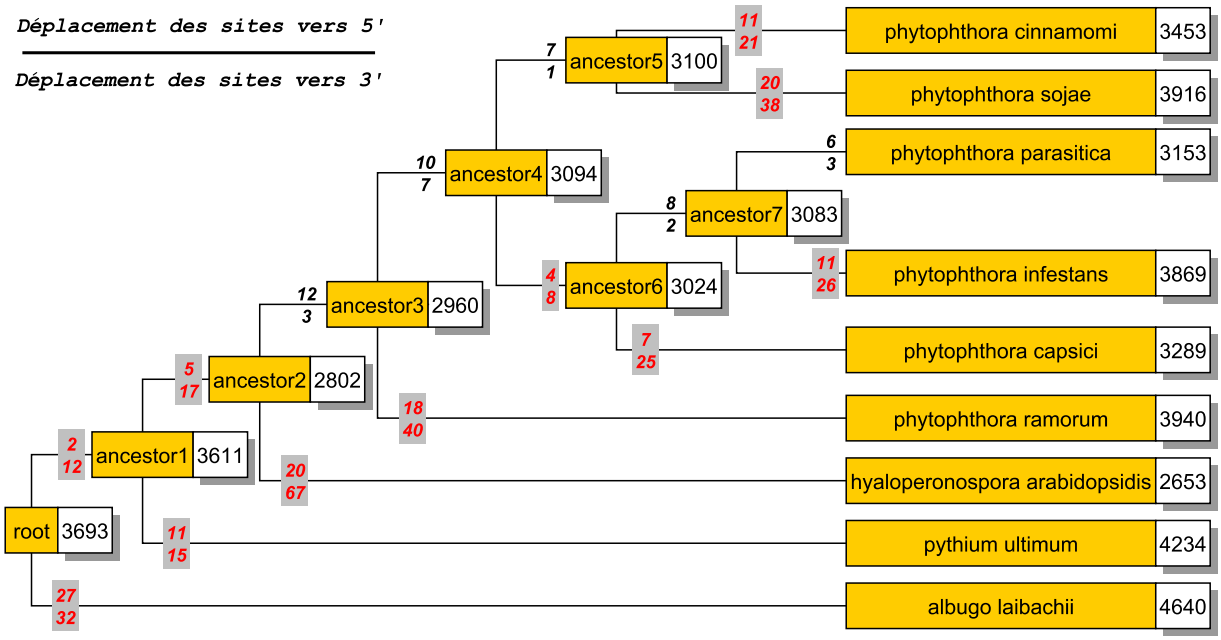


Figure 3.7 – Nombre de déplacements de sites d'épissage vers les extrémités des gènes, comptés sur chaque branche de l'arbre. Le nombre d'introns trouvés dans les familles strictement orthologues pour chaque espèce est affiché à droite du nœud correspondant. Le nombre de déplacements vers le côté 5' est affiché en haut du nombre de déplacements vers le côté 3'. Les étiquettes en rouge sur fond gris indiquent les branches sur lesquelles il y a plus de déplacements vers le côté 3' que vers le côté 5'.

CHAPITRE 4

CONCLUSION

4.1 Perspectives

La méthode que nous avons conçue permet à l'heure actuelle de détecter les évènements évolutifs qui affectent les introns dans les familles de protéines strictement orthologues d'un ensemble d'espèces données. La recherche des évènements est basée sur une hypothèse qui a pour but de trouver les mutations introniques locales, mais les évènements concrètement détectés incluent certainement beaucoup de mutations introniques globales, et probablement beaucoup de vrais négatifs, notamment parmi les évènements de type « E+E/E+E ». Plusieurs éléments de la méthode sont donc améliorables pour augmenter son efficacité.

Notamment, il pourrait être intéressant de généraliser la méthode pour qu'elle puisse étudier aussi les gènes qui codent pour des ARNs, et non plus seulement les gènes qui codent pour des protéines. Pour le faire, l'utilisation des gènes à la place des protéines pour déterminer les familles et construire les arbres doit être évaluée. Il serait aussi idéal de pouvoir analyser toutes les familles de gènes disponibles, donc aussi les familles de gènes paralogues ou les familles homologues qui ne sont pas strictement orthologues. L'étude de toutes les familles fournirait plus d'évènements, et donc plus de données statistiques analysables. Pour généraliser l'étude des familles, il faudrait pouvoir obtenir les arbres des gènes pour les familles de gènes, et étudier une famille avec son arbre de gènes associé. Cependant, dans la version actuelle de la méthode, on pourrait déjà étendre l'étude aux familles partiellement orthologues, c'est-à-dire les familles qui contiennent au plus 1 protéine par espèce, mais pas forcément une protéine pour toutes les espèces. Il suffirait en effet d'utiliser pour ces familles l'arbre des espèces disponible, en enlevant pour chaque famille de ce genre les feuilles de l'arbre qui correspondent aux espèces qui n'apparaissent pas dans la famille. Cette piste sera prochainement explorée.

