INSA Lyon - Biosciences - 4 BIM

Rapport de stage :

Développement d'un outil statistique visant à améliorer les inférences phylogénétiques à partir de données temporelles.

Auteur : Amaury Prin

Tuteur : Adrien Rieux

Référant : Fabien Subtil

Du 3 Mai 2018 au 31 Juillet 2018

Sommaire

Remerciements

Je souhaite remercier toutes les personnes qui ont contribué directement ou indirectement au succès et au bon déroulement de mon stage.

En premier lieu, je remercie vivement mon maître de stage, Adrien Rieux, pour son accueil chaleureux, pour la confiance accordée et pour les nombreux conseils avisés qui m'ont permis de mener à bien ce projet.

Je remercie également l'équipe du CIRAD pour le très bon accueil et la coopération professionnelle au cours de ces trois mois. En particulier, Frédéric Chiroleu, pour son expertise en statistique ainsi que sa connaissance pointue du logiciel R et Pierre Lefeuvre, pour ses conseils en programmation, qui m'ont été particulièrement bénéfiques dans les moments délicats de mon implémentation.

Un grand merci également à Sylvain Falala et Guillaume Cornu, deux formateurs et spécialistes du package Shiny, pour le temps passé depuis la Métropole et les conseils prodigués pour améliorer mon application.

Enfin, je tenais à remercier l'équipe de direction du département Biosciences de l'INSA Lyon pour m'avoir donné accès à cette proposition de stage.

1 Unité et organisme d'accueil

1.1 Le CIRAD

Dans le cadre de ma formation à l'INSA, j'ai choisi d'effectuer mon stage de 4ème année au sein d'un organisme de recherche : le Centre de coopération Internationale en Recherche Agronomique pour le Développement (CIRAD). Ce stage a commencé le 3 mai 2018 et s'est conclu le 31 juillet 2018.

Crée en 1984, le CIRAD est un établissement public français à caractère industriel et commercial (EPIC). C'est un organisme spécialisé dans la recherche agronomique et la collaboration internationale, destiné au développement durable dans les zones tropicales et méditerranéennes (`https://www.cirad.fr`). Il réalise des actions de recherche qui sont appliquées à l'agriculture, à l'alimentation, à l'environnement et à la gestion des territoires. C'est un établissement qui se démarque par son engagement dans de nombreux réseaux européens et internationaux, ainsi que par ses nombreuses applications dans le domaine industriel. La principale mission du CIRAD est de développer une expertise et une connaissance suffisamment forte dans l'objectif

d'aider au développement agricole. L'ambition du CIRAD, à long terme, est de faire concilier d'une part, une agriculture durable, capable d'approvisionner 10 milliards d'êtres humains d'ici une trentaine d'années et d'autre part, d'acquérir une capacité d'adaptation aux changements climatiques, tout en préservant l'environnement.

1.2 L'UMR PVBMT

Le CIRAD est structuré en 3 départements scientifiques et 33 unités de recherche. J'ai effectué mon stage au sein de l'unité mixte de recherche « Peuplements végétaux et Bio-agresseurs en Milieu Tropical » (UMR-PVBMT) basée à Saint-Pierre de la Réunion. Cette unité qui abrite le « Pôle de Protection des Plantes » (3P) a été créée en 2003 et regroupe des chercheurs du CIRAD, de l'INRA, de l'Université de la Réunion et du Muséum national d'histoire naturelle de Paris (MNHN). Les recherches effectuées par l'unité portent sur la protection des cultures et la sauvegarde de la biodiversité des écosystèmes terrestres. L'activité scientifique de l'unité est structurée en trois thématiques : 1) Génomique et épidémiologie des agents pathogènes émergents, 2) Diversité et utilisation durable des ressources génétiques végétales en milieu tropical et 3) Dynamiques écologiques en milieu insulaire.

1.3 Origine et déroulé du projet

Mon intérêt pour le projet s'est manifesté assez rapidement du fait de mon attrait pour la phylogénie. Le sujet d'étude laissait également la possibilité de développer une interface graphique sous forme de package, ce que je trouvais particulièrement plaisant. De plus, l'utilisation du logiciel R, un langage de programmation familier que j'affectionne particulièrement a renforcé mon choix. Enfin, ce premier stage professionnel me permettait de mettre en application mes connaissances diverses acquises durant mes études supérieures, à la fois sur un plan biologique, au niveau de la programmation informatique et en terme de compréhension statistique, le tout au sein d'une équipe de recherche.

Bien que la majorité des études menées au 3P ciblent (directement ou indirectement) les espèces végétales, la thématique de mon stage s'est inscrite dans un cadre plus généraliste, avec des applications à des modèles d'études diversifiés. Dès mon arrivée, j'ai été pris en charge par mon encadrant, Adrien Rieux, qui est un chercheur en génétique des populations, spécialisé dans l'étude et le suivi des agents pathogènes à l'origine de maladies infectieuses. Mon stage s'est subdivisé en plusieurs séquences chevauchantes. La première phase a concerné l'approche bibliographique pour la compréhension du sujet et l'initiation à la phylogénie sur le langage de programmation R. La seconde phase a consisté en l'implémentation progressive d'une application interactive. Enfin, une dernière phase m'a permis de tester différents jeux de données sur cette application, de comprendre les tenants et les aboutissants de mon étude et de mettre en avant les perspectives qui découlent de ce projet.

2 Introduction

Une maladie infectieuse émergente (MIE) est une maladie infectieuse, nouvellement apparue ou déjà existante, pour laquelle l'incidence, la virulence ou la portée géographique a considérablement augmenté au cours des dernières années ([Morens et al., 2004]). Les causes de l'émergence de ces maladies sont multiples et diverses. Des études ont montré que ces dernières peuvent être corrélées à des facteurs socio-économiques, évolutifs, écologiques et environnementaux ([Jones et al., 2008]). Parmi ceux-ci, on retrouve des changements climatiques, des transformations démographiques, des adaptations microbiennes ou encore le retour de maladies ré-émergentes. Ces maladies infectieuses, qui touchent aussi bien l'Homme que les animaux et les végétaux, constituent un réel danger à la fois pour la santé et la sécurité alimentaire mais aussi pour la biodiversité et l'économie agricole ([Wilkinson et al., 2011]).

Afin de mieux contrôler les MIE actuelles et prévenir les épidémies futures, il est nécessaire de mieux comprendre les facteurs contrôlant l'émergence, l'adaptation et la diffusion des agents pathogènes ([Gilligan and van den Bosch, 2008]). Pour ce faire, une stratégie possible consiste à adopter une approche « d'épidémiologie moléculaire » qui se définit par l'utilisation de marqueurs génétiques pour mesurer la diversité et la structure des populations d'agents pathogènes et ainsi tenter de reconstuire leur histoire évolutive passée ([Croucher and Didelot, 2015]).

Parmi les différentes méthodes permettant de reconstruire l'histoire évolutive passée des populations d'agents pathogènes à partir de données génétiques, les inférences phylogénétiques se sont montrées particulièrement efficaces et utiles ([Li et al., 2014]). Reconstruire un arbre phylogénétique consiste à reconstituer les relations de parenté d'une famille de gènes ou d'une espèce. Les outils qui permettent de générer de tels arbres reposent sur différents types de méthodes statistiques ([Saitou and Imanishi, 1989]). A partir d'une phylogénie, il devient possible de réaliser plusieurs types d'inférences comme la datation des ancêtres communs, l'estimation du taux de mutation, la reconstruction des états ancestraux ou encore une estimation de la démographie passée. Comme illustrées dans la figure 1 (tirée du papier de [Li et al., 2014]), de telles inférences représentent des outils complémentaires pour élucider l'origine et le mode de transmission d'un agent infectieux.

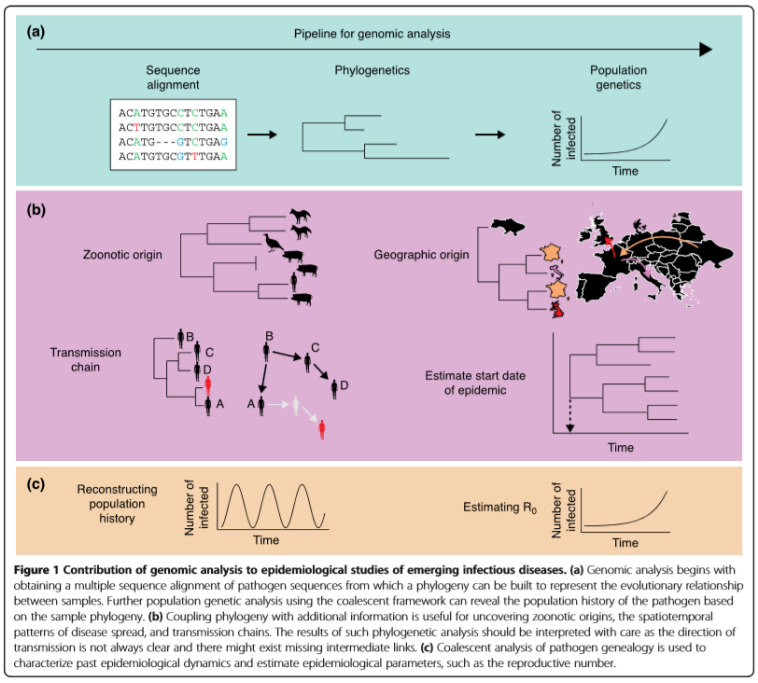


Figure 1 Contribution of genomic analysis to epidemiological studies of emerging infectious diseases. (a) Genomic analysis begins with obtaining a multiple sequence alignment of pathogen sequences from which a phylogeny can be built to represent the evolutionary relationship between samples. Further population genetic analysis using the coalescent framework can reveal the population history of the pathogen based on the sample phylogeny. **(b)** Coupling phylogeny with additional information is useful for uncovering zoonotic origins, the spatiotemporal patterns of disease spread, and transmission chains. The results of such phylogenetic analysis should be interpreted with care as the direction of transmission is not always clear and there might exist missing intermediate links. **(c)** Coalescent analysis of pathogen genealogy is used to characterize past epidemiological dynamics and estimate epidemiological parameters, such as the reproductive number.

Les analyses génomiques apportent des informations précieuses dans les études épidémiologiques. En premier lieu, les alignements multiples de séquences permettent de reconstruire une phylogénie **(a)**.
Ces phylogénies peuvent être ensuite utilisées pour estimer l'origine, les chaines de transmissions et la date d'origine d'une épidémie **(b)**. Enfin, la dynamique d'une population et des paramètres épidémiologiques peuvent aussi être estimés à l'aide de cette méthode **(c)**.

FIGURE 1 – Quelques applications épidémiologiques d'une étude phylogénétique.

Par le passé, les études phylogénétiques ont souvent permis de mieux comprendre la date et l'origine d'une épidémie. Un exemple célèbre reste le cas du VIH. Dans une étude récente de [Faria et al., 2014], des séquences de ce virus provenant de souches échantillonnées à différents moments (entre 1959 et 2015) et endroits dans le monde ont permis de reconstruire l'histoire épidémique de la maladie de façon plus précise. Plus particulièrement, les analyses phylogénétiques réalisées dans cette étude ont mis en évidence une origine probable de la pandémie dans la ville de Kinshasa (RDC) autour des années 1920-1930. Les résultats de cette étude ont permis de suggérer également que différents facteurs socio-culturels comme des campagnes locales de vaccination contre d'autres maladies telles que les hépatites (faisant usage de matériel médical mal stérilisé) ou la mise en service d'un réseau ferroviaire en RDC pourraient avoir favorisé la diffusion du virus à plus grande échelle. Cet exemple historique illustre l'utilité des inférences phylogénétiques dans la compréhension des facteurs de diffusion d'un agent pathogène. Dans le cadre de mon stage, je me suis tout particulièrement intéresser à l'analyse et aux conditions d'applications des datations phylogénétiques.

Le concept de datation phylogénétique a été introduit pour la première fois en 1962 par [Zuckerkandl and Pauling, 1962], qui ont suggéré que le temps de divergence entre des espèces pouvaient être approximé par le nombre de mutations existant entre leurs séquences nucléotidiques. En phylogénie, calibrer un arbre phylogénétique revient à convertir cette divergence moléculaire en temps absolu. La datation d'un arbre phylogénétique peut s'effectuer à l'aide de différentes approches, qui peuvent être utilisées de façon indépendante ou conjointe :

- « **Rate dating** » : lorsque le taux de mutation de l'espèce étudiée ou d'une espèce phylogénétiquement proche est connu (de part une estimation réalisée dans le cadre d'une

autre étude ou obtenue grâce a des experiences *in vitro/vivo*), il est possible d'utiliser ce taux de mutation (exprimé en nombre de substitutions par unité de temps) pour convertir la divergence moléculaire (exprimée en nombre de substitutions) en unité de temps absolu. Cette approche est discutable dans la mesure où l'on connait rarement avec précision les taux de mutations. De plus, ces derniers peuvent varier en fonction de différents facteurs. Cette méthodologie peut ainsi générer des datations incertaines voir erronées ([Ho and Duchêne, 2014]).

