LAAB
Laboratoire Associatif
d'Art & de Botanique

Végéta - Temps et expression du végétal

Rapport de TER

Anaëlle Dambreville, Nabil Ouhi, Idriss Roudmane et Darren Samreth

M1 MIASHS

Encadrement par Sophie Lèbre et François-David Collin

16/04/2018

I - Introduction

Le LAAB (Laboratoire Associatif d'Art et de Botanique) est une structure collective dédiée à l'organisation et la création en art contemporain. Elle est destinée aux artistes expérimentant l'art autour de la botanique. Le flux de sève est l'une des expressions végétales privilégiées par ces artistes qui créent des oeuvres de son et lumière combinés à l'expression du végétal. Les oeuvres sont conçues telles des expériences scientifiques. Par exemple, la Station Affleure 360° est une plateforme de transduction de signaux physiques et biologiques à l'échelle humaine. Elle permet de rendre visible les fluctuations climatiques et leurs incidences sur un *Ginkgo biloba*. La vitesse de rotation et la couleur d'un gyrophare qui surplombe l'installation reflète l'activité de montée de sève de l'arbre.

Il existe un certain nombre de capteurs dédiés à la mesure de différents types de variables des plantes. Ces capteurs fournissent essentiellement des séries temporelles dans des expériences scientifiques aux objectifs précis (Granier, 1987). Le flux de sève mesuré par ces capteurs est une variable complexe à appréhender car elle varie en fonction de plusieurs facteurs environnementaux et propres à la plante (l'espèce de l'arbre par exemple). Au cours d'une même journée, le flux de sève augmente au début du jour puis diminue dans l'après-midi (Cermak et al., 2007). En fonction des saisons, le flux de sève va aussi être plus ou moins élevé. Ces évolutions caractéristiques des séries temporelles couplées au nombre important de variables explicatives rendent complexe l'analyse et la visualisation des flux de sève.

L'objectif principal de notre TER est de déployer toute la problématique de "Data Visualization" sur des données de flux de sève. Ce qui nous intéresse ici, est d'élargir le cadre de ces données à une approche typée Big Data, en faisant ressortir les processus intrinsèques aux plantes, en dégageant des comportements, ou encore des rythmes non triviaux. Notre travail permettra d'apporter un soutien visuel à l'intuition artistique et scientifique.

Pour atteindre ces objectifs, nous avons testé différents types de modélisation (régression linéaire, régression sur composantes principales…) qui couplés à des outils web permettent au final une visualisation simple et interactive du flux de sève en fonction de plusieurs variables environnementales.

II - Données et méthodes utilisées

A - Les données

Nous avons récupéré notre jeu de données à partir du site : **European Fluxes Database Cluster**, qui est une grosse base de données qui héberge dans une seule infrastructure des mesures des flux entre les écosystèmes et l'atmosphère. Elle a été créée dans l'objectif de fournir des outils de traitement et de partage de données standard et de bonne qualité.

Les données sur lesquelles nous avons travaillé sont issues d'une expérimentation dans la forêt domaniale de Puéchabon, à proximité des gorges et de la vallée de l'Hérault, à 45 km de Montpellier (43.74139 (latitude), 3.595833 (longitude)). Le site est soumis à un climat méditerranéen typique comprenant une importante sécheresse estivale. La forêt, dominée par le chêne vert, est étudiée par les chercheurs du CNRS (CEFE).

Les données de flux de sève ont été mesurées à l'aide d'un capteur "thermocouple" qui calcule la vitesse du flux de sève dans un arbre entre deux électrodes (Figure 1; Ben Aïssa et al., 2008; Granier, 1987). Ce capteur est constitué de deux aiguilles séparées de 10 cm environ. L'aiguille du haut est chauffée et la température mesurée par l'aiguille du bas traduit la vitesse du flux de sève (Oishi et al., 2016). Cette mesure a été réalisée pour un chêne vert toutes les 30 minutes pendant un an (2010). En plus du flux de sève, 27 variables environnementales comme la température (TA) et l'humidité de l'air (RH) ont été mesurées (Tableau 1).