La détection des évènements évolutifs autour des introns peut aussi être améliorée. Tout d'abord, il faudrait développer des techniques qui permettraient de différencier les mutations introniques globales et les mutations introniques locales, pour pouvoir réaliser des études comparatives entre ces deux catégories de mutations, et surtout pour analyser spécifiquement et avec confiance les mutations introniques locales qui nous intéressent. Parallèlement à cette différenciation, il faudra mettre en place la détection des artifices. Les artifices désignent des évènements évolutifs apparents, mais qui n'existent pas en réalité pour diverses raisons, telles que des erreurs de séquençage, des erreurs d'assemblage, des erreurs d'annotation des génomes, ou des erreurs d'alignements. Les erreurs d'alignements sont ici mises en évidence grâce à l'ajout des introns dans les alignements de protéines. L'étude de l'évolution des introns pourrait donc indirectement permettre d'améliorer les algorithmes d'alignement utilisés en bio-informatique.

Enfin, la méthode s'arrête actuellement à la détection des évènements, mais ne fournit pas d'outils pour l'analyse proprement dite de la collection d'évènements détectés. Nous avons montré que l'analyse du fichier TSV finalement généré, dans lequel les occurrences des types d'évènements sont comptées pour chaque branche de l'arbre des espèces, peut fournir beaucoup d'informations statistiques sur les tendances et les variations relatives aux types d'évènements, et nous pensons qu'un programme complet dédié à l'analyse de ce fichier devrait être écrit. Les autres fichiers disponibles, qui recensent les évènements détectés pour chaque famille de protéines, pourraient être analysés pour chercher d'autres tendances et corrélations dans les évolutions des introns. Nous pourrions notamment vérifier si les occurrences des types d'évènements varient en fonction des familles de protéines, et donc si l'évolution des introns dans une famille peut être soumise à la sélection subie par les protéines de ladite famille. Enfin, nous comptons évaluer la possibilité d'utiliser une base de données SQL pour sauvegarder les évènements détectés, au lieu des fichiers actuellement disponibles, car une base de données SQL permet généralement une meilleure organisation des données, et offre des options puissantes pour extraire, via des requêtes SQL, de nombreuses informations dérivées à partir des données initiales.

4.2 Résumé et disponibilité

La méthode que nous avons développée passe donc par plusieurs étapes et utilise plusieurs programmes externes combinés à des programmes et des scripts que nous avons conçus dans le cadre du projet. L'algorithme général de la méthode est récapitulé sous la forme d'un organigramme dans la figure 4.1, qui met aussi en évidence à quelles étapes nous avons utilisé des logiciels tiers, et à quelles étapes nous avons conçu des programmes spécifiques.

La méthode se présente actuellement sous la forme d'un ensemble de programmes JAVA et de scripts PHP à utiliser avec les logiciels tiers dans un ordre donné, décrit dans la figure 4.1. L'ensemble des codes nécessaires à l'utilisation de la méthode est disponible sur un dépôt en ligne publiquement accessible [11], qui contient également un guide d'utilisation détaillé pour permettre à tout utilisateur d'exécuter la méthode sur son propre jeu de données.