- « **Node dating** » : cette approche consiste à associer à certains noeuds internes de l'arbre (représentant des ancêtres communs à certains clades ou souches) des dates à l'aide d'informations sur le passé tels que des fossiles datés et/ou des évènements géologiques ou historiques connus ([Ho et al., 2011]). Cette approche a rarement été appliquée dans le cas des micro-organismes pour la raison principale que ces derniers ne laissent que très rarement des traces observables dans les restes fossiles.

- « **Tip dating** » : cette approche consiste à associer des dates aux noeuds externes d'un arbre phylogénétique (appelés feuilles ou extrémités - *tips* en anglais). Le « tip-dating » n'est applicable que dans le cas où les individus qui constituent les feuilles de l'arbre ont été échantillonnés à différents moments dans le temps (échantillonnage hétérochrone). Aujourd'hui, cette approche est devenue très répandue grâce aux développements réalisés dans le domaine de la « paléogénomique » qui consiste en l'étude de l'ADN ancien collecté à partir de matériel *post-mortem* ([Leonardi et al., 2017]).

Dans le cadre de mon stage, je me suis particulièrement préoccupé des conditions requises pour pouvoir appliquer de façon robuste la méthode du « tip-dating ». En plus d'avoir un jeu de données hétérochrone, cette approche nécessite que le signal temporel existant au sein d'un jeu de données soit suffisant. Tester la présence de signal temporel dans un jeu de données revient à vérifier si l'accumulation progressive des mutations dans les génomes des individus échantillonnés à différentes dates est mesurable. Une des méthodes communément utilisée pour tester la présence d'un signal temporel suffisant repose sur un test paramétrique : la régression linéaire (cf.figure 2) entre les distances feuilles-racines et les dates d'échantillonnages ([Buonagurio et al., 1986], [Drummond et al., 2003]). A l'heure actuelle, cette méthode présente le défaut de devoir *a priori* choisir l'échelle évolutive à laquelle la régression est effectuée, ce qui comme illustré par la figure 2 peut mener à des signaux contradictoires. Or, rappelons que la vérification de l'intensité du signal temporel dans un jeu de données conditionne la validité des inférences phylogénétiques réalisées en aval. Dans la figure 2, le signal temporel est absent à l'échelle de l'arbre phylogénétique (pente en *pointillé noir* de la régression qui est négative). En revanche, ce signal temporel est présent (pente positive) à certaines échelles évolutives locales de l'arbre (*pointillé bleu et rouge*).

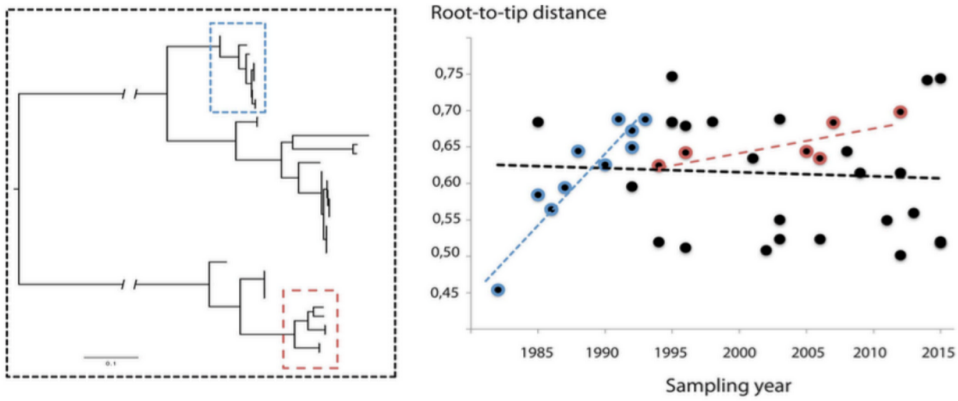


FIGURE 2 – Contrôle du signal temporel à différentes échelles évolutives.

L'objectif de ce stage est d'améliorer la méthode actuelle d'évaluation du signal temporel en identifiant de façon spécifique l'échelle évolutive (le/les noeud(s)) pour laquelle le signal est significativement positif dans un arbre phylogénétique. Sur la forme, ce projet s'effectuera par le développement d'un outil statistique, simple d'utilisation, qui permettra de vérifier de façon visuelle, l'échelle évolutive à laquelle le signal temporel est statistiquement positif. Cet outil permettra à n'importe quel utilisateur de connaître l'échelle évolutive à laquelle une calibration de l'arbre phylogénétique par la date des feuilles (tip-dating) pourra être effectuée.

3 Matériels, méthodes et modèles utilisés

3.1 Choix du langage de programmation

L'implémentation de cette interface peut se réaliser à l'aide de différents langages de programmation. Pour cette mission, le choix du logiciel R a semblé le plus judicieux. D'une part, il permet une bonne visualisation et manipulation des arbres phylogénétiques à l'aide de différents packages. D'autre part, il offre la possibilité de construire des applications webs dynamiques de façon intuitive et rapide. Enfin, dans ce sujet qui implique l'exploration d'arbres phylogénétiques et l'emploi de méthodes statistiques, ce langage nous a paru le plus adapté. Pour l'ensemble de ce projet, nous avons utilisé RStudio (`http://www.rstudio.com/`), un environnement de développement intégré (EDI) adapté à la programmation avec R.

3.2 Description d'un jeu de données « type »

La comparaison de séquences moléculaires (obtenues à partir de données de séquençages) permet de reconstruire la phylogénie d'organismes vivants. En bioinformatique, il est classique d'utiliser des alignements de séquences (au format FASTA) pour construire des arbres phylogénétiques (au format Nexus ou Newick). Le langage R présente l'avantage de pouvoir lire les deux formats, et de représenter ainsi la topologie d'un arbre sous forme de texte. Ces deux formats fournissent au minimum des informations sur le nom des feuilles, la longueur des branches

et la robustesse de chaques noeuds comme illustrés dans la figure 3.

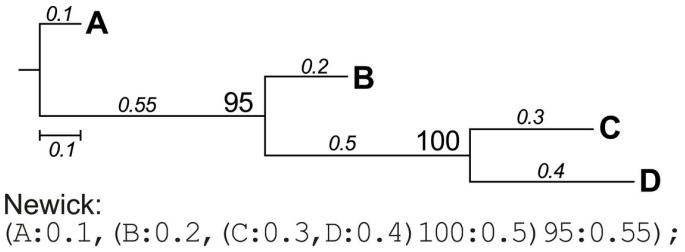


FIGURE 3 – Arbre phylogénétique au format Newick (tiré de [Stephens et al., 2016]).

Notre application prend en entrée un arbre phylogénétique au format Nexus ou Newick sur lequel l'intensité du signal temporel sera évaluée à chacun des noeuds. Parallèlement, l'utilisateur peut aussi fournir un alignement de séquences (au format FASTA) qui a permis de construire l'arbre. En cas de présence de signal temporel au sein d'un noeud interne de l'arbre, l'utilisateur peut alors récupérer un sous-échantillonage des séquences des feuilles concernées, à partir de l'alignement de départ. Il sera alors possible de procéder à des analyses phylogénétiques sur ce sous-jeu de données. La démarche complète est illustrée de façon schématique en figure 4.

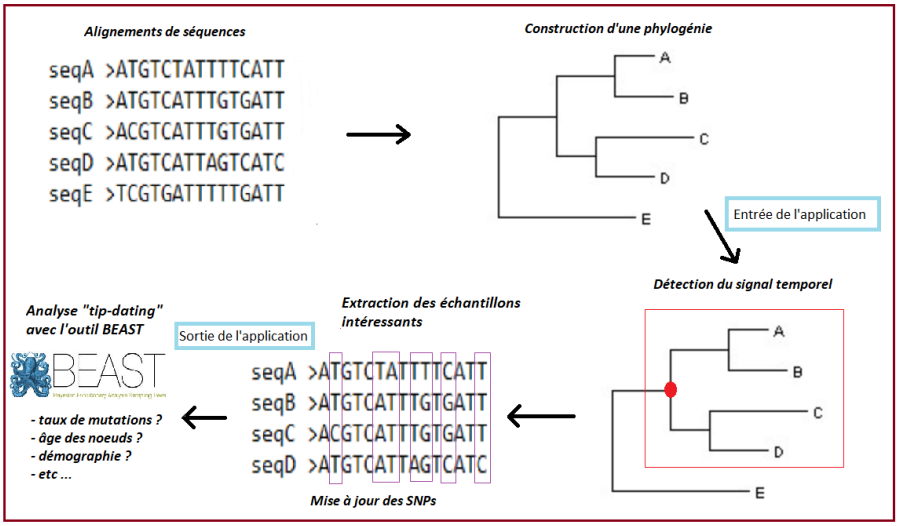


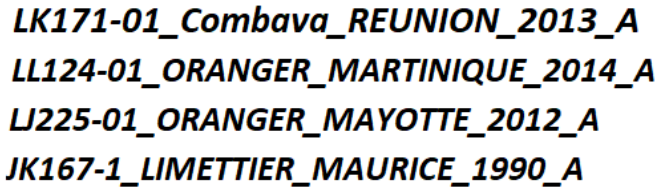
FIGURE 4 – Méthodologie utilisée et intérêt de l'application.

3.3 Explication des grandes étapes

Dans cette sous-partie, je vais m'attacher à introduire les grandes étapes de mon implémentation en justifiant particulièrement le choix des outils et de la méthodologie utilisée. Le script complet du code qui lance l'application est en appendice (voir partie 7).

La première étape consiste à lire un arbre phylogénétique sur R. Ces arbres doivent être au format Nexus ou Newick. Pour ce faire, nous avons utilisé les fonctions `read.nexus` ou `read.tree`, utilisable avec le package `ape`. Dans un second temps, nous calculons l'intégralité

des distances feuilles-racines pour toutes les feuilles de l'arbre. Cette tâche implique l'utilisation de la fonction `distRoot` du package `adephylo`. Une fois l'intégralité des distances calculées, il convient de récupérer la date d'échantillonnage de nos feuilles. Or, cette date d'échantillonnage se trouve généralement de manière standard dans le nom d'une feuille. Le nom d'une feuille est indiqué sous forme d'une chaine de caractères, qui peut être constituée de différents éléments descriptifs de l'échantillon, qui sont séparés par un séparateur (cf.figure 5). Il est alors possible à l'aide de la fonction `strsplit` de spécifier le séparateur et la position de la date d'échantillonage (sous réserve qu'elle s'y trouve), pour récupérer la date. Une fois la distance feuilles-racine mesurée pour chaques noeuds internes de l'arbre, nous récupérons sa liste des feuilles descendantes, la distance feuille-racine et la date d'échantillonage pour tous les descendants. Pour ce faire, nous utilisons la fonction « getDescent » issue du package `adephylo`. Cette liste va servir de point de départ pour construire des régressions, ainsi que pour chercher et explorer l'intensité du signal temporel au sein d'un arbre.



Le nom des feuilles est organisé en différents éléments descriptifs de l'échantillon. Dans cet exemple, le séparateur est un underscore.

Un identifiant de souche, un lieu et une date d'échantillonage peuvent être présents dans le nom d'une feuille.

FIGURE 5 – Description du format d'une « feuille ».

Sous certaines conditions, nous pouvons alors calculer une régression linéaire pour chaques noeuds internes de l'arbre en utilisant les distances feuilles-racines (en ordonnée) et les dates d'échantillonage (en abscisse) des descendants du noeud considéré. Il est important de noter que deux conditions doivent être respectées :

- Une régression peut être effectuée pour un noeud si ce noeud a plus de 2 feuilles descendantes. En effet, une droite de régression peut être construite avec au moins 3 points.
- Parmis ces trois points minimum, les coordonnées x et y ne doivent pas être chevauchantes (identiques) auquel cas, on se retrouve dans la situation d'effectuer une régression uniquement entre deux points. Pour éviter un tel cas de figure, nous avons utilisé la fonction `unique` de R appliquée simultanément aux deux coordonnées.

En finalité, pour chacune des régressions réalisées, nous récupérons les paramètres de pente, p-value et R^2 (coefficient de détermination). A partir de ces valeurs, il est possible de déterminer les noeuds pour lesquels le signal temporel est suffisant (c'est-à-dire les noeuds pour lesquels la pente de la régression est positive et la p-value significative).

3.4 Développement de l'application

Afin de visualiser de façon interactive la localité du signal temporel au sein d'un arbre, j'ai développé une application web dynamique grâce au package Shiny (`http://shiny.rstudio.com/`). Shiny permet de construire des applications de façon rapide et intuitive en utilisant R comme moteur de calcul. Il est constitué notamment d'une grande diversité de widgets pour générer rapidement des interfaces. Autre avantage, l'application peut être hébergée sur des serveurs de ShinyApps (service offert par RStudio). C'est le cas de mon application qui est désormais mise en ligne sur le serveur shinyapps.io et disponible pour tous au lien : `https://localtemporalsignal.shinyapps.io/LocalTemporalSignal/`.

