



**ESCUELA DE INGENIERÍA AGRARIA Y FORESTAL**

**TRABAJO DE FIN DE MÁSTER**

**MÁSTER UNIVERSITARIO EN GEOINFORMÁTICA PARA LA GESTIÓN DE  
RECURSOS NATURALES**

Ajuste y automatización de un sistema de seguimiento de daños en plantaciones de clones de chopo (*Populus spp.*) empleando sensores ecofisiológicos e Internet de las Cosas (IoT). Aplicabilidad como sistema de alerta temprana

Adjustment and automation of a damage monitoring system in poplar clone (*Populus spp.*) plantations using ecophysiological sensors and the Internet of Things (IoT). Applicability as an early warning system

Estudiante:

Isabel Cristina Grisales Sánchez

Tutores:

Flor Álvarez Taboada

Rodrigo Arthus Bacovich

Universidad de León – Campus de Ponferrada

Convocatoria de Septiembre, 2022

## RESUMEN

El aumento significativo de plagas, enfermedades forestales y el impacto del cambio climático en las plantas, requieren del desarrollo de sistemas el seguimiento y detección temprana de los daños y efectos que provocan sobre las masas forestales, para minimizar su incidencia negativa sobre las mismas. La aplicación de la tecnología del internet de las cosas (IoT) junto con sensores ecofisiológicos son capaces de medir variables que se pueden emplear como indicadores de estrés biótico o abiótico, que permiten el seguimiento y monitoreo en tiempo casi real y de forma remota del estado sanitario de los árboles.

Este trabajo ha tenido como objetivo el desarrollo un sistema de alerta temprana y seguimiento de daños, calibrado en plantaciones de clones de chopo (*Populus spp.*) basado en el IoT y sensores ecofisiológicos TreeTalkers (TT+), proporcionando un reporte final que contiene gráficas de las principales variables y un resumen de las alertas, para que su utilización sea operativa y permita la toma de decisiones por parte del gestor forestal. Para ello, se han desarrollado dos códigos con el lenguaje de programación en R, según el origen de los datos a emplear, que permiten la automatización del procesamiento de los datos registrados por los sensores y el reporte de alertas por valores anómalos. Estos códigos tienen la ventaja de que pueden ser aplicados a otros ecosistemas de sensores con otras especies y en otros lugares.

Se ha realizado la calibración del sistema para choperas, empleando 6 sensores localizados en tres plantaciones de la provincia de León. Analizando los datos con una frecuencia de cada hora entre abril y agosto de 2022, se pudo establecer valores umbral para registrar las alertas de daños/estrés y de funcionamiento del propio sistema. Comparando los análisis de la densidad del flujo de savia medida por los sensores, con los crecimientos en diámetro medidos en campo semanalmente y con el estado sanitario determinado semanalmente de forma visual, ha permitido verificar la validez de esta variable como indicador de alerta temprana de daños en choperas. Además, se han incluido como indicadores de posibles daños los crecimientos radiales del fuste por debajo de un umbral y el ángulo de volcamiento del árbol mayor de 45°, como indicador de posibles daños mecánicos. Se prevé incluir en un futuro otros indicadores de alerta como la radiación recibida por el sensor (indicador de defoliación) o la humedad relativa del fuste.

**Palabras clave:** calibración de sensores, Internet de las Cosas (IoT), TreeTalker, R, alerta temprana, chopo, sanidad forestal.

# ÍNDICE

RESUMEN.....	II
ÍNDICE.....	III
Índice de Figuras .....	V
Índice de Tablas.....	VI
1. INTRODUCCIÓN .....	1
2. OBJETIVOS.....	8
2.1.    Objetivo general.....	8
2.2.    Objetivos específicos.....	8
3. MATERIAL Y MÉTODOS .....	9
3.1.    Zona de estudio y red de sensores.....	9
3.2.    Sensores ecofisiológicos: características, configuración e instalación en campo.....	13
3.3.    Captura de datos.....	18
3.3.1.    Datos de la red de sensores ecofisiológicos .....	18
3.3.2.    Otros datos de campo .....	21
3.4.    Procesamiento de los datos.....	22
3.4.1.    Obtención de los datos de los TT+ .....	24
3.4.2.    Lectura y separación de las bases de datos.....	25
3.4.3.    Revisión de los <i>TTCloud</i> .....	26
3.4.4.    Revisión de los TT+ .....	27
3.4.4.1.    Fecha y hora.....	27
3.4.4.2.    Voltaje de las baterías de los TT+.....	28
3.4.4.3.    Temperatura y humedad relativa del aire.....	29
3.4.4.4.    Densidad del flujo de savia .....	29
3.4.4.5.    Crecimiento medio semanal del radio del fuste del árbol .....	31
3.4.4.6.    Estabilidad del árbol (Ángulo de oscilación).....	32
3.4.4.7.    Radiación electromagnética recibida bajo la copa del árbol .....	34
3.4.4.8.    Humedad relativa contenida en el fuste del árbol (Índice de saturación relativo).....	35
3.4.5.    Alerta de TT+.....	36
3.4.6.    Salidas (outputs).....	37
4. RESULTADOS Y DISCUSIÓN.....	39
4.1.    Código para la adquisición y el procesamiento de los datos de los TT+ .....	39
4.1.1.    Consideraciones generales para utilizar los códigos <i>TT_Check_from_Server.R</i> y <i>TT_Check_from_TXT.R</i> .....	39
4.1.2.    Utilización paso a paso de los códigos: parámetros de entrada en cada uno de los procesos principales .....	40
4.1.2.1.    Parte 1: Obtención de los datos.....	41

4.1.2.2.	Parte 2: lectura y separación de la base de datos.....	42
4.1.2.3.	Parte 3: revisión de los <i>TTCloud</i> .....	42
4.1.2.4.	Parte 4: revisión de los TT+.....	43
4.1.2.5.	Parte 5: alertas de los TT+.....	45
4.1.2.6.	Parte 6: gráficas.....	46
4.1.2.7.	Parte 7: reporte con RMarkdown.....	48
4.2.	Comparación con otras herramientas existentes.....	48
4.3.	Resultados de la aplicación del código desarrollado en choperas (calibración).....	50
4.3.1.	Voltaje de las baterías de los TT+.....	51
4.3.2.	Temperatura y humedad relativa del aire.....	52
4.3.3.	Densidad del flujo de savia.....	56
4.3.4.	Crecimiento medio semanal del radio del fuste del árbol.....	59
4.3.5.	Estabilidad del árbol (Ángulo de oscilación).....	64
4.3.6.	Radiación electromagnética recibida bajo la copa del árbol.....	65
4.3.7.	Humedad relativa contenida en el fuste del árbol (Índice de saturación relativo).....	68
4.4.	Resultados de las alertas.....	70
4.4.1.	Resultados de las alertas del <i>TTCloud</i> .....	70
4.4.2.	Resultados de las alertas de los TT+.....	71
4.5.	Integración en sistema de alerta temprana.....	75
5.	CONCLUSIONES.....	78
6.	REFERENCIAS BIBLIOGRÁFICAS.....	80
	ANEXOS.....	87
	Anexo 1. Tabla de nombres de referencia de las variables de las bases de datos en formato CSV.....	87
	Anexo 2. Código A: <i>TT_Check_from_Server.R</i> .....	88
	Anexo 3. Código B: <i>TT_Check_from_TXT.R</i> .....	111
	Anexo 4. Código RMarkdown para <i>TTReport</i> ( <i>TTReport.Rmd</i> ).....	131
	Anexo 5. Reporte ( <i>TTReport.pdf</i> ).....	135

## Índice de Figuras

Figura 3-1. Ubicación de los TT+ en las 3 parcelas correspondientes (a, b y c). .....	10
Figura 3-2. Partes de un TT+. Fuente: modificado a partir de Matasov <i>et al.</i> (2020). .....	15
Figura 3-3. Instalación de un TT+ en los pies VM_PIE20 (izq.) y VM_PIE15 (der.).....	17
Figura 3-4. Disposición de la instalación del <i>TTCloud</i> en Villamañán (C0210043). .....	18
Figura 3-5. Pie VS_PIE12 sensorizado con el TT+, el sensor de temperatura HOBO Pendant y la cinta dendrométrica.....	22
Figura 3-6. Partes para el procesamiento de los datos de la red de sensores con el código desarrollado en R.....	24
Figura 3-7. Movimientos en los ejes X ( $\theta$ ), Y ( $\psi$ ) y Z ( $\phi$ ). .....	32
Figura 3-8. Disposición del TT+ a 20° del tronco del árbol. ....	33
Figura 4-1. Ingreso de parámetros para Parte 1 del código A en R.....	41
Figura 4-2. Ingreso de parámetros para Parte 1 del código B en R.....	42
Figura 4-3. Parámetros de la Parte 3 del código A en R. ....	43
Figura 4-4. Primeras líneas de la Parte 4 de los códigos A y B. ....	43
Figura 4-5. Resultados de la Parte 4 de los códigos A y B.....	44
Figura 4-6. Captura de pantalla desde RStudio con los nombres de las columnas de dbTT.....	44
Figura 4-7. Generar gráficas temporales para la revisión de las variables. ....	45
Figura 4-8. Parámetros de entrada para la Parte 5 de los códigos A y B.....	46
Figura 4-9. Parámetros de entrada para generar las gráficas en la Parte 6 de los códigos A y B. ....	47
Figura 4-10. Gráfica de la densidad del flujo de savia para una semana con 4 TT+.....	47
Figura 4-11. Parámetro de entrada de la Parte 7 en el código A y B. ....	48
Figura 4-12. Historial del estado de las baterías de los TT+ en mV.....	52
Figura 4-13. Temperaturas registradas por los TT+ de Villamañán durante una semana en julio, 2022. ....	53
Figura 4-14. Temperatura del aire (°C) en Villamañán a partir del VM PIE15. ....	54
Figura 4-15. Humedad relativa del aire (%) en Villamañán a partir del VM PIE15.....	54
Figura 4-16. Gráfico de dispersión de la temperatura de los TT+ vs. temperatura del HOBO en Villamañán. ....	55
Figura 4-17. Gráfico de dispersión de la temperatura de los TT+ vs. temperatura del HOBO en Villasabariego. ....	56
Figura 4-18. Densidad del flujo de savia para el VM_PIE15 fuertemente afectado por estrés hídrico entre junio y julio (Figura 4-19).....	57
Figura 4-19. Aspecto del pie VM_PIE15 el 14 de julio de 2022, con amarilleamiento y pérdida de	

hojas por estrés hídrico. ....	57
Figura 4-20. Densidad de flujo de savia durante un periodo de una semana en la parcela de VM.....	59
Figura 4-21. Distancia entre el sensor decrecimiento radial con la superficie de la corteza del pie 15 (TV_PIE1), de acuerdo al orden de llegada de los datos durante el periodo de estudio.....	61
Figura 4-22. Crecimiento semanal del radio del fuste entre pies de diferentes parcelas. ....	62
Figura 4-23. Gráfico de dispersión del crecimiento del diámetro semanal entre el TT+ y la cinta diamétrica, en el VS_PIE12. ....	63
Figura 4-24. Gráfico de dispersión del crecimiento del diámetro semanal entre el TT+ y la cinta diamétrica, en el VM_PIE20. ....	63
Figura 4-25. Parte de los resultados del ángulo de oscilación para el TV_PIE1 desde el archivo <i>TTDB.csv</i> . ....	65
Figura 4-26. Radiación electromagnética recibida por los pies de Villamañán (VM) y Villasabariego (VS) entre el 24 y el 30 julio 2022. ....	66
Figura 4-27. Cubierta de hojas de los árboles en la parcela de Villamañán, antes (arriba) y después (abajo) del estrés hídrico en verano de 2022. ....	67
Figura 4-28. Resultados de las 12 bandas espectrales calibradas en la base de datos de resultados <i>TTDB.csv</i> . ....	68
Figura 4-29. Saturación relativa de la humedad en el fuste del árbol (todos ellos son clones diferentes). ....	69
Figura 4-30. Ejemplo de alertas del <i>TTCloud</i> . ....	71
Figura 4-31. Reporte de alertas de las baterías desde el <i>TTReport.pdf</i> . ....	72
Figura 4-32. Reporte de alertas de la densidad de flujo de savia bajas del <i>TTReport.pdf</i> . ....	73
Figura 4-33. Parte de la tabla de resultados del <i>TTReport.pdf</i> con las alertas de la tasa de crecimiento semanal. ....	73
Figura 4-34. Reporte de alertas para el ángulo del fuste del árbol de <i>TTReport.pdf</i> . ....	75

## Índice de Tablas

Tabla 1. Características de la red de sensores e información general de los TT+. ....	11
Tabla 2. Características de los sensores incluidos en los TT+ y variables de interés en este estudio calculadas a partir de ellos, de acuerdo con las especificaciones de los sensores. ....	14
Tabla 3. Tipos de datos provenientes del TT+ y correspondencia con la secuencia de datos del ejemplo. ....	19
Tabla 4. Tipos de datos provenientes del <i>TTCloud</i> y correspondencia con la secuencia de datos del	

ejemplo.....	20
Tabla 5. Umbrales de datos para las alertas de los <i>TTClouds</i> .....	26
Tabla 6. Fórmula de calibración para cada dato según su banda espectral.....	34
Tabla 7. Umbrales de datos para las alertas de los TT+.....	36
Tabla 8. Información de las salidas de los procesos con R. ....	37

## 1. INTRODUCCIÓN

En la actualidad, el aumento significativo de plagas, enfermedades forestales y el impacto del cambio climático en las plantas, plantean la necesidad de realizar un seguimiento y detección temprana de los daños y efectos que podrían causar en los ecosistemas forestales, para este modo, minimizar su incidencia sobre estos.

La reducción de los ecosistemas naturales, forestales y agrícolas debido a fenómenos como el cambio climático ha sido ampliamente estudiado y cómo estos a su vez, se relaciona con un incremento en el riesgo de plagas, si bien se ha visto que clima es un factor determinante en la propagación de las plagas cuando se encuentran las condiciones ideales de temperatura, humedad, luz y viento combinadas, que favorecen el ciclo de vida de estos mismos (IPPC Secretariat, 2021). Sin embargo, a pesar de los numerosos estudios que han evaluado la disminución de los bosques y los factores que predisponen a la mortalidad de los árboles a nivel de masa, todavía nos falta una comprensión profunda de los mecanismos ecofisiológicos subyacentes de las especies, la influencia de las diversas condiciones ambientales relacionadas con el suelo, la competencia y el microclima, además de, estrategias específicas de la especie para hacer frente al estrés ambiental prolongado (Buras et al., 2021). Esto implica que las actividades de análisis de riesgo de daños y de plagas que amenazan la salud de las plantas deben intensificarse a nivel nacional, regional e internacional, que a la vez requiere la instalación oportuna de equipos de monitoreo con actividades de vigilancia y seguimiento, con el fin de mejorar los programas de vigilancia multilateral para contrarrestar las amenazas fitosanitarias (IPPC Secretariat, 2021).

A pesar de que frecuentemente se pueden identificar las causas ambientales esenciales propias de la disminución de los bosques en masas forestales, la respuesta individual de los árboles a las condiciones ambientales extremas permanece en gran medida sin explicación (Buras et al., 2021). Como consecuencia, carecemos de una explicación mecánica de las respuestas de crecimiento individual observadas con frecuencia a las condiciones climáticas adversas que, en última instancia, pueden resultar en la muerte y supervivencia de árboles vecinos de la misma especie (Cailleret et al., 2017). Aunque al mismo tiempo, puede estar relacionada con procesos ecofisiológicos clave, como el flujo de savia, la regulación del potencial hídrico de las plantas y los cambios en la conductividad del xilema,



que no pueden estudiarse retrospectivamente en árboles muertos (Buras et al., 2021). Por eso, también se requiere de un estudio individual, oportuno y detallado de los árboles, para entender puntualmente y con rapidez el alcance de esos efectos negativos en las plantas.

Desde este punto de vista, los daños bióticos y abióticos causan una pérdida del crecimiento de los árboles con el consiguiente efecto económico sobre la plantación, por lo que es necesario desarrollar un sistema que permita evaluar el grado de afección de los mismos, y actuar como sistema de alerta temprana y prevención para evitar su evolución o propagación en caso de ser una plaga o enfermedad. De modo que, con el objetivo de realizar un seguimiento del estado sanitario de las masas forestales, es necesario conocer el estado en el que se encuentra, dónde se localiza el posible daño, cuán severo es y cuál es su causa. Por otra parte, la rapidez con la que se detecte el problema influye en gran medida en la detención y ralentización del avance para minimizar así sus efectos en la masa forestal. Por lo tanto, un sistema de seguimiento de daños que posee múltiples factores de estrés en la salud de los ecosistemas forestales y que, en su mayoría son complejos, no lineales y multiescalar, deben ser capaces reconocer varias de esas características espaciales o a nivel de organismo, temporales y de procesos en diferentes escalas como lo indica el estudio de Lausch *et al.* (2018). Que, a su vez, deben de reflejar el nivel de estrés, daño, estado y sus límites de recursos; además que, la información recogida sea estandarizable, comparable, repetible y fácil de compartir; por último, que sea posible de registrar esa información desde diferentes enfoques de monitoreo in situ o remotos (Lausch *et al.*, 2018; Wulder, 2012).

Se espera que a partir de ese mismo reconocimiento de los daños, y con suficiente información recogida, se pueda evaluar el estado de la vegetación y por medio de una clasificación que consta de 3 fases diferentes, ya sea como “sana”, “inestable” o “enferma”, cada una de acuerdo a características que dependen de su diversidad estructural, taxonómica y de rasgos funcionales, comparados con las propiedades características de los procesos o daños como la intensidad, amplitud, consistencia, según cómo influyan en la resiliencia forestal (Lausch *et al.*, 2018; Trumbore *et al.*, 2015). Esta clasificación, ha sido ampliamente utilizada en el monitoreo forestal, como el Sistema de Información Forestal para Europa (FISE, 2022).

Una ventaja que representa este método de sensorización de los árboles in situ, es que, debido a su capacidad de obtención de datos, posibilita la detección temprana del estrés

en una etapa previsual, quiere decir que, se reflejan en características ecofisiológicas internas del individuo, que posteriormente pueden ser reflejadas externamente, por ejemplo, con la caída de las hojas. Capacidad que no se consigue por medio de drones o sistemas remotos y que son de gran utilidad para mejorar las estrategias de gestión forestal y poder responder a los daños inminentes con prontitud (Ecke *et al.*, 2022).

Por otra parte, y dentro del ámbito forestal, España es el tercer país de Europa con mayor cantidad de choperas, siendo Castilla y León la primera región europea en producción de madera de chopo. Se estima que hay unas 44.000 hectáreas de plantaciones de chopo en la región, que suponen un 60% de la superficie cultivada a escala estatal (Chopo en Castilla y León, 2021). Estas plantaciones generan en la comunidad 532.000 m<sup>3</sup> de madera por valor de 32 millones de euros y generan 14.500 puestos de trabajo (CHD, 2020). Por lo tanto, el chopo es una materia prima transcendental para la industria de transformación de la madera, tanto regional como nacional, y prueba de ello es que, en esta región se ubican 7 empresas del sector del envase y tablero contrachapado. Desde un punto de vista comercial, disponer de un sistema de seguimiento de daños como los descritos anteriormente, permitiría una mejora cuantitativa en el sector del chopo en Castilla y León, en términos de productividad. Además de los aspectos comerciales, debe tenerse en cuenta, que detectar de forma temprana y monitorizar los daños en las choperas favorece el éxito en la gestión de las plantaciones existentes, haciéndolas más sanas, más resilientes y por lo tanto, capaces de crecer a un mayor ritmo, tanto para producir madera, como para proporcionar mayores beneficios en los servicios ecosistémicos como la fijación de carbono, y demás aspectos alineados con los Objetivos de Desarrollo Sostenible 2030, como el trabajo decente y el crecimiento económico, producción de consumo responsable y, la acción por el clima (Naciones Unidas, 2022).

En España el sistema nacional de seguimiento del estado sanitario de las masas forestales (naturales y plantaciones) se basa en las observaciones de campo de la red de parcelas IPC-Forests, que a escala europea y nacional proporciona información sobre el estado sanitario de las masas forestales Michel *et al.*, 2020). Ambas redes (Nivel 1 y Nivel 2) suman menos de 650 parcelas. La Red Europea de Seguimiento de Daños en los Bosques (Red de Nivel I). La Red Europea de Seguimiento de Daños en los Bosques se estableció en 1987 para el seguimiento de los daños detectados en los bosques, mediante la revisión de los puntos de una red de 16 x 16 km, sistemática y aleatoria, tendida sobre la superficie forestal

europea. En el Inventario de Daños Forestales (IDF) se presentan los datos de la revisión efectuada en 2019 para 620 parcelas (14.480 árboles) evaluadas en España, así como su evolución respecto a años anteriores). Además, a escala nacional, se trata de la única Red de estas características que existe. En algunos países se ha planteado una nueva Red basada en una complementariedad entre la Red de Nivel I existente y la Red del Inventario Forestal Nacional. Sin embargo, en España, la armonización no se ha llevado a cabo por el momento y la Red a gran escala de seguimiento de Bosques sigue estando formada por los 620 puntos de la Red de Nivel I. La muestra de árboles es constante (siempre se evalúan los mismos árboles), por lo que las comparativas respecto a la evolución histórica de la misma son apropiadas, si bien no tanto la extrapolación al conjunto de las masas forestales españolas (AIEF, 2019). En ningún caso existe más de una parcela (punto de muestreo) en una misma plantación, de modo que no se podrían emplear para hacer el seguimiento de la susceptibilidad a daños de diferentes clones de chopo.

Además, en España se obtiene información del estado fitosanitario de las masas forestales a través del Inventario Forestal Nacional (IFN). En el IFN4 el estudio del estado fitosanitario del arbolado se realizó a partir de datos observados en las parcelas de campo identificando pie a pie, cuando correspondió, el agente causante del daño y la importancia del mismo, así como los elementos del árbol afectados. Se hace sólo en las parcelas del IFN, y con la frecuencia con la que se miden datos en el IFN, en el caso de Castilla y León, los datos de campo del IFN4 son de 2018-19 y los del anterior (IFN3) de 2002-2004, lo que supone 15 años de intervalo (IFN, 2021). Esta baja frecuencia hace imposible emplear el IFN como sistema de alerta temprana y seguimiento de daños.

Aunque las mediciones en campo pueden ser una fuente muy exacta de información sobre el nivel de daño, no solucionan el problema de la detección temprana, además de tener limitaciones en cuanto al nivel de detalle con el que proporciona la información.

La solución alternativa a la planteada podría ser, por lo tanto, el establecimiento de una red de parcelas de campo que se muestrease cada semana o cada mes, pero esto no es viable económicamente ni posible en el caso de choperas de gran extensión. Además, con este sistema, no se puede realizar una alerta temprana (no es posible hasta que los síntomas sean visibles). Por lo tanto, ninguno de los sistemas implantados en España para el seguimiento de daños a través de inventario de campo permite realizar una detección

temprana y en tiempo casi real de daños bióticos y/o abióticos.

En los últimos años se han desarrollado tecnologías que permiten realizar la monitorización de los ecosistemas forestales en tiempo real usando sistemas inteligentes conocidos como “Internet de las cosas” (IoT) (por sus siglas en inglés “Internet of Things”). El término “Internet de las cosas” fue utilizado por primera vez en 1999 por el pionero tecnológico británico Kevin Ashton para describir un sistema en el que los objetos del mundo físico podían conectarse a Internet mediante sensores sin necesidad de la intervención humana (Rose *et al.*, 2015). De modo que, las herramientas IoT proporcionan nuevas formas de conectar los ecosistemas a los sistemas inteligentes de monitoreo, por lo que estas tecnologías se han utilizado en la gestión ambiental, como el mapeo de cambios en la composición y estructura de la vegetación o en la agricultura de precisión (Matasov *et al.*, 2020), así como en la detección de incendios, fuentes de contaminación y seguimiento de la deforestación ilegal (Cui, 2020). Adicionalmente, con la monitorización de los rasgos funcionales de los árboles (como el flujo de savia o la humedad relativa del tronco) y de su respuesta a factores bióticos y abióticos, se desea mejorar la gestión de estas masas y su resiliencia (Zorzi, *et al.*, 2021).

En base a estos sistemas IoT se ha desarrollado un dispositivo multifuncional de bajo coste para la observación en tiempo real de parámetros físicos y biológicos de árboles aplicable al seguimiento de bosques llamado “TreeTalker” (ahora versión TreeTalker+; TT+). Este dispositivo, fue diseñado por Nature 4.0 (Valentini *et al.*, 2019) y es capaz de medir simultáneamente y a cada hora parámetros del árbol como el crecimiento radial del fuste del árbol (con un dendrómetro), el transporte de agua en los árboles (a través de un sensor de flujo de savia), las características espectrales de la radiación electromagnética recibida bajo la copa del árbol (con un espectrómetro), la estabilidad de los árboles (con un acelerómetro), así como parámetros microclimáticos como la temperatura y humedad del aire, así como también, la humedad del tronco y su temperatura en tiempo casi real (Zorzi *et al.*, 2021; Valentini *et al.*, 2019; Matasov *et al.*, 2020). La disponibilidad de estos datos ayuda a la toma de decisiones informadas en tiempo casi real para mejorar la gestión de las masas forestales (Valentini *et al.*, 2019) teniendo en cuenta su estado fisiológico.

En la actualidad hay varias redes de TreeTalkers en el mundo realizando labores de monitorización de ecosistemas forestales. Una de las más extensas es la localizada en Italia,

el proyecto TreeTalkerNetwork (ITT-Net), que monitorea más de 600 árboles individuales, ubicados en 37 sitios desde los Alpes nororientales hasta Sicilia, que tienen como objetivo de estudio evaluar el impacto de los cambios climáticos en los servicios ecosistémicos de los bosques y la muerte regresiva de estos (Castaldi *et al.*, 2020). Otra red, se encuentra ubicada en Norfolk, Reino Unido, desde el 2021, con más de 60 TT+, que desarrolla un sistema de monitoreo Sentinel Treescape en múltiples escalas y multimodal, diseñados para detectar signos tempranos de estrés en los árboles a través de las escalas del paisaje y a nivel de árbol individual, que permiten una mejor gestión de los árboles, incluida la protección o preservación de árboles de importancia cultural y de conservación, y la detección de nuevas plagas y enfermedades invasoras (Gaulton *et al.*, 2022). Además, Moscú cuenta con una red de más de 200 TT+ desde el año 2020, en 7 lugares diferentes dentro de la capital, principalmente en árboles urbanos, con el fin de determinar qué tan peligroso o útil es el árbol y desarrollar algunas recomendaciones de tratamiento (SUN Lab, 2020).