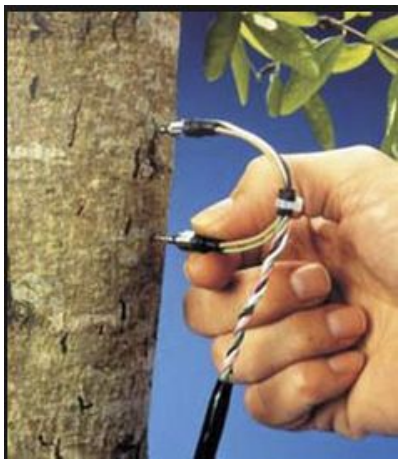


Figure 1 : Thermocouple

Tableau 1 : Variables mesurées à Puechabon

Variable	Description	Unité
TIMESTAMP_START	Date de début	timestamp
TIMESTAMP_END	Date de fin	timestamp
DTime	Date sur 366 jours	
NETRAD	Radiation nette (incluant radiation solaire et infrarouge)	W m^{-2}
P	Précipitation	mm
PA	Pression atmosphérique	Kpa
PPFD_DIF	PPFD diffuse incidente (400-700 nm)	µmol m^{-2} s^{-1}
PPFD_IN	Densité de Flux Photon Photosynthétique (PPFD)	µmol m^{-2} s^{-1}
PPFD_OUT	PPFD réfléchi (400-700 nm)	µmol m^{-2} s^{-1}
RH	Humidité relative	%
SW_IN	Radiations des ondes courtes incidentes (0.3 to 4.5 micron)	W m^{-2}
SW_OUT	Radiations des ondes courtes sortantes (0.3 to 4.5 micron)	W m^{-2}
TA	Température de l'air	°C
TS	Température du sol	°C
WD	Direction du vent	Decimal degree
WS	Vitesse du vent	m s^{-1}
CO2	Concentration du CO2	ppm
FC	Flux CO2	µmol CO_2 m^{-2} s^{-1}
H	Flux de chaleur sensible	W m^{-2}
H2O	Eau	mmol H_2O mol^{-1}
LE	Flux de chaleur latente	W m^{-2}
SB	Stock de chaleur dans la biomasse	W m^{-2}
SC	Flux de Stockage de CO2	µmol CO_2 m^{-2} s^{-1}
SH	Flux de stockage de chaleur sensible	W m^{-2}
SLE	Flux de stockage de chaleur latente	W m^{-2}
TAU	Momentum flux	Kg m^{-1} s^{-2}
USTAR	Vitesse de frottement	m s^{-1}
ZL	Paramètre de stabilité	Sans dimension
PRI	Indice de réflexion photochimique	Sans dimension
G	Flux de chaleur du sol	W m^{-2}
SAP_FLOW	Vitesse flux de sève	mmol H_2O m^{-2} s^{-1}

Le déficit de pression de vapeur (VPD, kPa) est une variable environnementale qui a un effet important sur la croissance des plantes. Nous avons calculé cette variable à partir de l'humidité relative et de la température de l'air à partir de la formule suivante :

$$VPD = ES - EA$$

avec $ES = 0.6108\, exp(\frac{17.17\, TA}{TA + 237.3})$ et $EA = \frac{RH\ ES}{100}$

où *ES* la pression de vapeur saturante, *EA* la pression de vapeur de l'air ambiant, *TA* la température de l'air (°C) et *RH* l'humidité relative de l'air (%).

Les données ont été nettoyées à l'aide de la bibliothèque Pandas de Python, en enlevant les lignes où il y avait uniquement des valeurs manquantes.

Ce jeu de données a été utilisé pour prédire le flux de sève d'une journée en fonction des variables environnementales et de l'heure (heure solaire). D'autre part, elles ont été utilisées pour calculer l'évolution moyenne du flux de sève observée sur une journée pour chaque mois où il y avait assez de mesures.

La prise en main et le nettoyage des données a été réalisé en à peu près 2 semaines.

B - La modélisation

La modélisation a été réalisée sur toutes les données de flux de sève regroupées sans tenir compte de la date ou de l'heure où la mesure a été faite. Au total, nous avons utilisé 9651 mesures de flux de sève.

Le logiciel R a été utilisé pour réaliser toutes les modélisations.

1 – Temps de latence de l'effet des régresseurs sur le flux de sève

Un intervalle de temps peut être observé entre un stimulus environnemental et la réponse physiologique de la plante. C'est ce qu'on appelle un temps de latence. Pour tous les régresseurs, nous avons testé plusieurs temps de latence allante de 0 à 3 heures (par pas de temps de 30 minutes). Pour chacun, nous avons sélectionné le temps de latence optimal en se basant sur la plus forte corrélation avec le flux de sève (Tableau 2).

Tableau 2 : Temps de latence observés pour les régresseurs

Temps de latence	Régresseurs
0	RH, TA, TS, TS_2, TS_3, WS, CO2, SB, G, VPD
30 min	SW_OUT, LE, TAU, USTAR
1 heure	P, PPFD_DIF, PPFD_IN, PPFD_OUT, SW_IN, FC
1 heure 30 min	NETRAD, WD, H
3 heures	PA, H2O, SC, SH, SLE, ZL

2 – Séparation en jeu de données d'entraînement et de test

Nous avons séparé notre jeu de données de façon aléatoire en jeu d'entraînement (70%) utilisé pour l'apprentissage des modèles et de test (30%) utilisé pour la prédiction des modèles.

3 – Étude des corrélations entre les variables

Nous voulons utiliser des régressions linéaires multiples afin de modéliser l'effet des différents régresseurs sur le flux de sève. Ce type de modélisation implique que la matrice des régresseurs doit être de plein rang, c'est à dire qu'aucune variable ne doit être une combinaison linéaire d'une autre variable. Dans un premier temps, nous avons donc étudié les corrélations entre les variables. Ces corrélations ont été calculées grâce au logiciel R (fonction cor()). Nous avons ensuite calculé une matrice de distance (distance euclidienne) afin d'identifier des clusters de régresseurs fortement corrélés entre eux (au delà de 90% de similarité).

Pour chaque cluster de variables fortement corrélées, nous avons gardé uniquement une variable (en se basant sur leur corrélation avec le flux de sève et sur leur pertinence biologique). Neuf régresseurs ont été supprimés à cette étape : NETRAD, PPFD_OUT, SW_IN, SW_OUT, TS_2, TS_3, TAU, G et RH.

4 – Modèle simple de régression linéaire multiple

Le modèle de régression linéaire multiple cherche à établir une relation linéaire entre une variable réponse, ici le flux de sève, et plusieurs variables explicatives, qui sont ici uniquement des régresseurs.

En utilisant les régresseurs sélectionnés précédemment, nous avons sélectionné un modèle de régression linéaire en utilisant la fonction de sélection de modèle regsubsets(). Ce modèle a été sélectionné sur le jeu de données d'apprentissage :

Flux de sève = 1.17 + 0.0023 PPFD_IN_1h -0.035 TA + 0.11 TS - 0.0009 WD_1h30 - 0.0046 CO2 - 0.033 FC_1h - 0.012 LE_30m + 0.017 SH_3h - 0.031 ZL_3h + 1.07 VPD

Un RMSE de 0.78 a été calculé avec les prédictions sur le jeu de données test pour ce modèle.