Au niveau de l'architecture, une application Shiny est structurée en deux parties :

- un fichier ui.R qui gère l'interface.
- un fichier server.R qui gère les calculs.

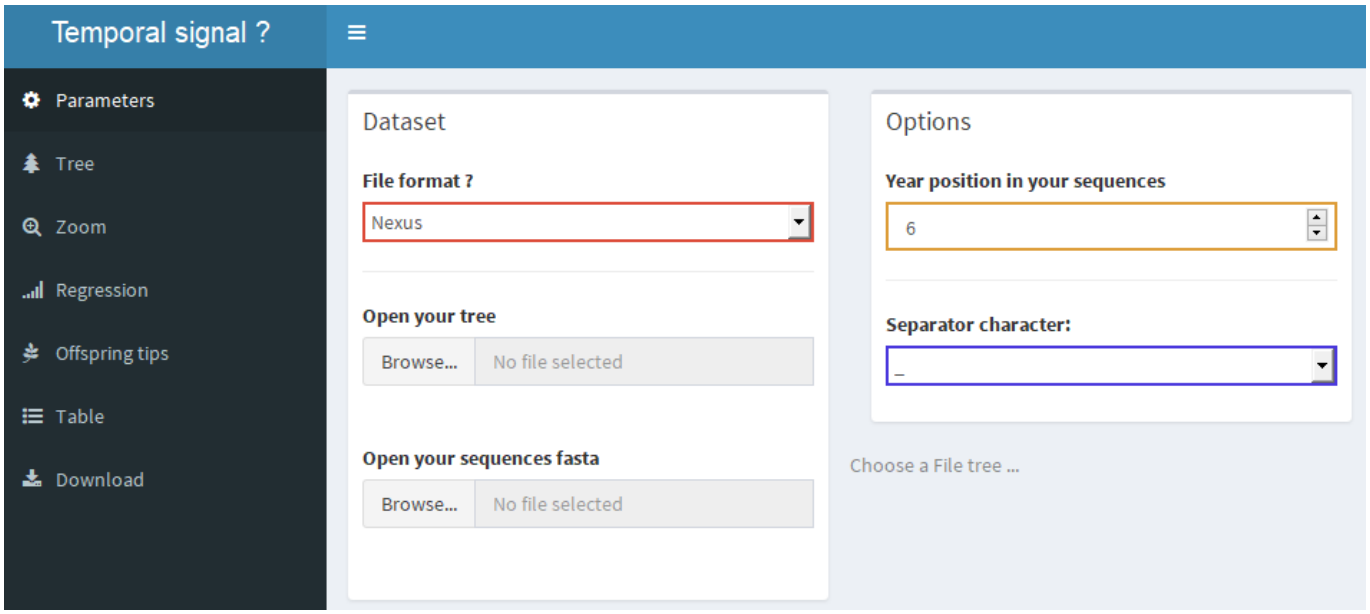


FIGURE 6 – Interface de mon application.

Avec Shiny, j'ai développé une interface interactive (en figure 6) qui permet à l'utilisateur de charger un arbre phylogénétique, et d'identifier sur celui-ci l'échelle évolutive à laquelle le signal temporel est statistiquement significatif. En particulier, l'utilisateur peut interagir avec l'arbre (par action de pointer-et-cliquer) et choisir le noeud de son choix pour visualiser la régression correspondante. Les noeuds les plus intéressants (pente positive et p-value significative) "s'allument" automatiquement dans l'arbre, et un tableau des paramètres statistiques pour ces noeuds est disponible. Enfin, notre outil intègre également une option d'agrandissement pour améliorer la lisibilité d'un arbre et une possibilité d'exporter notre arbre et la table des noeuds intéressants.

3.5 L'outil BEAST

Une fois la condition du signal temporel validée, il devient possible de réaliser des analyses bayésiennes en utilisant des séquences moléculaires. Ici, ces analyses ont été effectuées avec l'outil BEAST (`http://beast.community/`) : un logiciel multi-plateforme qui permet d'estimer des paramètres d'intérêt biologique en se reposant sur des méthodes de Monte-Carlo par chaînes de Markov (MCMC). Ces méthodes algorithmiques permettent d'estimer une valeur approchée pour un ensemble de paramètres (qui est la plus vraisemblable par rapport à un jeu de données empirique) qui sera généré par rapport à une distribution de probabilité. Ainsi, plusieurs analyses lancées avec cet outil en utilisant le même jeu de données et les mêmes paramètres renverront des résultats non identiques. Afin de vérifier la bonne convergence de l'algorithme, j'ai effectué 3 répétitions indépendantes pour chacune des analyses.

Le logiciel BEAST génère deux types de fichiers de résultats qu'il est possible d'analyser à l'aide de deux autres logiciels spécialisés :

- les valeurs de paramètres estimés lors de chacune des itérations de la chaine MCMC peuvent être résumées grâce au logiciel **Tracer** (`http://beast.community/tracer`).
- l'ensemble des arbres phylogénétiques est résumé en un arbre consensus avec le logiciel **TreeAnnotator** (`http://beast.community/treeannotator`) afin de pouvoir visualiser des informations précieuses comme la date estimée et la robustesse pour chaques noeuds.

L'objectif est de mettre en valeur la différence de qualité des estimations obtenus avec et sans la condition du signal temporel valide et de montrer en définitive l'importance cruciale d'avoir une application capable de détecter l'échelle évolutive à laquelle l'analyse est applicable.

3.6 Un cas d'étude : *Xanthomonas citri* pv. *citri*

Dans le cadre de ce projet, j'ai étudié l'origine et le mode de transmission d'une maladie bactérienne particulièrement répandue dans les zones tropicales et subtropicales : le chancre bactérien des agrumes. L'agent pathogène responsable de cette maladie est une bactérie phytopathogène à Gram positif : *Xanthomonas citri* pv. *citri* (Xcc). Le chancre bactérien des agrumes attaque les feuilles, les fruits et les tiges des agrumes, ce qui peut entraîner dessus des lésions nécrotiques circulaires ([Brunings and Gabriel, 2003]). Dans les régions où la maladie est endémique, des stratégies de luttes sont nécessaires afin de limiter ou éradiquer cet agent pathogène, qui provoquent des défoliations, des chutes précoces des fruits et qui peut altérer la qualité du fruit. Par ailleurs, il existe des restrictions qui empêchent l'export et la vente de fruits en provenance des régions où la bactérie est présente ([Gottwald et al., 2002]). C'est pourquoi la compréhension de l'origine et du mode de transmission de Xcc revêt un intérêt considérable afin de limiter les pertes économiques qui sont liées directement ou indirectement au chancre

bactérien des agrumes.

Dans cette étude, j'ai utilisé un alignement de 71 séquences d'ADN provenant de la bactérie Xcc et échantillonnés à différents moments (cf. figure 7) et endroits dans le monde. L'échantillon le plus ancien date de 1974 et le plus récent de 2017. La plupart des échantillons ont été récupérés dans les zones tropicales de l'océan Indien où la maladie est très présente (Réunion, Comores, Rodrigues, Maurice et Mayotte). On trouve également des échantillons du continent américain (Floride et Martinique) et asiatique (Inde et Oman). Dans notre étude, ces échantillons peuvent être subdivisés en pathotypes : groupes d'organismes pathogènes qui n'appartiennent pas toujours à la même classification mais qui ont le même spectre d'hôte. Trois pathotypes sont présents dans notre jeu de données. Le pathotype A est le plus répandu dans le monde et affecte la totalité des espèces du genre `citrus`. Le pathotype A^* est surtout présent en Asie mais a émergé en Afrique de l'Est. Enfin, le pathotype A^W a émergé en Floride mais sévit actuellement en Inde ([Pruvost et al., 2014]). Dans la figure 7, nous présentons aussi l'arbre que prend en entrée l'application avec une structuration génétique visible des différents pathotypes au sein de l'arbre.

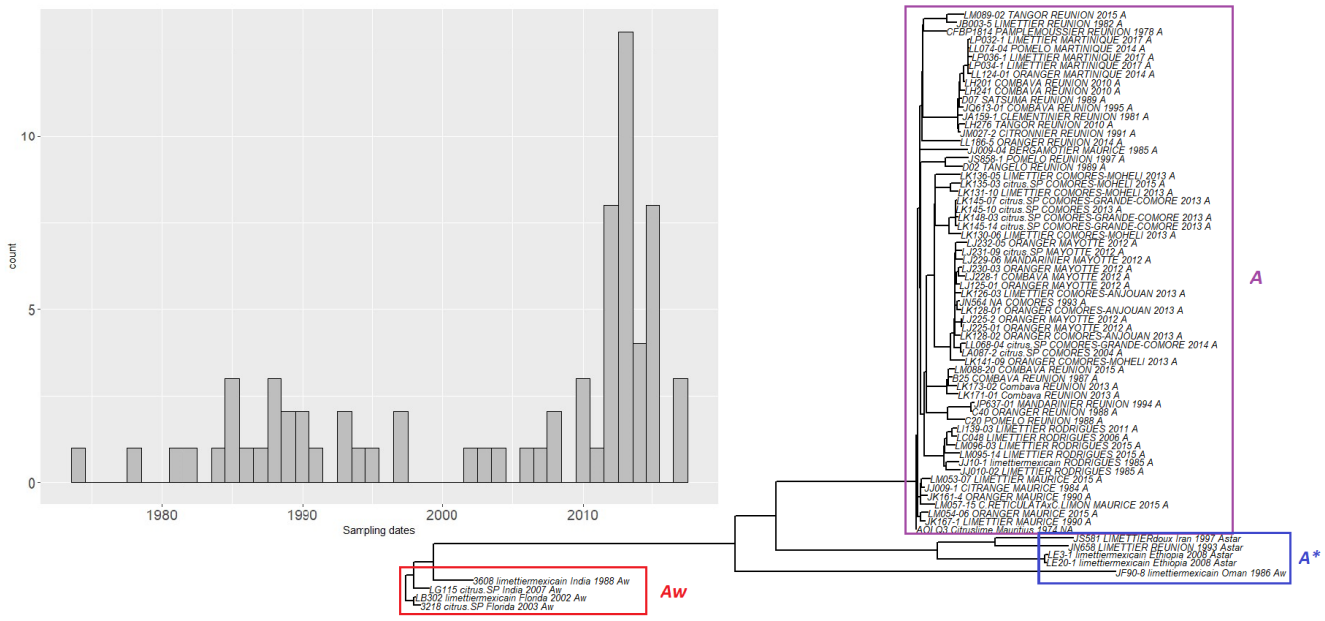


FIGURE 7 – Répartition des dates des échantillons et arbre phylogénétique avec la structuration génétique visible des 3 pathotypes.

4 Résultats obtenus

4.1 Visualisation du signal temporel

J'ai utilisé mon application dans le but de tester le jeu de données complet de Xcc et pour visualiser l'échelle évolutive à laquelle nous avons du signal temporel de façon significative. Pour une meilleure lisibilité du signal, le nom des feuilles n'est pas affiché dans l'application.

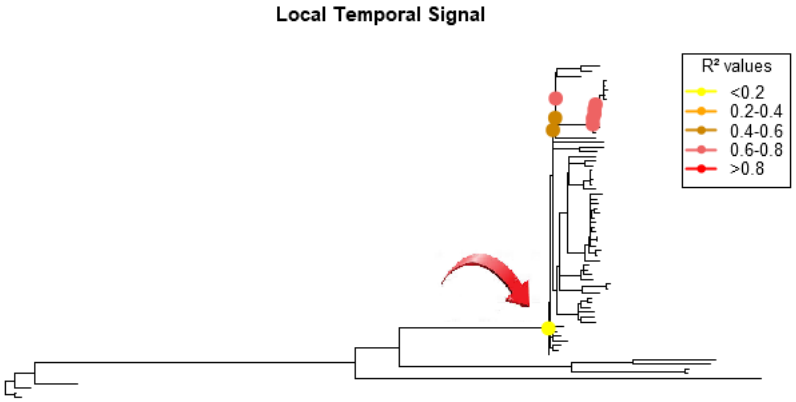


FIGURE 8 – Noeuds internes de l'arbre avec du signal temporel (issue de l'application).