Estos sistemas de monitorización de árboles constituidos por sensores permiten el estudio y la medida de variables ecofisiológicas requieren de una alta frecuencia de obtención de datos para que el sistema trabaje en tiempo casi real (Zorzi *et al.*, 2021). En consecuencia, cada TT en cada árbol registra una elevada cantidad de datos por hora, lo que hace difícil realizar de forma automática (y por parte de un usuario no experto) una evaluación rápida de los datos capturados por los TT presentes en la masa forestal y actuar si se detectan problemas (Zorzi *et al.*, 2021).

Debido a que el uso de los TT+ no está generalizado y es relativamente nuevo, no se dispone de una aplicación de lectura y análisis de datos propia, de modo que el procesamiento de los datos brutos depende del propio propietario/gestor de los TT+. Esto es un inconveniente para los usuarios que no tienen formación o conocimiento del procesamiento de lo que se conoce como el Big data, ya que, al generarse una cantidad de datos tan grande, no es fácil hacer un seguimiento de manera tradicional sino se cuentan con conocimientos de análisis de datos masivos (Kubick, 2012; SAS, 2022). Para solucionar esta necesidad, hasta este momento únicamente los investigadores de la Universidad de Firenze (Zorzi *et al.*, 2021) y de la Free University of Bolzano (Tomelleri *et al.*, 2022) han desarrollado y publicado herramientas (desarrolladas en lenguaje R) con el propósito de permitir al usuario de los TT+ depurar y transformar los datos de manera casi automática, de modo que se obtengan

resultados en poco tiempo, listos para realizar un análisis técnico de las variables ecofisiológicas y ambientales registradas. Ninguno de estas herramientas está diseñado para proporcionar alertas tempranas cuando alguna las variables ecofisiológicas medidas por el TT+ registra valores anómalos y que pueden ser indicadores de una situación de estrés abiótico o biótico en plantaciones de chopo.

Por ello, es necesario desarrollar una herramienta que realice una lectura y procesamiento de los datos recogidos por los TT+ de forma sencilla, sin la necesidad de tener conocimientos avanzados del lenguaje de programación para su uso. Adicionalmente, y para la integración real de estos sistemas en un sistema de alerta temprana de daños en plantaciones de clones de chopo (*Populus spp.*), se requiere que disponga de un sistema de alertas y seguimiento de daños calibrado para estas masas forestales productivas, y que proporcione un reporte final que contenga gráficas de las principales variables y un resumen de las alertas, para que su utilización sea operativa y permita la toma de decisiones por parte del gestor forestal.

## 2. OBJETIVOS

### 2.1. Objetivo general

El trabajo tiene como objetivo el ajuste y automatización de un sistema de seguimiento de estrés y daños en plantaciones de clones de chopo (*Populus spp.*) empleando sensores ecofisiológicos e Internet de las Cosas (IoT).

### 2.2. Objetivos específicos

- ✓ Desarrollar un código en R que permita la automatización del procesamiento de los datos registrados por los sensores (toma de múltiples datos cada hora), así como para el reporte de alertas por valores anómalos.
- ✓ Realizar la calibración del sistema empleando tres ecosistemas de sensores ecofisiológicos para chopo, localizados en la provincia de León, en varias plantaciones con diferentes clones de chopo.
- ✓ Determinar la posibilidad de emplear los datos de los sensores como parte de un sistema de alerta temprana de daños en choperas.

### 3. MATERIAL Y MÉTODOS

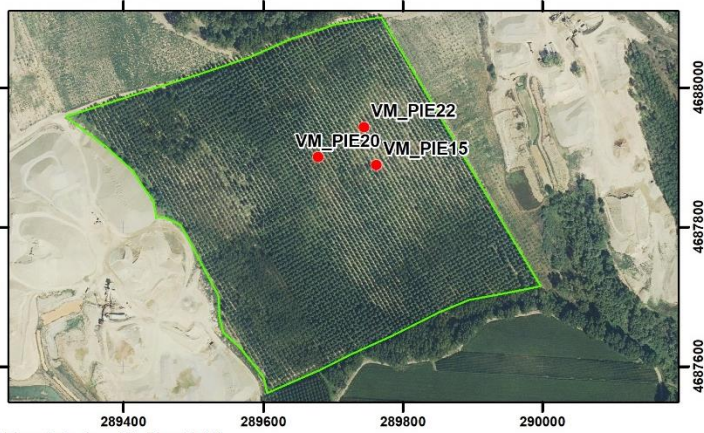
#### 3.1. Zona de estudio y red de sensores

Teniendo en cuenta la importancia de la industria del chopo en la región de Catilla y León, se eligió como zona de estudio la provincia de León, concretamente tres parcelas localizadas en plantaciones de clones híbridos de chopo localizadas en los municipios de Villamañán, Villasabariego y Toral de los Vados (Figura 3-1). Las parcelas de Villamañán y Villasabariego están gestionadas por la empresa Bosques y Ríos y forman parte de la red de seguimiento de daños establecida por la Escuela de Ingeniería Agraria y Forestal (EIAF) de la Universidad de León en abril de 2021, donde se realizan evaluaciones semanales de crecimiento en diámetro y del estado sanitario de 64 pies de diferentes clones de chopo. En estas parcelas durante el año 2021 se detectaron crecimientos semanales en diámetro anormalmente bajos y daños en algunos de los pies, lo que las convirtió en candidatas al establecimiento de sensores ecofisiológicos. La parcela de Toral de los Vados pertenece a la empresa Maderas Castañeiras y se incorporó a la red de seguimiento de daños de la EIAF en marzo de 2022, con el seguimiento semanal de dos pies.

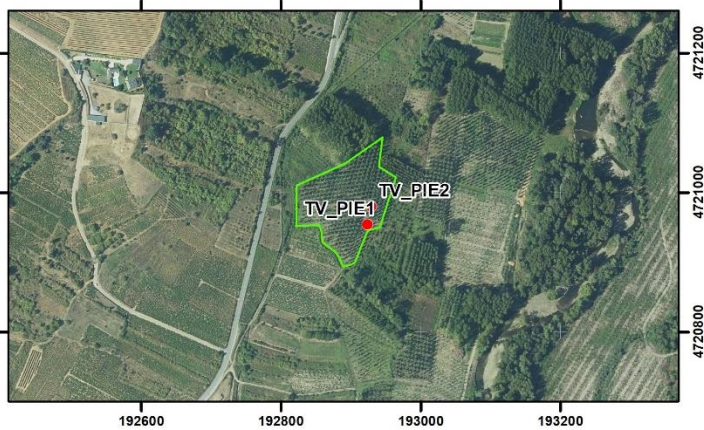
En cada una de las tres parcelas se ubicó un ecosistema de sensores ecofisiológicos TreeTalker TT+ versión 3.3 (TT+) con uno o más sensores (Tabla 1) creados por la compañía italiana Nature 4.0 (Valentini *et al.*, 2019). Los TT+ son plataformas de IoT basadas en microprocesadores, empleando Arduino 1.8.19, construidas sobre un chip ATMEGA328p (Atmel Corp) y equipadas con un transmisor LoRa para la transmisión por radio a un portal (*gateway*) central; capturan información del árbol en el que se instalan y la envían a un receptor/emisor (*TTCloud*) a través de tecnología GSM/GPRS (Matasov *et al.*, 2020; Valentini *et al.*, 2019).



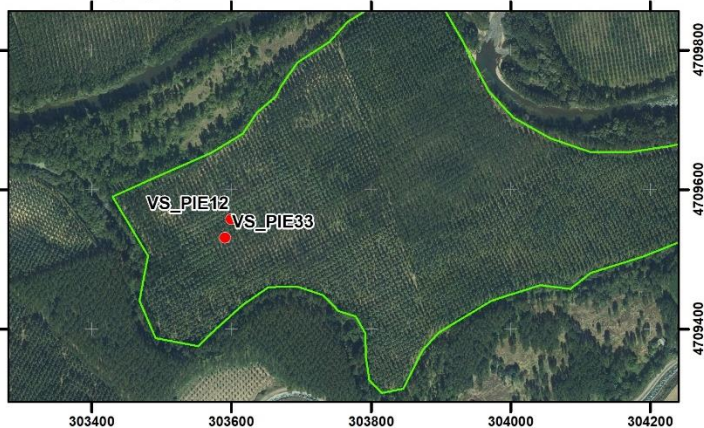
a) Villamañán (VM)



b) Toral de los Vados (TV)



c) Villasabariego (VS)





Provincia de León



España y C. A. de Castilla y León



#### Dispositivo de campo

-  Parcelas de estudio
-  Pies sensorizados



0 75 150 300 m

Escala 1:8.000

Sistema de Coordenadas:  
EPSG 25830

Figura 3-1. Ubicación de los TT+ en las 3 parcelas correspondientes (a, b y c).

De acuerdo con su ubicación y número de identificación de los pies en la red de parcelas de seguimiento, a cada TT+ se le otorgó un código (TTid). El TTsn, corresponde al número de serie (serial number) con que se encuentra registrado cada uno de estos dispositivos desde su fábrica. Se instalaron un total de siete TT+. Sin embargo, tiempo después uno de ellos, localizado en la parcela de Toral de los Vados, por un fallo del sensor, no se dispone de datos de abril a julio, y por lo tanto no se ha incluido en los resultados de este

trabajo. Se han instalado además dos TTCloud, que permiten recoger los datos de los TT+ en tiempo casi real y consultarlos desde el servidor (ver apartado 3.2. Sensores ecofisiológicos: características, configuración e instalación en campo, para una descripción más detallada del sistema).

Tabla 1. Características de la red de sensores e información general de los TT+.

TTsn	TTid	TTCloud	Clon	Edad (años)	Municipio	Fecha de instalación en campo
621B0392	VM_PIE20	C0210043	Raspalje	12	Villamañán	23/03/2022
621B0393	VM_PIE22	C0210043	Beaupre	12	Villamañán	23/03/2022
621B0394	VM_PIE15	C0210043	I-124	12	Villamañán	23/03/2022
621B0395	TV_PIE1	C0210044	Unal	8	Toral de los Vados	31/03/2022
621B0398	TV_PIE2*	C0210044	Unal	8	Toral de los Vados	31/03/2022
621B0396	VS_PIE12	-	Raspalje	11	Villasabariego	23/03/2022
621B0397	VS_PIE33	-	USA 184/411	11	Villasabariego	23/03/2022

\*TT+ no incluido en el estudio.

En una parcela localizada en el municipio de Villamañán, al sureste de la provincia de León, se instalaron tres TT+ (Figura 3-1) y un TTCloud. Esta se encuentra a 744 msnm (MDT del IDECyL, 2022), en una zona predominantemente llana, debido en gran parte a que el río Esla pasa por las inmediaciones con dirección de norte a sur. Considerando la estación climática más cercana a la plantación en el Aeropuerto de León (AEMET, 2022a), esta zona, posee una precipitación media anual de 556 mm, siendo agosto el mes con mayor sequía y diciembre el mes más lluvioso. En cuanto a la temperatura, es de 10,9 °C la media anual, con valores máximos medios de 26,8 °C en el mes de agosto y mínimos de -0,8 °C en el mes de enero.

La parcela de Villamañán (VM) es una plantación multiclonal de chopo establecida en el año 2011 a raíz profunda, con una extensión de 18,1 ha y un marco de plantación de 6 m x 6 m. Entre otros clones, cuenta con pies de tres de los clones más plantados en Castilla y León: I-124 (*Populus x euramericana*), Raspalje y Beaupre (*Populus x interamericana*). Se trata de una chopera en producción, que se ha gradeado y podado de acuerdo con los esquemas clásicos de populicultura aplicados en Castilla y León. Desde el punto de vista sanitario y de daños, a finales de julio y durante agosto de 2021 se detectaron crecimientos semanales en

diámetro anormalmente bajos, e incluso falta de crecimiento y defoliación en varias zonas de la chopera, afectando a varios pies de I-124 y Raspalje. Empleando imágenes de Sentinel 2 MSI y la aplicación “PoplarAlert” desarrollada en la Escuela de Ingeniería Agraria y Forestal, se confirmó que algunos de esos pies (I-214) habían sufrido episodios de estrés también durante el verano de 2020. Por ello se sensorizaron tres pies de esta parcela, dos de ellos (VM\_PIE22 y VM\_PIE15) con episodios de estrés confirmados en 2021 y uno de ellos (VM\_PIE20) sin incidencias relevantes en 2021 respecto a su estado sanitario. Tanto el pie VM\_PIE22 como VM\_PIE20 estuvieron afectados por roya (*Melampsora spp.*) entre agosto y octubre de 2021. Además, para poder determinar si existen comportamientos diferentes en los clones en cuanto a las variables que miden los sensores ecofisiológicos, se seleccionó un pie del clon I-214, otro de Beaupre y otro de Raspalje (Tabla 1). La sensorización de la parcela (instalación y conexión de los sensores y el *cloud* al servidor) se realizó el 23 de marzo de 2022 y se mantiene hasta la actualidad (septiembre 2022). La Figura 3-1 muestra la ubicación de los pies en la parcela.

La segunda parcela de este estudio está ubicada en una chopera del municipio del Toral de los Vados (TV), en el oeste de la provincia de León. Con una elevación de 458 msnm (MDT del IDECyL, 2022), cerca de la desembocadura del río Burbia Siendo la estación climatológica de AEMET en Ponferrada la más cercana a esta parcela (menos de 15 km), ésta se caracteriza por tener una precipitación media anual de 668 mm, siendo diciembre el mes más lluvioso (92 mm medio mensual) y julio el más seco (24 mm medio mensual); la temperatura media anual es de 12,6 °C, alcanzando una media máxima de 28,9 °C en julio y mínimas mensuales de 1,0 °C durante el mes de enero (AEMET, 2022b).

La tercera parcela de este estudio está ubicada en una chopera del municipio del Villasabariego, en el este de la provincia de León (Figura 3-1). Se encuentra a una altitud de 798 msnm (MDT del IDECyL, 2022), ubicada a 17 km de León, también posee un clima muy similar al de Villamañán si se tiene en cuenta la estación climatológica más cercana que es la del Aeropuerto de León (AEMET, 2022a). En esta parcela se sensorizaron 2 pies mediante la instalación de 2 dispositivos TT+ (Figura 3-1 y Tabla 1). Por falta de disponibilidad del equipo, no se pudo instalar un TTCloud que permita enviar la información a la red por medio del IoT, por lo tanto, requiere de visitas a campo para descargar las bases de datos directamente desde los TT+.

Esta parcela de Villasabariego (VS) es una plantación multiclonal de chopo establecida en el año 2012 a raíz profunda, con una extensión de 7,89 ha y un marco de plantación de 6 m x 6 m. Se trata de una chopera en producción, que se ha gradeado y podado de acuerdo con los esquemas clásicos de populicultura aplicados en Castilla y León. Cuenta con más de 30 clones diferentes, entre los que se encuentra USA 184/411, un clon experimental que ha mostrado una gran susceptibilidad al escolítido *Trypophloeus spp.*, que, tras reiterados ataques puede llegar a causar la muerte del árbol. Esta parcela forma parte de la red de seguimiento de daños de la EIAF y Bosques y Ríos desde abril de 2021, y ya en primavera y verano de 2021 se detectaron ataques del escolítido en varios pies de USA 184/411. Uno de esos pies se seleccionó para la sensorización (VS\_PIE33), por su alta probabilidad de sufrir otro ataque y el interés que supone el poder detectarlo con estos sensores. Se sensorizó además un pie del clon Raspalje (VS\_PIE12) (Tabla 1), perteneciente a la red de seguimiento de daños de la EIAF, que no mostró problemas de estrés en 2021, salvo la afección por roya (*Melampsora spp.*) entre agosto y octubre de 2021. La sensorización de la parcela (instalación de los sensores) se realizó el 23 de marzo de 2022 y se mantiene hasta la actualidad (septiembre 2022). La Figura 3-1 muestra la ubicación de los pies en la parcela.

### **3.2. Sensores ecofisiológicos: características, configuración e instalación en campo**

En este trabajo se emplearon como sensores ecofisiológicos los TreeTalkers TT+ versión 3.3. Se trata de dispositivos desarrollados por la compañía Nature 4.0 (Nature 4.0, 2020a y 2020b; Valentini *et al.*, 2019) ubicada en Italia, para el seguimiento de rasgos funcionales de los árboles y que permiten analizar la respuesta de los árboles a ciertas variables ambientales. Una vez instalado el TT+ en un árbol, es capaz de registrar la siguiente información en tiempo casi real (Matasov *et al.*, 2020; Zorzi *et al.*, 2021), con una frecuencia temporal de una hora o menor, que puede ser fijada y modificada en cualquier momento:

- (1) La densidad del flujo de savia, usando el método de disipación térmica transitoria basado en un ciclo de calentamiento/enfriamiento de 10 min cada 1 h (de acuerdo con la configuración empleada en este trabajo), conocido como el método de Granier (Granier, 1987).
- (2) Los espectros de reflectancia de luz a través del dosel en 12 bandas espectrales, usando dos espectrómetros (VIS y NIR).

- (3) Crecimiento del diámetro con un dispositivo óptico de pulsos IR.
- (4) Posición del tronco y oscilación en tres ejes con un acelerómetro incorporado.
- (5) Temperatura del aire al nivel de un solo árbol.
- (6) Humedad del aire al nivel de un solo árbol.
- (7) Humedad relativa de la madera.

La Tabla 2 muestra las características de los sensores integrados en los TT+ de este estudio y que permiten el cálculo de las variables anteriores. La Figura 3-2 muestra el esquema de la integración de estos sensores en el TT+.

Tabla 2. Características de los sensores incluidos en los TT+ y variables de interés en este estudio calculadas a partir de ellos, de acuerdo con las especificaciones de los sensores.

Sensor	Rango	Precisión	Variable
Acelerómetro	0 - 360° (0 - 8 g)	± 0,01°	Estabilidad del árbol (°)
Dendrómetro (sensor de crecimiento radial)	0 - 1 cm	± 200 µm	Crecimiento medio semanal del radio del fuste (mm) a 1,30 m
Sondas de temperatura de referencia y de calentamiento	-40 - +40 °C	± 0,1 °C	Temperatura de la madera (°C) y densidad del flujo de savia (L dm <sup>-2</sup> h <sup>-1</sup> )
Sonda de humedad	0 - 100%	± 2% v/v (resolución no fijada aún)	Humedad relativa de la madera (%)
Espectrómetro de visible	400 - 700 nm	± 20nm a mitad de ancho de banda (HBW) (450, 500, 550, 570, 600, 650, nm) ± 5 nm pico	Reflectancia de la radiación electromagnética que atraviesa el dosel de copas (espectro visible)
Espectrómetro de infrarrojo cercano	700 - 900 nm	± 10nm mitad de ancho de banda (HBW) (610, 680, 730, 760, 810, 860 nm)	Reflectancia de la radiación electromagnética que atraviesa el dosel de copas (espectro de infrarrojo cercano)
Sensor de temperatura y humedad del aire	-10 - +85 y 0 - 100%	± 1 °C y ± 5%	Temperatura del aire (°C) y humedad relativa del aire (%)

Fuente: modificado a partir de Valentini *et al.* (2019), Matasov *et al.* (2020) y Nature 4.0 (2021).

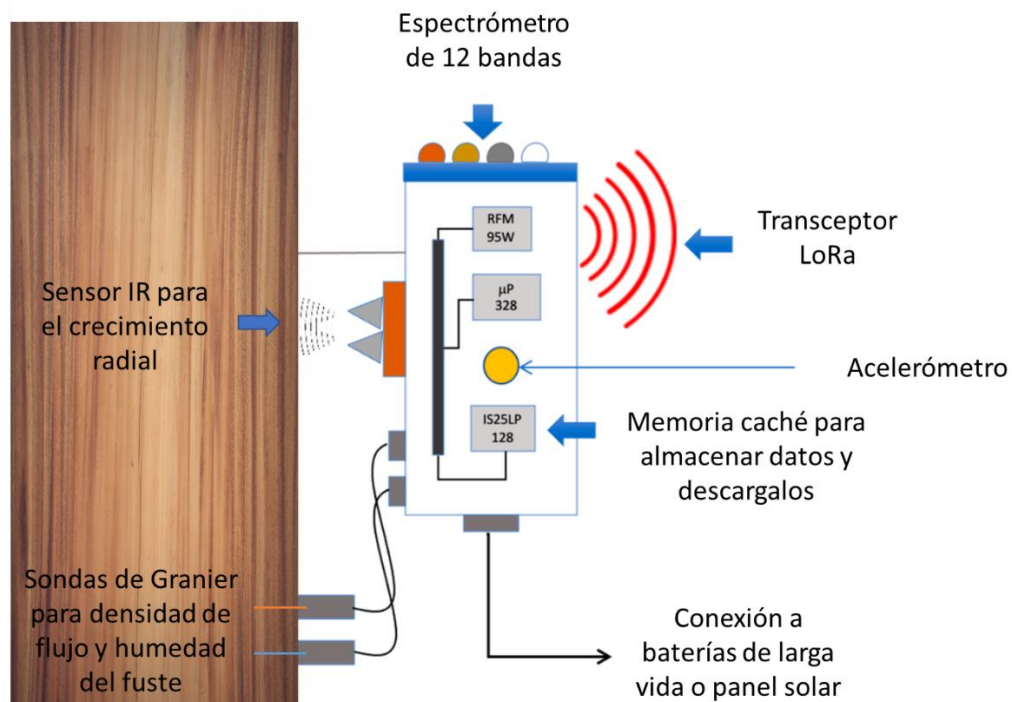


Figura 3-2. Partes de un TT+. Fuente: modificado a partir de Matasov *et al.* (2020).

Previo a la instalación en campo, se realizó la configuración en oficina de cada uno de los ecosistemas. Denominamos “ecosistema” al conjunto formado: Un router/modem llamado *TTCloud* con su batería, panel solar y antena, encargado del registro de los datos provenientes del resto de los sensores y los TT+ y sus baterías. La configuración del “ecosistema” en oficina se realizó de acuerdo con lo indicado en el manual proporcionado por Nature 4.0 (Nature 4.0, 2020b; Asgharinia *et al.*, 2021b). Para cada uno de los TT+ se conectó su batería externa correspondiente y se comprobó el correcto funcionamiento de los sensores, descargando directamente los datos y revisando los registrados durante dos días. Además, se configuró la conexión a internet del *TTCloud* a la red móvil de España mediante una tarjeta SIM de la compañía 1NCE y se estableció en 60 minutos la frecuencia de toma de datos de los TT+, empleando Arduino 1.8.19. Para el resto de los parámetros de medición del TT+ y del *TTCloud* se mantuvieron los que aparecen por defecto, según recomendación del fabricante (Nature 4.0, 2020<sup>a</sup>, 2020b; Asgharinia *et al.*, 2021b). Se comprobó que se producía de forma correcta la conexión entre cada uno de los dos *TTCloud* disponibles y los TT+ asociados a cada una de ellas (Tabla 1), de modo que la información registrada por cada TT+ era recibida por su *TTCloud*. Se comprobó que los datos enviados por los el *TTCloud* sobre su propio funcionamiento y los datos registrados por los TT+ eran recibidos a tiempo y de manera periódica por el servidor de Altervista proporcionado por Nature 4.0 y que se podía acceder a

ellos desde cualquier navegador que tenga conexión a internet. Además, a través de la plataforma 1NCE, que sirve para cambiar los parámetros de los ecosistemas de los TT en forma remota, sin necesidad de hacerlo físicamente en campo, se actualizó el *timestamp*, para que las mediciones de los TT+ y la información del *TTCloud* se correspondiesen con la fecha y hora real. De este modo se configuraron los ecosistemas de las parcelas de Villamañán y Toral de los Vados. Para la parcela de Villasabariego, se realizó la vinculación de los TT+ al *TTCloud* de Villamañán, para comprobar el correcto funcionamiento de todas las funciones de los TT+, si bien no se instaló ningún *TTCloud* en esa parcela, por solo disponer de dos de estos (Tabla 1).

La instalación en campo de los ecosistemas se realizó teniendo en cuenta que un *TTCloud* puede cubrir un radio de hasta 2 km de distancia en condiciones relativamente plana y sin muchos obstáculos que interfieran con la señal. Cada *TTCloud* puede vincularse con un máximo de 20 dispositivos TT+ al mismo tiempo, pero es recomendable que se coloquen un máximo de 16 dispositivos para evitar el choque de datos (Nature 4.0, 2020b). Cada uno de los TT+ en este estudio se instaló en un árbol a menos de 50 m de distancia del *TTCloud*, según lo indicado en la Figura 3-1 y Tabla 1, a una altura aproximada de 1,30 m del suelo como se ve en la Figura 3-3, en el lado norte del fuste del árbol para minimizar la incidencia de la radiación solar en las horas centrales del día, y minimizar de este modo el posible sobrecalentamiento de sus piezas por el sol. La batería externa del TT+ se colocó a una altura aproximada de 2 m, en la cara sur del fuste del árbol, para favorecer la incidencia de la radiación solar, puesto que tiene unas pequeñas placas solares que contribuyen al mantenimiento del nivel de batería. Las sondas, el sensor de crecimiento y el radiómetro se colocaron de acuerdo con las instrucciones del manual proporcionado por Nature 4.0 (2020a; 2021). En la Figura 3-3, se muestra la disposición del TT+ y de la cinta dendrométrica permanente que de modo complementario se colocó en cada uno de los pies para realizar mediciones analógicas de crecimiento del diámetro normal con frecuencia semanal.