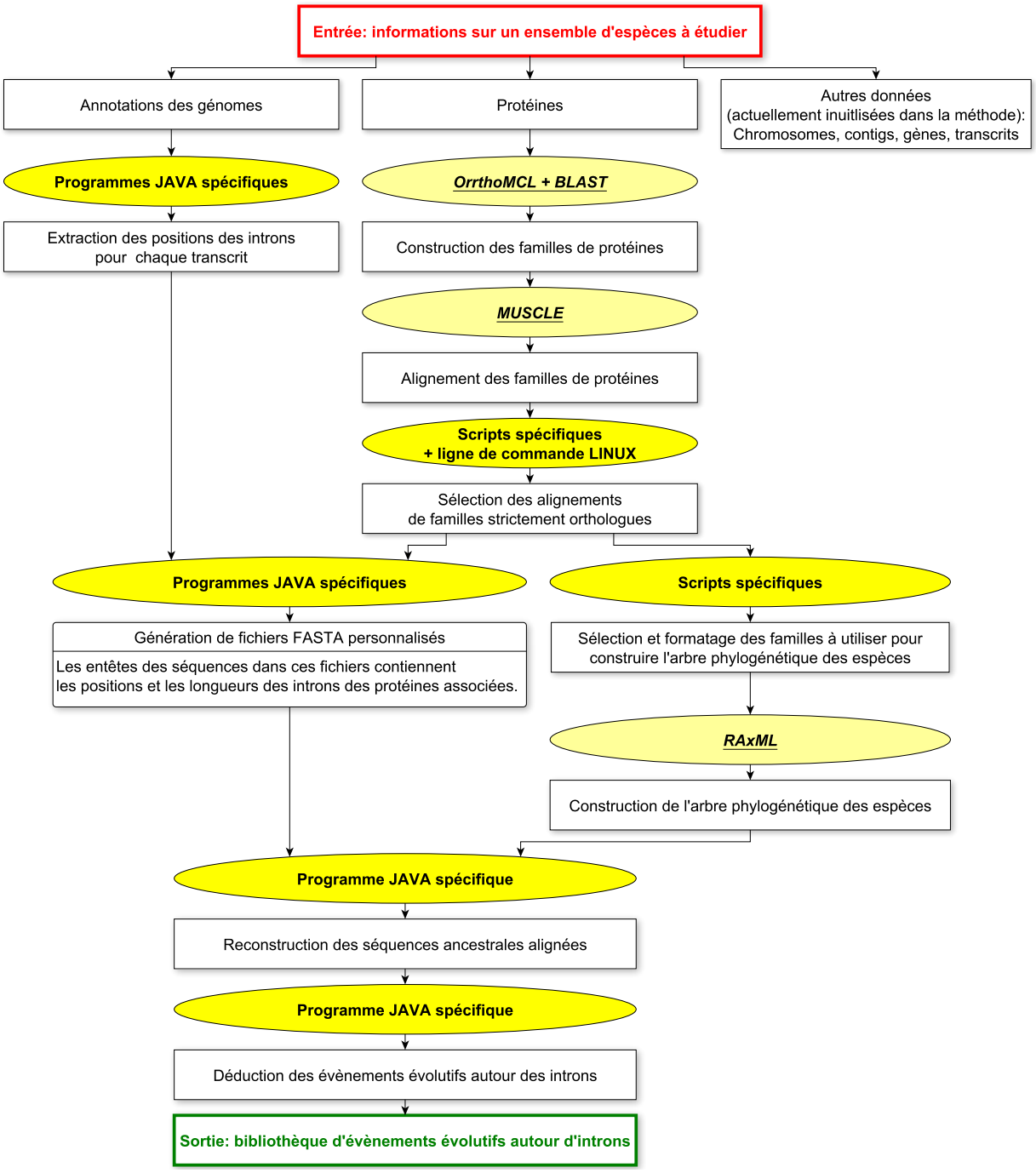


Figure 4.1 – Algorithme général de la méthode.

BIBLIOGRAPHIE

[1] Ingi Agnarsson et Jeremy A. Miller. Is acctran better than deltran ? *Cladistics*, 24(6): 1032–1038, 2008. ISSN 1096-0031. URL (consulté le 30 août 2015) http://dx.doi.org/10.1111/j.1096-0031.2008.00229.x.

[2] S. F. Altschul, W. Gish, W. Miller, E. W. Myers et D. J. Lipman. Basic local alignment search tool. *J Mol Biol*, 215(3):403–10, 1990. ISSN 0022-2836 (Print) 0022-2836 (Linking). URL (consulté le 30 août 2015) http://www.ncbi.nlm.nih.gov/pubmed/2231712.

[3] K. E. Baker et R. Parker. Nonsense-mediated mrna decay : terminating erroneous gene expression. *Curr Opin Cell Biol*, 16(3):293–9, 2004. ISSN 0955-0674 (Print) 0955-0674 (Linking). URL (consulté le 30 août 2015) http://www.ncbi.nlm.nih.gov/pubmed/15145354.