5 – Modèle de régression linéaire multiple avec des variables transformées

Certains régresseurs ont une relation non linéaire avec le flux de sève. Afin d'améliorer nos prédictions, nous avons transformé ces variables pour rendre leur relation avec le flux de sève linéaire (Tableau 3).

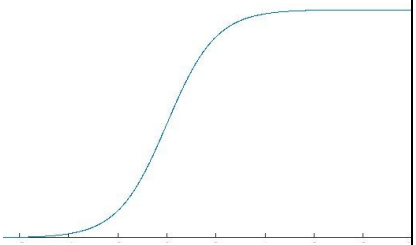
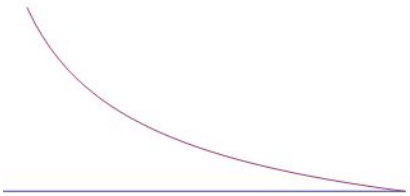
Comme pour le modèle précédent, nous avons sélectionné le meilleur modèle avec la fonction regsubsets() :

Flux de sève = -1.04 - 0.083 P_1h + 0.002 PPFD_IN_1h + 0.054 TS - 0.0004 WD_1h30 + 0.14 WS + 0.015 SH_3h - 0.62 USTAR_30m - 0.025 ZL_3h + **0.031 SB_trans** + **0.44 LE_30m_trans** + **0.12 VPD_trans** + **0.25 CO2**

Les régresseurs en **gras** sont transformés dans ce modèle.

Un RMSE de 0.71 a été calculé avec les prédictions sur le jeu de données test pour ce modèle.

Tableau 3 : Transformation des variables pour le modèle de régression linéaire

Régresseur	Transformation	Type de la relation
TA	$\frac{9.5}{1+e^{\frac{20-TA}{4}}}$	Sigmoïde 
SB	$\frac{9.5}{1+e^{\frac{8-SB}{4}}}$	
G	$\frac{9.5}{1+e^{\frac{6-G}{3}}}$	
LE	$\frac{9}{1+e^{\frac{92-LE}{34}}}$	
VPD	$\frac{9}{1+e^{\frac{1.06-VPD}{0.33}}}$	
CO2	$2.10^{-11}e^{\frac{10000}{CO2+0}}$	Log-inverse 
FC	$0.01\ e^{\frac{210}{FC+50}}$	
TAU	$0.01\ e^{\frac{1}{TAU+0}}$	

6 – Modèle de régression sur composantes principales (PCR)

Nous avons choisi de faire une régression sur composantes principales comme troisième type de modèle afin de prédire l'effet des régresseurs sur le flux de sève.

La régression sur les composantes principales comprend deux étapes successives :

- Une analyse en composantes principales (ACP) qui permet d'explorer les données en les projetant dans un espace à plusieurs nouvelles dimensions
- Une régression linéaire sur les nouvelles composantes obtenues

Pour la première étape (ACP), les valeurs manquantes ont été remplacées par les valeurs moyennes des régresseurs. La figure 2 montre les pourcentages de variance expliquée pour chaque composante et le graphe des corrélations entre les régresseurs.