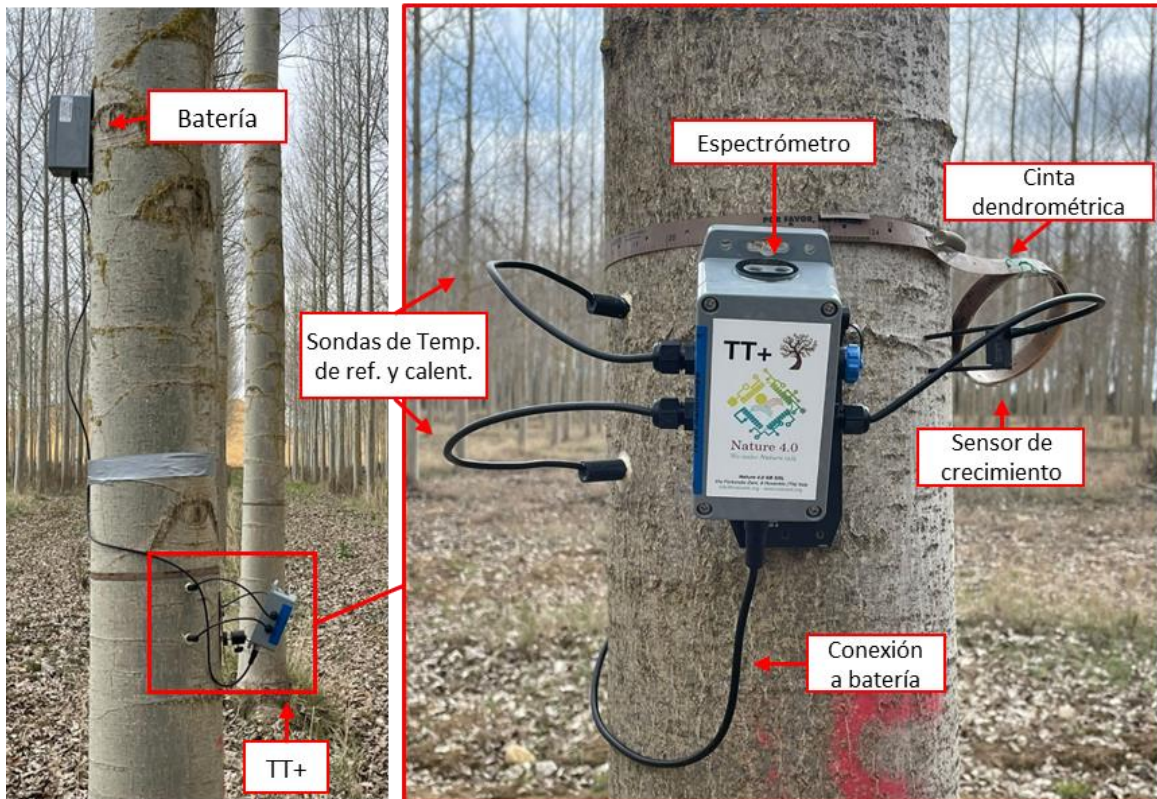


Figura 3-3. Instalación de un TT+ en los pies VM\_PIE20 (izq.) y VM\_PIE15 (der.).

Además, en las parcelas de Villamañán y Toral de los Vados se instaló su correspondiente *TTCloud*. Para ello se eligió un pie de una zona despejada de la parcela, a menos de 50 m de distancia de los TT+, y se colocaron en el fuste el *TTCloud* (a una altura de 1,5 m aproximadamente), su batería externa (a la misma altura aproximadamente), el panel solar (con orientación sur para recibir más radiación y a una altura de 2 m) y la antena de comunicación con los TT+ (también a unos 2 m de altura). De este modo, el *TTCloud* recibe los datos de los TT+ vinculados y los envía por medio del internet a un servidor en línea, con el fin de que se tenga acceso a los datos de los TT+ y del propio funcionamiento del *TTCloud* según la frecuencia programada. La Figura 3-4 muestra la disposición de los *TTCloud* de la parcela de Villamañán.



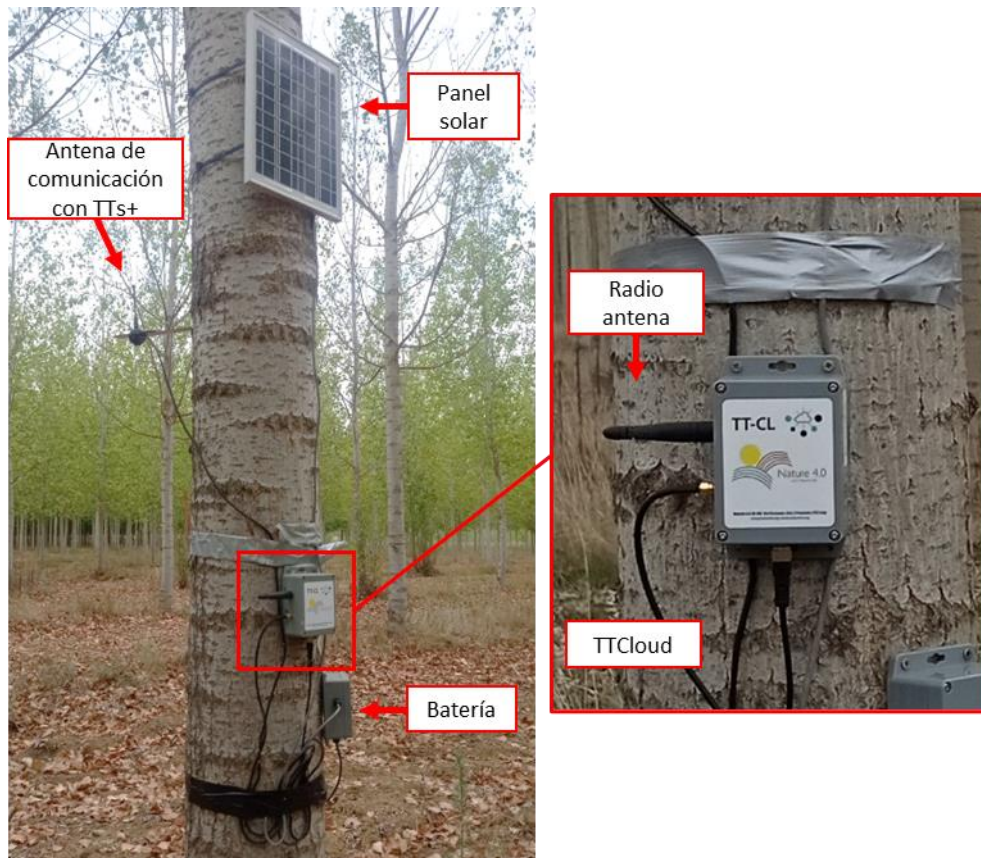


Figura 3-4. Disposición de la instalación del TTCloud en Villamañán (C0210043).

### 3.3. Captura de datos

#### 3.3.1. *Datos de la red de sensores ecofisiológicos*

Cada TT+ graba los datos que captura como una secuencia alfanumérica (que en el caso de ser números lo son con signo) separada por comas y puntos y comas, agrupándolos en dos tipos: (i) '55' o '4D', siendo el primero de la versión más reciente usada en este estudio (TT+ versión 3.3.), y el tipo '4D' el de las versiones anteriores y, (ii) '49', que registran los valores del espectrómetro en 12 bandas electromagnéticas de luz visible y del infrarrojo cercano. A continuación, se muestra un ejemplo de cómo se reciben los datos de un TT+ correspondientes a una medición, y en la Tabla 3 se indica la variable con la que se corresponde cada uno de los valores registrados.

Secuencia de datos recibida del TT 621B0395 a través del servidor:

15.06.22 11:11:20,621B0395;1F68;55;1655283600;33723;33689;46800;  
43781;17;58;249;-3980;0;181;0;-1215;0;33495;30826;1708024;75760

15.06.22 11:11:36,621B0395;1F69;49;1655283600;7398;3369;16618;  
18496;21117;19742;10696;8504;17551;12772;14507;6125;50;3

Tabla 3. Tipos de datos provenientes del TT+ y correspondencia con la secuencia de datos del ejemplo.

Tipo de dato '55' / '4D'	Ejemplo	Tipo de dato '49'	Ejemplo
Server Date	15.06.22	Server Date	15.06.22
Server Time	11:11:20	Server Time	11:11:36
TT+ ID	621B0395	TT+ ID	621B0395
record_number	1F68	record_number	1F69
string type	<b>55</b>	string type	<b>49</b>
Timestamp	1655283600	Timestamp	1655283600
Tref_0 [d.n.]	33723	AS7263_610 [d.n.]	7398
Theat_0 [d.n.]	33689	AS7263_680 [d.n.]	3369
growth sensor [d.n.]	46800	AS7263_730 [d.n.]	16618
adc_bandgap [d.n.]	43781	AS7263_760 [d.n.]	18496
number of bits	17	AS7263_810 [d.n.]	21117
Air relative humidity [%]	58	AS7263_860 [d.n.]	19742
Air temperature [10*° C]	249	AS7262_450 [d.n.]	10696
Ax.out [d.n.]	-3980	AS7262_500 [d.n.]	8504
Ax.out(std.dev) [d.n.]	0	AS7262_550 [d.n.]	17551
Ay.out [d.n.]	181	AS7262_570 [d.n.]	12772
Ay.out (std.dev) [d.n.]	0	AS7262_600 [d.n.]	14507
Az.out[d.n.]	-1215	AS7262_650 [d.n.]	6125
Az.out (std.dev) [d.n.]	0	integration time	50
Tref_1 [d.n.]	33495	gain	3
Theat_1 [d.n.]	30826		
ECf_stem (Hz)	1708024		
adc_Vbat [d.n.]	75760		

El significado de cada una de las variables puede consultarse de forma detallada en Asgharina *et al.* (2021).

Además, en aquellas parcelas con *TTCloud*, se registra y se recibe, además de la información de los TT+, una secuencia alfanumérica de datos propios del *TTCloud* de dos tipos: '4B' y '4C'. A continuación, se muestra un ejemplo de cómo se reciben los datos de un *TTCloud*

correspondientes a una hora, y en la Tabla 4 se indica la variable con la que se corresponde cada uno de los valores registrados.

Secuencia de datos recibida del *TTCloud* C0210044 a través del servidor:

15.06.22 11:00:11,C0210044;1F66;**4B**;1655283600;8037;0;214;1;1;16;  
4114;rel.5.1L

15.06.22 11:00:27,C0210044;1F67;**4C**;1655283600;0;0;-137;-57;  
0;0

Tabla 4. Tipos de datos provenientes del *TTCloud* y correspondencia con la secuencia de datos del ejemplo.

Tipo de dato "4B"	Ejemplo	Tipo de dato "4C"	Ejemplo
Server Date	15.06.22	Server Date	15.06.22
Server Time	11:00:11	Server Time	11:00:27
<i>TTCloud</i> ID	C0210044	<i>TTCloud</i> ID	C0210044
record_number	1F66	record_number	1F66
String Type	<b>4B</b>	String Type	<b>4C</b>
Timestamp	1655283600	Timestamp	1655283600
Accumulated records in memory	8037	TBL_LOCKED (0: tbl_locked=off, 1: tbl_locked=on)	0
Number of records that still needs to be send to the server	0	n° first sensor of the list	0
MCC telephone operator	214	RSSI radio signal strenght between TT+1 & <i>TTCloud</i>	-137
MNC telephone operator	1	RSSI radio signal strenght between TT+2 & <i>TTCloud</i>	-57
GSM registration (1: registered to the network, 0: not registered)	1	Not connected device	0
GSM field level (from 0 to 32)	16	Not connected device	0
Battery level (mv)	4114	Not connected device	0
Firmware Version	rel.5.1L	Not connected device	0

El significado de cada una de las variables puede consultarse de forma detallada en Asgharinia *et al.* (2021).

Para la realización este trabajo se han empleado las medidas registradas por la red de sensores desde su instalación (23 o 31 de marzo de 2022, ver Tabla 1) hasta el 18 de agosto

de 2022, con una frecuencia temporal de una hora, lo que supone aproximadamente 140 días de mediciones. Teniendo en cuenta que un solo TT+ puede recoger 43 datos por hora (Tabla 3) (1.032 por datos por día) y 28 datos/hora por TTCloud (672 por datos por día), que para el periodo total de estudio son aproximadamente 144.000 datos para cada TT+ y 90.000 para cada *TTCloud* instalados en la red de seguimiento de este trabajo.

### **3.3.2. Otros datos de campo**

Para cada uno de los pies sensorizados se realizaron las siguientes medidas, con periodicidad semanal, desde la fecha de instalación de los sensores:

- Crecimiento en diámetro a 1,30 m, empleando dendrómetros permanentes, con precisión de décimas de milímetro. En la parcela de Toral de los Vados las mediciones se iniciaron el 27 de julio 2022, mientras que en Villamañán y Villasabariego se realizaron desde el 23 de marzo de 2022 hasta la actualidad.
- Estado sanitario de cada uno de los pies sensorizados (afecciones foliares y/o corteza).

Además, en las parcelas de Villamañán y Villasabariego se instalaron los sensores de temperatura Hobo Pendant MX2201, configurados para registrar la temperatura del aire cada hora, con precisión de centésima de °C. En la parcela de Villamañán se instaló en el pie VS\_PIE22 y en la parcela de Villasabariego en el pie VS\_PIE12. En ambos casos los datos capturados se consideran representativos de la parcela. Se instalaron el 21 de julio en cada parcela y continúan activados. Los datos se descargaron con periodicidad semanal empleando la aplicación móvil HoboConnect, además de realizar una descarga del total de los datos el 18 de agosto de 2022.

La Figura 3-5 muestra el pie VS\_PIE12 con el TT+, el sensor de temperatura HOBO Pendant y la cinta dendrométrica.



Figura 3-5. Pie VS\_PIE12 sensorizado con el TT+, el sensor de temperatura HOBO Pendant y la cinta dendrométrica.

Estos datos de campo se usaron de soporte para realizar una comparación y posible calibración de los sensores TT+, comparando las mediciones del sensor de crecimiento de los TT+ con las tomadas por las cintas dendrométricas, las medidas del sensor de temperatura del aire de los TT+ con las del HOBO Pendant, y para calibrar/comprobar la correspondencia entre medidas de TT+ con variaciones en el estado sanitario de cada pie.

### **3.4. Procesamiento de los datos**

Para ajustar y automatizar de un sistema de seguimiento de daños en plantaciones de clones de chopo empleando sensores ecofisiológicos e utilizar la información proporcionada por los TT+ en un sistema de alerta temprana es fundamental disponer de una herramienta que permita gestionar de forma automática y sencilla los datos capturados y transmitidos por los TT+ y los *TTCloud*. Todos esos datos en números digitales (*d.n.*) deben ser transformados en información con unidades físicas mensurables, especialmente para las variables con mayor interés en un sistema de detección y seguimiento de daños.

Las variables que se han considerado en este trabajo como más relevantes para detectar anomalías en el estado sanitario de clon de chopo son la densidad del flujo de savia (con valores promedio diarios bajos en la época de crecimiento, por su relación con una

disminución de la evapotranspiración y la pérdida de capacidad fotosintética del árbol (Poyatos, 2016) y el crecimiento en diámetro (con valores bajos o próximos a cero en la época de crecimiento, como consecuencia de estrés hídrico o causado por plagas/enfermedades). Como variables secundarias de interés en el sistema propuesto se consideraron las siguientes: variaciones en radiación transmitida a través de las copas de los árboles (como indicador de una posible defoliación o pérdida de hojas), cambios en la inclinación del árbol (como posible indicador de daños mecánicos o por viento), temperatura y humedad relativa del aire (por su posible relación con un aumento de plagas y enfermedades), cambios en la temperatura y humedad de la madera del árbol (por su posible relación con estrés hídrico y/o daños por plagas/enfermedades).

Además, para poder mantener el sistema de sensores operativo, se requiere que la herramienta de gestión de datos permita acceder a información sobre el nivel de batería de los TT+ y del *TTCloud*, y a si los TT+ siguen conectados o no al sistema.

Otro de los requerimientos que debe cumplir la herramienta que procese los datos de los TT+ es que proporcione dos tipos de alertas al usuario: (i) alertas de funcionamiento del sistema como nivel de batería, intensidad de la señal de transmisión, acumulación de datos sin enviar al servidor, valores anómalos de alguna variable debido al mal funcionamiento del sensor; y (ii) alertas por posible daño como valores muy bajos en el flujo de savia, crecimiento semanal del radio del fuste anómalos o muy próximos a 0, movimientos abruptos en la inclinación del árbol. Estas alertas y la información más relevante del estado fisiológico del pie en estudio deben estar accesible al usuario en forma de informe y en un formato sencillo.

Se comprobó que ninguna de las soluciones previamente planteados para el procesamiento de datos de los TT+ en el lenguaje de programación de R (Zorzi *et al.*, 2021; Tomelleri *et al.*, 2022), cumple los requerimientos planteados, por lo que se decidió desarrollar un código propio para convertir los datos brutos de los TT+ en información relevante para el sistema de seguimiento, realizando los cálculos de manera rápida, pudiendo seleccionar el periodo de análisis y el/los pies a monitorizar, y mostrando los resultados por medio de gráficas representativas de las variables y a través de la generación de un reporte en PDF.

Para ello se desarrolló un código para el procesamiento y tratamiento de los datos de los sensores TT+, empleando el lenguaje de programación en R (versión 4.2.1.) y con el

programa de RStudio, versión 2022.07.1+554. Este código, se ejecuta en 7 partes o procesos principales, que se muestran de forma esquemática en la Figura 3-6 y que se desarrollan de forma detallada en los siguientes apartados.

	Partes (funciones)	Salidas
1	Obtención de los datos	<ul style="list-style-type: none"> <li><input type="checkbox"/> C:/TreeTalkerDB</li> <li><input type="checkbox"/> Raw_fullDB.csv</li> <li><input type="checkbox"/> TTsn_site.txt*</li> </ul>
2	Lectura y separación de las bases de datos	<ul style="list-style-type: none"> <li><input type="checkbox"/> Raw_TTDB.csv</li> <li><input type="checkbox"/> Raw_CloudDB.csv*</li> </ul>
3	Revisión de los TTCloud*	<ul style="list-style-type: none"> <li><input type="checkbox"/> /TTResults</li> <li><input type="checkbox"/> Cloud.csv*</li> <li><input type="checkbox"/> CloudAlert.csv*</li> </ul>
4	Revisión de los TT+	<ul style="list-style-type: none"> <li><input type="checkbox"/> Date_TTDB.csv</li> </ul>
5	Alertas de TT+	<ul style="list-style-type: none"> <li><input type="checkbox"/> Date_TTAlert.csv</li> </ul>
6	Gráficas	<ul style="list-style-type: none"> <li><input type="checkbox"/> /TTPlots</li> <li><input type="checkbox"/> /Date_to_Date_plots</li> <li><input type="checkbox"/> Gráficas </li> </ul>
7	RMarkdown	<ul style="list-style-type: none"> <li><input type="checkbox"/> TTReport.pdf</li> </ul>

\*Solo en descargas desde el servidor

Figura 3-6. Partes para el procesamiento de los datos de la red de sensores con el código desarrollado en R.

### 3.4.1. Obtención de los datos de los TT+

El primer objetivo a lograr a través del código de R fue convertir los datos brutos de los TT+ en unidades físicas, para poder analizar los valores de las variables y así calibrar e los sensores para su empleo en el sistema de alerta por medio de los dispositivos TT+.

Se desarrollaron dos códigos en R para permitir el procesamiento de los datos desde dos formas diferentes, dependiendo de los datos de partida, es decir, de cómo se obtuvieron los datos brutos de los TT+ (A o B).

- A)** Por medio de la descarga de los datos desde el servidor de Nature Talkers y Altervista. Se accede a través de un enlace como el siguiente:

<http://www.naturetalkers.altervista.org/CXXXXXXX/TTCloud.txt>, donde es necesario especificar el identificador del *TTCloud* (número de serie) y posteriormente, definir en el código de R el nombre otorgado a cada uno de los sitios de estudio. El código desarrollado para este tipo de acceso se denominó ***TT+\_Check\_from\_Server.R***.

- B)** Directamente desde los TT+, empleando un monitor de Arduino, conectando el ordenador por medio de un cable USB y un conversor FTDI32 a cada TT+, guardando los datos en formato TXT. El código desarrollado para este tipo de acceso se denominó ***TT+\_Check\_from\_TXT.R***.

La diferencia principal de estos códigos fue, por lo tanto, en si se dispone de un *TTCloud* para la obtención de los datos o si se descargan de manera directa de los TT+, de modo que, la organización de los datos brutos en una sola base de datos será realizada por el código de diferentes maneras. Además, el proceso de revisión de los datos generados por el propio *TTCloud* no tiene lugar para el código ***TT+\_Check\_from\_TXT.R***. Sin embargo, el resto del código funciona con el mismo procedimiento para ambos.

### ***3.4.2. Lectura y separación de las bases de datos***

En este paso el código desarrollado se encarga de separar en dos bases de datos los datos brutos recogidos del paso anterior, que de ser descargadas desde el servidor toda la información que incluye todos los *TTClouds* y sus TT+, quedarían almacenadas en el archivo *Raw\_FullDB.csv*. Luego, se dividen los datos en dos tipos de bases de datos, una con la información de los TT+ (*Raw\_TTDB.csv*) y la otra, con la información de los *TTClouds* (*Raw\_CloudDB.csv*).

Hay que tener presente que ambos dispositivos almacenan los datos de acuerdo con cuatro tipos diferentes de cadenas de datos, *4B* y *4C* corresponden a los datos de los *TTClouds*; *55* o *4D* y *49* corresponden a los datos de los TT+. De este modo, son separados los datos en dos partes correspondientes a cada tipo de dispositivo. En el caso en que se tengan múltiples sistemas, toda la información estará junta representada en las mismas dos bases de datos.



### 3.4.3. Revisión de los TTCloud

A partir de los datos que provee cada *TTCloud*, en este proceso, principalmente se revisa la siguiente información con ayuda del manual del *TTCloud* (Nature 4.0, 2020b).

- ❖ Señal GSM: mide la señal del *TTCloud*. El valor correspondiente está entre 32 (máximo) y 0 (mínimo). Una señal buena está por encima de 15, por debajo de este valor hasta 9 puede presentar fallos.
- ❖ Número de datos pendientes por enviar al servidor: normalmente el 0 a menos de que haya un fallo en la conexión a internet vía GPRS.
- ❖ Nivel de batería: se muestra directamente en milivoltios (mV). En el caso de los *TTClouds*, este valor puede oscilar entre 3300 mV a 4100 mV o más dependiendo de la capacidad de la batería.
- ❖ Señal RSSI (*Radio Signal Strength*): señal entre el *TTCloud* y los TT+ conectados, con un rango entre -137 a -53, siendo el valor -100 una señal pobre y el valor de -50 una señal excelente. Valores de -137 indican que el *TTCloud* no recibe señal del TT+ correspondiente.

Transformando estos valores de números digitales, a sus variables correspondientes, es creada la base de datos llamada *Cloud.csv*, que incluye la fecha del momento en el que el archivo es creado. Posteriormente, en este mismo proceso de revisión del *TTCloud* (paso 3 de la Figura 3-6), es ejecutada la funcionalidad *CloudAlert* que está incluida sólo en el código de R *TT+\_Check\_from\_Server.R*, tomando en cuenta el trabajo de Zorzi *et al.* (2021). Con el objetivo de crear un archivo con alertas en relación con los datos de funcionamiento del *TTCloud* por defecto para el día anterior a la ejecución del código, pero con la posibilidad de conseguir el historial completo desde la instalación de los dispositivos en campo. Estas alertas son reportadas para las variables seleccionadas y según los umbrales de la Tabla 5. Estos umbrales podrían modificarse en el código en caso de ser necesario.

Tabla 5. Umbrales de datos para las alertas de los *TTClouds*.

Variable	Alerta cuando
Señal GSM	< 9

Variable	Alerta cuando
Batería del <i>TTCloud</i>	< 4000
Señal RSSI	< -100
Número de datos pendientes por enviar al servidor	> 10

Para valores que cumplen con los umbrales de las alertas, sus datos serán reportados en el archivo de salida *CloudAlert.csv* y se incluirá la descripción de la alerta o fallo, la identificación del *TTCloud* y la fecha (más adelante ver el apartado 3.4.6 de las salidas u outputs).

#### **3.4.4. Revisión de los TT+**

En esta parte del proceso número 4 como lo indica la Figura 3-6, el código en R convierte los datos brutos de *d.n.* (*digital numbers*) de todas las variables a variables mensurables de datos reales (en sus unidades) y se almacena la información de cada uno de los TT+ en una sola base de datos (*TTDB.csv*) de todo el historial. La mayoría de las ecuaciones y las transformaciones aplicadas están descritas principalmente en el manual de los dispositivos TT+ de Nature 4.0 de la versión 3.0 y posteriores (Asgharinia *et al.*, 2021); no obstante, algunos de los cálculos se modificaron para adecuarlos al objetivo principal de este trabajo. En los apartados siguientes se muestran las ecuaciones y transformaciones integradas en el código R para el cálculo de las variables definidas como de interés en este trabajo. Los nombres de las variables calculadas con el código R se han definido en inglés, así como las propias anotaciones del código, para que su uso no esté limitado a usuarios hispanohablantes.

##### **3.4.4.1. Fecha y hora**

En los datos brutos recogidos por los TT+ y los *TTClouds*, aparecen la fecha y hora de la llegada de los datos al servidor como el *Server Date* y *Server Time*, y, a su vez, está grabado el momento exacto de la lectura de datos dado por el *Timestamp* según la frecuencia configurada, si es cada hora, será tomada exactamente en cada cambio de hora, es decir,

cuando los minutos y segundos estén en ceros. Así que estos valores no necesariamente estas dos fechas serán iguales, ya que los datos no son enviados al servidor inmediatamente, puede tardar unos minutos y si no son enviados debido a una baja señal RSSI entre los TT+ y el *TTCloud*, los datos pueden quedar almacenados en la memoria del TT+. De este modo, en el caso de los TT+ que no tienen un *TTCloud* solo será posible ver la fecha el *Timestamp* en la cadena de datos, la otra no.

El *Timestamp*, también conocido como el tiempo UNIX o tiempo EPOCH, cuenta el número de segundos a partir de la medianoche del 1 de enero de 1970, UTC, y es universalmente usado en los sistemas computacionales (Asgharinia *et al.*, 2021a). Según el valor de esta variable, serán tomados los datos de los dispositivos de acuerdo con la periodicidad definida por el usuario, que en este estudio fue de una hora. Así, se podría comprobar en los datos brutos, una lectura de datos cada 3600 segundos en el *Timestamp*, de acuerdo con el tipo de dato.

La fórmula para transformar el *Timestamp* de los datos brutos a fecha y hora es la Ec. 1 (Asgharinia *et al.*, 2021a). Se debe realizar un ajuste según la zona horaria, por defecto esta está en hora central europea (GMT), por lo que, para España, se realizó un ajuste de 2 horas (GMT +2) que se integró en el código desarrollado

$$Date = (((Timestamp/60)/60)/24) + DATE(1970,1,1) \quad [Ec. 1]$$

También se utilizó la siguiente página web para realizar una conversión rápida de un solo dato y comprobar la fecha y hora: <https://www.epochconverter.com/>.

#### **3.4.4.2. Voltaje de las baterías de los TT+**

En el caso de las baterías de los TT+, se usan las variables obtenidas de los datos brutos como *adc\_Vbat[d.n.]* y *adc\_bandgap[d.n.]* las cuales se transformaron para obtener la carga de la batería en milivoltios (mV) con la Ec. 2 (Asgharinia *et al.*, 2021a).

$$Battery\_Voltage(mV) = 2 * 1100 * \left( \frac{adc\_Vbat[d.n.]}{adc\_bandgap[d.n.]} \right) \quad [Ec. 2]$$

Se debe asegurar que la carga de los TT+ sea superior a los 3,5 mV en todo momento, ya que si es inferior a esta carga el microprocesador y otros componentes pueden presentar fallas en la toma de datos y no se garantiza un funcionamiento adecuado (Asgharinia *et al.*,

2021a).