[4] D. L. Black. Mechanisms of alternative pre-messenger rna splicing. *Annu Rev Biochem*, 72: 291–336, 2003. ISSN 0066-4154 (Print) 0066-4154 (Linking). URL (consulté le 30 août 2015) http://www.ncbi.nlm.nih.gov/pubmed/12626338.

[5] Scott D. Boyd. Everything you wanted to know about small rna but were afraid to ask. *Lab Invest*, 88(6):569–578, 2008. ISSN 0023-6837. URL (consulté le 30 août 2015) http://dx.doi.org/10.1038/labinvest.2008.32.

[6] Broadinstitute.org. Broad institute of mit and harvard, 2015. URL (consulté le 30 août 2015) http://www.broadinstitute.org.

[7] Broadinstitute.org. Phytophthora parasitica inra-310 sequencing project, broad institute of harvard and mit (http ://www.broadinstitute.org/), 2015. URL (consulté le 30 août 2015) http://www.broadinstitute.org/annotation/genome/Phytophthora_parasitica/MultiDownloads.html.

[8] J. Castresana. Selection of conserved blocks from multiple alignments for their use in phylogenetic analysis. *Mol Biol Evol*, 17(4):540–52, 2000. ISSN 0737-4038 (Print) 0737-

4038 (Linking). URL (consulté le 30 août 2015) http://www.ncbi.nlm.nih.gov/pubmed/10742046.

[9] Y. F. Chang, J. S. Imam et M. F. Wilkinson. The nonsense-mediated decay rna surveillance pathway. *Annu Rev Biochem*, 76(1):51–74, 2007. ISSN 0066-4154 (Print) 0066-4154 (Linking). URL (consulté le 30 août 2015) http://www.ncbi.nlm.nih.gov/pubmed/17352659.

[10] Wikipedia contributors. Alternative splicing, 30 Septembre 2013 2013. URL (consulté le 30 août 2015) http://en.wikipedia.org/wiki/Alternative_splicing.

[11] Miklós Csűrös et Steven Bocco. Méthode bio-informatique pour la détection et l'analyse des mutations des sites d'épissage des introns eucaryotiques., 2015. URL (consulté le 30 août 2015) https://github.com/notoraptor/LIMA.

[12] M. Csuros. Malin : maximum likelihood analysis of intron evolution in eukaryotes. *Bioinformatics*, 24(13):1538–9, 2008. ISSN 1367-4811 (Electronic) 1367-4803 (Linking). URL (consulté le 30 août 2015) http://www.ncbi.nlm.nih.gov/pubmed/18474506.

[13] M. Csuros, J. A. Holey et I. B. Rogozin. In search of lost introns. *Bioinformatics*, 23(13): i87–96, 2007. ISSN 1367-4811 (Electronic) 1367-4803 (Linking). URL (consulté le 30 août 2015) http://www.ncbi.nlm.nih.gov/pubmed/17646350.

[14] Dev.mysql.com. Mysql : : Mysql 5.1 reference manual : : 1.3.1 what is mysql ?, 2015. URL (consulté le 30 août 2015) http://dev.mysql.com/doc/refman/5.1/en/what-is-mysql.html.

[15] R. C. Edgar. Muscle : multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Res*, 32(5):1792–7, 2004. ISSN 1362-4962 (Electronic) 0305-1048 (Linking). URL (consulté le 30 août 2015) http://www.ncbi.nlm.nih.gov/pubmed/15034147.

[16] Evolution.genetics.washington.edu. The newick tree format, 2015. URL (consulté le 30 août 2015) http://evolution.genetics.washington.edu/phylip/newicktree.html.

[17] A. Fedorov, G. Suboch, M. Bujakov et L. Fedorova. Analysis of nonuniformity in intron phase distribution. *Nucleic Acids Res*, 20(10):2553–7, 1992. ISSN 0305-1048 (Print) 0305-1048 (Linking). URL (consulté le 30 août 2015) http://www.ncbi.nlm.nih.gov/pubmed/1598214.