Les noeuds qui s'allument représentent les noeuds pour lesquels la régression est positive et statistiquement significative (présence de signal temporel). Dans notre étude, nous avons fixé un risque de première espèce α de 5%. L'utilisateur peut modifier ce seuil dans l'application à l'aide d'un curseur. L'échelle de couleur est fonction du R^2, qui évalue d'un point de vue statistique la qualité de la régression. Sur le plan biologique, cette valeur est un indicateur de l'hétérogénéité du taux de mutation entre plusieurs échantillons de l'arbre. Ce paramètre sera d'autant plus élevé que le noeud considéré sera récent : les individus sont plus proches génétiquement entre eux, ce qui peut expliquer un taux de mutation alors plus proche (cf. figure 8). Parmi les noeuds colorés, on peut observer qu'il existe du signal temporel à différentes échelles évolutives dans l'arbre. Le noeud le plus ancien avec du signal temporel (représenté par une flèche) correspond à la racine du clade *A* (cf.figure 7). Sur la figure de notre application, l'utilisateur peut directement cliquer sur le noeud et afficher les informations statistiques de la régression ainsi que le plot de cette régression (cf.figure 9) caractéristique de ce noeud.

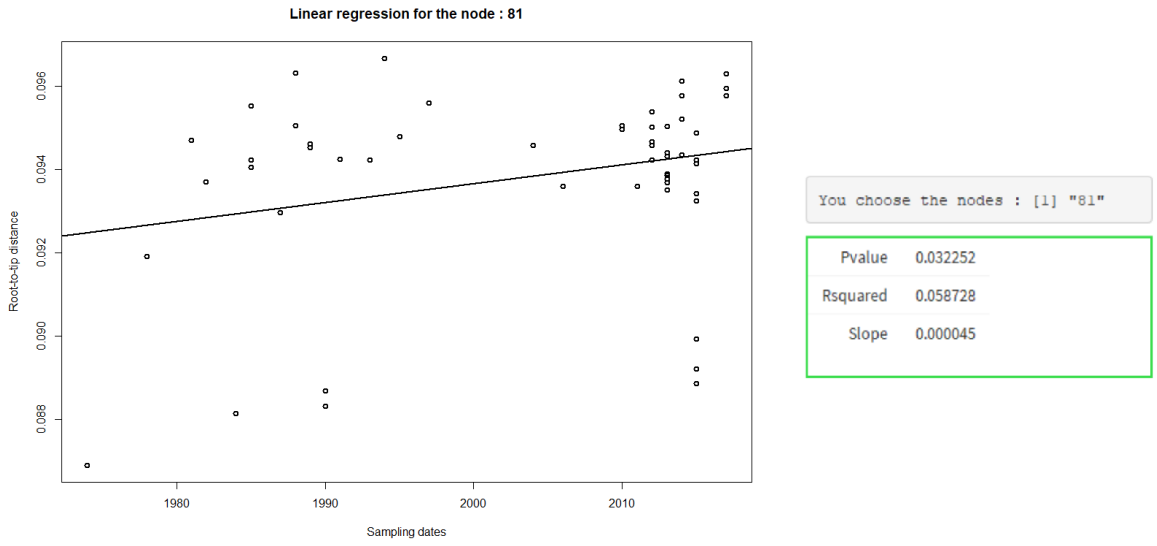


FIGURE 9 – Régression linéaire et paramètres statistiques de la racine du clade *A* (extraits de l'application).

Un onglet de l'application permet de visualiser l'allure du nuage de points de la régression. Elle constitue un indicateur de la validité d'une régression. Par exemple, il faut éviter d'avoir une courbure nette dans l'écart des points à la courbe. Pour le noeud d'intérêt ici, l'allure de la régression est acceptable. On note néanmoins un groupe de points qui se démarquent dans la partie inférieure du graphe et qui correspondent au clade des échantillons de l'île Maurice.

4.2 Estimation de paramètres biologiques

J'ai ensuite lancé l'analyse BEAST séparément sur 2 jeux de données : le premier représente notre alignement avec l'intégralité des séquences (71 échantillons structurés par les 3 clades : A, A^W et A^*). Le second ne présente en revanche que les séquences présentes dans le clade A, c'est-à-dire à l'échelle du noeud le plus ancien pour lequel nous avons détecté du signal temporel. Il est possible d'avoir une estimation de nombreux paramètres avec BEAST. Dans cette étude, nous nous sommes concentrés sur l'estimation de la valeur de deux paramètres biologiques. D'une part, **l'âge de la racine du clade A**, qui pourrait être un indicateur de la date à laquelle la bactérie est arrivée dans la zone de l'Océan Indien. D'autre part, **le taux de mutation dans la population du clade A** de Xcc, pour mieux comprendre l'évolution de la diversité de cette phytobactérie. Dans les figures 10 et 11, nous avons représenté les intervalles de confiance à 95% sur l'estimation des deux paramètres. Trois analyses indépendantes ont été effectuées sur l'échantillon complet (où le signal temporel est absent) et sur le sous-échantillon correspondant au clade A (où le signal temporel est présent).

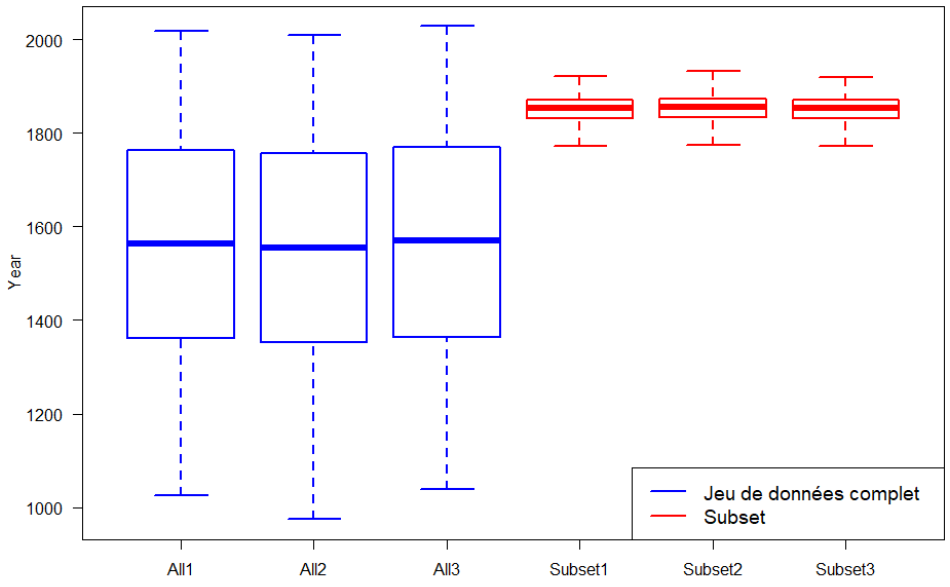


FIGURE 10 – Estimation de la date de l'ancêtre commun du clade A sur les 2 jeux de données (issue de l'outil BEAST).

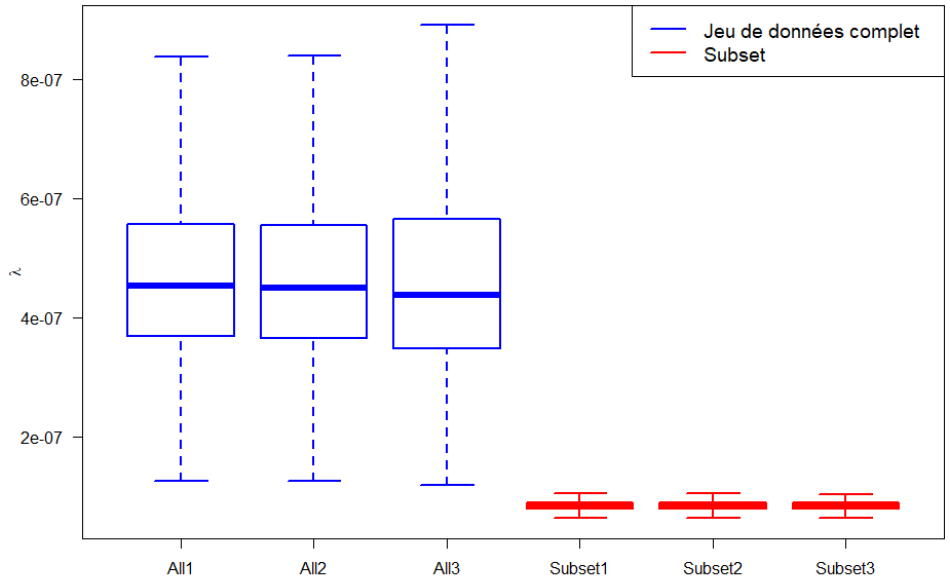
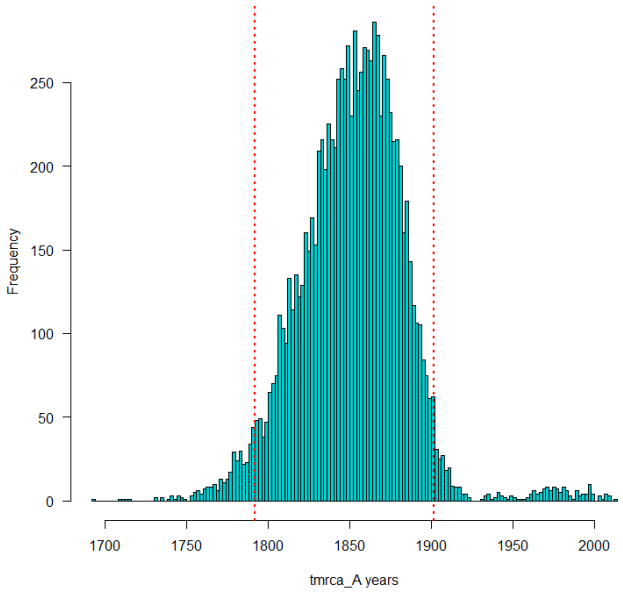


FIGURE 11 – Estimation du taux de mutation sur les 2 jeux de données (issue de l'outil BEAST).

En effectuant l'analyse sur le jeu de données complet (pas de signal temporel), il est possible d'estimer l'âge de l'ancêtre commun du clade A (figure 10), qui serait datée entre 448 et 462 ans (entre les années 1555 et 1569) selon les analyses. Néanmoins, l'intervalle de confiance est très important. Sur le sous-échantillonage (avec signal temporel), l'âge de l'ancêtre commun se situe cette fois à une période de temps de 164 à 167 ans (entre les années 1850 et 1853), avec un intervalle de confiance beaucoup plus restreint. Dans la figure 12, j'ai représenté la distribution des valeurs de cette date, estimée par BEAST, après ajustement des données observées (séquences) au modèle sous-jaçent et aux paramètres fixés *apriori*.

En ce qui concerne l'estimation du taux de mutation chez Xcc (figure 11), on a également une meilleure précision sur l'estimation lorsque le signal temporel est statistiquement positif. Sans signal, le taux de mutation est estimée entre 4.7 et 5.2 $\times 10^{-7}$ substitutions par site et par an. La prise en compte du signal réduit ce taux à une estimation située entre 8.3 et 8.4$\times 10^{-8}$ substitutions par site et par an.

La différence entre les estimations obtenus sur chaques jeux de données (sans et avec signal temporel) a été testé d'un point de vue statistique. Ce faisant, j'ai utilisé le test non paramétrique de Kolmogorov-Smirnov pour vérifier si les jeux de données sont issus d'une même loi. Les résultats du test renvoie une p-value de l'ordre de 2×10^{-16}, ce qui est inférieure au risque α de l'ordre de 5%. On peut alors rejeter l'hypothèse nulle et conclure à une significativité des différences entre les deux jeux de données. La prise en compte du signal temporel dans l'estimation des paramètres s'avère donc cruciale pour réaliser une analyse "tip-dating".



L'estimation de la date de l'ancêtre commun le plus récent suit une distribution *a posteriori*. C'est la distribution des valeurs de cette date estimée par BEAST obtenue en ajustant les données observées (séquences) au modèle sous-jaçent et aux paramètres fixés *apriori* (priors). Cette distribution a l'allure d'une distribution d'une loi normale. La probabilité qu'une valeur de paramètre soit correcte sachant les données n'est pas la même selon les valeurs testées par BEAST. En pointillé rouge, j'ai indiqué les intervalles de confiance à 95% fournis par BEAST pour la distribution.

FIGURE 12 – Distribution de la date de l'ancêtre commun le plus récent du clade (issue de l'outil BEAST).

4.3 Topologie d'un arbre consensus

Le logiciel TreeAnnotator permet de construire un arbre consensus. Cette arbre est construit en maximisant la somme des *posteriors* à chaques noeuds : un paramètre statistique qui est représentatif ici de la robustesse d'un noeud. L'analyse permet également de calculer une estimation de la date de chaques noeuds de l'arbre (en année). Les feuilles de l'arbre que j'ai obtenu (voir figure 13) ont été colorées par zone géographique.

L'arbre consensus obtenu montre que la zone géographique permet de structurer des échantillons en clade : les échantillons issus de Martinique sont regroupés en un clade distinct. On observe la même structure pour les individus provenant de Rodrigues. En revanche, les échantillons issus de l'île de la Réunion sont structurés en 3 clusters au sein de l'arbre. Les échantillons issus des Comores et de Mayotte sont localisés dans la partie supérieure de l'arbre, avec un individu d'Aujouan (île appartenant aux Comores) qui a la particularité de se situer parmi le clade des échantillons mahorais. L'arbre consensus apporte une autre information précieuse : une estimation de la date de chaques noeuds. Ainsi, la datation de l'ancêtre commun du clade *A* est situé au milieu de 19ème siècle ($\approx$ 1851), ce qui peut donner une indication temporelle sur l'année d'émergence de cette souche de Xcc dans la zone de l'Océan Indien.