#### **3.4.4.3. Temperatura y humedad relativa del aire**

Ambas variables permiten monitorear fenómenos abióticos que pueden ser comparables con los de las otras variables propias de la fisiología del árbol medidas por los TT+, como por ejemplo el flujo de savia o el crecimiento (Potayos *et al.*, 2016). A partir de los datos descargados con números digitales, se calculó la temperatura en grados centígrados (°C) aplicando la Ec. 3 y la humedad relativa del aire en porcentaje (%) de acuerdo con la Ec. 4 (Asgharina *et al.*, 2021a).

$$\text{Air temperature (}^\circ\text{C)} = \text{Air temperature [10 * }^\circ\text{C]} / 10 \quad [\text{Ec. 3}]$$

$$\text{Air relative humidity (\%)} = \text{Air relative humidity [\%]} \quad [\text{Ec. 4}]$$

#### **3.4.4.4. Densidad del flujo de savia**

La tasa de transpiración de las plantas se aproxima mucho a la tasa de flujo de savia en el tallo o tronco principal (Nature 4.0, 2020), permite conocer cómo está funcionando este proceso físico y químico del árbol para consumir su energía y crecer. En la estimación de la densidad del flujo de savia (DFS), los TT+ usan un método de disipación térmica con ciclos de calentamiento repetidos, conocido como el método de Granier (Granier, 1987; Isarangkool Na Ayutthaya *et al.*, 2010). Do y Rocheteau (2002) encontraron una relación empírica entre la densidad de flujo de savia ( $J_s$ ;  $l\ dm^{-2}\ h^{-1}$ ) y un índice de cambio en la diferencia de temperatura, denominado índice de flujo alternativo ( $K_a$ ; adimensional).

Por defecto, las sondas de los dispositivos TT+ producen un calentamiento del tejido vivo por 10 minutos, luego tendrá 50 minutos de enfriamiento para determinar el movimiento del flujo de savia. De esta manera, se obtienen las siguientes 4 variables de los siguientes dos tipos de sensores:

- *Tref\_0* [d.n.]
- *Theat\_0* [d.n.]
- *Tref\_1* [d.n.]
- *Theat\_1* [d.n.]

Donde,  $T_{ref}$  es la temperatura de referencia del fuste y  $Theat$  pertenece a la lectura de la sonda de calentamiento en temperatura de números digitales ( $d.n.$ ) del antes (en 0) y después de estar en calentamiento (en 1) según el periodo establecido, que por defecto y en este trabajo, está establecido para un ciclo de cada hora el cual es suficiente si se tiene un promedio diario no superior a  $15 \text{ l dm}^{-2}\text{h}^{-1}$  (Asgharinia *et al.*, 2021a). El calentamiento pulsado o continuo es también usado para rastrear los cambios de temperatura causados por el transporte de calor por convección al mover la savia en diferentes órganos de la planta a lo largo de la vía de transpiración (Burguess *et al.*, 2000). Así bien, cada uno de estos datos, se transforman con el código R desarrollado en valores de temperatura ( $^{\circ}\text{C}$ ) aplicando la Ec. 5 (Asgharinia *et al.*, 2021a).

$$\text{Temperature } (^{\circ}\text{C}) = 127.6 - 0.006045 * d.n. + 1.26 \times 10^{-7} * d.n.^2 - 1.15 \times 10^{-12} * d.n.^3 \quad [\text{Ec. 5}]$$

En este caso, donde se procesan múltiples dispositivos TT+ en una sola base de datos, se agrupó por separado cada uno de estos para que los estadísticos del filtrado de los datos no se mezclen entre las propiedades físicas de cada pie y los valores que se tienen de cada uno. Seguidamente, se aplicaron dos tipos de filtros en los datos de temperatura: el primer filtro, elimina los valores anómalos teniendo en cuenta sólo las temperaturas entre los  $-20$  y los  $50$   $^{\circ}\text{C}$ ; con el segundo filtro, se aplicó el método de Savitzky Golay para suavizar el ruido de los datos de la temperatura para el flujo de savia (RDocumentation, 2022). En RStudio, se utilizó el paquete de R *prospectr*, aplicando la función de `savitzkyGolay()`, que calcula una regresión polinómica local, del grado que se le indique, para determinar el nuevo valor de cada punto equidistante y así suavizar los datos.

Posteriormente, se calculó la diferencia entre las temperaturas  $\Delta T$  que se puede evaluar en función de los datos de la temperatura de las sondas del calentador (Ec. 6) o de los datos de la sonda de referencia y la del calentador (Ec. 7). Cualquiera de las dos ecuaciones puede ser usadas para aplicar en la ecuación K y obtener los resultados densidad de flujo de savia.

$$\Delta T_1 = Theat_1 - Theat_0 \quad [\text{Ec. 6}]$$

$$\Delta T_2 = (Theat_1 - Tref_1) - (Theat_0 - Tref_0) \quad [\text{Ec. 7}]$$

Luego, se calculó el Índice de flujo alternativo ( $K$ ) (Granier, 1987), de acuerdo con el

sistema de enfriamiento de 50 min después de 10 min de calentamiento de las sondas con la Ec. 8 (Asgharinia *et al.*, 2021a).

$$K = \frac{\Delta T_{max(24hrs)} - \Delta T_i}{\Delta T_i} \quad [\text{Ec. 8}]$$

Donde  $\Delta T_{max}$  (°C) es la máxima diferencia de temperatura obtenida a lo largo de un solo día, esta sucede cuando la velocidad del flujo de savia es cero o son mínimas, cuando el flujo disminuye, la diferencia entre temperaturas también disminuye (Granier, 1987; Isarangkool Na Ayutthaya *et al.*, 2010). Aparte,  $\Delta T_i$  (°C), es la diferencia de temperaturas bajo las condiciones de savia medidas.

Por último, para calcular la DFS, se utilizó la Ec. 9 (Asgharinia *et al.*, 2021a) en unidades de densidad de savia por tiempo ( $l \, dm^{-2} h^{-1}$ ).

$$\text{Sap flux density } (l \, dm^{-2} h^{-1}) = 4.79 * K1 \quad [\text{Ec. 9}]$$

El coeficiente 4,79 proviene de la calibración del laboratorio de TT+ de Nature 4.0 utilizando un banco hidráulico en condiciones de laboratorio.

#### **3.4.4.5. Crecimiento medio semanal del radio del fuste del árbol**

Los TT+ permiten conocer la variación del crecimiento radial del fuste como un indicador de la transformación y fotosíntesis del carbono como biomasa, a partir del sensor infrarrojo (IR) de pulsaciones que se encuentra ajustado por dos perforaciones con dos varas de fibra de carbono en el xilema (Figura 3-3), midiendo así la distancia entre este sensor y la superficie del fuste del árbol. Con los datos obtenidos del *growth\_sensor [d.n]*, en el código de R primero se aplicó un filtro que eliminara los valores por superiores a 85.000 e inferiores a 30.000 que son los valores de referencia como la capacidad del sensor (Nature 4.0, 2020a). Posteriormente, se aplicó la Ec. 10 (Asgharinia *et al.*, 2021a), con la variante de que se multiplicó por diez para obtener la distancia en milímetros (mm).

$$\text{Distance from sensor to the bark (mm)} = \frac{237908.4541 + (-1.1171 * \text{growth sensor [d.n]})}{199.4330 + \text{growth sensor [d.n]}} * 10 \quad [\text{Ec. 10}]$$

De este modo, el incremento radial del tallo se traduce en una reducción de la distancia entre el sensor y la superficie del tronco del árbol (Valentini *et al.*, 2019). Luego, en cada TT+, se

calculó la mediana de la variación semanal de la distancia del sensor a la corteza (mm/semana) y así obtener un valor más representativo del cambio de esta variable. Por último, se normalizaron los datos tomando como referencia el máximo valor histórico de la distancia entre el sensor y la superficie del árbol, de este modo, conseguimos el crecimiento medio semanal del radio del fuste en mm desde la instalación de los TT+.

#### 3.4.4.6. Estabilidad del árbol (Ángulo de oscilación)

Con esta variable se espera detectar los cambios en el ángulo del árbol en pie en función de la oscilación gravitacional de un sistema de coordenadas esférico. Sabiendo esto, se puede valorar la estabilidad del árbol y detectar volcamientos potenciales. Para esto, el acelerómetro del TT+ mide la fuerza de aceleración en la unidad  $g$  en tres planos (X, Y y Z) de acuerdo con su dirección. Del mismo modo, estas variables están representadas en los datos brutos como  $A_{x.out}$ ,  $A_{y.out}$  y  $A_{z.out}$ . A partir de estos datos, se pueden calcular tres movimientos o inclinaciones diferentes que tenga el árbol, determinando el ángulo individual de cada eje del acelerómetro desde una posición de referencia, siendo esta, la orientación típica de un dispositivo TT+, con los ejes X y Y en el plano del horizonte ( $0 g$ ) y el eje Z ortogonal al horizonte ( $1 g$ ) (Fisher, 2011).

Si bien, la orientación del dispositivo puede ser entendida con 3 movimientos básicos desde cada uno de los ejes como se observa en la Figura 3-7, donde  $\theta$  es el ángulo entre el horizonte y el eje X del acelerómetro,  $\psi$  es el ángulo entre el horizonte y el eje Y del acelerómetro, y  $\phi$  es el ángulo entre el vector de gravedad y el eje Z.

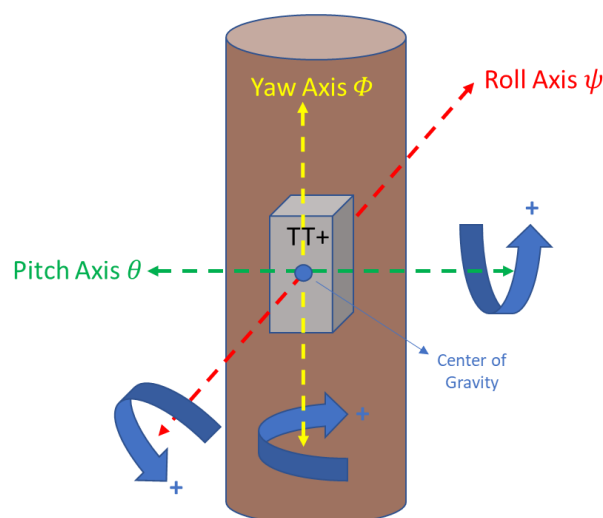


Figura 3-7. Movimientos en los ejes X ( $\theta$ ), Y ( $\psi$ ) y Z ( $\phi$ ).

Según el manual de Nature 4.0, para los TT+ (Asgharinia *et al.*, 2021), se puede aplicar la Ec. 11, Ec. 12 y Ec. 13 para calcular estos mismos tres ángulos de oscilación del árbol.

$$\text{Pitch Angle: } \theta = \tan^{-1} \left( \frac{A_{x.out}}{\sqrt{A_{y.out}^2 + A_{z.out}^2}} \right) \quad [\text{Ec. 11}]$$

$$\text{Roll Angle: } \psi = \tan^{-1} \left( \frac{A_{y.out}}{\sqrt{A_{x.out}^2 + A_{z.out}^2}} \right) \quad [\text{Ec. 12}]$$

$$\text{Yaw Angle: } \Phi = \tan^{-1} \left( \frac{\sqrt{A_{x.out}^2 + A_{y.out}^2}}{A_{z.out}} \right) \quad [\text{Ec. 13}]$$

El ángulo de inclinación  $\theta$ , es una variable de interés en las alertas tempranas para detectar daños por volcamiento del árbol. Conocer el movimiento del árbol por medio de los cambios de los ángulos puede ser aplicable de igual manera para estudios de monitoreo de ecosistemas forestales y su resiliencia a los impactos del viento (James y Hallam, 2013).

Se tomó en cuenta la inclinación con la que se instala el TT+, siguiendo las directrices del manual de instalación (Asgharinia *et al.*, 2021a), cada uno se colocó una inclinación de  $20^\circ$  en dirección opuesta del tronco del árbol con el propósito de reducir el efecto de la sombra del mismo tronco sobre el espectrómetro (Figura 3-8).



Figura 3-8. Disposición del TT+ a  $20^\circ$  del tronco del árbol.



### 3.4.4.7. Radiación electromagnética recibida bajo la copa del árbol

La conversión de la cadena de datos de tipo 49, hace referencia a la cantidad de radiación que pasa a través de la copa del árbol en relación con la radiación fraccional absorbida por sus hojas y la determinación de su frondosidad según su fisiología (Nature 4.0, 2020a). Estos datos son tomados como la radiación correspondiente las 12 bandas espectrales siguientes, según la longitud de onda de: 610, 680, 730, 760, 810, 860 nm ( $\pm 10$  nm) y de 450, 500, 550, 570, 600, 650 nm ( $\pm 20$  nm).

Antes de hacer la conversión se introdujo en el código un filtro, para eliminar los *d.n.* superiores a un valor de 65.000, para eliminar el ruido que se produce en algunas ocasiones en las lecturas, especialmente, cuando la luz se presenta en su mayor intensidad. Aunque de ser necesario (en el caso de obtener la mayoría de los valores por encima de ese umbral), se podría calibrar la sensibilidad del espectrómetro cambiando el tiempo de integración y ganancia (*integration/gain time*) (Asgharinia *et al.*, 2021a).

Para realizar esta transformación, se utilizaron las fórmulas de conversión de la Tabla 6 (Asgharinia *et al.*, 2021a).

Tabla 6. Fórmula de calibración para cada dato según su banda espectral.

Bandas espectrales	Banda (pico de longitud de onda, nm)	Fórmula de calibración
Infrarrojo cercano	AS7263_610 [d.n.]	$-312.45 + (1.6699 * d.n.)$
	AS7263_680 [d.n.]	$-561.56 + (1.5199 * d.n.)$
	AS7263_730 [d.n.]	$-1511.2 + (1.6209 * d.n.)$
	AS7263_760 [d.n.]	$-1012.5 + (1.4549 * d.n.)$
	AS7263_810 [d.n.]	$91.58 + (0.8414 * d.n.)$
	AS7263_860 [d.n.]	$334.88 + (0.531 * d.n.)$
Espectro de luz visible	AS7262_450 [d.n.]	$-212.62 + (0.4562 * d.n.)$
	AS7262_500 [d.n.]	$-232.13 + (0.6257 * d.n.)$
	AS7262_550 [d.n.]	$-842.1 + (1.0546 * d.n.)$
	AS7262_570 [d.n.]	$-666.72 + (1.0462 * d.n.)$
	AS7262_600 [d.n.]	$-328.08 + (0.8654 * d.n.)$
	AS7262_650 [d.n.]	$202.77 + (0.7829 * d.n.)$

Fuente: Asgharinia *et al.*, 2021a.

Con el propósito de mostrar los datos por medio de una gráfica que represente la radiación correspondiente a cada una de las bandas espectrales, solo se utilizaron los datos recogidos entre las 11 am y las 2 pm, que es cuando se presentan los picos máximos de luz durante el día. Se realizó un promedio correspondiente a cada hora, y también de cada día en caso de que se realice un gráfico correspondiente a un periodo más prolongado que el de un solo día.

#### **3.4.4.8. Humedad relativa contenida en el fuste del árbol (Índice de saturación relativo)**

Este es un indicador de la funcionalidad hidráulica del árbol que se relaciona con el contenido de humedad en la madera (Pfausch, 2016). Para estimarlo se necesitan las medidas de la temperatura, niveles de humedad y conocer propiedades físicas de la anatomía de la madera dependiendo de la especie en estudio, debido a esto, sería necesario calibrar una ecuación específica en relación con la especie para estimar esta variable. En el momento de la redacción de este trabajo, la compañía desarrolladora de los TT+, Nature 4.0, aún no tenía estas ecuaciones calibradas para los chopos con ésta última versión (3.3) de los dispositivos TT+ usados en este estudio, con sensores que capturan los datos brutos de la humedad del tronco ( $ECf_{stem}$ ) en una banda de frecuencia de 3 MHz (3.000.000 Hz).

Sin embargo, se puede hacer uso de un índice de humedad relativo a partir de datos de alta frecuencia y largos periodos de tiempo que permitan conocer el comportamiento de la humedad del tronco como se propuso en el estudio de Asgharinia *et al.* (2022). Este índice, considera el valor mínimo reportado por la banda de frecuencia del sensor ( $ECf_{stem}$ ) que muestra el punto de saturación, y disminuye a medida que la temperatura aumenta en la sonda de referencia del tallo ( $T_{ref}$ ); asimismo, su valor máximo se muestra cuando la condición de la madera se encuentra en su estado más seco. Teniendo en cuenta estos aspectos, se requiere el registro de frecuencia mínima y la temperatura del tallo que le corresponde en ese momento, para determinar la frecuencia esperada en el punto de saturación de la fibra en un tiempo determinado; de este modo se calculó el  $ECf_{adj}(Hz)$ , aplicando la Ec. 14.

$$ECf_{adj}(Hz) = m_1 * (T_i + T_{sat}) + ECf_{sat} \quad [Ec. 14]$$

Donde  $m_1$  es la pendiente de sensibilidad a la temperatura, la cual depende de las propiedades físicas del tejido vivo de cada especie de árbol, aunque en este trabajo, se consideró el punto de saturación del agua como  $m_1 = -81 \frac{\Delta ECf}{\Delta T}$ , recomendado por el mismo artículo en mención (Asgharinia *et al.*, 2021a, 2022), mientras que se desconozca esta propiedad para la especie *Populus spp.* Posteriormente, se calculó la saturación relativa de la madera (%) con la Ec. 15, donde el  $ECf (Hz)_i$  corresponde al dato obtenido  $ECf\_stem$  del TT+ en un momento determinado.

$$Relative\ stem\ saturation\ (\%) = \left( 1 - \frac{ECf\ (Hz)_i - ECf_{adj}(Hz)}{ECf_{adj}(Hz)} \right) * 100 \quad [Ec. 15]$$

### 3.4.5. Alerta de TT+

Esta funcionalidad (parte 5 de la Figura 3-6) está incluida en ambos códigos de R con el objetivo de reproducir bases de datos que indiquen alertas por defecto en relación con los datos obtenidos por los TT+ históricamente, aunque con la posibilidad de obtener los resultados en un periodo seleccionado. En este trabajo, las alertas son reportadas para las variables seleccionadas y según los umbrales de la Tabla 7. Estos umbrales podrían modificarse en el código en caso de ser necesario. En un futuro se prevé incorporar alertas en la cantidad de radiación recibida, para detectar procesos de defoliación o pérdida parcial/total de la copa; además, se estudia la posibilidad de relacionar una alerta con el nivel de humedad del fuste, aunque cabe resaltar que es una variable muy con mucha especificidad a la especie misma del árbol.

Tabla 7. Umbrales de datos para las alertas de los TT+.

Variable	Alerta cuando	Archivo de salida
Batería del TT+	< 3,5 mV	BatteryTTAlert.csv
Densidad del flujo de savia	Máximo valor por semana (en época de crecimiento) < 1,0 l dm <sup>-2</sup> h <sup>-1</sup>	SapFDTTAlert.csv
Crecimiento radial del fuste	Máximo valor por semana (en época de crecimiento) < 0,1 mm	GrowthTTAlert.csv

Variable	Alerta cuando	Archivo de salida
Ángulo de volcamiento del árbol	> 45 °	AngleTTAlert.csv

### 3.4.6. Salidas (outputs)

El código desarrollado en este trabajo, como resultado de la ejecución de las funciones de la Figura 3-6, descritas en los apartados anteriores, crea numerosas salidas con la información generada, que se resumen en la Tabla 8. El código crea una carpeta nueva en el disco C del computador llamada “TreeTalkersDB”, donde se almacenan las bases de datos brutos y los resultantes del procesamiento de datos, en formato CSV, gráficas en formato PNG y el reporte generado por *RMarkdown* en formato PDF. En cuanto a las alertas para los dispositivos TT+ y *TTCloud*, cada archivo con su información de alertas correspondiente descrita en la Tabla 8.

Principalmente, se obtienen bases de datos con los resultados de las transformaciones de las variables para la calibración de los TT+ que quedaron almacenados en el archivo *TTDB.csv*; por otro lado, los archivos guardados en la carpeta “TTResults” tienen integrados la fecha que corresponde a la creación de este mismo, esto para que no sea confundido con otros reportes finales si se vuelve a ejecutar los códigos en un día diferente.

Tabla 8. Información de las salidas de los procesos con R.

Nombre del archivo	Contenido
Raw_fullDB.csv	Datos brutos de todos los <i>TTCloud</i> y todos los TT+
TTsn_site.txt*	Datos brutos de cada uno de los <i>TTCloud</i> y sus TT+
Raw_TTDB.csv	Datos brutos de todos los TT+, cadena de datos 4D/55 y 49
Raw_CloudDB.csv*	Datos brutos de todos los <i>TTCloud</i> , cadena de datos 4B y 4C
Cloud.csv*	Datos de todos los <i>TTCloud</i> del día anterior
CloudAlert.csv*	Anomalidad de los datos, CloudID, hora y fecha, sitio
TTDB.csv	Variables transformadas: Ttid, TTsn, fecha y hora, sitio, humedad relativa (%), temperatura del aire (°C), batería (mV), densidad del flujo de savia (l dm <sup>-2</sup> h <sup>-1</sup> ), crecimiento medio semanal del radio del fuste (mm), ángulo del fuste del árbol ( $\theta$ ), saturación relativa de la humedad de la madera (%), 12 bandas espectrales (nm)

Nombre del archivo	Contenido
Gráficas	Humedad relativa (%), temperatura del aire (°C), batería (mV), densidad del flujo de savia (l dm <sup>-2</sup> h <sup>-1</sup> ), crecimiento medio semanal del radio del fuste (mm), saturación relativa de la humedad de la madera (%), promedio de la radiación electromagnética diaria bajo la copa del árbol (nm)
TTAlert.csv	Anomalidad de los datos en: batería de TT+ (mV), densidad del flujo de savia (l dm <sup>-2</sup> h <sup>-1</sup> ), crecimiento medio semanal del radio del fuste (mm), ángulo del fuste del árbol ( $\theta$ )
Reporte Rmarkdown	Resultados del <i>CloudAlert.csv</i> , <i>TTAlert.csv</i> y las gráficas en el reporte <i>TTReport.pdf</i>

\*Solo con descargas desde el servidor.

Los posibles nombres de las variables resultantes de las bases de datos (archivos CSV) y sus correspondientes unidades, están adjuntos en el Anexo 1 de este trabajo.

## 4. RESULTADOS Y DISCUSIÓN

### 4.1. Código para la adquisición y el procesamiento de los datos de los TT+

Se obtuvieron 2 códigos (A y B), desarrollados en el entorno de R, para la adquisición y el procesamiento de los datos de los TT+. El código A para datos desde el servidor (*TT+\_Check\_from\_Server.R*) cuenta con aproximadamente 1.130 líneas de código; y el código B (*TT+\_Check\_from\_TXT.R*), para procesar datos provenientes de las descargas directas de los TT+, cuenta con casi 955 líneas de código. En cada uno de los códigos se incluyen, de forma anotada, cada uno de los pasos y procedimientos realizados, con el fin de que otra persona pueda ejecutar los códigos para su propio sistema de sensores ecofisiológicos si lo desea, cambiando únicamente los parámetros de entrada de cada función. El código y las anotaciones se han desarrollado en inglés, para que su alcance y utilización sean las máximas posibles. Ambos, se encuentran en los ANEXOS de este documento (Anexo 2 y Anexo 3), así mismo, están disponibles de manera libre en el repositorio de GitHub con el siguiente enlace: <https://github.com/Isa-Gris/TreeTalkersCode.git>.

El código A, está configurado para que se pueda realizar una prueba con los TT+ tratados en este trabajo y que están conectados al servidor. Aun así, hay que considerar los parámetros que deben indicar en la entrada de cada una de las partes del código descritos en los dos siguientes apartados (4.1.1 y 4.1.2).

#### **4.1.1. Consideraciones generales para utilizar los códigos *TT\_Check\_from\_Server.R* y *TT\_Check\_from\_TXT.R***

Uno de los inconvenientes más comunes que se pueden presentar al momento de ejecutar un código en R, es no contar con las últimas actualizaciones del software (R y RStudio) y las últimas versiones de los paquetes que son utilizados. Así mismo, es imperativo mantener el código actualizado y realizar revisiones periódicas en este, debido a que las funciones o paquetes de R podrían ser modificadas en cualquier momento, por ejemplo, desde el CRAN (R-project, 2022). De manera que estos dos códigos (*TT\_Check\_from\_Server.R* y *TT\_Check\_from\_TXT.R*), fueron desarrollados con las especificaciones que aparecen al principio de los mismos (ver en ANEXOS), y se recomienda tener esas mismas versiones o superiores su correcto funcionamiento. Por otro lado, se recomienda eliminar la carpeta “TreeTalkersDB” antes de que se ejecute el código otra vez por completo con datos nuevos.

Los paquetes de R utilizados en ambos códigos y que se incluyen con su instalación en las líneas del mismo código, sirven para realizar el filtro de datos, cambios en las fechas, diseño de gráficas, impresión de códigos y resultados, además de la modificación de objetos y entre otras funciones son:

- "signal"
- "lubridate"
- "dplyr"
- "ggplot2"
- "scales"
- "prospectr"
- "reshape2"
- "knitr"
- "rmarkdown"

Adicionalmente, para generar el reporte con *RMarkdown* en un PDF, es necesario tener instalado el software de TeX/LaTeX (versión MikTeX 22.7 usada) en el sistema operativo de Windows o la versión de MacTeX para Mac.

#### **4.1.2. Utilización paso a paso de los códigos: parámetros de entrada en cada uno de los procesos principales**

A pesar de que ambos códigos de por sí parecen extensos, éstos están diseñados para que sean ejecutados por una función en R por cada una de sus partes, quiere decir, en los mismos de la Figura 3-6. Así que el usuario sólo necesita modificar o ejecutar los parámetros de entrada que por defecto que tiene cada procedimiento, las pocas líneas que están al principio de cada paso, y de resto, podrá ver el paso a paso de lo que está haciendo el código, con su explicación detallada al respecto, que a la vez permite la modificación de sus parámetros o fórmulas si fuera necesario. Al ser funciones, estas requieren de la definición de algunos parámetros de entrada, de manera que los resultados estén más acordes con los análisis que se desean. De todos modos, estas tienen definidos algunos de sus campos por

defecto y como ejemplo. Se puede ver en el Anexo 2 y Anexo 3 de referencia de los códigos, que estas funciones están enumeradas como sus 7 partes (*Part*). A continuación, veremos de qué se trata cada una de estas partes como se muestra en la Figura 3-6.