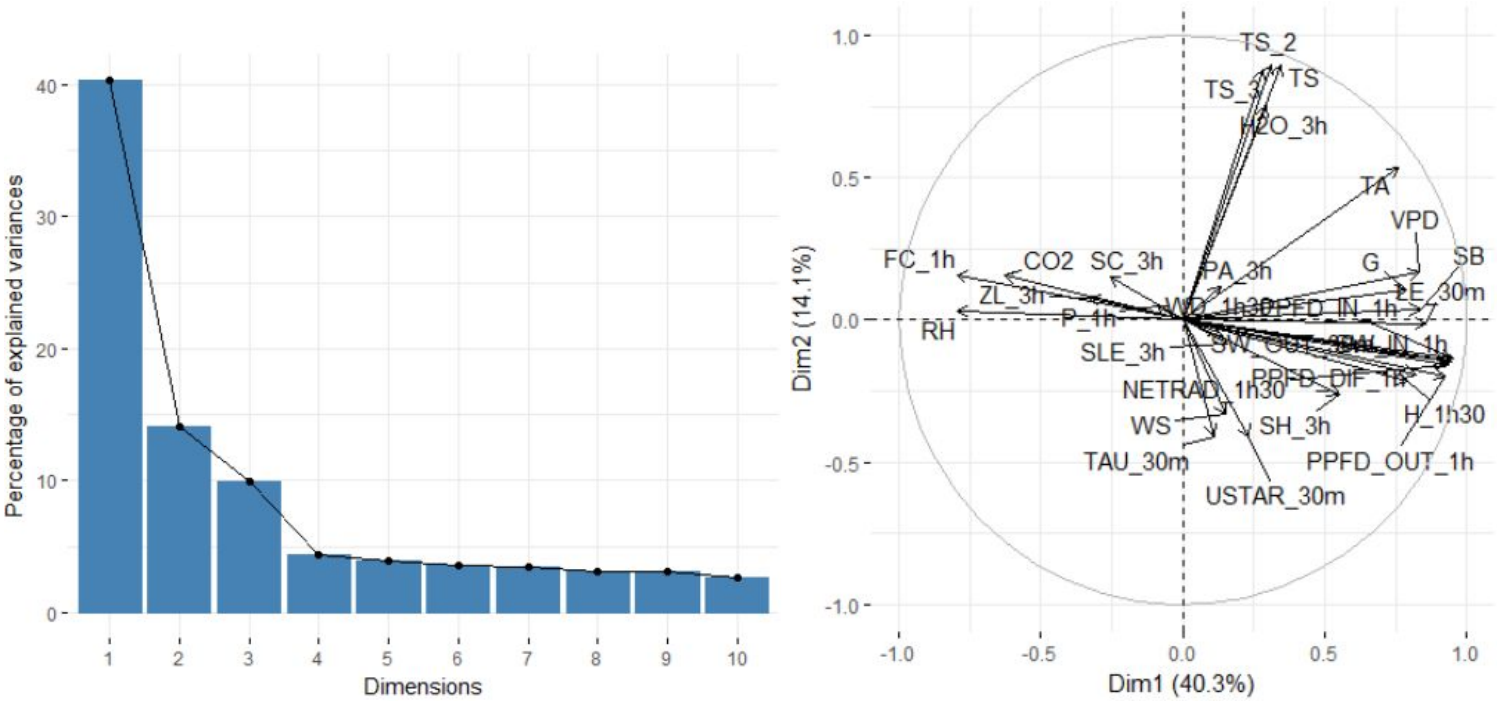


Figure 2 : Analyse en composante principale. Pourcentage de variance expliqué en fonction des dimensions (gauche) et graphe des régresseurs (droite).

Nous choisissons de garder 3 dimensions permettant d'expliquer 64% de la variance observée. Le modèle final est le suivant :

Flux de sève = 1.78 + 0.74 dim1 + 0.015 dim2 - 0.14 dim3

Un RMSE de 0.96 a été calculé avec les prédictions sur le jeu de données test pour ce modèle.

8 – Focus sur l'effet du déficit de pression de vapeur (VPD)

Le VPD est un régresseur qui a effet important sur le flux de sève. Comme il présentait un fort intérêt pour le commanditaire du projet, nous avons décidé de réaliser une modélisation spécifique au VPD.

La relation du VPD avec le flux de sève étant non linéaire, nous avons ajusté une courbe logistique sur cette relation. La fonction a été ajustée grâce à un ajustement non linéaire (fonction nls()). L'ajustement est le suivant :

$$Flux\ de\ sève\ = \frac{8.5}{1+e^{\frac{0.76-VPD}{0.1}}}$$

Le VPD a été transformé en utilisant cette relation puis une régression linéaire a été réalisée entre le flux de sève (variable réponse), et le VPD et PPFD_IN.

Le modèle final est le suivant :

Flux de sève = -0.018 + 0.0031 PPFD_IN_1h + 0.51 VPD

C - La visualisation

Notre objectif principal était de proposer une représentation visuelle de la variation du flux de sève. Il a fallu trouver un outil permettant une visualisation efficace, tout en étant interactif. De plus, nous voulions pouvoir utiliser des données stockées sur un fichier externe. En effet, nos données sont stockées dans des fichiers CSV (mesures, coefficients de régression).

Nous avons décidé de présenter notre travail sous la forme d'un service web. Chaque onglet permet de présenter un type de visualisation différent. Les graphiques doivent être interactifs, car l'objectif est de pouvoir faire varier les variables et de visualiser le résultat de ces variations en temps réel.

1 – Premier outil testé : Python

Python est un langage de programmation flexible qui offre de nombreuses possibilités. L'avantage est que son utilisation aurait permis de développer plusieurs outils dans le même environnement : traitement et analyses des données avec Pandas, modélisation de la régression grâce à Scikit-learn ou Statsmodels, mis en place d'un serveur WEB avec à Flask et visualisation grâce à Dash.

Ces outils auraient facilement pu être connectés entre eux. Ainsi, les données traitées grâce à Pandas dans un premier temps, auraient pu générer un modèle statistique sur Scikit-learn, qui lui-même aurait été utilisé par Dash pour la visualisation.

Toutefois, nous avons finalement penché en faveur d'une deuxième solution de visualisation, qui s'inscrivait directement dans notre programme du semestre, et qui nécessitait moins de prérequis en programmation Python. Celle d'une application Web,

développé directement en HTML et Javascript. Ce type d'application a pour avantage qu'aucune installation ne soit nécessaire, si ce n'est celle d'un navigateur Web. Il a donc été nécessaire de créer une interface web facile d'utilisation.

2 – L'application WEB

a - Bootstrap , le framework

Nous avons utilisé Bootstrap comme framework (qui est une librairie pour le front-end), qui nous permet d'avoir une application à la fois simple d'utilisation et à la fois organisée sur nos pages html. Bootstrap inclus des fichiers ".css" ce qui permet d'avoir un style plus épuré. Il permet de créer entre autres une barre de navigation fixe qui prend souvent trop de place sur un petit écran. Il faut donc parfois la cacher et la montrer uniquement lorsqu'elle est nécessaire. Un menu déroulant s'affiche lorsque l'utilisateur clique sur le bouton de la barre de navigation (option pour les petits écrans ex: smartphone). La fonctionnalité principale pour laquelle nous avons retenu bootstrap comme framework est la fonctionnalité de grille et d'organisation des éléments d'une page permettant de disposer plus aisément les textes, les graphiques et les sliders.

b - Les fonctions Javascript

Différents éléments composent les outils de visualisations :

- Les courbes (associées aux modèles et aux fonctions de calculs)
- Les axes de coordonnées et leur echelles
- Les sliders qui permettent de faire varier l'effet des régresseurs

Pour construire et assembler ces différentes composantes, nous avons utilisé des fonctions javascript et la librairie D3.js. Pour le javascript, c'est un ensemble de boucles et de fonctions qui parcourt le jeu de données, et qui les rend exploitable pour la fonction de modélisation. Nous récupérons ensuite ces données qui sont des coordonnées, puis nous les affichons avec D3 et des balises svg sur la page web.

c - La librairie D3.js

D3.js est une bibliothèque Javascript de représentation graphique interactive. C'est aujourd'hui un des outils les plus populaires de visualisation qui est souvent utilisé pour

l'exploration de données biologiques (Wang et al., 2015). De plus, il est relativement accessible : étant codé en Javascript, son utilisation nécessite juste de savoir comment utiliser la librairie et des connaissances en programmation Web (SVG-Javascript-CSS -HTML).