[18] S. Fischer, B. P. Brunk, F. Chen, X. Gao, O. S. Harb, J. B. Iodice, D. Shanmugam, D. S. Roos et Jr. Stoeckert, C. J. Using orthomcl to assign proteins to orthomcl-db groups or to cluster proteomes into new ortholog groups. *Curr Protoc Bioinformatics*, Chapter 6:Unit 6 12 1–19, 2011. ISSN 1934-340X (Electronic) 1934-3396 (Linking). URL (consulté le 30 août 2015) http://www.ncbi.nlm.nih.gov/pubmed/21901743.

[19] W. M. Fitch. Toward defining the course of evolution : Minimum change for a specific tree topology. *Systematic Biology*, 20(4):406–416, 1971. ISSN 1063-5157 1076-836X. URL (consulté le 30 août 2015) http://sysbio.oxfordjournals.org/content/20/4/406.

[20] M. B. Gerstein, C. Bruce, J. S. Rozowsky, D. Zheng, J. Du, J. O. Korbel, O. Emanuelsson, Z. D. Zhang, S. Weissman et M. Snyder. What is a gene, post-encode ? history and updated definition. *Genome Res*, 17(6):669–81, 2007. ISSN 1088-9051 (Print) 1088-9051 (Linking). URL (consulté le 30 août 2015) http://www.ncbi.nlm.nih.gov/pubmed/17567988.

[21] W. Gilbert. Genes-in-pieces revisited. *Science*, 228(4701):823–4, 1985. ISSN 0036-8075 (Print) 0036-8075 (Linking). URL (consulté le 30 août 2015) http://www.ncbi.nlm.nih.gov/pubmed/4001923.

[22] M. Gouy, S. Guindon et O. Gascuel. Seaview version 4 : A multiplatform graphical user interface for sequence alignment and phylogenetic tree building. *Mol Biol Evol*, 27(2):

221–4, 2010. ISSN 1537-1719 (Electronic) 0737-4038 (Linking). URL (consulté le 30 août 2015) http://www.ncbi.nlm.nih.gov/pubmed/19854763.

[23] P. J. Kersey, J. E. Allen, M. Christensen, P. Davis, L. J. Falin, C. Grabmueller, D. S. Hughes, J. Humphrey, A. Kerhornou, J. Khobova, N. Langridge, M. D. McDowall, U. Maheswari, G. Maslen, M. Nuhn, C. K. Ong, M. Paulini, H. Pedro, I. Toneva, M. A. Tuli, B. Walts, G. Williams, D. Wilson, K. Youens-Clark, M. K. Monaco, J. Stein, X. Wei, D. Ware, D. M. Bolser, K. L. Howe, E. Kulesha, D. Lawson et D. M. Staines. Ensembl genomes 2013 : scaling up access to genome-wide data. *Nucleic Acids Res*, 42(Database issue):D546–52, 2014. ISSN 1362-4962 (Electronic) 0305-1048 (Linking). URL (consulté le 30 août 2015) http://www.ncbi.nlm.nih.gov/pubmed/24163254.

[24] L. Li, Jr. Stoeckert, C. J. et D. S. Roos. Orthomcl : identification of ortholog groups for eukaryotic genomes. *Genome Res*, 13(9):2178–89, 2003. ISSN 1088-9051 (Print) 1088-9051 (Linking). URL (consulté le 30 août 2015) http://www.ncbi.nlm.nih.gov/pubmed/12952885.

[25] K. H. Lim, L. Ferraris, M. E. Filloux, B. J. Raphael et W. G. Fairbrother. Using positional distribution to identify splicing elements and predict pre-mrna processing defects in human genes. *Proc Natl Acad Sci U S A*, 108(27):11093–8, 2011. ISSN 1091-6490 (Electronic) 0027-8424 (Linking). URL (consulté le 30 août 2015) http://www.ncbi.nlm.nih.gov/pubmed/21685335.

[26] N. Lopez-Bigas, B. Audit, C. Ouzounis, G. Parra et R. Guigo. Are splicing mutations the most frequent cause of hereditary disease ? *FEBS Lett*, 579(9):1900–3, 2005. ISSN 0014-5793 (Print) 0014-5793 (Linking). URL (consulté le 30 août 2015) http://www.ncbi.nlm.nih.gov/pubmed/15792793.

[27] K. W. Lynch et T. Maniatis. Assembly of specific sr protein complexes on distinct regulatory elements of the drosophila doublesex splicing enhancer. *Genes Dev*, 10(16):2089–101, 1996. ISSN 0890-9369 (Print) 0890-9369 (Linking). URL (consulté le 30 août 2015) http://www.ncbi.nlm.nih.gov/pubmed/8769651.