#### 4.1.2.1. Parte 1: Obtención de los datos

El código comienza con la instalación y/o activación de los paquetes de R mencionados. Luego, está la **Parte 1** del código que indica cuál de los 2 códigos se debe de usar, recordar el factor para tener en cuenta según el apartado 3.4.1 (Obtención de los datos de los TT+). De acuerdo con el código A (*TT\_Check\_from\_Server.R*), para la función `DownloadTTDB_A()`, se modifican solo los 3 tipos de parámetros que están de color verde en el ejemplo de la Figura 4-1. Si se quieren modificar o integrar más sensores ecofisiológicos, se escriben en el mismo orden con su enlace de servidor, el identificador que elijamos para el lugar y por último el número de identificación del *TTCloud*. Después, simplemente se ejecuta la función.

```
# (A.1) Write/modify the following 3 inputs (examples):  
  
# Server data links for each Cloud  
server <- c('http://www.naturetalkers.altervista.org/C0210043/ttcloud.txt',  
           'http://www.naturetalkers.altervista.org/C0210044/ttcloud.txt')  
# Give a name to each cloud (site or location name)  
site <- c('VM',  
         'TV')  
# Serial number of each cloud  
cloud <- c('C0210043',  
          'C0210044')
```

Figura 4-1. Ingreso de parámetros para Parte 1 del código A en R.

En el código B (*TT\_Check\_from\_TXT.R*), es necesario eliminar manualmente las primeras 29 líneas de los datos que son directamente descargados de los TT+ en un archivo \*.TXT, o hasta el símbolo ">". Estos datos solo muestran especificaciones del TT+, por lo tanto, no son necesarios. Luego, se debe tener en cuenta que, de acuerdo con el nombre del archivo TXT que va a ser procesado por el código, ese mismo será el nombre que le otorgaremos al TT+ (TTid) en los análisis (ej. *VM\_PIE20.txt*). Ahora, solo nos queda definir en el código (Figura 4-2) modificando las letras verdes, la ruta de la carpeta que almacena los archivos TXT. En esta misma carpeta podemos poner todos los datos de cada TT+ que necesite ser analizado.



Después, correr la función `DownloadTTDB_B()`.

```
# (B.1) Write/modify the input directory, folder with TXT files from TTs  
(example):  
setwd("C:/Users/Isabel/Documents/TFM - Tree Talkers/Códigos de R/DatosTFM")
```

Figura 4-2. Ingreso de parámetros para Parte 1 del código B en R.

Con cualquiera de los dos procedimientos anteriores, según el caso, obtendremos una base de datos con todos los datos brutos compilados por TT+ y por *TTCloud* (si los tiene) en `C:/TreeTalkerDB/Raw_fullDB.csv`.

#### **4.1.2.2. Parte 2: lectura y separación de la base de datos**

En la **Parte 2** de los códigos, según lo indica la Figura 3-6, consta de la lectura y separación de los datos brutos en dos tipos bases de datos: datos de los TT+ (*Raw\_TTDB.csv*) y datos de los *TTCloud* (*Raw\_CloudDB.csv*). El código A contiene una función llamada `ReadServerData()` y el otro código B, una función llamada `ReadTXTData()`. Simplemente se ejecuta la función que le corresponde a cada código. No es necesario modificar nada ni ingresar parámetros en específico. El tipo de código B, lógicamente no genera una base de datos del *TTCloud* porque no tiene esa información. Ambas funciones se encargan de organizar los datos, eliminar duplicados y darles nombres a las columnas de los datos brutos según el tipo de dato (Tabla 3 y Tabla 4).

#### **4.1.2.3. Parte 3: revisión de los TTCloud**

La **Parte 3**, solo está en el código A, ya que revisa el estado de los *TTClouds* y los *TTCloud Alerts* en la misma función llamada `checkCloud()`. Los dos parámetros que se deben de tener en cuenta y que se deben definir según la preferencia del usuario son los que están en la Figura 4-3. Si se desea que se tomen en cuenta todos los *TTClouds* y que se revise el estado del día anterior a la ejecución del reporte, entonces se deben de dejar ambos parámetros por defecto. Sin embargo, estas dos entradas dejan la posibilidad de que se realice un filtro por *TTCloud* en la `dbCloud` si se desea, o en el informe más reciente de todos los *TTCloud*, que incluye sus niveles de batería, estará en el archivo de salida *Cloud.csv*.

```
# Define variables for the function  
dbCloud <- ccloudDB          # Copy Cloud DB  
  
# Date of interest for clouds, by default yesterday. Format: (YYYY-MM-DD)  
whichDay = Sys.Date() - 1  
  
checkCloud <- function(dbCloud, whichDay) {
```

Figura 4-3. Parámetros de la Parte 3 del código A en R.

#### 4.1.2.4. Parte 4: revisión de los TT+

A partir de la **Parte 4** en ambos códigos (A y B) (Figura 3-6), todos los procedimientos son aplicados de la misma manera, solo hay una diferencia casi que imperceptible entre ambos códigos en esta parte que corresponde a los nombres de las variables, pero que son aquí homologadas para que al final de este procedimiento obtengamos los mismos resultados. Se podría decir que esta función `checkTT(dbTT)`, con la variable `dbTT` procedente de la **Parte 2** como la base de datos brutos de los TT+, es la parte más importante del código porque es la responsable de que las transformaciones sí correspondan a los datos reales de las variables que calibramos, según las mediciones de los TT+. Los resultados serán demostrados a mayor profundidad en el siguiente apartado (4.3 Resultados de la aplicación del código desarrollado en choperas). En cuanto al código en R (Figura 4-4), primero se realiza una copia de la salida del proceso de la **Parte 2** por defecto (de `ttDB` a `dbTT`), esto en caso de que se desee hacer un filtrado por TT+ antes del análisis o, si después de este paso se hace algún cambio de alguno de los parámetros internos que hacen parte de la función, con `ttDB` se podría restaurar la base de datos inicial con los datos brutos y volver a correr la función.

```
# Define variables for the function  
dbTT <- ttDB # Copy TT's DB from previous step.  
          # Filter by TT here if it is necessary  
  
checkTT <- function(dbTT) {
```

Figura 4-4. Primeras líneas de la Parte 4 de los códigos A y B.

Ahora bien, como resultado principal, tendremos la base de datos `TTDB.csv` que será almacenada en el computador, pero, por otro lado, como archivos temporales en R tendremos en la consola de RStudio, dos tipos de objetos en el entorno de variables (Figura 4-5).

```
# Return as a variable in the global environment
dbTT <- dbTT # db with some step by step calculations
dbTT_Results <- dbTT_Results # db with the main results
```

Figura 4-5. Resultados de la Parte 4 de los códigos A y B.

El marco de datos (*data frame*) `dbTT_Results`, corresponde los datos almacenados como *TTDB.csv*, que solo muestran los resultados finales obtenidos a partir de la transformación de las variables de *d.n.* a variables con unidades físicas (ver Tabla 8); por otro lado, tendremos el marco de datos `dbTT` con 61 columnas, contiene los datos brutos y la mayoría de las variables intermedias calculadas para obtener la base de datos definitiva `dbTT_Results` (la Figura 4-6 muestra el contenido de sus columnas). Por decir un ejemplo, en el cálculo de la densidad de flujo de savia, cuando se suavizan los datos con el método de Savitzky Golay (Tomelleri *et al*, 2022), se obtienen las temperaturas de las sondas de referencia y calentamiento con el mismo nombre de columna, pero con terminación en “\_filt” que hace referencia a la temperatura “filtrada”. En el Anexo 1 se adjunta una la tabla que contiene todos los nombres de las posibles variables en las bases de datos y su respectivo significado.

```
> colnames(dbTT)
 [1] "TTid"           "TTsn"           "timeStamp"      "Location.x"      "stringType.x"
 [6] "tref_0"         "theat_0"        "growth"          "adc_bandgap"     "nbits"
[11] "relativeHumidity" "temperature"    "gzm"            "gzsd"           "gym"
[16] "gysd"           "gxm"           "gxsd"           "tref_1"         "theat_1"
[21] "stwc"           "adc_vbat"      "Location.y"     "stringType.y"   "B1"
[26] "B2"             "B3"            "B4"             "B5"             "B6"
[31] "B7"             "B8"            "B9"             "B10"            "B11"
[36] "B12"           "intTime"       "gain"           "batteryLevel"   "tref_0filt"
[41] "tref_1filt"    "theat_0filt"   "theat_1filt"    "sapFluxDensity" "DistGrowth"
[46] "YYYYww"        "dendrometer"   "TreeMovement"   "B610"           "B680"
[51] "B730"          "B760"          "B810"           "B860"           "B450"
[56] "B500"          "B550"          "B570"           "B600"           "B650"
[61] "RelSteamSat"
```

Figura 4-6. Captura de pantalla desde RStudio con los nombres de las columnas de `dbTT`.

La **Parte 4** también ofrece unas líneas que por defecto se encuentran desactivadas para generar gráficas temporales (`plot`) en RStudio, con el fin de observar el historial completo de los datos en relación con una variable y así comparar entre tendencias pies, o detectar visualmente anomalías en los datos. Un ejemplo para la temperatura de los datos se puede ver en la Figura 4-7, donde activando el código en la parte superior de esta imagen (borrando los #), los datos son graficados en orden para cada uno de los 5 TT+ que se tenían en funcionamiento en las parcelas de VM y de TV a partir de la fecha de instalación. Se debe recordar que las gráficas definitivas se crean en la Parte 6 de los códigos.

```
# Plot to visualize temperature data
#plot(dbTT$temperature,
#     type = "l",
#     main = "Temperature",
#     xlab = "Data in order",
#     ylab = "Temperature (°C)")
```

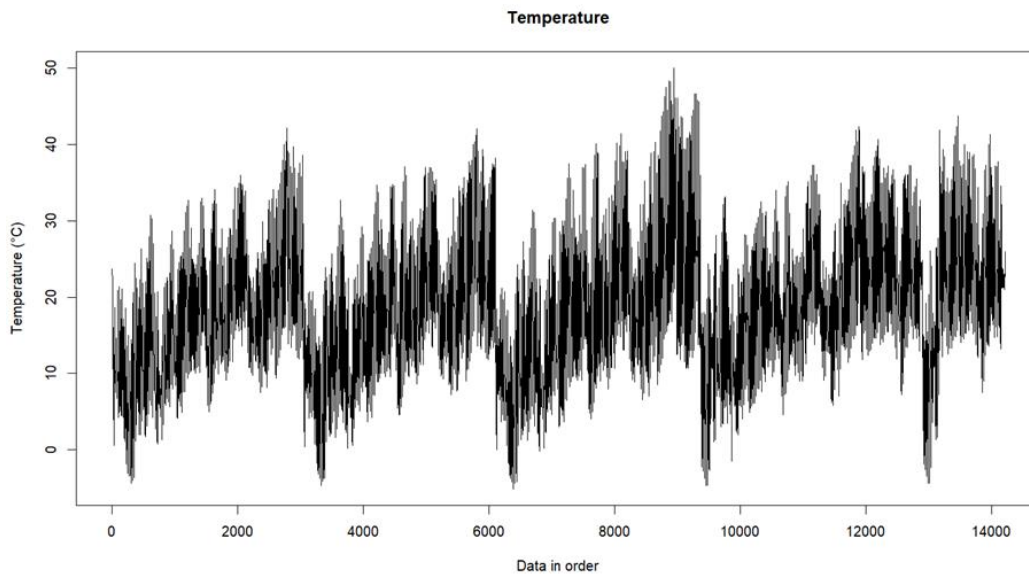


Figura 4-7. Generar gráficas temporales para la revisión de las variables.

#### 4.1.2.5. Parte 5: alertas de los TT+

La **Parte 5** del código tiene como resultado las alertas de los TT+ a partir de los umbrales definidos para los datos de las variables (Tabla 7). La función `TTAlert(dbTT_alert)` por defecto, creará las alertas basándose en el historial completo de los resultados de la calibración de las variables de los TT+ (`dbTT_alert` copia del objeto `dbTT`). Asimismo, en el código brinda las instrucciones para modificar los rangos de las fechas de análisis, ya sea para a un solo día o para un periodo de tiempo de varios días (Figura 4-8).

```
# Copy complete data base history
dbTT_alert <- dbTT
```

Activate one of the next 2 parameters options if data results is desired for one day:

```
#Day_alert <- as.Date("2022-08-10") # Day to check the alerts
#Day_alert <- as.Date(Sys.Date()-1) # yesterday
```

And activate next 2 lines to subset data frame by a date (Day\_alert)

```
#data_alert <- as.Date(dbTT_alert$timeStamp) == Day_alert
#dbTT_alert <- dbTT_alert[data_alert,]
```

Entry dates for next lines if data results is desired for a range of dates, format: "YYYY-MM-DD"

```
#Date0 <- as.Date("2022-04-07") # Start
#Date1 <- as.Date("2022-08-17") # End
#dbTT_alert <- dbTT_alert[dbTT_alert$timeStamp >= Date0 &
dbTT_alert$timeStamp <= Date1,]
```

```
TTAlert <- function(dbTT_alert) {
```

Figura 4-8. Parámetros de entrada para la Parte 5 de los códigos A y B.

Después de definir los parámetros, y ejecutar la función de la **Parte 5**, se consiguió cada uno de los archivos de salida de la Tabla 7, que se guardaron automáticamente en la carpeta *C:\TreeTalkerDB\TTResults*, además, se verán como parte del reporte final en el PDF generado por *RMarkdown*, como *TTRreport.pdf* (ver ejemplo en Anexo 5). Adicionalmente, los resultados se tratarán a profundidad en el apartado 4.4.2.

#### 4.1.2.6. Parte 6: gráficas

La **Parte 6** del código, genera por lo tanto las gráficas finales de las 7 variables descritas en la Tabla 8. Esta parte del código que se ve en la Figura 4-9, para la función `Plot(TTdb_today, Date0, Date1)`, que incluye la copia de la base de datos del resultado del tratamiento de datos de los TT+, también está descrita para que se sigan las instrucciones de acuerdo con la definición de los periodos de tiempo para crear las gráficas, como la fecha de inicio (`Date0`) y la fecha final (`Date1`) del análisis; además, el código permite seleccionar algunos TT+ o separar los ecosistemas si se definen por su número serial o de fábrica (`TTsn`) si se desea. Por defecto va a considerar todos los TT+ de entrada y la fecha que está de color verde en la Figura 4-9 que éstas si se deben de ser modificadas por el usuario.

```
# Define db for the function
```

Graphics for complete data (leave by default)

```
TTdb_today <- dbTT # Copy TT's DB from previous step
```

Filter by one TT here if it is necessary. Examples:

```
#unique(dbTT$TTsn)
#TTdb_today <- dbTT[dbTT$TTsn == "621B0393",]
```

Two or more TTs at the same time

```
#TTdb_today <- dbTT[dbTT$TTsn == "621B0396" | dbTT$TTsn == "621B0397",]
```

```
# Entry dates for graphics, format: "YYYY-MM-DD"
```

```
Date0 <- as.Date("2022-05-01") # Start
```

```
Date1 <- as.Date("2022-05-08") # End
```

plotting is recommended for 1 week or 2 for all variables, disable line that starts with 'scale\_x' (\*\*) on each variable when plotting > 2 weeks. If the plotting fails, be sure that the timestamp from TTs are on time.

```
Plot <- function(TTdb_today, Date0, Date1) {
```

Figura 4-9. Parámetros de entrada para generar las gráficas en la Parte 6 de los códigos A y B.

Debido a que los datos contienen un alto volumen de información, se recomienda realizar gráficas por periodos de tiempos cortos, menos de 2 semanas, o según la necesidad, ya que si se realizan las gráficas con todos los TT+ tomando periodos largos, la gráfica estará muy sobresaturada. En la Figura 4-10 se muestra un ejemplo para un periodo de 1 semana.

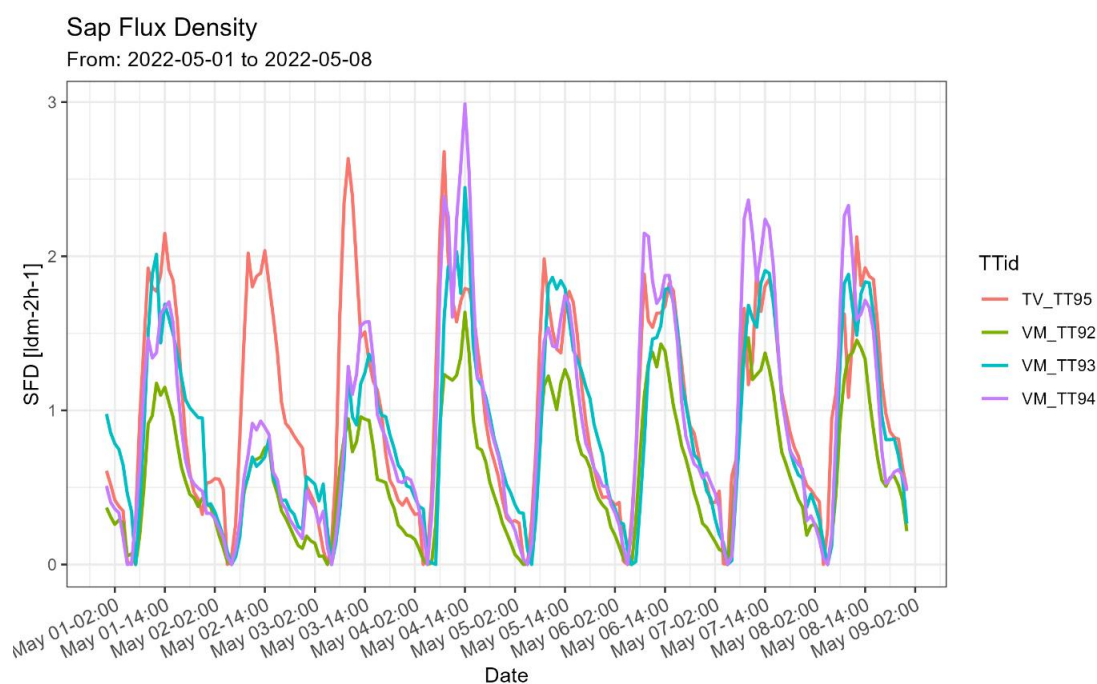


Figura 4-10. Gráfica de la densidad del flujo de savia para una semana con 4 TT+.

Ejemplos de las demás figuras generadas, al igual que los otros resultados estarán en el reporte en PDF (*Report.pdf*) en los ANEXOS. De todos modos, todas las gráficas serán almacenadas en la carpeta con ruta en *C:\TreeTalkerDB\TTPlots*, y cada vez que se cambie la fecha en los parámetros de entrada de la función se creará una carpeta nueva dentro de *TTPlots* con las fechas nuevas y sus gráficas correspondientes, quiere decir que, se puede jugar con los periodos de tiempo o la cantidad de TT+, pero antes de obtener el reporte final en PDF, se deben de dejar sólo las gráficas que sí se desean.

#### **4.1.2.7. Parte 7: reporte con RMarkdown**

Por último, la **Parte 7**, consiste en generar el reporte con todos los resultados de la Tabla 8 con las alertas de los *TTCloud* y TT+; y las gráficas del paso anterior. Este reporte está generado automáticamente como *TTReport.pdf* (ver ejemplo en Anexo 5). Fue diseñado otro código en R, de formato RMD, con *RMarkdown* con el fin de construir este reporte. La escritura del código completo también está incluida en los anexos de este documento (Anexo 4). Éste archivo es llamado *TTReport.RMD*. Es de alta importancia tenerlo junto con los códigos A y B para que estos finalmente generen el reporte.

Para hacer correr el código en la **Parte 7** se tiene que definir la ruta de dónde está en el archivo *TTReport.RMD* que realizará la renderización de los resultados como se muestra en la Figura 4-11. Sólo se debe de modificar la primera parte que está escrita de color verde. Después, ejecutar la siguiente línea que está en el código.

Input the folder path with the "TTReport.Rmd" file downloaded to create the report with RMarkdown

```
render_path <- file.path("C:/Users/Isabel/Downloads", "TTReport.Rmd")
```

Figura 4-11. Parámetro de entrada de la Parte 7 en el código A y B.

De no realizarse este paso, de todos modos, todos los resultados estarán almacenados en el *C:\TreeTalkerDB*, no obstante, ésta última parte es también de gran utilidad para comprobar en un solo documento el estado de nuestros sistemas de sensores ecofisiológicos, revisar las alertas y analizar las tendencias de nuestros datos.

## **4.2. Comparación con otras herramientas existentes**

Una de las mejoras alcanzadas con el código desarrollado en este trabajo con respecto

al desarrollado por de Zorzi *et al.* (2021), es que éste aquí, está adaptado y funciona con la versión más reciente de los TT+ (TT+3.3), en los que la cadena de datos que antes era “4D” en las versiones TT+ 3.2 y ahora es “55” en esta versión (3.3). Otra mejora clave del código desarrollado con este informe, es que descarga procesa y resume en intervalos temporales información de variables del TT+ no analizadas por la versión de Zorzi *et al.* (2021), como son la humedad relativa en el tronco, la estabilidad del árbol y la radiación electromagnética recibida bajo la cubierta para diferentes longitudes de onda. Si bien en ambos trabajos se introduce concepto del uso de TT+ para las alertas en cuanto a anomalías en los datos, Zorzi *et al.* (2021) enfocan las alertas únicamente al correcto funcionamiento de los *TTCloud* y de TT+ (cuestiones que también se han tenido en cuenta en este trabajo), mientras que el código desarrollado en este trabajo de fin de máster (TFM), presenta como novedad y extra la alerta en variables relacionadas con el estado sanitario del árbol, como valores anómalos del flujo de savia o de inclinación del árbol. Además, los códigos desarrollados en este TFM permiten elegir las fechas de revisión de los datos y de análisis (función de filtrado), así como obtener un informe completo de todo el historial desde la instalación de los dispositivos, se supone como una gran ventaja desde el punto de vista objetivo de este estudio de seguimiento del estado sanitario de la chopera. Esto supone mejora respecto al sistema de Zorzi *et al.* (2021), que proporciona la información proveniente sólo desde los recursos de los servidores, y que por ello no se podría emplear si falla la emisión de *TTCloud* o si no se dispone de ella, y sólo hace una evaluación de los datos del día anterior a la ejecución del paquete en R.

En cuanto a las opciones de tratamiento previo y suavizado de los datos registrados, el código de Zorzi *et al.* (2021) no lo permite, mientras que el código de Tomelleri *et al.* (2022) sí lo considera, e incluye una función para cada una de las variables de los TT+, haciendo énfasis en la aplicación de métodos de depuración y suavizado en los datos. El trabajo desarrollado en este TFM profundizó en esa misma línea, aplicando funciones de suavizado y filtrado como Savitzky Golay, y tomando como punto de partida algunas de las metodologías de Tomelleri *et al.* (2022), que fueron adaptadas a los objetivos de este trabajo.

Adicionalmente, el código desarrollado, resume y presenta un informe con alertas para la identificación de condiciones anómalas de los árboles que pueden ser utilizadas con el fin de identificar y analizar en tiempo casi real (dependiendo del tiempo de llegada de la información al servidor) si el individuo está en condiciones poco favorables, ya sea por



factores bióticos o abióticos, y que pueden ser redimensionadas a escala de rodal. Sin olvidar que, a partir de estas mismas alertas, se pueden detectar posibles fallas en el buen funcionamiento de los sensores o en sus baterías y que merezca la revisión de los dispositivos, ya que quizá sea causado por alguna interferencia externa.

Además, los dos códigos resultado de este trabajo brindan la posibilidad de procesar datos con TT+ que no tengan un *TTCloud* disponible (como en el caso de la parcela de Villasabariego) y que, por lo tanto, no se pueda acceder a los datos de manera remota y se requiera de la visita en campo para descargar la información directamente desde los TT+. Esto también puede ser muy útil en situaciones donde el *TTCloud* esté fallando en la recepción y transmisión de datos (como ocurre en este caso con la parcela de Villamañán), para poder emplear los datos para el sistema de detección y seguimiento de daños. Estas dos posibilidades no habían sido abordadas ni están operativas en los códigos desarrollados por Zorzi *et al.* (2021) ni Tomelleri *et al.* (2022).

El mayor inconveniente que puede surgir con los datos de los TT+ que no están conectados a un *TTCloud* es que no tengan el *Timestamp correcto al momento de la toma de los datos presentando* errores en la fecha y la hora, siendo reiniciadas de un momento a otro. Este fue un conveniente que se presentó con los dos TT+ de Villasabariego. Para solventar este problema, se consultó la hora y fecha de la instalación de los dispositivos, se asignaron las fechas corregidas con el mismo sistema UNIX usando RStudio y después, se comprobó que estuvieran los datos sincronizados al momento de la descarga. De manera más práctica, se recomienda que todos los ecosistemas de sensores tengan un *TTCloud* asignado, esto hace más fácil el tratamiento de los datos, más preciso y eficiente, ya que estarían disponibles en cualquier momento sin necesidad de ir a campo desde el servidor. De todos modos, el *Timestamp* de los TT+ que estén conectados al *TTCloud* pueden sufrir de la pérdida de sincronización en el momento en que se cambien las baterías o pierda su carga por completo; aunque por medio del IoT se puede calibrar el *Timestamp* siguiendo las instrucciones del manual (Nature 4.0, 2020a; Nature 4.0, 2020b).

#### **4.3. Resultados de la aplicación del código desarrollado en choperas (calibración)**

Debido a que los sensores ecofisiológicos, y en específico los dispositivos TT+ son una

tecnología desarrollada recientemente, estos están permanentemente sujetos a constantes actualizaciones. En consecuencia, también se deben de considerar los cambios en relación con las variables y los parámetros de calibración para transformar los datos registrados por los sensores en información (por ejemplo, el cálculo de humedad relativa del tronco), lo que hace prever que los códigos en R requieran actualizaciones y adaptaciones a las especies y ecosistemas en los que se empleen. Es de destacar que no existe un conjunto de parámetros y de valores estándar (*set de calibración*) para el cálculo de todas las variables derivadas, y que en este trabajo se han empleado como punto de partida los de los manuales de los TT+, los empleados en otras publicaciones y los transmitidos por el personal de Nature 4.0. y de IDAF Consulting según su experiencia con otras especies forestales.

Las gráficas y análisis descritos en esta sección son las resultantes de aplicar la metodología y los parámetros (*set de calibración*) definidos el apartado 3.4.4 (Revisión de los TT+), proceso incluido en la **Parte 4** del código en R con mucho más detalle (Anexo 3), resultados que son posteriormente graficados en la **Parte 6** de los mismos códigos (Figura 3-6). Aunque no se presentarán los análisis de cada TT+ y cada gráfica con los datos completos, porque toda la información recogida es muy voluminosa, se mostrarán a continuación los resultados más relevantes a partir de la aplicación de los códigos en R y su posible aplicabilidad en el sistema de alerta temprana para plantaciones de chopo.