3 – La visualisation du flux de sève au cours d'une journée

Nous n'avons pas utilisé de dates ou d'heures dans nos modèles, car nous avons sélectionné des régresseurs qui permettent de modéliser intrinsèquement l'effet des saisons (par exemple la température ou la quantité d'eau) et des heures de la journée (grâce notamment aux relevés d'exposition au soleil) sur le flux de sève.

L'évolution du flux de sève sur une journée « moyenne » a été prédite en calculant la moyenne de chaque régresseur (quelque soit la saison) pour 24 heures avec un pas de temps de 30 min.

4 – Les visualisations choisies pour le site web

Nous avons décidé de mettre en place six types de représentation du flux de sève :

- Un modèle linéaire qui permet de prédire l'évolution du flux de sève en fonction de plusieurs régresseurs
- Un deuxième modèle linéaire dont certaines variables ont été transformées afin d'obtenir une relation linéaire avec le flux de sève
- Une régression sur composantes principales
- Une visualisation des trois courbes précédemment citées sur un seul graphe, qui permet de comparer ces trois modèles
- Une représentation qui présente en particulier l'impact du déficit de pression vapeur (VPD) sur l'évolution du flux de sève
- Une représentation qui présente l'évolution moyenne du flux de sève sur 24 heures en fonction des mois (de mai à novembre)

Chaque visualisation est présentée dans un onglet du site.

Nous avons commencé par concevoir deux visualisations : celle du modèle linéaire, et celle de la représentation mensuelle. Ces deux représentations sont totalement indépendantes étant donné que la première permet de représenter la prédiction d'un

modèle en fonction des modalités choisies par l'utilisateur, alors que la deuxième restitue uniquement des moyennes de données observées. Ainsi, nous pouvions développer ces deux visualisations en parallèle, ce qui explique que même si l'utilisation de la librairie D3.js (et plus globalement de Javascript) reste centrale, le déroulement de ces deux scripts ne sont pas identiques. Cette différence est effective dès l'importation du fichier CSV.

Dans la représentation mensuelle, le fichier CSV (qui contient une ligne pour valeur de flux de sève moyens à une heure et un mois donnés) est importé directement par la fonction *d3.csv()*. Les lignes correspondant au mois sélectionné par l'utilisateur sont filtrées, et les valeurs correspondantes au flux de sève et à l'heure sont stockées dans deux tableaux. La courbe est ensuite tracée en utilisant ces deux tableaux de données avec d3.js. Ce processus est réutilisé à chaque fois qu'un nouveau mois est sélectionné par l'utilisateur, par le biais d'une liste de mois que l'on peut cocher. Ainsi, il est possible d'afficher plusieurs mois différents simultanément : chaque mois ayant une couleur unique.

Dans la représentation du modèle linéaire, nous importons les coefficients du modèle de régression et les valeurs moyennes observées sur 24 heures des régresseurs par le biais d'objet *XMLHttpRequest* (qui permet d'importer ces données stockées en format CSV). Le texte récupéré est ensuite mis en forme pour qu'il soit utilisable, puis les valeurs sont stockées dans deux dictionnaires. Pour chaque demi-heure, les valeurs moyennes de chaque régresseur sont multipliées au coefficient multiplicateur choisi par l'utilisateur grâce aux sliders (0 : pas d'effet de la variable, 0.5 : effet divisé par 2, 2 : effet double de la variable). Si la valeur obtenue du régresseur n'est pas observable dans la réalité, la valeur maximum (ou minimum, en fonction des situations) de la variable est utilisée. Sinon, le résultat du produit est utilisé. Ainsi, nous assurons la transposition des coefficients multiplicateurs à des données observables dans la réalité. Enfin, la valeur précédemment obtenue est multipliée par le coefficient du régresseur associé dans le modèle. Le modèle prédit un flux de sève ponctuel (un point) et il est appelé plusieurs fois pour produire toute la courbe (un point toutes les 30 minutes). A chaque variation d'un slider, la courbe est recalculée avec le nouveau coefficient multiplicateur. La figure 3 présente les différentes étapes suivies pour la création et la mise à jour des courbes.

Toutes les autres visualisation des modèles sont dérivées de celle du modèle linéaire. La plus grande difficulté a été de gérer les sliders qui sont créés de façon dynamique pour chaque modèle. Car en effet, chaque modèle a ses propres sliders. Pour créer la visualisation qui permet de comparer nos trois modèles statistiques, nous utilisons un dictionnaire pour chaque modèle qui permet d'affecter correctement les facteurs multiplicatifs des sliders aux régresseurs correspondants de chaque modèle concerné.

Les valeurs affichées à côté des sliders correspondent aux valeurs des régresseurs multipliées par le facteur multiplicatif (valeur moyenne sur la journée). Ces valeurs permettent à l'utilisateur d'avoir un aperçu direct du changement qu'il effectue sur le régresseur grâce au slider.

Nous avons aussi ajouté une fonction qui permet de conserver la position d'une courbe, afin de la comparer à une nouvelle prédiction. Pour cela, nous utilisons l'arborescence de notre HTML. En effet, nous savons que la courbe est créé dans une balise "path" de classe "courbe" qui est elle-même contenue dans une balise "g" avec un identifiant "courbes". Ainsi, nous utilisons ces caractéristiques pour récupérer le contenu de la balise "path" et en faire une copie avec une opacité plus faible.