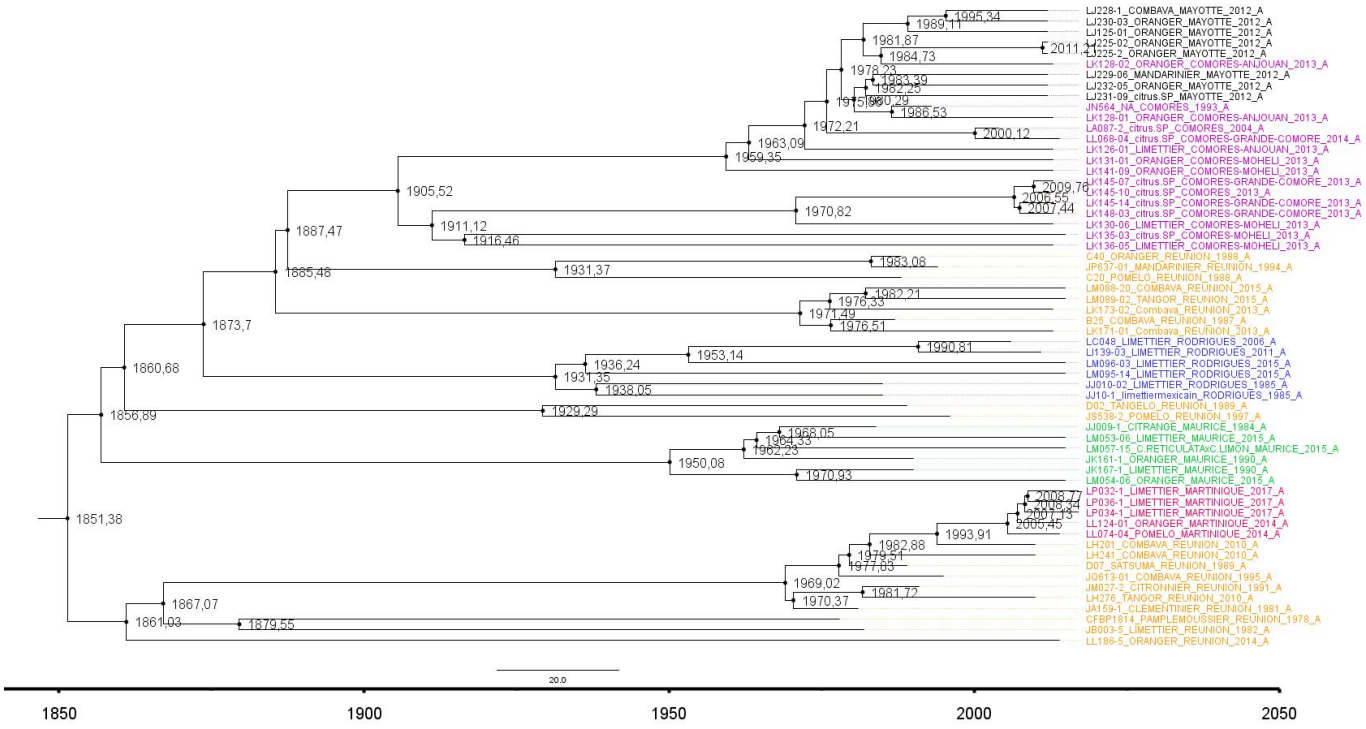


FIGURE 13 – Topologie de l'arbre consensus construit par l'outil BEAST.

5 Discussion des résultats

5.1 Choix de l'échelle évolutive

L'intérêt de l'application développée dans le cadre de mon stage était de localiser l'échelle évolutive au sein d'un arbre phylogénétique à laquelle une analyse de « tip-dating » (réalisée ici avec l'outil BEAST) est réalisable, c'est-à-dire l'échelle évolutive à laquelle l'intensité du signal temporel est statistiquement positive. Cet outil est générique dans la mesure où il peut s'appliquer à des jeux de données diversifiés (*e.g* bactéries, virus, eucaryotes ...). La nouveauté de cette application réside dans le fait qu'elle permet d'évaluer l'intensité du signal temporel à toutes les échelles évolutives de l'arbre et pas seulement à sa racine comme proposé par les outils existants ([Rambaut et al., 2016] : `http://tree.bio.ed.ac.uk/software/tempest/`). En outre, l'intérêt est souvent de localiser le noeud le plus basal (*i.e* ancien) pour lequel on détecte un signal temporel statistiquement significatif. Le seuil de p-value peut être augmenté ou diminué par l'utilisateur pour rendre plus ou moins contraignant la détection du signal. Le choix de l'échelle évolutive dans un arbre peut s'avérer déterminant dans la qualité des résultats d'une analyse tip-dating effectuée en aval. Nous avons montré cette différence à travers l'étude de cas d'une bactérie phytopathogène : *Xanthomonas citri* pv. *citri* (Xcc).

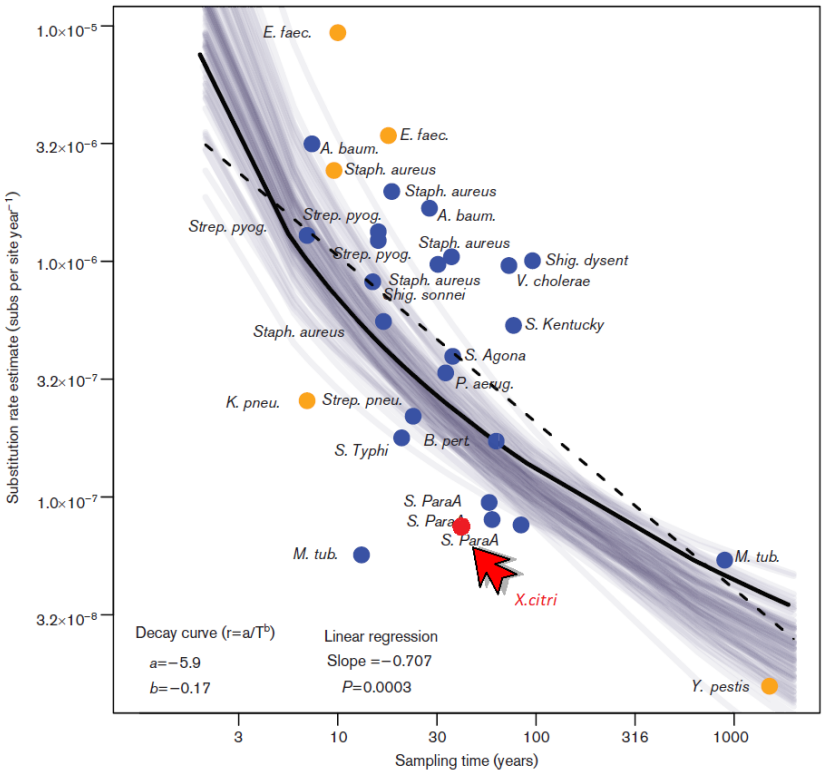
Pour ce faire, nous avons réalisé une analyse BEAST de « tip-dating » à deux échelles évolutives : celle détectée comme présentant du signal temporel à l'aide de l'application développée, et une échelle plus large au sein de laquelle le signal temporel semblait inexistant. Nos résultats

indiquent que la précision sur les estimations des paramètres biologiques (figures 10 et 11) est bien meilleure lorsque la condition du signal temporel est valable. En plus d'être plus précises, les estimations moyennes du taux de mutation et de l'âge de l'ancêtre commun des souches du clade A diffèrent largement entre les deux analyses. La prise en compte du signal temporel modifie donc considérablement la valeur et la précision que l'on peut obtenir lors de l'estimation de paramètres biologiques par une approche de « tip-dating ».

Dans une étude récente, [Murray et al., 2016] ont simulé dans un premier temps plusieurs jeux de données qui diffèrent par l'intensité du signal temporel avant dans un second temps, de les analyser avec BEAST. Dans leur étude, le fait de considérer des jeux de données simulés leur permet de connaître les paramètres biologiques (comme le taux de mutation ou l'âge de la racine) ayant servi à générer les données. Leurs résultats indiquent que les estimations réalisées sur des jeux de données sans signal temporel s'écartent significativement des attendus simulés, ce qui n'est pas le cas lorsque le signal est présent. Nos résultats obtenus sur un jeu de données empirique semblent donc cohérents avec ces observations. Toutefois, est-ce que la prise en compte de cette condition du signal temporel dans nos analyses permet d'avoir des résultats cohérents avec la réalité biologique ? Il est difficile de répondre à cette question dans la mesure où le domaine de recherche sur Xcc est encore très récent. En effet, les taux de mutation des bactéries phytopathogènes ainsi que l'historique de l'invasion du chancre critique dans la zone de l'Océan Indien restent largement méconnus. Dans une analyse ultérieure, il pourrait donc être intéressant de générer des jeux de données simulés permettant de comparer l'effet de prendre ou non en compte le signal temporel au sein d'un même arbre (l'analyse réalisée par [Murray et al., 2016] était basée sur l'analyse de plusieurs arbres différents montrant ou non du signal temporel).

5.2 Explication biologique et historique aux résultats

Le taux de mutation moyen estimée sur Xcc se situerait entre 8.3 et 8.4 $\times 10^{-8}$ substitutions par site et par an. Cette estimation, obtenue en présence de signal temporel, est presque d'ordre dix fois plus grand (situé entre 4.7 et 5.2 $\times 10^{-7}$) lorsqu'il est absent. Rappelons que le taux de mutation sur des phytobactéries n'est pas connue et qu'établir un point de comparaison reste une tâche ardue. Néanmoins, dans les deux cas de figure, ces taux de mutations restent en accord avec les gammes de taux de mutations recensées par ([Duchêne et al., 2016]) sur des bactéries pathogènes chez les animaux et l'Homme : entre 10^{-5} et 10^{-9} substitutions par site et par an. J'ai cherché à mettre en relation le taux de mutation estimée ici avec des taux connus chez ces bactéries. Comme indiqué dans la figure 14, notre estimation du taux de substitution chez Xcc semble cohérente avec celles précédemment réalisées chez d'autres bactéries pathogènes.



Le taux de mutation chez Xcc est estimée dans notre étude à ($\approx 8.3 \times 10^{-8}$) sur une période donnée correspondant à la profondeur temporelle de notre clade (2017-1974 = 43 années) dans notre arbre.
Ce faisant, nous avons comparé ces paramètres avec des valeurs obtenues sur des pathogènes d'animaux. Dans la figure ci-contre 14, la structure temporelle de différents pathogènes est évaluée (bleu : forte /orange : modérée). Un point correspondant à Xcc a été ajouté en rouge sur la figure.

FIGURE 14 – Structure temporelle de différents pathogènes (modifié de [Duchêne et al., 2016])

Nos résultats (figures 10 et 13) ont également permis d'estimer l'âge de l'ancêtre commun des souches du clade *A*. Ce noeud est daté autour des années 1850 [1792.084 ;1901.656]. Ce résultat laisse penser que la bactérie a été introduite dans l'océan Indien au cours de cette période. Différentes hypothèses peuvent être émises en ce qui concerne l'origine de cette apparition. D'un point de vue global, on peut constater que cette période coincide avec la forte **croissance économique mondiale des années 1850** qui a entraîné une explosion des exportations à l'échelle mondiale. Dans ce contexte, les agrumes sont concernés par cette intensification des flux de marchandises. De façon plus locale, cette date coincide avec l'**Engagisme** à la Réunion et à l'île Maurice, mis en place juste après l'abolition de l'esclavage (1848) pour pallier au manque de main d'oeuvre, et qui consiste pour les propriétaires de terres à faire venir de l'étranger (notamment pour beaucoup d'entre eux d'Inde) des travailleurs pour travailler dans les champs. Cette immigration intensive s'est étendue de 1830 à 1880 et pourrait expliquer l'introduction de la maladie (via le flux d'immigrants) dans la région de l'Océan Indien. En effet, comme le suggère [Beaujard, 2011], certaines plantes infectées ont pu être ramenées par ces travailleurs sur l'île au cours de cette période.

En outre, j'ai cherché à mettre en relief la structuration génétique de l'arbre consensus (figure 13) avec des explications géographiques/historiques. Dans l'arbre, les échantillons comoriens et mahorais sont assez proches, ce qui s'explique par la proximité géographique entre les deux régions : Mayotte est une île située dans l'archipel des Comores. On peut même noter qu'un échantillon de l'île d'Anjouan se trouve au sein du cluster des échantillons mahorais.