#### **4.3.1. Voltaje de las baterías de los TT+**

Conocer el voltaje de las baterías de cada TT+ es esencial para asegurarse de un adecuado funcionamiento de estos, para esto, se debe tener en cuenta que las baterías deben ser recargadas antes de que lleguen al umbral de 3.500 mV como indica el manual (Asgharinia *et al.*, 2021, Nature 4.0, 2020a). Para cada TT+ se calculó su nivel de batería correspondiente y a su vez se graficaron los resultados para ver su rendimiento a lo largo del tiempo (Figura 4-12). En este caso, se puede ver que las baterías redijeron su carga significativamente después de estar cerca del umbral de 3,75 mV. Por otro lado, se cree que las baterías también fueron afectadas por una ola de calor en verano que ocurrió al mismo tiempo en que la carga descendió de manera abrupta, las temperaturas superaron los 40 °C en la primera quincena del mes de julio (Figura 4-14), que pudo haber generado un sobrecalentamiento de estas y, por tanto, su descarga repentina. Ahora con la alerta temprana del TTAAlert, se espera conocer

esta situación a tiempo y actuar más oportunamente.

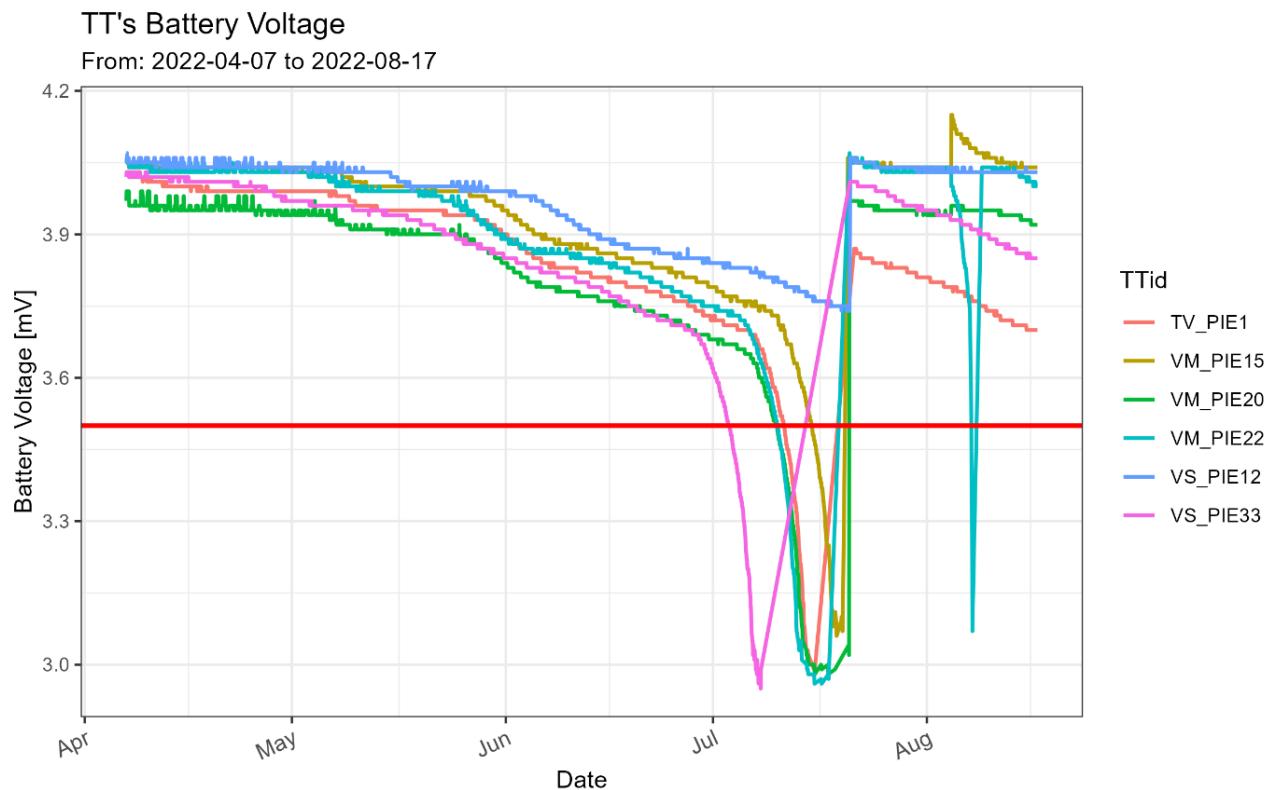


Figura 4-12. Historial del estado de las baterías de los TT+ en mV.

Como consecuencia de la descarga de las baterías, los datos no fueron capturados durante este tiempo por los TT+. Así que, se puede encontrar un vacío en los datos o un salto en las gráficas en este periodo de tiempo que es comprendido entre el 13 y el 20 de julio para la mayoría de los dispositivos.

#### 4.3.2. Temperatura y humedad relativa del aire

Primero debemos de considerar que ambas variables presentan fluctuaciones a lo largo de un solo día, y que éstas, también dependen de fenómenos climáticos que ocurran y de variaciones según la estación del año. Por otro lado, en función de los datos, se pueden describir situaciones que evidentemente pueden perjudicar el crecimiento y desarrollo de los árboles en relación con estas variables. Por ejemplo, durante este estudio en el mes de julio se registró una ola de calor en España como se mencionó anteriormente, referenciado por el AEMET (Periódico de España, 2022) como el julio más cálido desde el año 1961, con una temperatura media de 25,6 °C. Esto provocó una evapotranspiración alta y por ende un estrés hídrico en los árboles; de este mismo modo, una reducción en la humedad relativa del aire

debido a los calores extremos y prolongados durante la primera tercera parte del mes. Adicionalmente, veremos más adelante cómo esta variable se relaciona con otras en este mismo estudio.

En una sola parcela, como era de esperarse, las condiciones de temperatura y humedad registradas por los TT+ son muy similares entre sí. En la Figura 4-13 se muestra un ejemplo de ello en temperaturas muy altas registradas en el mes de julio. Pueden variar un poco dependiendo del efecto borde en las plantaciones o de la exposición al sol del TT+ en el árbol. Sin embargo, estas temperaturas serán confrontadas más adelante en este mismo documento, en un análisis de correlación con las temperaturas del sensor HOBO.

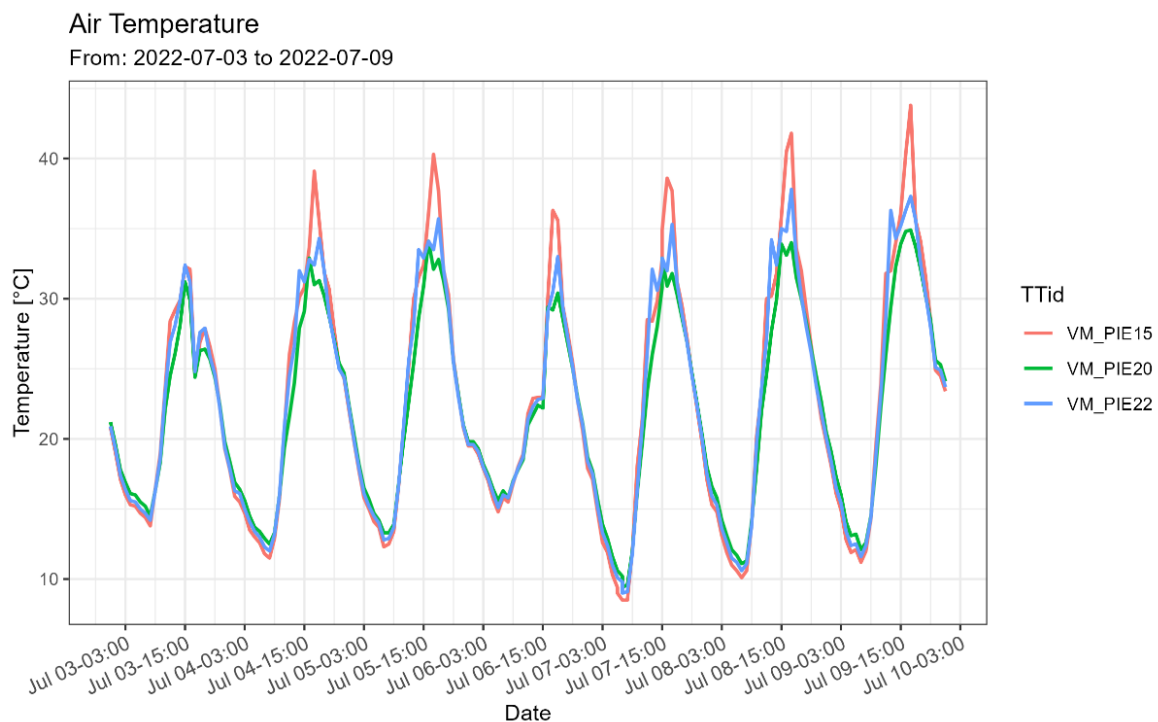


Figura 4-13. Temperaturas registradas por los TT+ de Villamañán durante una semana en julio, 2022.

Por consiguiente, podemos considerar un solo TT+ para describir el historial de datos de una parcela en cuanto a su temperatura del aire (Figura 4-14) y la humedad relativa del aire (Figura 4-15), en ambas gráficas se puede ver un salto entre los datos que no fueron tomados al bajar el nivel de batería. Estos datos son muy relevantes por su relación con la aparición y evolución de plagas y enfermedades, y se pueden emplear durante el seguimiento de otras enfermedades a escala de rodal/parcela como las royas (*Melapsora* spp.) o plagas como *Trypophloeus* spp.

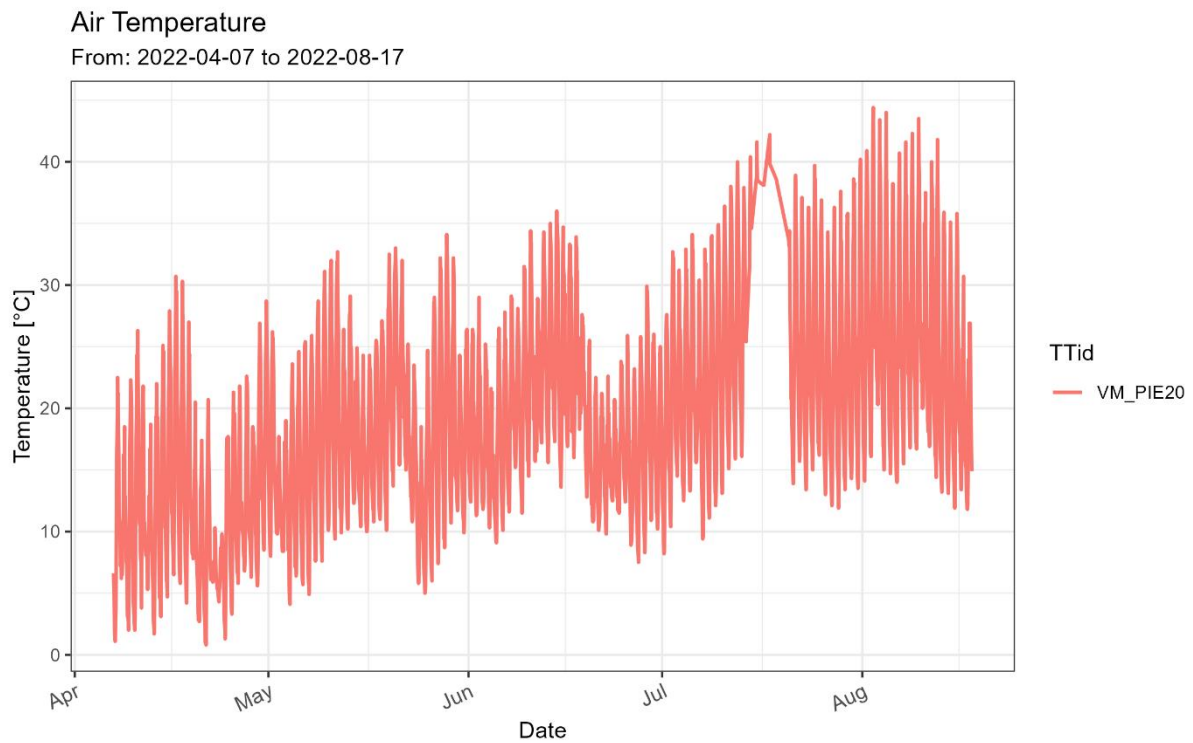


Figura 4-14. Temperatura del aire (°C) en Villamañán a partir del VM\_PIE20.

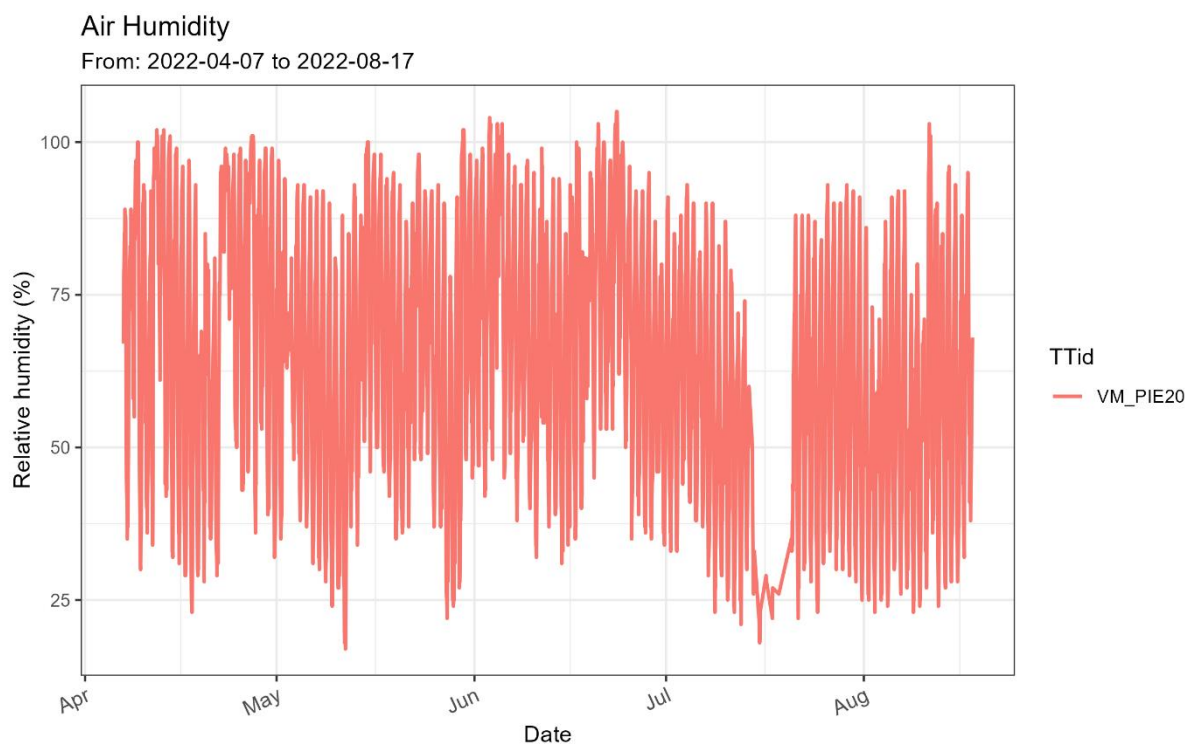


Figura 4-15. Humedad relativa del aire (%) en Villamañán a partir del VM\_PIE20.

Adicionalmente, se calculó un modelo de regresión lineal simple para calibrar la

temperatura de los TT+ y comprobar la correlación entre la temperatura promedio de los TT+ por sitio, y la temperatura tomada por el dispositivo HOBO a cada hora igualmente.

En Villamañán, la ecuación de la regresión fue estadísticamente significativa (Ec. 16), con un valor de  $R^2$  ajustado = 0,83 con un valor  $p < 0,001$ , lo que indica que el 83% de la temperatura de los TT+ puede ser explicada por la temperatura del HOBO (Figura 4-16). También se obtuvo un coeficiente de correlación de Pearson entre los dos métodos de medición, positivo y muy alto del 0,91 (valor  $p < 0,001$ ), con 694 grados de libertad. La ecuación de la regresión fue:

$$\text{Temperatura del TT en VM } (^{\circ}\text{C}) = -0,74814 + 1,2027 * (\text{Temperatura del HOBO}, ^{\circ}\text{C}) \text{ Ec. 16}$$

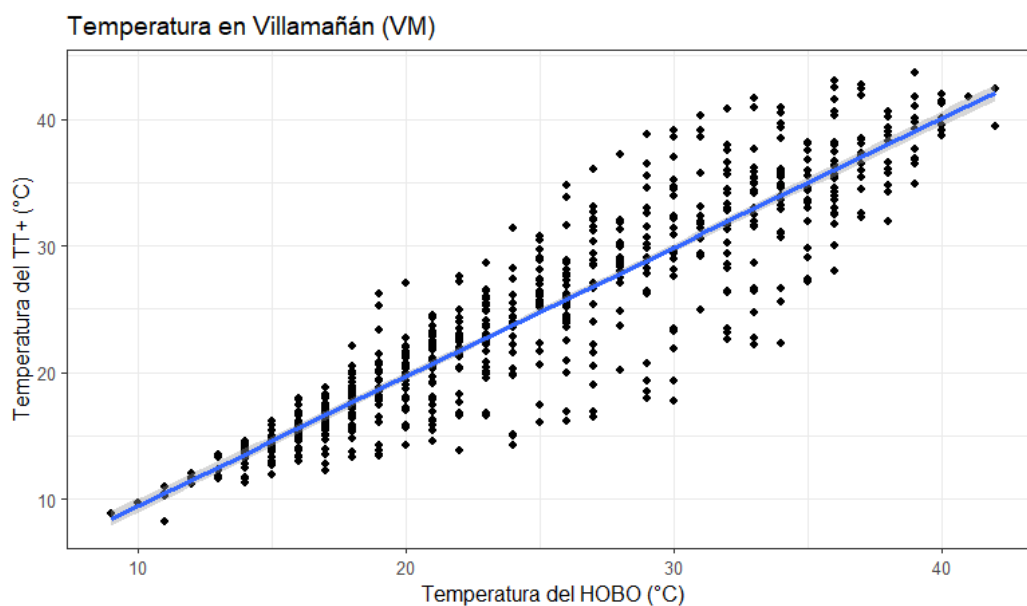


Figura 4-16. Gráfico de dispersión de la temperatura de los TT+ vs. temperatura del HOBO en Villamañán.

En Villasabariego, la ecuación de la regresión fue estadísticamente significativa (Ec. 17), con un valor de  $R^2$  ajustado = 0,88 con un valor  $p < 0,001$ , lo que indica que el 88% de la temperatura de los TT+ puede ser explicada por la temperatura del HOBO (Figura 4-17). También se obtuvo un coeficiente de correlación de Pearson entre los dos métodos de medición, positivo y muy alto del 0,94 (valor  $p < 0,001$ ), con 688 grados de libertad. La ecuación de la regresión fue:

$$\text{Temperatura del TT en VS } (^{\circ}\text{C}) = 1,49461 + 0,95839 * (\text{Temperatura del HOBO}, ^{\circ}\text{C}) \text{ Ec. 17}$$

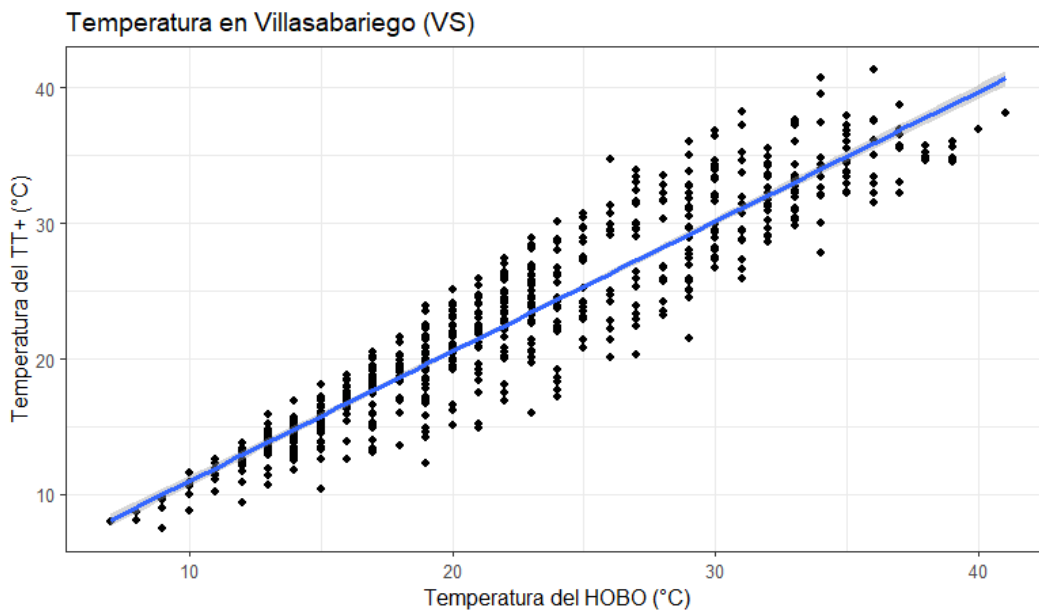


Figura 4-17. Gráfico de dispersión de la temperatura de los TT+ vs. temperatura del HOBO en Villasabariego.

En ambos casos se observa que hay ciertas discrepancias entre los datos tomados por los dos sensores y que es necesario continuar con su análisis en un periodo de tiempo más extenso para extraer conclusiones sobre los límites de su funcionamiento, especialmente en el caso de temperaturas elevadas.

#### 4.3.3. *Densidad del flujo de savia*

Investigaciones dirigidas a cuantificar el uso de agua y la salud de las plantas en términos de crecimiento pueden responderse mediante el uso de dendrómetros y sensores de flujo de savia (Mencuccini *et al.*, 2017). De este modo, se deben de entender las fluctuaciones diarias del flujo de savia y también identificar las tendencias “normales” que demuestren el desarrollo del árbol a lo largo de las estaciones del año; las desviaciones entre los valores esperados (“normales”) y los registrados es la base para detectar un problema fisiológico o un daño de forma temprana (Figura 4-18 y Figura 4-19).

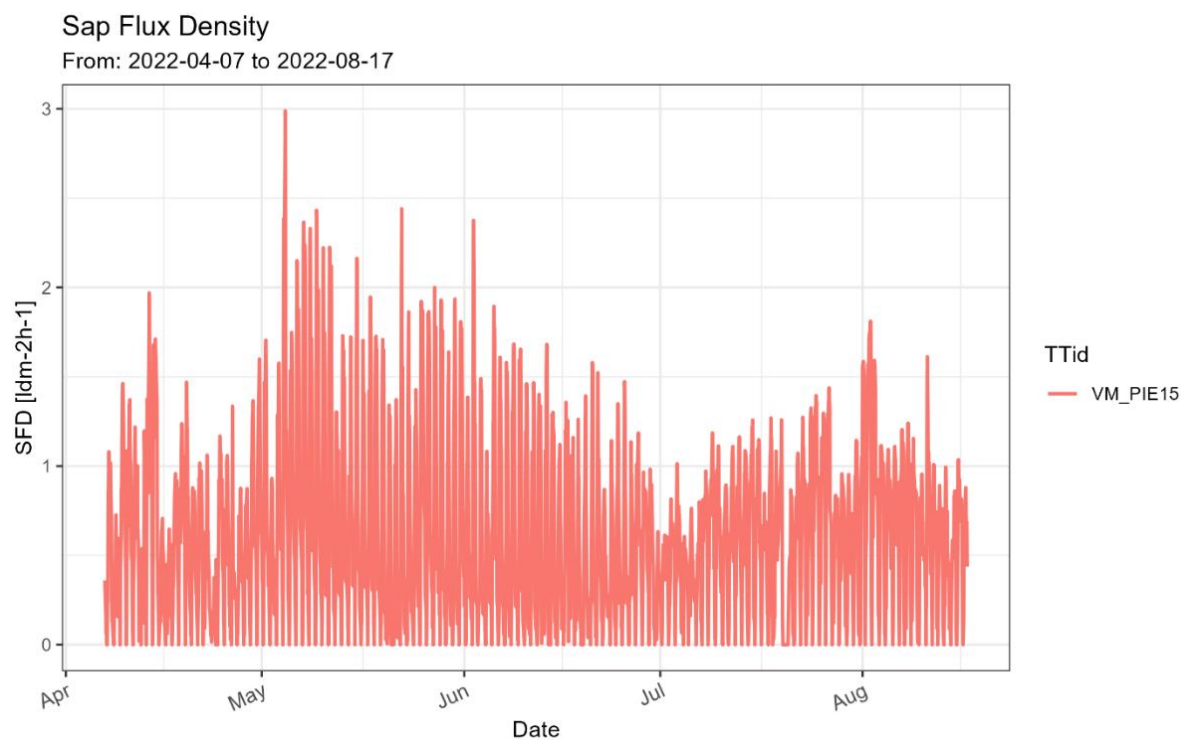


Figura 4-18. Densidad del flujo de savia para el VM\_PIE15 fuertemente afectado por estrés hídrico entre junio y julio (Figura 4-19).



Figura 4-19. Aspecto del pie VM\_PIE15 el 14 de julio de 2022, con amarilleamiento y pérdida de hojas por



estrés hídrico.

Con el fin de cuantificar esta variable de la manera más precisa, la mayor dificultad está en distinguir cuándo ocurre un bajo flujo de savia o nulo, así que, se recomienda considerar las medidas durante la noche bajo ciertas condiciones que afecten esta medida a tal punto que se considere como un flujo en cero, sobre un tronco de madera de árbol cortado (Becker, 1997). No obstante, la máxima diferencia entre las dos temperaturas obtenidas de las sondas de referencia y calentamiento, con mediciones de alta frecuencia, también nos puede indicar un flujo de savia en cero, aplicado en el índice de flujo K (Ec. 8 y Ec. 9), obteniendo así, los datos de densidad de flujo de savia por árbol normalizados.