[28] Mblab.wustl.edu. Gtf2 : Mouse/human annotation collaboration : Submission format, 2015. URL (consulté le 30 août 2015) http://mblab.wustl.edu/GTF2.html.

[29] T. Mourier et D. C. Jeffares. Eukaryotic intron loss. *Science*, 300(5624):1393, 2003. ISSN 1095-9203 (Electronic) 0036-8075 (Linking). URL (consulté le 30 août 2015) http://www.ncbi.nlm.nih.gov/pubmed/12775832.

[30] H. Nordberg, M. Cantor, S. Dusheyko, S. Hua, A. Poliakov, I. Shabalov, T. Smirnova, I. V. Grigoriev et I. Dubchak. The genome portal of the department of energy joint genome institute : 2014 updates. *Nucleic Acids Res*, 42(Database issue):D26–31, 2014. ISSN 1362-4962 (Electronic) 0305-1048 (Linking). URL (consulté le 30 août 2015) http://www.ncbi.nlm.nih.gov/pubmed/24225321.

[31] Marcin Nowicki, Majid R. Foolad, Marzena Nowakowska et Elzbieta U. Kozik. Potato and tomato late blight caused by phytophthora infestans : An overview of pathology and resistance breeding. *Plant Disease*, 96(1):4–17, 2011. ISSN 0191-2917. URL (consulté le 30 août 2015) http://dx.doi.org/10.1094/PDIS-05-11-0458.

[32] S. Osawa, T. H. Jukes, K. Watanabe et A. Muto. Recent evidence for evolution of the genetic code. *Microbiol Rev*, 56(1):229–64, 1992. ISSN 0146-0749 (Print) 0146-0749 (Linking). URL (consulté le 30 août 2015) http://www.ncbi.nlm.nih.gov/pubmed/1579111.

[33] Q. Pan, O. Shai, L. J. Lee, B. J. Frey et B. J. Blencowe. Deep surveying of alternative splicing complexity in the human transcriptome by high-throughput sequencing. *Nat Genet*, 40(12):1413–5, 2008. ISSN 1546-1718 (Electronic) 1061-4036 (Linking). URL (consulté le 30 août 2015) http://www.ncbi.nlm.nih.gov/pubmed/18978789.

[34] Ann B. Petersen et SØren Rosendahl. Phylogeny of the peronosporomycetes (oomycota) based on partial sequences of the large ribosomal subunit (lsu rdna). *Mycological Research*, 104(11):1295–1303, 2000. ISSN 0953-7562. URL (consulté le 30 août 2015) http://www.sciencedirect.com/science/article/pii/S0953756208614823.

[35] The Sequence Ontology Project. Generic feature format version 3 (gff3), 2013. URL (consulté le 30 août 2015) http://www.sequenceontology.org/resources/gff3.html.

[36] Protists.ensembl.org. Albugo laibachii - ensembl genomes, 2015. URL (consulté le 30 août 2015) http://protists.ensembl.org/Albugo_laibachii/Info/Index/.

[37] Protists.ensembl.org. Hyaloperonospora arabidopsidis - ensembl genomes, 2015. URL (consulté le 30 août 2015) http://protists.ensembl.org/Hyaloperonospora_arabidopsidis/Info/Index/.

[38] Protists.ensembl.org. Phytophthora infestans t30-4 - ensembl genomes, 2015. URL (consulté le 30 août 2015) http://protists.ensembl.org/Phytophthora_infestans/Info/Index/.

[39] Protists.ensembl.org. Phytophthora ramorum - ensembl genomes, 2015. URL (consulté le 30 août 2015) http://protists.ensembl.org/Phytophthora_ramorum/Info/Index/.

[40] Protists.ensembl.org. Phytophthora sojae - ensembl genomes, 2015. URL (consulté le 30 août 2015) http://protists.ensembl.org/Phytophthora_sojae/Info/Index/.

[41] Protists.ensembl.org. Pythium ultimum daom br144 - ensembl genomes, 2015. URL (consulté le 30 août 2015) http://protists.ensembl.org/Pythium_ultimum/Info/Index/.