Or, l'île d'Anjouan, appartenant à l'archipel des Comores, est l'île la plus proche de Mayotte (70kms de distance), ce qui nous amène à penser qu'il y a eu de nombreux échanges de personnes et de marchandises entre les deux îles au cours des dernières décennies. La proximité spatiale peut expliquer aussi la contiguité entre les clades des échantillons de Maurice, Rodrigue et La Réunion, appartenant tous à l'archipel des Mascareignes au sein de l'Océan Indien. Les échantillons de Xcc provenant de l'île de la Réunion sont structurés en trois clusters à différents endroits de l'arbre. Cette observation tend à montrer que la bactérie a été introduite sur l'île à trois périodes distinctes, ce qui est cohérent avec la datation des ancêtres communs des trois clusters (1861/1885/1929). Une autre hypothèse, moins probable ici, aurait été d'avoir eu une introduction unique des individus appartenant aux trois sous-structures génétiques au sein d'une même structure initiale. Enfin, les échantillons de l'arbre ne sont pas tous issus de l'Océan Indien. En effet, l'intégralité des échantillons de Xcc provenant de Martinique forme un cluster. La datation de l'ancêtre commun entre les échantillons réunionnais et martiniquais coincide avec le début du 21ème siècle (au plus tard 2005). Ce résultat tendrait à montrer que la bactérie a été incorporé de la Réunion vers la Martinique (entre DOM) dans un passé très récent. Ce constat est concomittant avec les premières observations de cas du chancre critique en Martinique qui date de 2014 (`http://daaf.martinique.agriculture.gouv.fr/Le-chancre-citrique-en-Martinique`). Nos résultats indiquent donc que la bactérie est probablement arrivée sur l'île une dizaine d'années avant que sa pathogénicité ne soit détectée, ce qui est un laps de temps classique lors d'arrivée d'un pathogène dans un nouvel environnement.

6 Perspectives

D'un point de vue spécifique à l'application que j'ai eu l'opportunité de développer, certaines améliorations aussi bien sur le fond que sur la forme pourront être apportées dans le futur. Tout d'abord, j'ai remarqué que l'arbre peut être difficilement lisible, en particulier lorsque ce dernier contient une structure composée de longues branches internes et de nombreuses feuilles génétiquement très proches. Une alternative intéressante pourrait être d'implémenter un outil de « zoom » pour que l'utilisateur puisse agrandir spécifiquement certaines parties de l'arbre. Ensuite, il pourrait être aussi très utile de pouvoir relier n'importe quel point de la régression directement avec l'échantillon associé sur l'arbre. Pour ce faire, une solution serait d'implémenter une fonction qui, lorsqu'une zone de régression serait sélectionnée, ferait apparaître les échantillons associés directement sur l'arbre. Toutes les suggestions énoncées ci-dessus me semblent réalisables à l'aide de la librairie Shiny.

D'un point de vue plus général, l'outil développé dans le cadre de cette étude devrait faciliter la tâche des biologistes cherchant à déterminer l'échelle évolutive à laquelle des inférences de type « tip-dating » pourront être réalisées de façon robuste sur leurs jeux de données. Nous avons montré que de telles analyses ont la capacité de fournir des estimations de paramètres

biologiques comme l'âge d'un ancêtre commun ou encore le taux de mutation. La connaissance de tels paramètres s'avère cruciale pour pouvoir prévoir et anticiper l'arrivée et la vitesse d'adaptation d'un pathogène dans un environnement. Dans le cadre des MIE, l'âge d'un ancêtre commun peut permettre d'élucider l'origine d'introduction d'une maladie infectieuse. Le taux de mutation apporte lui une information capitale sur la capacité d'un pathogène à s'adapter. Il peut, par exemple, être utilisé dans le but de prédire la vitesse à laquelle un agent pathogène pourrait acquérir une résistance à un traitement ou à un vaccin. Ces informations sont donc précieuses pour aider les services publics à la prise de décision lors d'une épidémie. C'est notamment pour cette raison qu'il existe de nombreuses restrictions phytosanitaires en ce qui concerne l'import de végétaux sur certains territoires (`http://www.douane.gouv.fr/articles/a11624-restrictions-phytosanitaires-applicables-aux-vegetaux-fruits-et-legumes-en-provenance-d-un-pays-tiers`).

7 Conclusion

Lors de ce stage, j'ai eu l'opportunité de développer une interface graphique pour aider la communauté des biologistes. Cette application doit permettre une utilisation plus raisonnée et robuste des méthodes de datations phylogénétiques. Les résultats obtenus ont souligné l'importance d'avoir du signal temporel dans un jeu de données, ce qui conditionne la validité des inférences phylogénétiques effectuées en aval. La bonne conduite de ces analyses mène à l'estimation de paramètres biologiques cruciaux dans la compréhension de l'origine et du mode de transmission de certains pathogènes. Ainsi, l'intérêt de l'étude que j'ai mené sur la bactérie phytopathogène de l'espèce *Xanthomonas citri* a permis de comprendre son histoire épidémiologique, de son introduction dans l'Océan Indien jusqu'à aujourd'hui.

Sur le plan professionnel, ce stage m'a apporté une première expérience dans un organisme de recherche et donc une première prise de connaissance du monde de la recherche. J'ai pu réaliser un projet qui m'a amené à développer une application graphique destinée à la communauté scientifique. Ce stage m'a permis d'améliorer mes connaissances en langage R, dans le domaine de la phylogénie ainsi que dans la rigourosité à prendre avant d'analyser tout résultat issu d'une analyse bioinformatique. Pour ma formation d'ingénieur, ce stage m'a été bénéfique pour surpasser mon appréhension de la programmation, pour apprendre à travailler en collaboration avec des chercheurs/stagiaires et pour développer mon esprit d'initiative dans l'optique de créer un outil biostatistique.

Références

[Beaujard, 2011] Beaujard, P. (2011). The first migrants to madagascar and their introduction of plants : linguistic and ethnological evidence. *Azania : Archaeological Research in Africa*, 46(2) :169–189.

[Brunings and Gabriel, 2003] Brunings, A. M. and Gabriel, D. W. (2003). Xanthomonas citri : breaking the surface. *Molecular plant pathology*, 4(3) :141–157.

[Buonagurio et al., 1986] Buonagurio, D. A., Nakada, S., Parvin, J. D., Krystal, M., Palese, P., and Fitch, W. M. (1986). Evolution of human influenza a viruses over 50 years : rapid, uniform rate of change in ns gene. *Science*, 232(4753) :980–982.

[Croucher and Didelot, 2015] Croucher, N. J. and Didelot, X. (2015). The application of genomics to tracing bacterial pathogen transmission. *Current opinion in microbiology*, 23 :62–67.

[Drummond et al., 2003] Drummond, A., Pybus, O. G., and Rambaut, A. (2003). Inference of viral evolutionary rates from molecular sequences. *Adv Parasitol*, 54 :331–358.

[Duchêne et al., 2016] Duchêne, S., Holt, K. E., Weill, F.-X., Le Hello, S., Hawkey, J., Edwards, D. J., Fourment, M., and Holmes, E. C. (2016). Genome-scale rates of evolutionary change in bacteria. *Microbial Genomics*, 2(11).

[Faria et al., 2014] Faria, N. R., Rambaut, A., Suchard, M. A., Baele, G., Bedford, T., Ward, M. J., Tatem, A. J., Sousa, J. D., Arinaminpathy, N., Pépin, J., et al. (2014). The early spread and epidemic ignition of hiv-1 in human populations. *science*, 346(6205) :56–61.

[Gilligan and van den Bosch, 2008] Gilligan, C. A. and van den Bosch, F. (2008). Epidemiological models for invasion and persistence of pathogens. *Annu. Rev. Phytopathol.*, 46 :385–418.

[Gottwald et al., 2002] Gottwald, T. R., Graham, J. H., and Schubert, T. S. (2002). Citrus canker : the pathogen and its impact. *Plant Health Progress*, 10 :32.

[Ho and Duchêne, 2014] Ho, S. Y. and Duchêne, S. (2014). Molecular-clock methods for estimating evolutionary rates and timescales. *Molecular ecology*, 23(24) :5947–5965.

[Ho et al., 2011] Ho, S. Y., Lanfear, R., Bromham, L., Phillips, M. J., Soubrier, J., Rodrigo, A. G., and Cooper, A. (2011). Time-dependent rates of molecular evolution. *Molecular ecology*, 20(15) :3087–3101.

[Jones et al., 2008] Jones, K. E., Patel, N. G., Levy, M. A., Storeygard, A., Balk, D., Gittleman, J. L., and Daszak, P. (2008). Global trends in emerging infectious diseases. *Nature*, 451(7181) :990.

[Leonardi et al., 2017] Leonardi, M., Librado, P., Der Sarkissian, C., Schubert, M., Alfarhan, A. H., Alquraishi, S. A., Al-Rasheid, K. A., Gamba, C., Willerslev, E., and Orlando, L. (2017). Evolutionary patterns and processes : lessons from ancient dna. *Systematic biology*, 66(1) :e1–e29.

[Li et al., 2014] Li, L. M., Grassly, N. C., and Fraser, C. (2014). Genomic analysis of emerging pathogens : methods, application and future trends. *Genome biology*, 15(11) :541.

[Morens et al., 2004] Morens, D. M., Folkers, G. K., and Fauci, A. S. (2004). The challenge of emerging and re-emerging infectious diseases. *Nature*, 430(6996) :242.

[Murray et al., 2016] Murray, G. G., Wang, F., Harrison, E. M., Paterson, G. K., Mather, A. E., Harris, S. R., Holmes, M. A., Rambaut, A., and Welch, J. J. (2016). The effect of genetic structure on molecular dating and tests for temporal signal. *Methods in Ecology and Evolution*, 7(1) :80–89.

[Pruvost et al., 2014] Pruvost, O., Magne, M., Boyer, K., Leduc, A., Tourterel, C., Drevet, C., Ravigné, V., Gagnevin, L., Guérin, F., Chiroleu, F., et al. (2014). A mlva genotyping scheme for global surveillance of the citrus pathogen xanthomonas citri pv. citri suggests a worldwide geographical expansion of a single genetic lineage. *PloS one*, 9(6) :e98129.

[Rambaut et al., 2016] Rambaut, A., Lam, T. T., Max Carvalho, L., and Pybus, O. G. (2016). Exploring the temporal structure of heterochronous sequences using tempest (formerly path-o-gen). *Virus evolution*, 2(1) :vew007.

[Saitou and Imanishi, 1989] Saitou, N. and Imanishi, T. (1989). Relative efficiencies of the fitch-margoliash, maximum-parsimony, maximum-likelihood, minimum-evolution, and neighbor-joining methods of phylogenetic tree construction in obtaining the correct tree.

[Stephens et al., 2016] Stephens, T. G., Bhattacharya, D., Ragan, M. A., and Chan, C. X. (2016). Physortr : a fast, flexible tool for sorting phylogenetic trees in r. *PeerJ*, 4 :e2038.

[Wilkinson et al., 2011] Wilkinson, K., Grant, W. P., Green, L. E., Hunter, S., Jeger, M. J., Lowe, P., Medley, G. F., Mills, P., Phillipson, J., Poppy, G. M., et al. (2011). Infectious diseases of animals and plants : an interdisciplinary approach.

[Zuckerkandl and Pauling, 1962] Zuckerkandl, E. and Pauling, L. (1962). Molecular disease, evolution and genetic heterogeneity.