Aunque la aplicación de este método, considerando los deltas máximos se menciona por varios autores en sus estudios (Asgharinia *et al.*, 2022 y 2021; Zorzi *et al.*; Granier, 1987), no era muy claro el periodo de tiempo que debía de ser considerado para conseguir este valor máximo de diferencias de temperaturas para calcular el índice de flujo, sin embargo, con la colaboración del equipo técnico de Nature 4.0, se definió éste como la máxima diferencia de temperaturas de las sondas de referencia y calentamiento dentro de un periodo de 24 horas. Así tenemos como resultado, las fluctuaciones diarias del flujo de savia indicando que este es cero en horas de la madrugada, y en contraste, el mayor flujo de savia con los valores máximos o picos, durante la época de verano se presentaron entre las horas de las 10 am y las 4 pm, en concordancia con el momento en que debería de ocurrir su mayor producción fotosintética (Figura 4-20). Debido a esto, se pueden aplicar métodos de flujo de savia para analizar patrones estacionales diurnos de transpiración y conductancia estomática, también, para cuantificar sus respuestas a la variabilidad hidroclimática (Phillips y Oren, 1998).

En esta imagen (Figura 4-20) se puede además observar que el intervalo de las horas centrales del día es el más adecuado para identificar valores anómalos: los valores de densidad del flujo de savia del pie 15 fueron menores que los del pie 22, lo que se vio reflejado en las mediciones semanales de diámetro. Posteriormente el pie 15 detuvo el crecimiento, amarillaron las hojas, hasta la pérdida total de las mismas en julio (Figura 4-19).

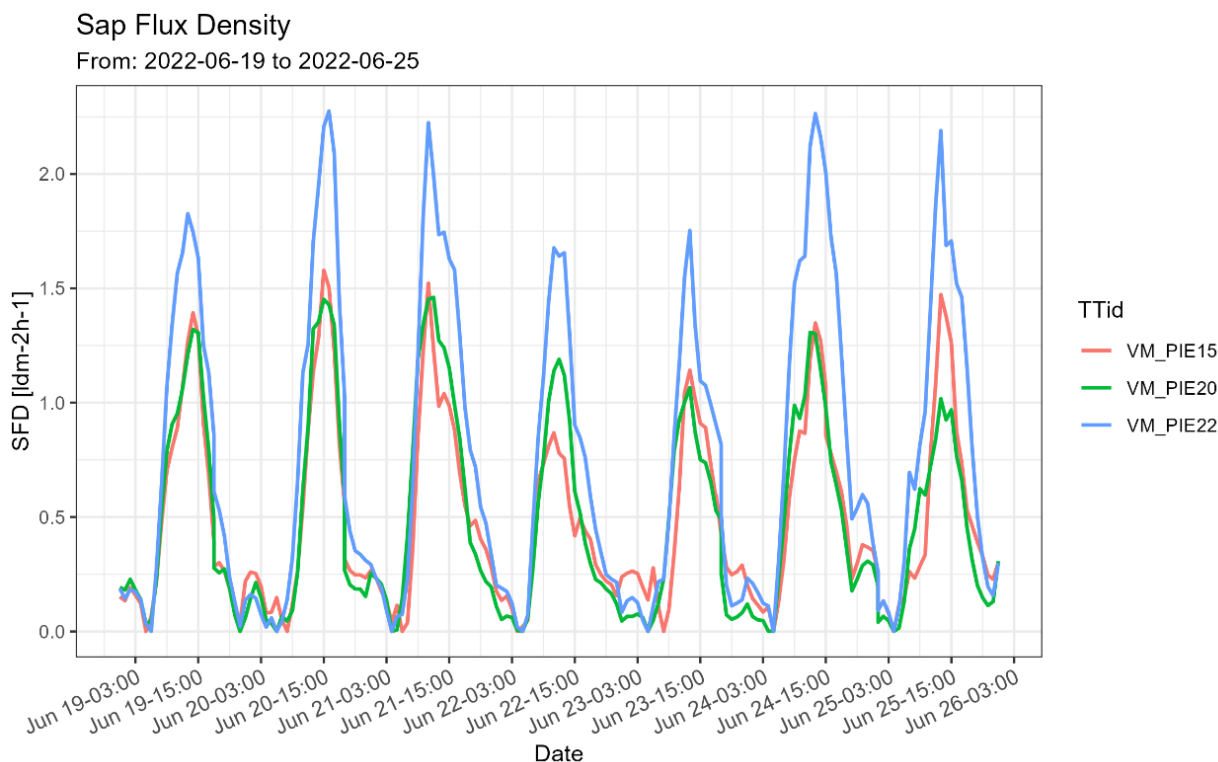


Figura 4-20. Densidad de flujo de savia durante un periodo de una semana en la parcela de VM.

El individuo VS\_PIE12 no se tuvo en cuenta para el análisis de esta variable, debido a que presentó fallos en las sondas de temperatura de referencia y de calentamiento, con lecturas anómalas, temperaturas excesivamente altas que fueron detectadas y luego se realizó la comprobación en campo de que los sensores estaban deteriorados, por lo tanto, no se tuvo en cuenta este parámetro para esta variable y tampoco para la humedad relativa contenida en el fuste del árbol.

#### 4.3.4. Crecimiento medio semanal del radio del fuste del árbol

La lectura de datos por los sensores en relación con el crecimiento radial del fuste a la altura normal (1,30 m) presenta mucha variabilidad durante el día y en cada hora que la medida es capturada. La constante expansión y contracción en el diámetro del fuste diariamente, refleja el movimiento del agua debido a la transpiración durante el día y la reposición de los tejidos del tallo durante la noche (Zweifel *et al.*, 2021; Vieira *et al.*, 2022); especialmente en la época de verano, donde se ha demostrado en otras especies de árboles que pueden cambiar la amplitud de los ciclos diarios hasta 10 veces más que en otras estaciones debido a las condiciones ambientales (Vieira *et al.*, 2022). En consecuencia, la medida que es tomada por los TT+ como la distancia del sensor a la superficie de la corteza

del árbol, no debe ser considerada como un valor absoluto cada medida por cada hora, sino que se debe llevar a términos del valor del crecimiento semanal a partir de su media (mediana), así como sugiere Valentini *et al.* (2019), Zorzi *et al.* (2021), Tomelleri *et al.* (2022), entre otros.

Debido a esa misma variabilidad, se pueden encontrar errores donde el crecimiento aparentemente disminuye, porque al normalizar los datos tomando como punto de partida la máxima distancia registrada entre el sensor y la corteza del árbol, puede que esta medida no haya sido registrada el día en que comenzaron las mediciones. Por esto mismo, se sugiere tomar esta medida manualmente en campo el día de la instalación de los TT+ con un instrumento como un calibre o micrómetro ( $\pm 0.01$  mm), y que las barras de carbono que sostienen el sensor queden bien incrustadas o ajustadas en el xilema del árbol para que tome una medida más precisa. Así se sugiere que se eliminaría el error del ajuste para la normalizar los datos. Sin embargo, al monitorear la variación reversible en el diámetro del tallo, es posible inferir el balance hídrico del árbol, obteniendo información sobre el funcionamiento fisiológico del árbol (Vieira *et al.*, 2022).

Además, también se observó mucho ruido en los datos, con demasiada variabilidad en poco tiempo, en muchas ocasiones de más de 15 mm, con la metodología aplicada, lo que quiere decir, que posiblemente se deberían de aplicar métodos de filtrado más estrictos. De las opciones que hay para generar gráficas con la función `Plot()` en el código en R, se consiguió la Figura 4-21 que muestra precisamente esta variabilidad diaria durante el periodo de estudio completo y se ve cómo se va reduciendo la distancia de interés. De todos modos, se ha encontrado en otros estudios (Vieira *et al.*, 2022) que la amplitud de los ciclos diarios de variación del tallo es mayor en verano, así como la duración de la fase de expansión, lo que significa que los árboles necesitan más tiempo para reponer el agua perdida por la transpiración durante el día, sin embargo, a medida que avanza la temporada de crecimiento y el ambiente se vuelve más cálido, el ciclo diario ajusta sus tiempos y amplitud a las condiciones ambientales del momento.

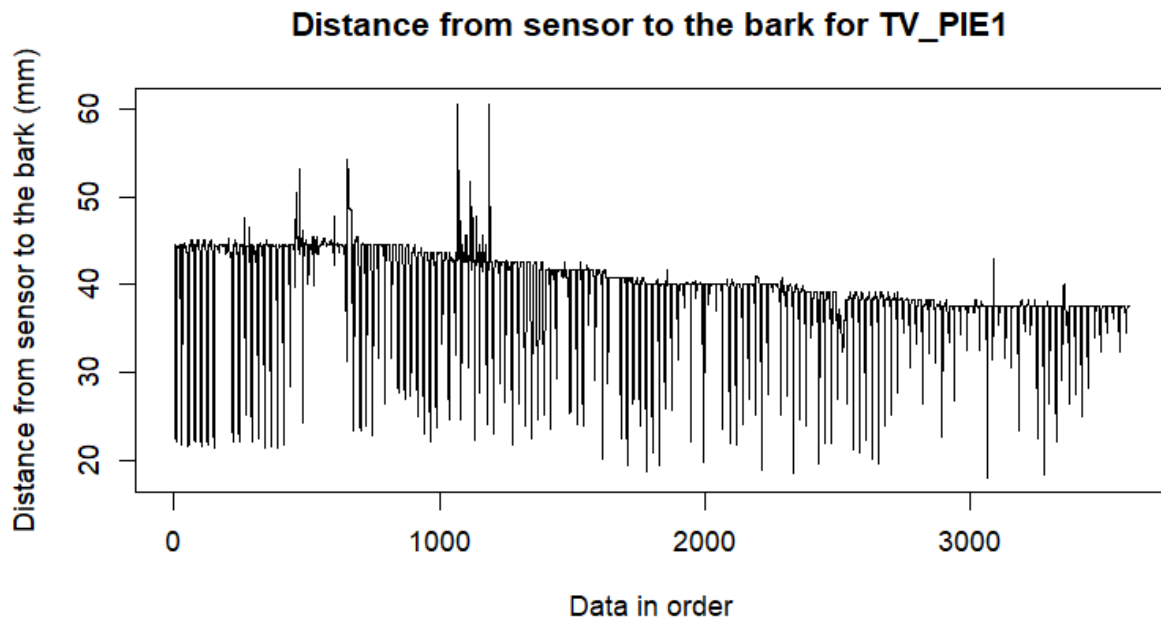


Figura 4-21. Distancia entre el sensor de crecimiento radial con la superficie de la corteza del pie 15 (TV\_PIE1), de acuerdo al orden de llegada de los datos durante el periodo de estudio.

Gracias a la toma de datos que se llevó a cabo en campo semanalmente con cintas dendrométricas, se pudo comprobar que los pies 1, 12 y 22 (Figura 4-22) presentaron un ajuste más cercano a las medias tomadas por los TT+. Vemos cómo coincide el periodo que tuvo el pie 12 donde dejó de crecer virtualmente desde mediados de junio. Aunque, los otros dos pies (20 y 22), muestran un gran salto a mediados de julio en la Figura 4-22, esto se debe a los datos que no se tomaron cuando las baterías estaban descargadas. Por otro lado, esta misma imagen, también demuestra que el TV\_PIE1 tiene una mayor tasa de crecimiento comparado con los demás árboles, siendo ésta de  $0,92 \pm 0,82$  mm/semana, aunque su crecimiento haya comenzado un poco tardío en relación con los otros.

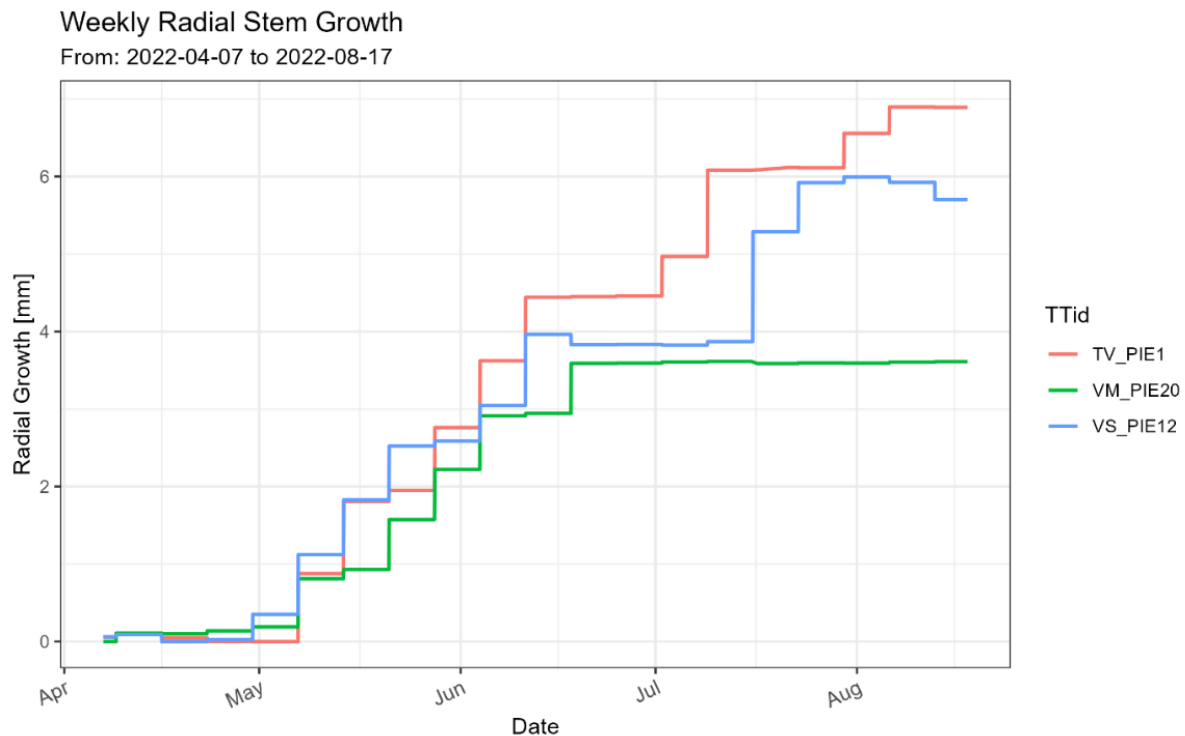


Figura 4-22. Crecimiento semanal del radio del fuste entre pies de diferentes parcelas.

En cuanto al crecimiento en diámetro que se registró en campo semanalmente con las cintas dendrométricas, se realizó una comparación de sus medidas con las medidas del TT+ que fue transformada de radio a diámetro para hacer posible su comparación, entre dos pies, uno de VS (PIE12) y otro de VM (PIE20).

A partir de las medidas del VS\_PIE12 (TT96), se obtuvo una correlación de Pearson positiva, de moderada a alta de 0,53 (valor  $p < 0,05$ ), con 13 grados de libertad. Calculando un modelo de regresión lineal simple (Ec. 18), se obtuvo un  $R^2$  ajustado = 0,22 con un valor  $p < 0,05$  estadísticamente significativo, lo que indica que solo el 22% del crecimiento por los sensores del TT+ puede ser explicado por las mediciones de las cintas dendrométricas (Figura 4-23). La ecuación de la regresión fue:

$$\text{Crecimiento por TT + en PIE12 (mm/semana)} = -0,2009 + 11,2288 * (\text{Crecimiento por cintas dendrométricas mm/semana}) \text{ Ec. 18}$$

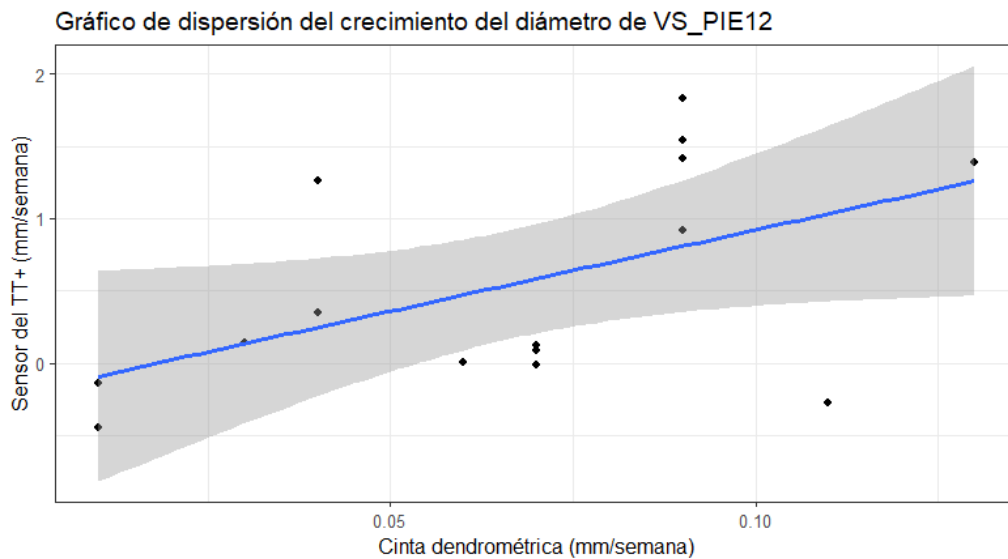


Figura 4-23. Gráfico de dispersión del crecimiento del diámetro semanal entre el TT+ y la cinta diamétrica, en el VS\_PIE12.

En el caso del VM\_PIE20 (TT92), se obtuvo una correlación de Pearson positiva alta de 0,69 (valor  $p < 0,01$ ), con 14 grados de libertad. Calculando un modelo de regresión lineal simple (Ec. 19), se obtuvo un  $R^2$  ajustado de = 0,44 con un valor  $p < 0,01$  estadísticamente significativo, lo que indica que el 44% del crecimiento semanal por los sensores del TT+ puede ser explicado por las mediciones de las cintas dendrométricas (Figura 4-24). La ecuación de la regresión fue:

$$\text{Crecimiento por TT + en PIE20 (mm/semana)} = 0,07442 + 7,49514 * (\text{Crecimiento por cintas dendrométricas mm/semana}) \text{ Ec. 19}$$

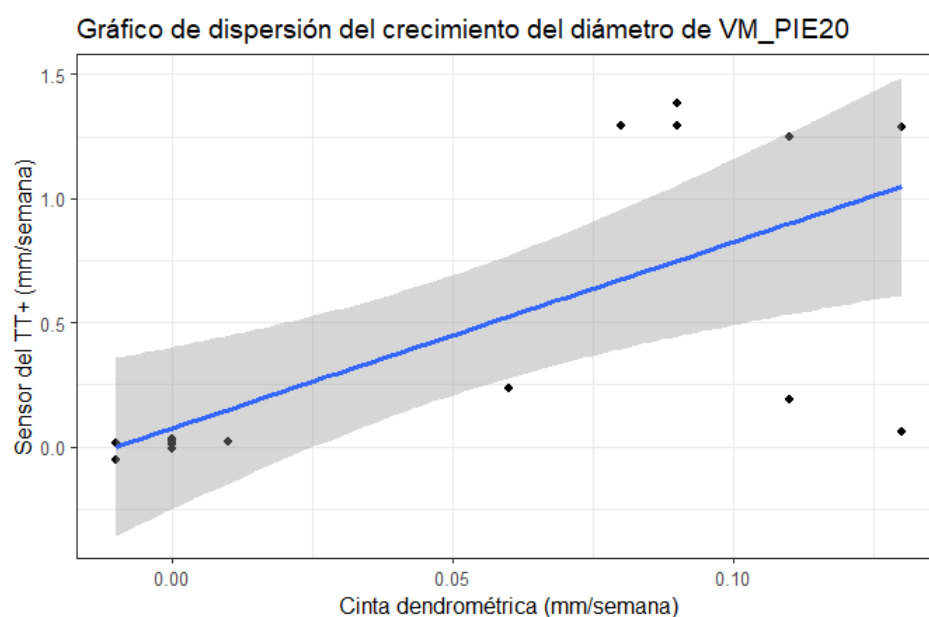


Figura 4-24. Gráfico de dispersión del crecimiento del diámetro semanal entre el TT+ y la cinta diamétrica, en

el VM\_PIE20.

En relación con las medidas tomadas con las cintas dendrométricas y el sensor de crecimiento de los TT+, se debe trabajar más en la calibración de estos, ya sea en la conversión de los números digitales a unidades mensurables o, en una forma de consolidación de los datos para filtrar y suavizar los datos, de manera en que se reduzca la variabilidad de esta medida y se obtengan datos más realistas y uniformes para el crecimiento radial del fuste.

Drew y Downes (2009) definen algunos de los factores que necesitan ser considerados en este tipo de dendrómetros para asegurarse de que los datos sean lo más precisos posibles:

(i) los datos medidos con dendrómetros automáticos deben verificarse o corregirse para detectar ajustes y varios posibles errores. Es común que se encuentren estos datos donde se “encoge” ocasionalmente el fuste, además de los factores de interferencia que son causados por insectos.

(ii) es importante asegurarse de que la alineación entre el dendrómetro y de la madera sean las correctas, porque incluir en el diámetro total el grosor de la corteza y el floema puede reducir la precisión. Además, es posible que el grosor de la corteza disminuya en ciertos momentos, lo que altera aún más las medidas.

Adicionalmente, los datos recogidos por un dendrómetro deberían de incluir de un tratamiento extra que incluye correcciones y calibraciones que generalmente está vinculado temporalmente con una variedad de otros tipos de datos (por ejemplo, clima, flujo de savia, agua del suelo) (Drew y Downes, 2009).

#### ***4.3.5. Estabilidad del árbol (Ángulo de oscilación)***

Esta variable se espera que sea de gran utilidad, ya que la detección de una inclinación progresiva de los árboles o una desviación de la respuesta normal bajo la carga del viento tiene un gran potencial de aplicación en el contexto de un sistema de alerta temprana para árboles en riesgo o con posibilidad de sufrir daños mecánicos (James y Hallam, 2013).

El cálculo del ángulo de inclinación  $\theta$ , que describe el ángulo entre el horizonte Y el eje X del acelerómetro, es suficiente para detentar el movimiento o inclinación de un árbol que se dirige al suelo. Así que no se realizó una gráfica para describir esta posición del árbol en este estudio, ya que no representaría un mayor cambio, sin embargo, si se quiere revisar por

alguna razón el ángulo de la posición del acelerómetro del TT+, se tiene el archivo de salida *TTDB.csv*, los resultados de la columna “TreeMovement” con unidades en grados. Por ahora solo se consideró esta variable en las alertas del TT+, con un informe que indique el volcamiento total o parcial del árbol. Por ejemplo, en la Figura 4-25 se muestra la columna que corresponde al ángulo de inclinación del pie 1, el primer dato señalado en rojo indica el momento en que se hizo un ensayo antes de la instalación en campo del TT+ con una troza de madera recostada en la superficie de una mesa; este dato se presenta en las alertas que provienen de la variable si se toma en cuenta todo el historial que tiene almacenado el TT+.

	A	B	C	D	J
1	TTid	TTsn	timeStamp	Location	TreeMovement
10371	TV_PIE1	621B0395	3/30/2022 20:00	TV_PIE1.txt	88.99319
10372	TV_PIE1	621B0395	3/31/2022 14:00	TV_PIE1.txt	14.35761
10373	TV_PIE1	621B0395	3/31/2022 15:00	TV_PIE1.txt	14.33424
10374	TV_PIE1	621B0395	3/31/2022 16:00	TV_PIE1.txt	14.33785
10375	TV_PIE1	621B0395	3/31/2022 17:00	TV_PIE1.txt	14.34798
10376	TV_PIE1	621B0395	3/31/2022 18:00	TV_PIE1.txt	14.44041
10377	TV_PIE1	621B0395	3/31/2022 19:00	TV_PIE1.txt	14.35440
10378	TV_PIE1	621B0395	3/31/2022 20:00	TV_PIE1.txt	14.32781
10379	TV_PIE1	621B0395	3/31/2022 21:00	TV_PIE1.txt	14.33794
10380	TV_PIE1	621B0395	3/31/2022 22:00	TV_PIE1.txt	14.34135
10381	TV_PIE1	621B0395	3/31/2022 23:00	TV_PIE1.txt	14.34477
10382	TV_PIE1	621B0395	4/1/2022 0:00	TV_PIE1.txt	14.30105
10383	TV_PIE1	621B0395	4/1/2022 1:00	TV_PIE1.txt	14.34487
10384	TV_PIE1	621B0395	4/1/2022 2:00	TV_PIE1.txt	14.33150
10385	TV_PIE1	621B0395	4/1/2022 3:00	TV_PIE1.txt	14.30817
10386	TV_PIE1	621B0395	4/1/2022 4:00	TV_PIE1.txt	14.32486
10387	TV_PIE1	621B0395	4/1/2022 5:00	TV_PIE1.txt	14.34185
10388	TV_PIE1	621B0395	4/1/2022 6:00	TV_PIE1.txt	14.34527
10389	TV_PIE1	621B0395	4/1/2022 7:00	TV_PIE1.txt	14.34859
10390	TV_PIE1	621B0395	4/1/2022 8:00	TV_PIE1.txt	14.33531
10391	TV_PIE1	621B0395	4/1/2022 9:00	TV_PIE1.txt	14.34869
10392	TV_PIE1	621B0395	4/1/2022 10:00	TV_PIE1.txt	14.40482
10393	TV_PIE1	621B0395	4/1/2022 11:00	TV_PIE1.txt	14.45770
10394	TV_PIE1	621B0395	4/1/2022 12:00	TV_PIE1.txt	14.44083

Figura 4-25. Parte de los resultados del ángulo de oscilación para el TV\_PIE1 desde el archivo *TTDB.csv*.

#### 4.3.6. Radiación electromagnética recibida bajo la copa del árbol

En este estudio, al igual que en Zorzi *et al.* (2021), la gráfica del espectro de la radiación electromagnética recibida bajo la copa del árbol sólo corresponde al promedio por día, precisamente con los promedios de cada una de las bandas entre las 11 am y las 2 pm por cada día de observación de la gráfica, ya que es una variable que está compuesta por 12 bandas espectrales. Se deberían de considerar otras herramientas alternativas que reflejen el comportamiento de todo el espectro en periodo de tiempo más prolongado, si eso es lo que desea ver un usuario que utilice estos códigos. Quizás, se podría revisar una sola banda



espectral entre la de 7300 nm o la de 760 nm, que en la intensidad de la radiancia recibida (en sus picos), se puede interpretar la cantidad de radiación que atraviesa la copa. Así que, a picos más altos, menor follaje interpuesto entre el receptor y la luz solar. En la Figura 4-26 se puede comparar el espectrograma de dos de las parcelas, donde se ve que los pies de VM sufrieron una pérdida de hojas importante (mayor en VM\_PIE20 que en VM\_PIE22) y que, por otro lado, los pies de VS (VS\_PIE12 y VS\_PIE33) estuvieron en mejor estado en comparación de las primeras y en condiciones muy homogéneas. En esas fechas se comprobó en campo que VM\_PIE22 ya sufría una defoliación severa, mientras que VM\_PIE20 sufría una pérdida de hojas menor; ambos pies tenían una cantidad de hojas menor a la esperada debido a los fenómenos de estrés hídrico detectados en las semanas anteriores, con crecimientos y flujos de savia anormalmente bajos. En campo se comprobó asimismo que en esas fechas los pies de VS tenían una copa con la frondosidad esperable en esa época. Este fenómeno también se encuentra interpretado de esta forma en base al manual de TT+ (Ashgarinia *et al.* 2021a).