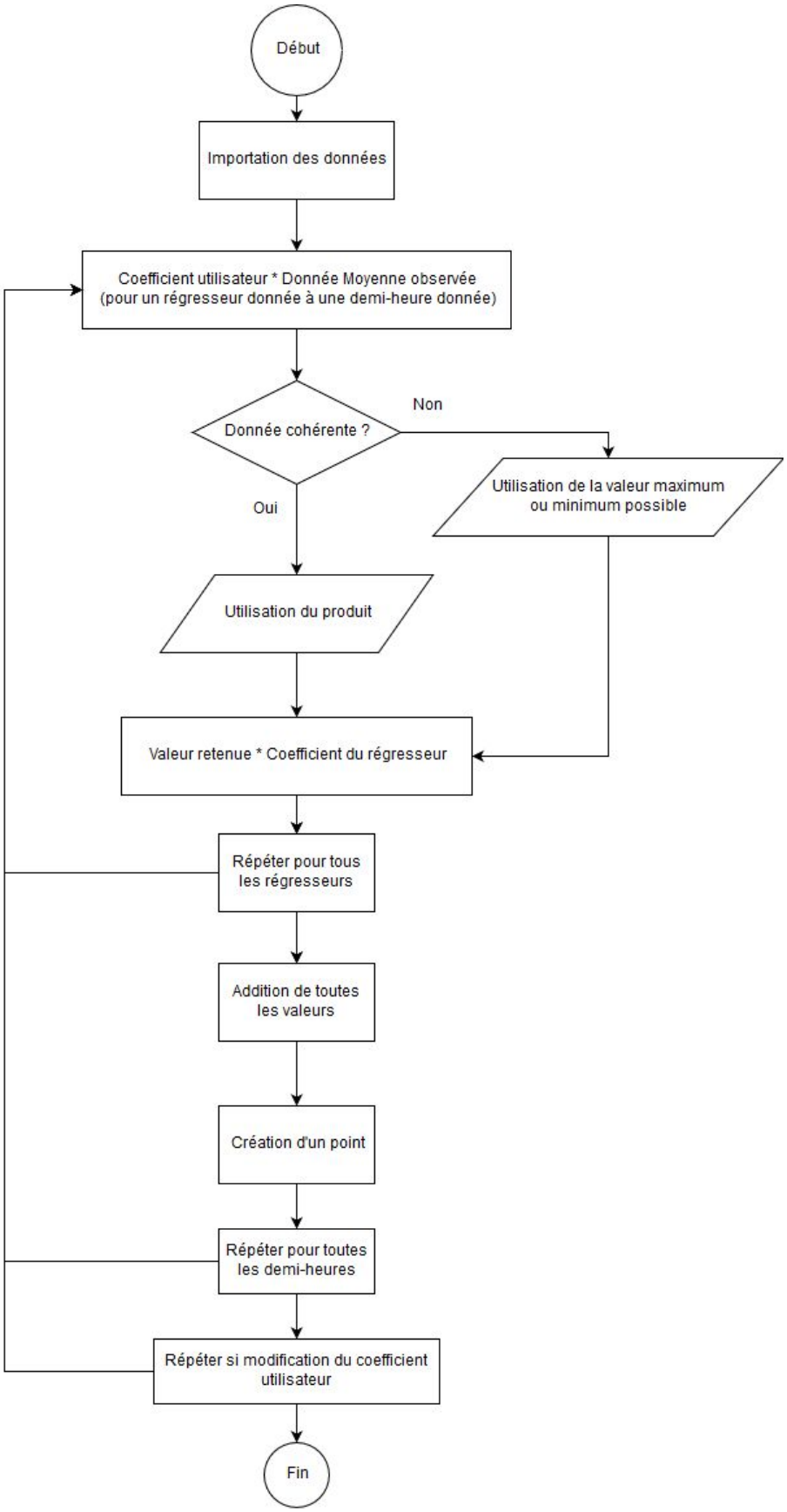


Figure 3 : Etapes suivies pour la création et la mise à jour des courbes du flux de sève prédit par les modèles.

III - Résultats obtenus

Notre résultat principal est une application web visible à cette adresse :

https://darrens34.github.io/Vegeta_Website/index.html

A l'heure actuelle, le site est optimisé pour un affichage avec Mozilla Firefox.

Le projet global y est présenté sur le premier onglet de la page d'accueil (Figure 4). Le résultat de chaque modèle est visualisé dans un onglet (ex pour le modèle linéaire et la PCR Figure 5). Ces visualisations sont basées sur une prédiction dynamique du flux de sève grâce aux modèles décrits précédemment.

Figure 4 : Page d'accueil

Figure 5 : Pages pour le modèle de régression linéaire multiple et pour la PCR.

Un onglet permet de comparer de façon interactive les prédictions des trois modèles (Figure 6).