Appendices

Chargement des librairies

```
library(ape)
library(ade4)
library(adephylo)
library(phytools)
library(shiny)
library(shinydashboard)
library(dashboard)
library(ggplot2)
```

Définition des fonctions

```
## Extrait les feuilles descendantes de chaques noeuds
get_descent <- function(arbre,nb=FALSE){
  out <- list()
  all <- mrca(arbre)
  N<- arbre$Nnode
  tip <- length(arbre$tip.label)
  for (i in 1:N)
  {
    if(!nb) out[[i]] <- rownames(all)[apply(all==(tip+i),1,sum)>0]
    if(nb) out[[i]] <- match(rownames(all)[apply(all==(tip+i),1,sum)>0],arbre$tip.label)
  }
  out
}

## Extrait la position d'un element dans une chaine de caractere
trim_names <- function(X,sep,pos,fix){
  strsplit(X,split=sep,fixed=fix)[[1]][pos]
}


## Calcul distance euclidienne
dist_coord <- function(X,xref,yref,ratio){
  ((X[2]-xref)**2 +  ((X[3]/ratio)-yref)**2)**0.5
}
```

Interface utilisateur

```
## Organisation des onglets de l'interface
ui <- dashboardPage(
  dashboardHeader(title = "Temporal signal ? "),
  dashboardSidebar(

    sidebarMenu(

      menuItem("Parameters", tabName="parameters", icon=icon("cog")),
      menuItem("Tree", tabName="tree", icon=icon("tree")),
      menuItem("Zoom", tabName="zoom", icon=icon("search-plus")),
      menuItem("Regression", tabName= "regression", icon=icon("signal")),
      menuItem("Offspring tips", tabName="offspring", icon=icon("pagelines")),
      menuItem("Table", tabName = "tabnodes", icon = icon("list-ul")),
      menuItem("Download", tabName="download", icon=icon("download"))

    )
  ),

  ## Constitution des onglets
  dashboardBody(

    tabItems(

      tabItem(tabName = "parameters",

              fluidRow(
```

```
27
28                box(
29                  title = "Dataset",
30                  tags$style("#format {border: 2px solid #dd4b39;}"),
31                  selectInput("format", "File format ?", choices=c("Nexus", "Newick"), width="100%", selectize = FALSE),
32
33                  hr(),
34                  fileInput("file1", "Open your tree",  accept = c("*",".*")),
35                  fileInput("file2","Open your sequences fasta", accept = c("*", ".*"))),
36
37                tags$style("#Year {border: 2px solid #dd9d39;}"),
38                box(title = "Options", numericInput("Year", "Year position in your sequences", 6, min = 1, max = 9),
39
40                    hr(),
41                    tags$style("#separator {border: 2px solid  #4b39dd;}"),
42                    selectInput("separator", "Separator character:",choices = c("_",",","-",";"),width="100%", selectize=FALSE
                          )),
43
44                plotOutput("DATES")
45
46              )
47        ),
48
49
50        tabItem(tabName = "tree",
51
52                checkboxInput("header", "Print Tips ?", TRUE),
53
54                conditionalPanel(
55
56                  condition = "input.header == true",
57                  plotOutput("contents",click = "plot_click", brush="plot_brush")),
58
59                conditionalPanel(
60                  condition = "input.header == false",
61                  plotOutput("contents_NF", click = "plot_click", brush="plot_brush")),
62
63
64                box(sliderInput("PV", "Seuil pvalue:", min=0.01, max=0.2, value=0.05)),
65                box(verbatimTextOutput("infos_nodes"),
66
67                    tags$style("#INFOS_NOEUDS {border: 2px solid #39dd4b;}"),
68                    tableOutput("INFOS_NOEUDS")),
69
70                checkboxInput("mantel", "Mantel test ?", FALSE),
71
72                conditionalPanel(
73                  condition = "input.mantel == true",
74                  tags$style("#infos_mantel {border: 2px solid #e8867a;}"),
75                  verbatimTextOutput("infos_mantel")
76                )
77
78        ),
79
80
81        tabItem(tabName = "zoom",
82
83                plotOutput("zoom",  click = "plot_click", width = "100%")
84
85        ),
86
87        tabItem(tabName = "regression",
88
89                plotOutput("plot2")
90
91        ),
92
93        tabItem(tabName = "offspring",
94
95                fluidRow(
96
97                  box(tableOutput("OFFSPRING"), plotOutput("DATES2"))
98
99                )),
100
101
102
103       tabItem(tabName = "tabnodes",
104
105               box(title = "Interesting nodes ... ",
```

```
                tableOutput("N"))

      ),

      tabItem(tabName = "download",

              box(title = "Export the table ",
                  downloadButton("downloadData", "Download")),

              box(title = "Export the tree ",
                  downloadButton("downloadPlot", "Download")),

              box(title = "Export subset",
                  downloadButton("downloadFasta", "Download")),

              box(title = "Export regression",
                  downloadButton("downloadRegression", "Download"))

      )
    )
  )
)
```

Serveur

Déclaration des variables réactives

```

server <- function(input, output) {

  # Toutes variables reactives dans Shiny se definissent de cette facon. Suite a une modification, la variable sera recalculee
      . Return en fin de description (ici Print_tree prendra toujours la valeur de tree2).

  ## Fonction qui renvoie l'arbre selectionne par l'utilisateur (objet de type phylo)

  Print_tree = reactive({

    inFile <- input$file1

    validate(
      need(input$file1 != "", "Choose a File tree ...")
    )
    if (input$format=="Nexus"){
      tree2 <- read.nexus(inFile$datapath)
    }
    else if (input$format=="Newick"){
      tree2 = read.tree(inFile$datapath)
    }
    tree2
  })


  ## Fonction qui renvoie pour chaques noeuds : le numero du noeud, le nom des feuilles descendantes, les distances feuilles-
      racine et la date d'echantillonage de tous les descendants.
  find.arbre.noeuds = reactive({

    distance  = distRoot(Print_tree(), tips = "all", method = c("patristic"))
    jeu  = as.data.frame(distRoot(Print_tree(), tips = "all", method = c("patristic")))
    date <- as.numeric(sapply(row.names(jeu),trim_names,sep=input$separator,pos=input$Year,fix=TRUE))

    validate(
      need(date != "", "Incorrect separator/Year position. \n Check your(s) option(s) ")
    )

    jeu <- cbind(jeu,date)
    colnames(jeu)=c("distance","date")

    # Recuperation de la liste des descendants pour chaques noeuds
    noeud = get_descent(Print_tree())
    des_by_node = unlist(lapply(noeud, length))
    noeud_num = rep(1:length(des_by_node), times = des_by_node)
    tips_tot = unlist(noeud)
    arbre = data.frame(cbind(noeud_num, tips_tot))

    # Creation de la colonne qui va servir de reference au merge
```

```
    jeu = cbind(tips_tot = rownames(jeu), jeu)

    arbre_complet = merge(arbre, jeu, by="tips_tot")

    arbre.noeud = split(arbre_complet, arbre_complet$noeud_num)
    arbre.noeud


  })


  ## Fonction qui renvoie un histogramme des dates d'echantillonages de notre jeu de donnees complet.

  hist_dates = reactive({

    distance  = distRoot(Print_tree(), tips = "all", method = c("patristic"))
    jeu  = as.data.frame(distance)
    date <- as.numeric(sapply(row.names(jeu),trim_names,sep=input$separator,pos=input$Year,fix=TRUE))

    validate(
      need(date != "", "Sampling dates unavailable. \n Check your(s) option(s)")
    )

    data=data.frame(value=date)
    ggplot(data) + geom_histogram(aes(x = value),  binwidth = 1, fill = "grey", color = "black") + xlab("Sampling dates")
  })


  ## Fonction qui renvoie l'histogramme des dates des descendants du noeud choisi par l'utilisateur.

  hist_dates_subset = reactive({

    distance  = distRoot(Print_tree(), tips = as.character(List_Offspring()), method = c("patristic"))
    jeu  = as.data.frame(distance)
    date <- as.numeric(sapply(row.names(jeu),trim_names,sep=input$separator,pos=input$Year,fix=TRUE))

    validate(
      need(date != "", "Sampling dates unavailable. \n Check your(s) option(s)")
    )

    data=data.frame(value=date)
    ggplot(data) + geom_histogram(aes(x = value),  binwidth = 1, fill = "grey", color = "black") + xlab("Sampling dates")
  })



  ## Fonction qui renvoie les parametres statistiques (pv, Rsquare et pente) de tous les noeuds pour lesquels la regression
       est possible.

  table_nodes = reactive({

    validate(
      need(input$file1 != "", "Please select a data set")
    )

    # Applique la regression pour chaques noeuds
    modele = lapply(find.arbre.noeuds(), function(x) lm(distance ~ date, data = x))

    resume = lapply(modele, summary)

    Tab = data.frame(des = sapply(find.arbre.noeuds(), nrow),
                     num = sapply(find.arbre.noeuds(), function(x) as.character(unique(x$noeud_num)))
    )
    Tab$num_ID = factor(as.numeric(as.character(Tab$num)) + (Print_tree()$Nnode + 1))

    # Recuperation des parametres statistiques sous reserve que les 3 conditions d'applications de la regression soient
         possibles
    autres = do.call(rbind, lapply(resume, function(x) c(Pvalue = x$coef[nrow(x$coef),4], Rsquared = x$adj.r, Slope = x$coef[
         nrow(x$coef),1])))

    autres[!sapply(find.arbre.noeuds(), function(y) nrow(y) > 2 & length(unique(y$date)) >1 & nrow(unique(data.frame(y$
         distance,y$date))) > 2) ,] = NA

    Tab = cbind(Tab, autres)
    Tab = Tab [!is.na(Tab$Slope),]

    Tab
  })
```

```
## Fonction qui renvoie la p-value du test de Mantel (si on choisit de la vouloir)
# Objectif : verifier la correlation structure temporelle et structure genetique dans un arbre.

mantel_essai = reactive({

  validate(
    need(input$file1 != "", "Test of Mantel unavailable")
  )

  for (i in 1:length(find.arbre.noeuds())){

    if (length(find.arbre.noeuds()[[i]][,1]) > 2 & length(unique(find.arbre.noeuds()[[i]]$date)) >1 & as.integer(as.vector(
        unique(find.arbre.noeuds()[[i]]$noeud_num)))+(Print_tree()$Nnode+1) == as.integer(numero_noeuds())){
      dat1 = find.arbre.noeuds()[[i]]$date
      m1 = dist(dat1)
      m2 = distTips(Print_tree(), tips=as.character(find.arbre.noeuds()[[i]]$tips_tot), method=c("patristic"), useC = FALSE)

      print(mantel.rtest(m1,m2)[5]) # Recuperation de la p-value du test de Mantel
    }
    else if (as.integer(as.vector(unique(find.arbre.noeuds()[[i]]$noeud_num)))+(Print_tree()$Nnode+1) == as.integer(numero_
        noeuds())){
      print("Mantel test impossible")
    }
  }
})

## Fonction qui renvoie le numero du noeud le plus proche du clic grace a un systeme de pointage.

numero_noeuds = reactive({

  validate(
    need(input$file1 != "", "Please select a data set")
  )
  lastPP <- get("last_plot.phylo", envir = .PlotPhyloEnv, inherits = FALSE)
  subedge <- lastPP$edge

  XX <- lastPP$xx[subedge[, 1]]
  YY <- lastPP$yy[subedge[, 2]]
  tab = cbind(XX,YY)

  # Recuperation des aretes et des coordonnnees des extremites des aretes
  df = data.frame(cbind(subedge,tab)) #

  toto =NULL
  for (i in 1:length(df$V1)){
    test=which(df$V1==df$V1[i])
    test = test[which(test!=i)] # Recuperation pour un noeud donne, de l'autre arete qui est en commun avec ce noeud
    toto = append(toto,test)}


  Xnodes = NULL
  Ynodes = NULL
  for (el in toto){
    Xnodes = append(Xnodes, df$XX[el])
    Ynodes = append(Ynodes, (df$YY[el]+df$YY[which(toto==el)])/2) # Coordonnees Y qui est une moyenne
  }

  # Recuperation des coordonnees de tous les noeuds
  coord_nodes = unique(data.frame(cbind(df$V1,Xnodes,Ynodes)))

  # Recuperation des aretes et des coordonnnees des extremites des aretes
  df = data.frame(cbind(subedge,tab)) #

  toto =NULL
  for (i in 1:length(df$V1)){
    test=which(df$V1==df$V1[i])
    test = test[which(test!=i)] # Recuperation pour un noeud donne, de l'autre arete qui est en commun avec ce noeud
    toto = append(toto,test)}


  Xnodes = NULL
  Ynodes = NULL
  for (el in toto){
    Xnodes = append(Xnodes, df$XX[el])
    Ynodes = append(Ynodes, (df$YY[el]+df$YY[which(toto==el)])/2) # Coordonnees Y qui est une moyenne
  }

  # Recuperation des coordonnees de tous les noeuds
  coord_nodes = unique(data.frame(cbind(df$V1,Xnodes,Ynodes)))

```

```
    ## ratio :  pour adapter l'echelle et ponderer l'importance d'un axe par rapport a l'autre dans le calcul de la distance
    ymax = max(coord_nodes$Ynodes)
    if (xmax < ymax){
      ratio = ymax/xmax
    } else {
      ratio = xmax/ymax
    }

    x = input$plot_click[[1]]
    y = input$plot_click[[2]]/ratio

    validate(
      need(input$plot_click != "", "Please click on nodes")
    )
    # Calcul du noeud le plus proche du clic a l'aide du systeme de pointage, de la fonction de distance et des coordonnees des noeuds.
    d <- apply(as.matrix(coord_nodes),1,dist_coord,x,y,ratio)
    names(d) = coord_nodes$V1
    id <- names(d)[order(d)][1]
    id

  })


  ## Renvoie la liste des feuilles descendantes du noeud choisi

  List_Offspring = reactive({

    for (i in 1:length(find.arbre.noeuds())){

      if (as.integer(as.vector(unique(find.arbre.noeuds()[[i]]$noeud_num)))+(Print_tree()$Nnode+1) == as.integer(numero_noeuds())){
        offspring = find.arbre.noeuds()[[i]]$tips_tot
      }
    }
    offspring
  })
```