[42] D. Rearick, A. Prakash, A. McSweeny, S. S. Shepard, L. Fedorova et A. Fedorov. Critical association of ncrna with introns. *Nucleic Acids Res*, 39(6):2357–66, 2011. ISSN 1362-4962 (Electronic) 0305-1048 (Linking). URL (consulté le 30 août 2015) http://www.ncbi.nlm.nih.gov/pubmed/21071396.

[43] Jeffrey Rogers et Richard A. Gibbs. Comparative primate genomics : emerging patterns of genome content and dynamics. *Nat Rev Genet*, 15(5):347–359, 2014. ISSN 1471-0056. URL (consulté le 30 août 2015) http://dx.doi.org/10.1038/nrg3707.

[44] I. B. Rogozin, L. Carmel, M. Csuros et E. V. Koonin. Origin and evolution of spliceosomal introns. *Biol Direct*, 7(1):11, 2012. ISSN 1745-6150 (Electronic) 1745-6150 (Linking). URL (consulté le 30 août 2015) http://www.ncbi.nlm.nih.gov/pubmed/22507701.

[45] M. F. Seidl, G. Van den Ackerveken, F. Govers et B. Snel. Reconstruction of oomycete genome evolution identifies differences in evolutionary trajectories leading to present-day large gene families. *Genome Biol Evol*, 4(3):199–211, 2012. ISSN 1759-6653 (Electronic) 1759-6653 (Linking). URL (consulté le 30 août 2015) http://www.ncbi.nlm.nih.gov/pubmed/22230142.

[46] A. Stamatakis. Raxml-vi-hpc : maximum likelihood-based phylogenetic analyses with thousands of taxa and mixed models. *Bioinformatics*, 22(21):2688–90, 2006. ISSN 1367-4811 (Electronic) 1367-4803 (Linking). URL (consulté le 30 août 2015) http://www.ncbi.nlm.nih.gov/pubmed/16928733.

[47] A. V. Sverdlov, V. N. Babenko, I. B. Rogozin et E. V. Koonin. Preferential loss and gain of introns in 3' portions of genes suggests a reverse-transcription mechanism of intron insertion. *Gene*, 338(1):85–91, 2004. ISSN 0378-1119 (Print) 0378-1119 (Linking). URL (consulté le 30 août 2015) http://www.ncbi.nlm.nih.gov/pubmed/15302409.

[48] R. Tarrio, F. J. Ayala et F. Rodriguez-Trelles. Alternative splicing : a missing piece in the puzzle of intron gain. *Proc Natl Acad Sci U S A*, 105(20):7223–8, 2008. ISSN 1091-6490 (Electronic) 0027-8424 (Linking). URL (consulté le 30 août 2015) http://www.ncbi.nlm.nih.gov/pubmed/18463286.

[49] T. Warnow. Standard maximum likelihood analyses of alignments with gaps can be statistically inconsistent. *PLoS Curr*, 4:RRN1308, 2012. ISSN 2157-3999 (Electro-

nic). URL (consulté le 30 août 2015) http://www.ncbi.nlm.nih.gov/pubmed/22453901.

[50] A. M. Weiner, P. L. Deininger et A. Efstratiadis. Nonviral retroposons : genes, pseudogenes, and transposable elements generated by the reverse flow of genetic information. *Annu Rev Biochem*, 55:631–61, 1986. ISSN 0066-4154 (Print) 0066-4154 (Linking). URL (consulté le 30 août 2015) http://www.ncbi.nlm.nih.gov/pubmed/2427017.

[51] Simon Whelan, Pietro Liò et Nick Goldman. Molecular phylogenetics : state-of-the-art methods for looking into the past. *Trends in Genetics*, 17(5):262–272, 2001. ISSN 01689525. URL (consulté le 30 août 2015) http://www.sciencedirect.com/science/article/pii/S0168952501022727.

[52] M. D. Wilkerson, Y. Ru et V. P. Brendel. Common introns within orthologous genes : software and application to plants. *Brief Bioinform*, 10(6):631–44, 2009. ISSN 1477-4054 (Electronic) 1467-5463 (Linking). URL (consulté le 30 août 2015) http://www.ncbi.nlm.nih.gov/pubmed/19933210.

[53] Scott William Roy et Walter Gilbert. The evolution of spliceosomal introns : patterns, puzzles and progress. *Nat Rev Genet*, 7(3):211–221, 2006. ISSN 1471-0056. URL (consulté le 30 août 2015) http://dx.doi.org/10.1038/nrg1807.