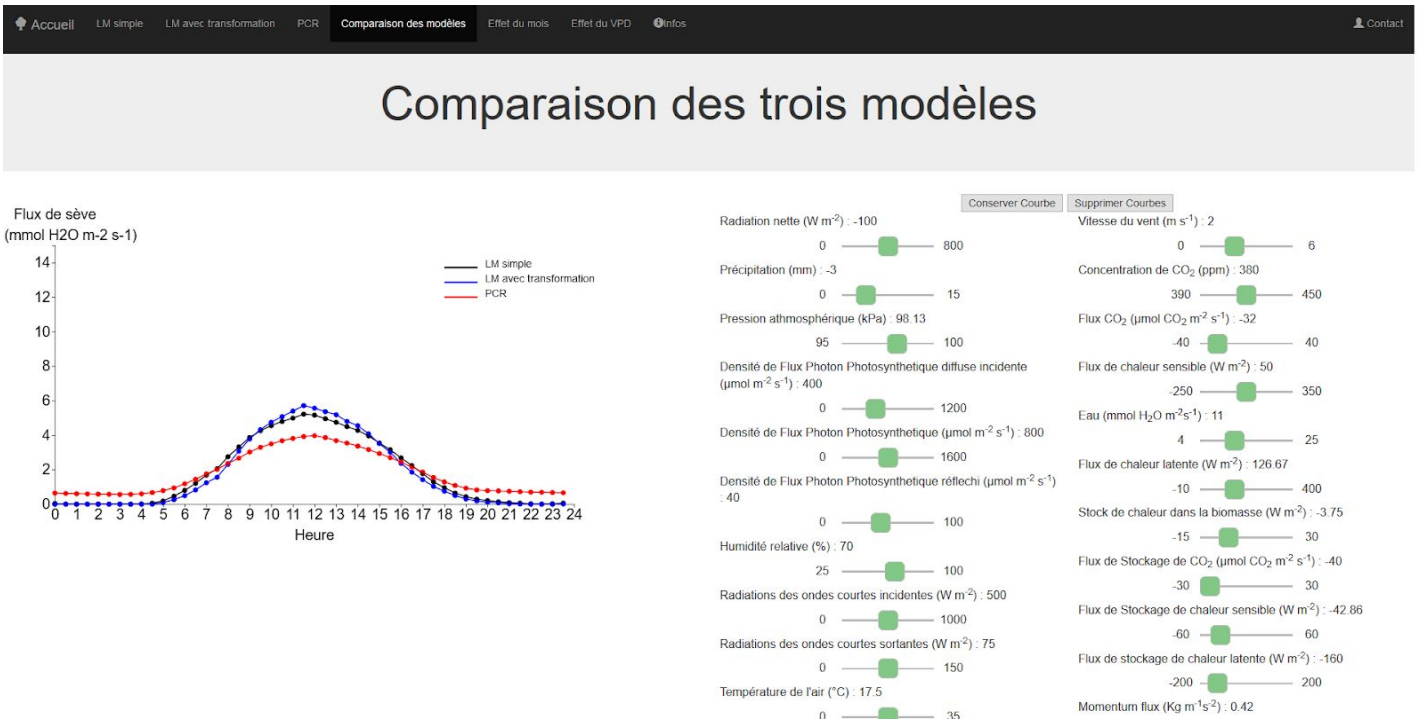


Figure 6 : Page pour la comparaison des modèles.

Les onglets “effet du mois” et “effet du VPD” présentent un focus sur les effets des mois et du VPD sur le flux de sève (Figure 7).

Enfin, un onglet “infos” détaille les différentes variables utilisées, les délais de latence observés pour certains régresseurs et les coefficients de régressions utilisés par les modèles.