Construction des différentes composantes (objets output) de l'application

```

  # Sorties (Output) qui vont s'afficher dans l'interface

  ## Affichage arbre phylogenetique (avec feuilles) avec les noeuds qui s'allument selon la PV et le Rsquare

  output$contents <- renderPlot({

    ## Pvalue significative et pente positive
    Tab2 = table_nodes()[(table_nodes()$Pvalue<input$PV  & table_nodes()$Slope>0),]


    coco = findInterval(Tab2$Rsquared, seq(0,0.8,0.2))
    couleur = c("yellow", "orange", "orange3", "indianred2", "red"  )

    plot(Print_tree(),show.tip.label=TRUE, cex=0.6, edge.lty = 1, main="Local Temporal Signal ")

    if (nrow(Tab2)>0){
      nodelabels("",as.numeric(as.character(Tab2$num_ID)), frame="none",pch=16, col = couleur[coco],cex=1.8)}
    else{NULL}
    legend("topright", legend=c("<0.2","0.2-0.4", "0.4-0.6", "0.6-0.8", ">0.8"), col=couleur, pch=16, lwd=2, title = "R? values")

  })



  ## Affichage arbre phylogenetique (sans feuilles) avec les noeuds qui s'allument selon la PV et le R?

  output$contents_NF <- renderPlot({

    ## Pvalue significative et pente positive
    Tab2 = table_nodes()[(table_nodes()$Pvalue<input$PV  & table_nodes()$Slope>0),]

    coco = findInterval(Tab2$Rsquared, seq(0,0.8,0.2))
    couleur = c("yellow", "orange", "orange3", "indianred2", "red"  )

```

```
36
37    plot(Print_tree(),show.tip.label=FALSE, main="Local Temporal Signal ")
38    if (nrow(Tab2)>0){
39      nodelabels("",as.numeric(as.character(Tab2$num_ID)), frame="none",pch=16, col = couleur[coco],cex=1.8)}
40    else{NULL}
41    legend("topright", legend=c("<0.2","0.2-0.4", "0.4-0.6", "0.6-0.8", ">0.8"), col=couleur, pch=16, lwd=2, title="R? values"
          )
42
43  })
44
45
46  ## Affiche un dataframe de tous les noeuds ou le signal temporel est statistiquement positif
47
48  output$N <- renderTable({
49
50    ## P-value significative et pente positive
51    Tab2 = table_nodes()[(table_nodes()$Pvalue<input$PV  & table_nodes()$Slope > 0),]
52
53    validate(
54      need(nrow(Tab2)>0, "No interesting node(s)"))
55    Tab2
56
57  },include.rownames=FALSE,include.colnames=TRUE,align='c',  digits=6)
58
59
60
61  ## Renvoie les parametres statistiques (pv,Rsquare,pente) du noeud le plus proche du clic
62
63  output$INFOS_NOEUDS <- renderTable({
64
65    Tab3 = table_nodes()[table_nodes()$num_ID == numero_noeuds(),]
66    Table_summary = t(Tab3[,c("Pvalue","Rsquared","Slope")])
67
68    validate(
69      need(Table_summary != "", "Unavailable informations"))
70
71    data.frame(Table_summary)
72
73  },include.rownames=TRUE,include.colnames=FALSE,align='c', digits=6)
74
75
76
77  ## Renvoie le numero du noeud selectionne
78
79  output$infos_nodes = renderPrint(
80    {
81      cat("You choose the nodes : ")
82      numero_noeuds()
83    }
84  )
85
86
87  ## Affichage du resultat du test de Mantel
88
89  output$infos_mantel = renderPrint(
90    {
91      cat("Mantel-Test p-value : \n")
92      mantel_essai()
93    }
94  )
95
96
97  ## Affichage du plot de la regression du noeud choisi (si les conditions sont valides)
98
99  output$plot2 = renderPlot({
100   for (n in 1:length(find.arbre.noeuds())){
101
102     if (length(find.arbre.noeuds()[[n]][,1]) > 2 & length(unique(find.arbre.noeuds()[[n]]$date)) >1 & as.integer(as.vector(
          unique(find.arbre.noeuds()[[n]]$noeud_num)))+(Print_tree()$Nnode+1) == as.integer(numero_noeuds())){
103
104       plot(find.arbre.noeuds()[[n]]$distance~find.arbre.noeuds()[[n]]$date, xlab="Sampling dates", ylab= "Root-to-tip
            distance", main=paste("Linear Regression for the node : ",(as.integer(as.vector(unique(find.arbre.noeuds()[[n]]$
            noeud_num)))+(Print_tree()$Nnode+1))))
105       reg = lm(find.arbre.noeuds()[[n]]$distance~find.arbre.noeuds()[[n]]$date)
106       abline(reg)
107
108     }
109     else if ((as.integer(as.vector(unique(find.arbre.noeuds()[[n]]$noeud_num)))+(Print_tree()$Nnode+1) == as.integer(numero_
          noeuds()))){
110       plot(NULL, xlim=c(0,1), ylim=c(0,1), ylab="", xlab="")
```

```
111         text(0.5,0.5,"REGRESSION UNAVAILABLE : \n insufficient number of points or identical sampling dates ", col="firebrick2
                ", cex=1.3, font=3)
112       }
113     }
114   })
115
116
117
118   ## Fonction pour exporter la table des noeuds interessants.
119
120   output$downloadData <- downloadHandler(
121     filename = function() {
122       paste(input$file1, ".txt", sep = "")
123     },
124     content = function(file) {
125       write.table(table_nodes(), file, row.names = FALSE, sep="\t", dec=".", quote=FALSE)
126     }
127   )
128
129
130   ## Fonction pour exporter les sequences FASTA des echantillons descendants du noeud choisi (pour une analyse "tip-dating" en
          aval).
131   # Cela implique d'avoir fourni la sequence FASTA du jeu de donnees complet au prealable
132
133   output$downloadFasta <- downloadHandler(
134
135     filename= function() "snp.fasta",
136
137     content = function(file) {
138
139       validate(
140         need(input$file2 != "", "Choose a File Fasta ...")
141       )
142       # Lecture d'un alignement FASTA fourni par l'utilisateur
143       inFile2 = input$file2
144       dna <- read.dna(file = inFile2$datapath, format="fasta", as.character = "TRUE", as.matrix="TRUE")
145       m = dna
146
147       selection = NULL
148
149       for (i in 1:length(find.arbre.noeuds())){
150
151         if (as.integer(as.vector(unique(find.arbre.noeuds()[[i]]$noeud_num)))+(Print_tree()$Nnode+1) == as.integer(numero_
                noeuds())){
152           selection = i
153         }
154       }
155
156       include_list = as.character(find.arbre.noeuds()[[selection]]$tips_tot)
157       # Recuperation des alignements des echantillons qui sont des descendants du noeud choisi par l'utilisateur
158       dna2 = m[include_list,]
159
160       isa=as.numeric(apply(dna2=="a"|dna2=="A",2,sum)>0)
161       ist=as.numeric(apply(dna2=="t"|dna2=="T",2,sum)>0)
162       isc=as.numeric(apply(dna2=="c"|dna2=="C",2,sum)>0)
163       isg=as.numeric(apply(dna2=="g"|dna2=="G",2,sum)>0)
164       # Somme. Si la valeur est superieure a 1, cela veut dire qu'il y a un snp
165       issnp=isa+ist+isc+isg
166
167       # On conserve la position des snp
168       snp=dna2[,issnp>1]
169       write.dna(snp,file,format="fasta",nbcol=-1,colsep="")
170     }
171   )
172
173
174
175   ## Export de l'arbre (avec les noeuds interessants).
176
177   output$downloadPlot <- downloadHandler(filename = function(){
178     paste("image","pdf",sep=".")
179   },
180
181   content = function(file) {
182     pdf(file)
183     Tab2 = table_nodes()[(table_nodes()$Pvalue<input$PV  & table_nodes()$Slope>0),]
184
185     coco = findInterval(Tab2$Rsquared, seq(0,0.8,0.2))
186     couleur = c("yellow", "orange", "orange3", "indianred2", "red"  )
187
```

```
    plot(Print_tree(),show.tip.label=TRUE, cex=0.6, edge.lty = 1, main="Local Temporal Signal ")

    if (nrow(Tab2)>0){
      nodelabels("",as.numeric(as.character(Tab2$num_ID)), frame="none",pch=16, col = couleur[coco],cex=1.8)}
    else{NULL}
    legend("topright", legend=c("<0.2","0.2-0.4", "0.4-0.6", "0.6-0.8", ">0.8"), col=couleur, pch=16, lwd=2, title = "R? values")
    dev.off()
  },
  contentType='pdf'
  )



  ## Export du plot de la regression

  output$downloadRegression <- downloadHandler(filename = function(){
    paste("regression","pdf",sep=".")
  },
  content = function(file) {

    pdf(file)

    for (n in 1:length(find.arbre.noeuds())){

      if (length(find.arbre.noeuds()[[n]][,1]) > 2 & length(unique(find.arbre.noeuds()[[n]]$date)) >1 & as.integer(as.vector(unique(find.arbre.noeuds()[[n]]$noeud_num)))+(Print_tree()$Nnode+1) == as.integer(numero_noeuds())){

        plot(find.arbre.noeuds()[[n]]$distance~find.arbre.noeuds()[[n]]$date, xlab="Sampling dates", ylab= "Root-to-tip distance", main=paste("Linear Regression for the node : ",(as.integer(as.vector(unique(find.arbre.noeuds()[[n]]$noeud_num)))+(Print_tree()$Nnode+1))))
        reg = lm(find.arbre.noeuds()[[n]]$distance~find.arbre.noeuds()[[n]]$date)
        abline(reg)

      }
      else if ((as.integer(as.vector(unique(find.arbre.noeuds()[[n]]$noeud_num)))+(Print_tree()$Nnode+1) == as.integer(numero_noeuds()))){
        plot(NULL, xlim=c(0,1), ylim=c(0,1), ylab="", xlab="")
        text(0.5,0.5,"REGRESSION UNAVAILABLE : \n insufficient number of points or identical sampling dates ", col="firebrick2", cex=1.3, font=3)
      }
    }

    dev.off()
  },
  contentType='pdf'
  )


  ## Affichage du zoom si l'on encadre manuellement une zone sur l'arbre

  output$zoom <- renderPlot({

    # Conservation des coordonnees du minimum et maximum en x et y de l'encadrement effectue par l'utilisateur
    xmin_range_str <- function(e) {
      if(is.null(e)) return("NULL\n")
      else (e$xmin)
    }

    xmax_range_str <- function(e) {
      if(is.null(e)) return("NULL\n")
      else (e$xmax)
    }
    ymin_range_str <- function(e) {
      if(is.null(e)) return("NULL\n")
      else(round(e$ymin, 3))
    }
    ymax_range_str <- function(e) {
      if(is.null(e)) return("NULL\n")
      else(round(e$ymax, 3))
    }

    plot(Print_tree())
    lastPP <- get("last_plot.phylo", envir = .PlotPhyloEnv)
    subedge <- lastPP$edge

    XX <- lastPP$xx[subedge[, 1]]
    YY <- lastPP$yy[subedge[, 2]]
    tab = cbind(XX,YY)

```

```
    df = data.frame(cbind(subedge,tab)) ## aretes + coordonnnes extremites aretes

    toto =NULL
    for (i in 1:length(df$V1)){
      test=which(df$V1==df$V1[i])
      test = test[which(test!=i)]
      toto = append(toto,test)}

    Xnodes = NULL
    Ynodes = NULL
    for (el in toto){
      Xnodes = append(Xnodes, df$XX[el])
      Ynodes = append(Ynodes, (df$YY[el]+df$YY[which(toto==el)])/2)
    }

    coord_nodes = unique(data.frame(cbind(df$V1,Xnodes,Ynodes)))

    xmax = max(coord_nodes$Xnodes)
    ymax = max(coord_nodes$Ynodes)
    if (xmax < ymax){
      ratio = ymax/xmax
    } else {
      ratio = xmax/ymax
    }

    validate(
      need(input$plot_brush != "", " No selected zoom ! \n Frame an area on your tree.")) # Gestion d'erreur si l'utilisateur
           ne zoome pas

    plot(coord_nodes$Xnodes,coord_nodes$Ynodes, pch=20,xlim=c(xmin_range_str(input$plot_brush),xmax_range_str(input$plot_brush
         )), ylim=c(ymin_range_str(input$plot_brush) , ymax_range_str(input$plot_brush)), main= "Expansion")

    text(coord_nodes$Ynodes~coord_nodes$Xnodes, labels = coord_nodes$V1, pos=4)
  })



  ## Affiche l'histogramme des dates d'echantillonage du jeu de donnees complet
  output$DATES = renderPlot({
    hist_dates()
  })

  ## Affiche l'histogramme des dates d'echantillonage du sous-echantillonage choisi
  output$DATES2 = renderPlot({
    hist_dates_subset()
  })

  ## Affiche la liste des feuilles descendantes
  output$OFFSPRING = renderTable({
    List_Offspring()
  })
}
```

Lancement de l'application

```
## Lecture des parties interfaces utilisateurs (ui) et serveur du code.
shinyApp(ui = ui, server = server)
```