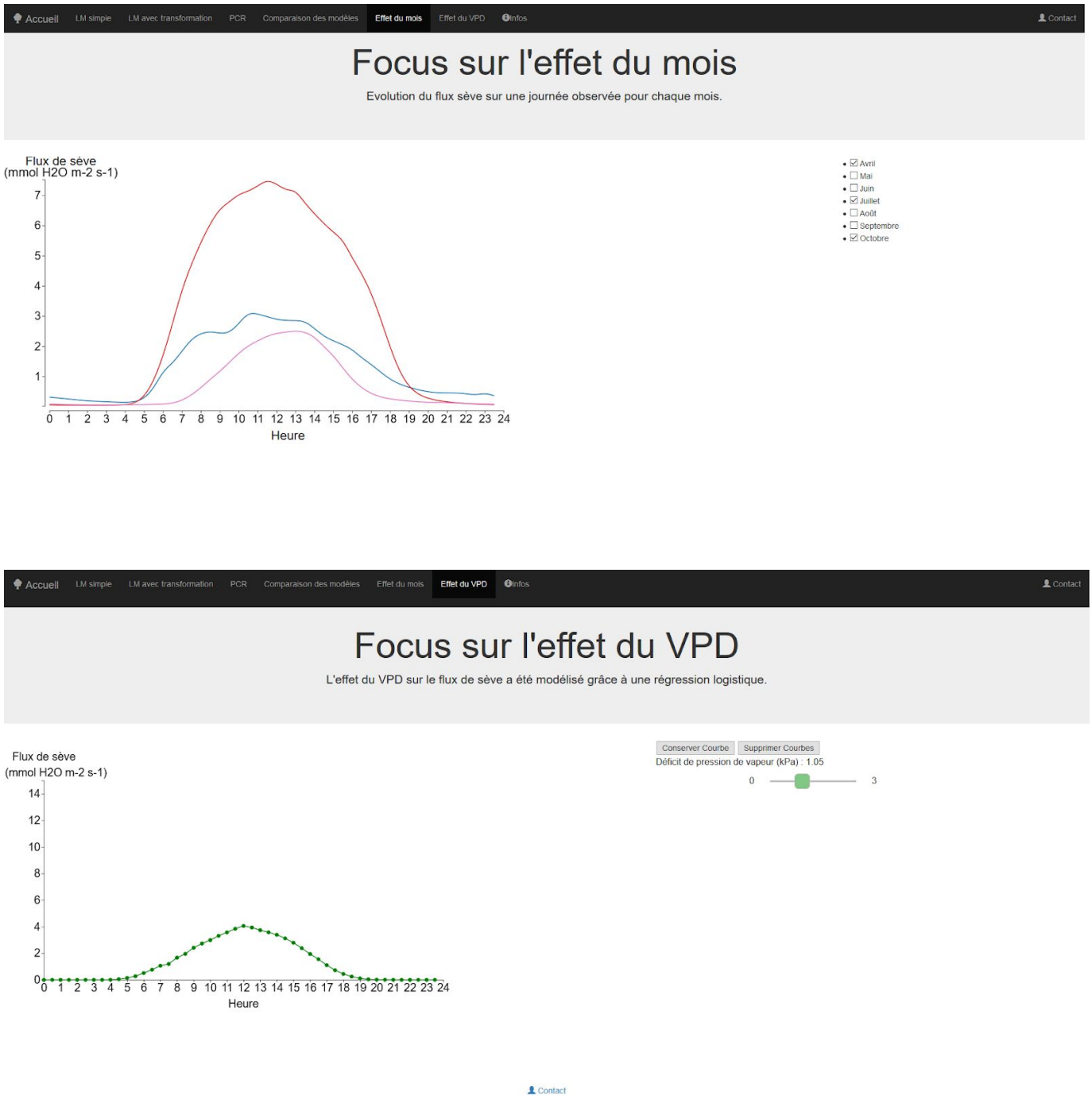


Figure 7 : Pages pour les effets du mois et du VPD.

IV - Bilan des tâches réalisées et problèmes rencontrés

Nous avons terminé toutes les tâches qui étaient prévues au début du projet. Il resterait uniquement des tâches de perfectionnement à réaliser par la suite (Tableau 4).

Tableau 4 : Tâches réalisées et à perfectionner.

Tâches réalisées	Tâches de perfectionnement
- Recueil des besoins du commanditaire - Recherche des données - Nettoyage et enrichissement des données - Modélisation : linéaire, linéaire avec variables modifiées, PCR - Focus sur l'effet du mois et du VPD - Création du site web - Représentation des graphiques en D3.js - Rédaction des rapports	- Optimisation et uniformisation des scripts de représentation graphique - Optimisation pour connexion au site avec d'autres navigateurs - Changer les unités du flux de sève pour le rendre plus explicite (exprimé actuellement en mmol H2O m-2 s-1)

Au cours du projet, nous avons dû faire face à plusieurs problèmes :

- Au niveau du recueil des données : pour trouver des jeux de données. Nous avons réalisé nos modèles uniquement sur un jeu de données. Nous avons essayé d'en trouver d'autres au début du projet mais sans succès (jeu de données avec beaucoup de valeurs manquantes, ou sans flux de sève, ou sans aucune variable environnementale).
- Au niveau de la visualisation : pour assembler sur un seul graphique plusieurs modèles qui dépendent de différents sliders (page de comparaison des modèles). De plus, les pages individuelles des modèles ayant été codées par des personnes différentes, cela a aussi compliqué la tâche d'intégration des modèles sur le même graphique.
- Au niveau de la visualisation : pour afficher à côté des sliders les valeurs des régresseurs modifiées par les sliders et non pas le facteur multiplicatif du slider directement.

V - Conclusion et perspectives

Au long de ce projet nous avons dans premier temps été amenés à explorer et à nettoyer les données brutes. La recherche de nouveaux jeux de données nous a pris plusieurs jours de travail même si au final elle s'est révélée infructueuse. Ces étapes qui sont laborieuses sont néanmoins indispensables à la bonne démarche du data scientist.

Dans un second temps, nous avons conçu un réel outil d'analyse exploratoire en implémentant différents modèles pour que cette application soit consultable par toute personne voulant étudier le comportement du flux de sève selon un ensemble de variables configurables. Nous avons proposé plusieurs types de modélisation du flux de sève. L'onglet de comparaison des modèles permet à un utilisateur d'appréhender facilement les effets des différents régresseurs tout en tenant compte de la variation observée en fonction des différents modèles.

Un outil intéressant qui aurait pu être utilisé pour notre projet est Shiny. Ce package R que nous avons découvert au second semestre permet de créer des sites web en intégrant directement les graphiques issus du logiciel R. Dans notre cas, les graphiques issus de R ont dûs être régénérés grâce à du javascript pour l'affichage sur la page web. L'utilisation de Shiny aurait donc grandement simplifié la mise en place du site web.

En perspective principale, il y a un travail d'homogénéisation et d'optimisation du code à effectuer pour que la maintenance puisse être plus facile par la suite.

Ce TER a été l'occasion pour nous d'appliquer dans un cadre plus réaliste et avec un jeu de données issu du monde réel, les connaissances acquises depuis le début de notre formation. En effet, il nous a permis d'aborder plusieurs disciplines (telles que la régression linéaire, les analyses en composantes principales et sémiologie graphique) et d'explorer des outils de visualisation comme D3js en profondeur.

Bibliographie

Ben Aïssa et al., 2008. Utilisation de la mesure thermique du flux de sève pour l'évaluation de la transpiration d'un palmier dattier. Actes du quatrième atelier régional du projet Sirma. Mostaganem, Algérie, 26-28 mai 2008.

Cermak et al., 2007. Tree water storage and its diurnal dynamics related to sap flow and changes in stem volume in old-growth Douglas-fir trees. Tree Physiology 27, 181–198.

Granier. 1987. Evaluation of transpiration in a Douglas-fir stand by means of sap flow measurements.Tree Physiology 3, 309–320.

Oishi et al., 2016. Baseliner: An open-source, interactive tool for processing sap flux data from thermal dissipation probes. SoftwareX 5, 139–143.

Wang et al., 2015. Open source libraries and frameworks for biological data visualisation: A guide for developers. Proteomics 15, 1356–1374.