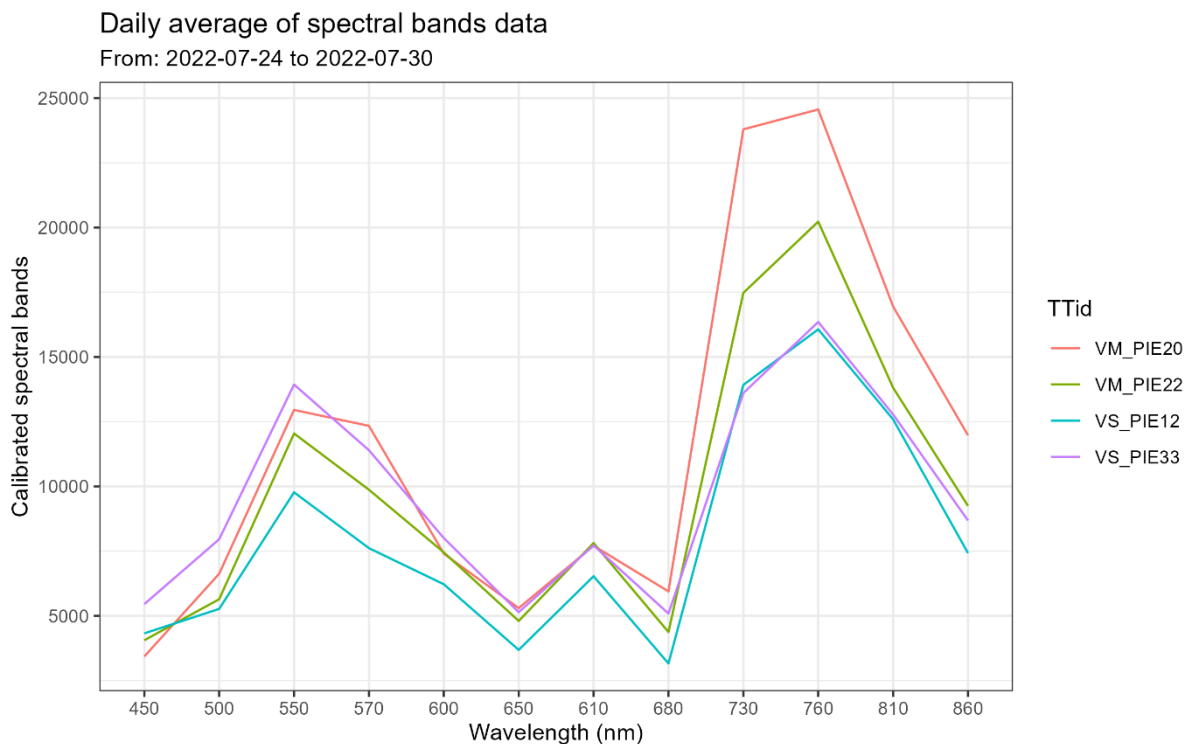


Figura 4-26. Radiación electromagnética recibida por los pies de Villamañán (VM) y Villasabariego (VS) entre el 24 y el 30 julio 2022.

Así que, visto de esta forma, la recepción de la radiación electromagnética también puede ser interpretada como la pérdida de la cobertura de hojas en el árbol que pueden ser atribuidas a graves afecciones de la roya (*Melampsora spp.*) o, por los efectos del estrés hídrico como ya ha sucedido en los años anteriores con algunos ejemplares en las parcelas de

estudio y sigue ocurriendo en el presente (Figura 4-27). Por eso ésta variable es de relevancia para la detección de alertas tempranas, sin embargo, esta calibración aún se encuentra en evaluación para la integración con otros sistemas remotos y mejorar su exactitud.



Figura 4-27. Cubierta de hojas de los árboles en la parcela de Villamañán, antes (arriba) y después (abajo) del estrés hídrico en verano de 2022.

Una limitación que se puede presentar con los TT+ es que, la lectura de la radiación electromagnética que puede verse sobresaturada en las horas donde se presenta una mayor intensidad de luz, por lo tanto, queda grabado con valores extremadamente altos, así que se

eliminaron visto que alteraba el resto de los datos en el caso de obtener un promedio de estos. En estos casos es donde se debe considerar la calibración de la sensibilidad del espectrómetro cambiando el tiempo de integración y ganancia (*integration/gain time*) (Asgharinia *et al.*, 2021).

En la Figura 4-28, se ve un ejemplo de los datos eliminados en la base de datos de los resultados (*TTDB.csv*). Aunque esto no sucedió todos los días, era común encontrar este defecto en los datos del espectro visible (bandas de 450, 500, 550, 570, 600 y 650 nm).

	A	B	C	L	M	N	O	P	Q	R	S	T	U	V	W
1	TTid	TTsn	timeStamp	B610	B680	B730	B760	B810	B860	B450	B500	B550	B570	B600	B650
14939	VS_PIE12	621B0396	5/14/2022 2:00	NA	NA	NA	NA	NA	334.88	NA	NA	NA	NA	NA	NA
14940	VS_PIE12	621B0396	5/14/2022 3:00	NA	NA	NA	NA	NA	334.88	NA	NA	NA	NA	NA	NA
14941	VS_PIE12	621B0396	5/14/2022 4:00	NA	NA	NA	NA	NA	334.88	NA	NA	NA	NA	NA	NA
14942	VS_PIE12	621B0396	5/14/2022 5:00	NA	NA	NA	NA	NA	334.88	NA	NA	NA	NA	NA	NA
14943	VS_PIE12	621B0396	5/14/2022 6:00	-305.77	-552.441	-1498.23	-1002.32	98.3112	339.128	-201.671	-218.99	-830.499	-659.397	-322.888	209.8161
14944	VS_PIE12	621B0396	5/14/2022 7:00	1157.062	809.3898	406.3247	675.184	1218.215	982.169	916.0188	1305.215	1232.298	948.6128	978.674	1413.916
14945	VS_PIE12	621B0396	5/14/2022 8:00	10059.3	8732.629	10491.56	9635.913	7144.195	4532.966	5385.866	8344.34	12507.03	10190.74	8716.215	8334.752
14946	VS_PIE12	621B0396	5/14/2022 9:00	17727.48	14944.46	18761.4	17077.73	12037.78	7502.318	9199.242	14102.03	22374.92	18175.34	15395.37	13973.2
14947	VS_PIE12	621B0396	5/14/2022 10:00	21928.95	15835.12	25097.49	23067.55	16244.78	9538.703	11174.13	15688.18	28222.68	21721.96	19159.86	15203.92
14948	VS_PIE12	621B0396	5/14/2022 11:00	35104.46	33310.93	36082.33	33650.49	22059.69	14386.73	17860.66	29920.98	45674.2	39110.85	30585.74	28329.24
14949	VS_PIE12	621B0396	5/14/2022 12:00	20634.78	20750.48	20605.98	20367.26	13538.83	9430.379	11193.75	19219.63	27208.15	23361.36	18044.36	18376.23
14950	VS_PIE12	621B0396	5/14/2022 13:00	6063.228	5343.252	4581.763	5421.068	4025.125	3132.188	3267.73	5423.572	7490.295	6412.915	5174.133	5447.417
14951	VS_PIE12	621B0396	5/14/2022 14:00	45574.73	45347.02	42679.4	39940.03	26280.16	17887.09	23944.08	NA	NA	NA	39167.91	NA
14952	VS_PIE12	621B0396	5/14/2022 15:00	33352.73	41450	41659.85	42017.62	25917.51	16403.47	19322.78	NA	NA	NA	31506.52	NA
14953	VS_PIE12	621B0396	5/14/2022 16:00	10480.11	8854.221	18388.59	22537.97	15273.8	8919.557	7474.35	12241.2	14781.8	13667.27	9978.834	8480.372
14954	VS_PIE12	621B0396	5/14/2022 17:00	11512.11	9536.656	19897.65	25006.93	15541.37	9340.109	9274.059	16278.22	20817.27	20293.9	14174.29	13195.78
14955	VS_PIE12	621B0396	5/14/2022 18:00	8761.787	11105.19	14459.53	17866.28	12191.75	7439.66	6561.95	11984.04	11365.95	11926.39	7860.335	9581.912
14956	VS_PIE12	621B0396	5/14/2022 19:00	3830.572	2808.058	6596.542	7607.783	5754.202	3234.14	3371.743	4690.252	5423.279	4607.174	3460.641	3596.642
14957	VS_PIE12	621B0396	5/14/2022 20:00	2244.167	2438.723	1492.328	2128.629	2239.674	1788.758	1818.382	2923.275	2412.396	2347.382	1934.941	3103.415
14958	VS_PIE12	621B0396	5/14/2022 21:00	-148.8	-260.62	-1138.39	-632.771	360.828	550.997	-105.869	-85.0905	-659.654	-511.882	-187.885	415.7188
14959	VS_PIE12	621B0396	5/14/2022 22:00	NA	NA	NA	NA	NA	334.88	NA	NA	NA	NA	NA	NA
14960	VS_PIE12	621B0396	5/14/2022 23:00	NA	NA	NA	NA	NA	334.88	NA	NA	NA	NA	NA	NA
14961	VS_PIE12	621B0396	5/15/2022 0:00	NA	NA	NA	NA	NA	334.88	NA	NA	NA	NA	NA	NA
14962	VS_PIE12	621B0396	5/15/2022 1:00	NA	NA	NA	NA	NA	334.88	NA	NA	NA	NA	NA	NA
14963	VS_PIE12	621B0396	5/15/2022 2:00	NA	NA	NA	NA	NA	334.88	NA	NA	NA	NA	NA	NA

Figura 4-28. Resultados de las 12 bandas espectrales calibradas en la base de datos de resultados *TTDB.csv*.

#### 4.3.7. Humedad relativa contenida en el fuste del árbol (Índice de saturación relativo)

Este es un parámetro que requiere de mayor investigación en el futuro y de una calibración bastante específica, ya que la variación del contenido de agua del fuste en los árboles puede depender de múltiples factores, como la especie, la forma de crecimiento, el tamaño, la competencia, la fenología y las condiciones ambientales abióticas (López-Bernal *et al.*, 2012; Asgharinia *et al.*, 2022). Así que, como se describió en la metodología, aún no se cuenta con la calibración de este parámetro asociado explícitamente a la especie del chopo. Lo que se hizo fue aplicar el índice de saturación relativo según el estudio de Asgharinia *et al.* (2022), que puede estimar la evolución temporal de los cambios relativos en el contenido de

agua del fuste, sin importar las condiciones ambientales o la especificidad del comportamiento ecofisiológico de la especie de árbol.

Así como muchas de las otras variables medidas por los TT+, las medidas de esta variable también presentan fluctuaciones regulares de acuerdo con el cambio diurno o nocturno, cumpliendo con la regla de que la humedad aumenta durante el día y cae durante la noche, dado que igualmente está relacionado con la transpiración del árbol producida por su actividad fotosintética (Asgharina *et al.*, 2022).

En la Figura 4-29 se muestra la saturación de humedad relativa del fuste de los árboles del estudio, en general aumentan desde abril hasta alcanzar su máximo en junio (próximo a 100%), cuando ya han brotado todas las hojas y tienen capacidad fotosintética máxima. Esta época se corresponde en campo con los máximos crecimientos semanales en diámetro (primera semana de junio). Posteriormente este valor empezó a disminuir, correspondiéndose con un descenso en el crecimiento medido en campo de los pies de la parcela de Villamañán (VM\_15, 20 y 22). Por ello, conocer estos patrones en particular para cada árbol (o clon) y sus condiciones ecofisiológicas, será de gran utilidad para identificar anomalías y alertas tempranas. Además, en el gráfico (Figura 4-29) se han detectado valores extremos para TV\_Pie1 y V; pie15 al inicio de las mediciones, que se tendrá que corroborar si realmente son o no anomalías durante el próximo año.

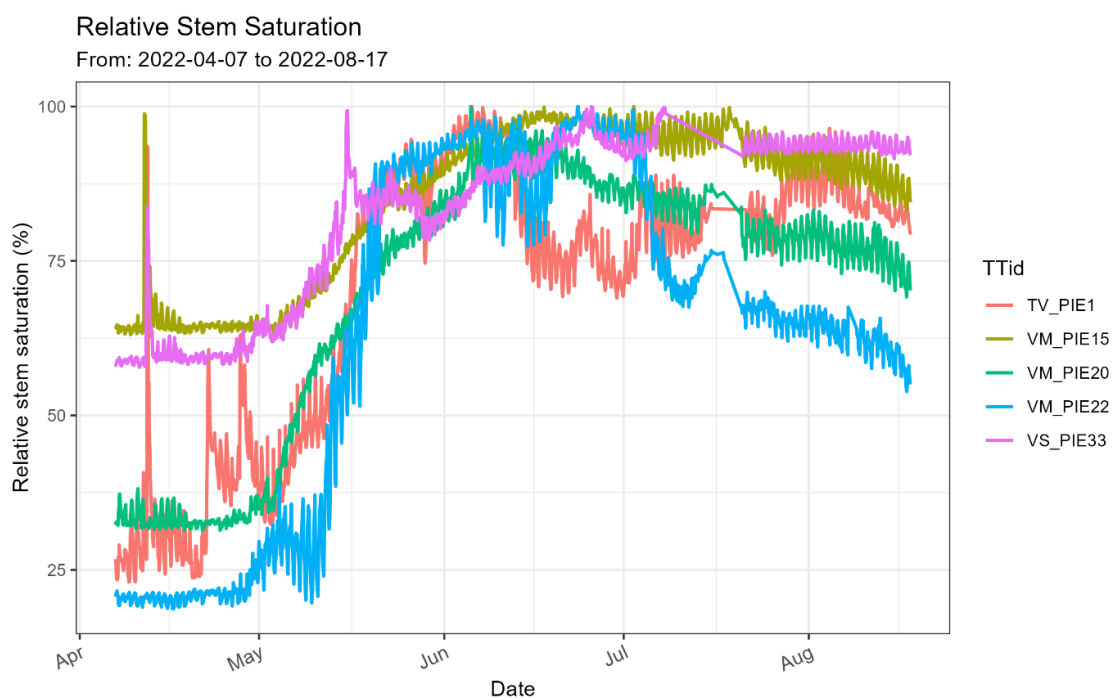


Figura 4-29. Saturación relativa de la humedad en el fuste del árbol (todos ellos son clones diferentes).

Todos los TT+, como en el caso de la densidad del flujo de savia, el VS\_PIE12 no se tuvo en cuenta para el análisis de esta variable, debido a que presentó fallos en las sondas de temperatura de referencia y de calentamiento.

#### **4.4. Resultados de las alertas**

##### ***4.4.1. Resultados de las alertas del TTCloud***

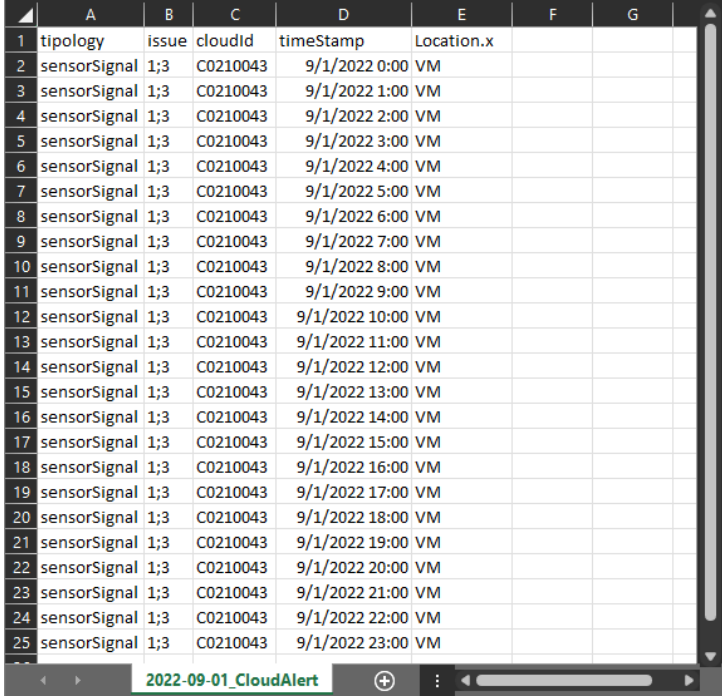
Si bien los análisis a partir de la aplicación de los umbrales contenidos en la Tabla 5 que definen 4 tipos de errores, estos están enfocados a detectar algunas de las características de un funcionamiento deficiente de los *TTClouds*. Este análisis, que sólo funciona en el código B, es realizado por defecto para el día anterior a la ejecución del código, para que así, muestre si hubo un problema a lo largo del último día de captura de datos. Esto es porque dependiendo de la señal GPRS, el *TTCloud* puede tardar entre segundos y horas en enviar la información al servidor (Zorzi *et al.* 2021). De todos modos, si el problema estuviera desde una fecha anterior, sería detectado en esas últimas 24 horas (si no se hubiese corregido).

La base completa del *TTCloud* del día anterior, queda almacenada en el *Cloud.csv* (Tabla 8), por otro lado, el análisis de las alertas, quedan guardados en la base de datos del archivo *CloudAlert.csv* (en la carpeta *C:/TreeTalkerDB/TTResults* también) y, una copia de este en el reporte en PDF (*TTReport.pdf*) (ver en Anexo 5).

En este estudio, el reporte de alertas muestra una señal RSSI inferior a -100 en el *TTCloud* C0210043, indicando que un dispositivo TT+ no se encuentra conectado al *TTCloud* (Figura 4-30), o que la señal es muy mala (Tabla 5). En este caso, se muestra como un ensayo, para demostrar qué pasa cuando se conecta con un TT+ y luego éste se encuentra apagado o desconectado. Estos mismos análisis para las alertas del *TTCloud*, fueron aplicados en el trabajo de Zorzi *et al.* (2021). Esta información es de mucha utilidad, porque puede detectar el robo de un dispositivo o un daño importante en el mismo.

En la Figura 4-30, se muestra los resultados registrados con los problemas mencionados anteriormente, el “1;3” hace referencia al número de sensores con este problema y la posición del sensor, quiere decir que el sensor número 3 es el único que no tiene señal con el *TTCloud*. Para ampliar esta misma información y al mismo tiempo ver los

sensores TT+ conectados se puede revisar el *Cloud.csv* (para ver la lista de nombres de variables y sus unidades, ver en Anexo 1).



	A	B	C	D	E	F	G
1	tipology	issue	cloudId	timeStamp	Location.x		
2	sensorSignal	1;3	C0210043	9/1/2022 0:00	VM		
3	sensorSignal	1;3	C0210043	9/1/2022 1:00	VM		
4	sensorSignal	1;3	C0210043	9/1/2022 2:00	VM		
5	sensorSignal	1;3	C0210043	9/1/2022 3:00	VM		
6	sensorSignal	1;3	C0210043	9/1/2022 4:00	VM		
7	sensorSignal	1;3	C0210043	9/1/2022 5:00	VM		
8	sensorSignal	1;3	C0210043	9/1/2022 6:00	VM		
9	sensorSignal	1;3	C0210043	9/1/2022 7:00	VM		
10	sensorSignal	1;3	C0210043	9/1/2022 8:00	VM		
11	sensorSignal	1;3	C0210043	9/1/2022 9:00	VM		
12	sensorSignal	1;3	C0210043	9/1/2022 10:00	VM		
13	sensorSignal	1;3	C0210043	9/1/2022 11:00	VM		
14	sensorSignal	1;3	C0210043	9/1/2022 12:00	VM		
15	sensorSignal	1;3	C0210043	9/1/2022 13:00	VM		
16	sensorSignal	1;3	C0210043	9/1/2022 14:00	VM		
17	sensorSignal	1;3	C0210043	9/1/2022 15:00	VM		
18	sensorSignal	1;3	C0210043	9/1/2022 16:00	VM		
19	sensorSignal	1;3	C0210043	9/1/2022 17:00	VM		
20	sensorSignal	1;3	C0210043	9/1/2022 18:00	VM		
21	sensorSignal	1;3	C0210043	9/1/2022 19:00	VM		
22	sensorSignal	1;3	C0210043	9/1/2022 20:00	VM		
23	sensorSignal	1;3	C0210043	9/1/2022 21:00	VM		
24	sensorSignal	1;3	C0210043	9/1/2022 22:00	VM		
25	sensorSignal	1;3	C0210043	9/1/2022 23:00	VM		

Figura 4-30. Ejemplo de alertas del *TTCloud*.

Por otro lado, podemos rectificar el funcionamiento de los *TTClouds* manualmente revisando en el enlace del servidor que los datos estén llegando a tiempo y en buen estado. Aunque un inconveniente que se tuvo con este mismo *TTCloud*, es que no fue posible calibrar el *Timestamp* de este enviando un SMS desde la plataforma del 1NCE según las indicaciones del manual del TT+ (Asgharinia *et al*, 2021a). Por lo tanto, es un problema que no es detectable por estos métodos, sino hasta que se realizan los respectivos ensayos de funcionamiento antes de la instalación en campo. De todos modos, este cumple con su tarea de enviar los datos al servidor mientras que se reemplaza por otro *TTCloud* y así fue usado en este estudio.

#### 4.4.2. Resultados de las alertas de los TT+

En este apartado, se ejecutó el código B con los datos obtenidos desde el servidor, con el fin de ver las alertas completas de los TT+ conectados a los *TTCloud*. En este caso, los TTid estarán identificados el sitio y con los últimos dos dígitos del TTsn (Tabla 1) (ej. TV\_TT95 es igual a TV\_PIE1), porque este nombre fue automatizado desde este código. Adicionalmente, de aquí mismo se obtuvo el *TTReport.pdf* de ejemplo que se muestra en los anexos de este trabajo (Anexo 5).

Así entonces, aplicando los condicionales descritos en el apartado 3.4.5 para generar las alertas de los TT+ según la Tabla 7, se obtuvieron los archivos de salida correspondientes a cada una de las variables establecidas y sus condicionales para generar una alerta (Tabla 8). Estas nos describen la condición de las baterías de los TT+, además del estado y el momento donde las variables ecofisiológicas del árbol son consideradas como anómalas o en estado de alerta, en el sentido que los pies requieren ser revisados o en caso tal el equipo, si presenta alteraciones fuera de lo común con las medidas.

Ahora, como alerta para el vital funcionamiento de los TT+, se tiene el nivel de batería en mV. El resultado en la Figura 4-31, muestra la fecha y el valor mínimo alcanzado de la última vez en que un TT+ pasó los límites del umbral de 3,5 mV; como vemos, solo se presentó para 4 pies, a partir de los datos que provienen de los dos servidores con sus TT+ correspondientes.

Table 1: Issues related with batteries (mV)

TTid	timeStamp	batteryLevel
TV_TT95	2022-07-15 18:00:00	2.99
VM_TT92	2022-07-20 17:00:00	3.02
VM_TT93	2022-07-17 18:00:00	2.97
VM_TT94	2022-07-19 19:00:00	3.07

Figura 4-31. Reporte de alertas de las baterías desde el *TTReport.pdf*.

En el caso de las alertas del flujo de savia, se estableció un límite para el valor máximo semanal que esté por debajo de  $1,0 \text{ l dm}^{-2} \text{ h}^{-1}$  indicando niveles muy bajos de la densidad de flujo de savia. Como resultado, se obtuvo la tabla de la Figura 4-32 en el reporte final, con el historial completo de los datos anormales. Si se tiene en cuenta la semana del año (YYYYww), vemos que el flujo de savia fue casi nulo cuando se hicieron los ensayos antes de la instalación (en la oficina, en un tronco muerto), y posteriormente entre las semanas 27 y 30, que corresponden entre el 4 al 30 de julio del año 2022 ([EpochConverter](#), 2022), se comprueba que a los pies VM\_PIE\_20, TV\_PIE2 y el VM\_PIE22 (TT92, TT98, TT93, respectivamente) les afectó en gran proporción el estrés hídrico que sufrieron durante esa semana, provocando una caída en el flujo de savia, situaciones altamente correlacionadas (Asgharina *et al.*, 2022).

Table 2: Issues related with sap flux density (l d m-2 h-1)

TTid	YYYYww	sapFluxDensity	timeStamp
TV_TT98	202212	0.0000	2022-03-22 16:00:00
VM_TT96	202212	0.0550	2022-03-21 22:00:00
VM_TT92	202227	0.6327	2022-07-02 16:00:00
VM_TT92	202228	0.7803	2022-07-13 19:00:00
TV_TT98	202229	0.7331	2022-07-21 18:00:00
VM_TT93	202229	0.7697	2022-07-21 19:00:00
VM_TT93	202230	0.8637	2022-07-25 18:00:01

 Figura 4-32. Reporte de alertas de la densidad de flujo de savia bajas del *TTReport.pdf*.

Este límite de flujo de savia fue determinado por la observación de los datos recogidos de manera objetiva, y funcionó al coincidir las alertas con las condiciones de estrés hídrico más extremas. Pero si el usuario desea que el límite sea menos restrictivo, se puede cambiar fácilmente en la variables límites del TAlert en el código (Parte 4 del Anexo 2 o Anexo 3).

En Villamañán se considera que las condiciones de estrés hídrico, además de ser causado por las condiciones de aridez climáticas, adyacentemente, pueden ser causa por la disminución del nivel del freático que puede estar relacionada con la proximidad de dos canteras de extracción de áridos. Es posible que esta sea la razón por la cual se reportan alertas en Villamañán y en Villasabariego no. En este último, no se ha presentado esa bajada extrema del nivel freático que ha probado defoliaciones.

En relación con el crecimiento semanal medio del radio del fuste, las alertas se generaron a partir de un cambio en el crecimiento inferior a 0,1 mm entre una semana a la siguiente semana (*deltaGrowth* en la Figura 4-33). Aunque solo se deben considerar las mediciones a partir del YYYYww = 202216, dos semanas después de la instalación en campo de todos los TT+ (en el código está la opción de definir la semana), porque la primera muestra la diferencia entre el tronco de ensayo y el pie definitivo.

Table 3: Issues related with the weekly radial growth (mm)

TTid	YYYYww	dendrometer	deltaGrowth
TV_TT95	202216	0.05	-0.06
TV_TT95	202217	0.00	-0.05
TV_TT95	202218	0.00	0.00
TV_TT95	202225	4.45	0.01
TV_TT95	202226	4.46	0.01
TV_TT95	202229	6.11	0.03
TV_TT95	202230	6.11	0.00

 Figura 4-33. Parte de la tabla de resultados del *TTReport.pdf* con las alertas de la tasa de crecimiento semanal.



Debido a la variabilidad que se presenta en la expansión y contracción del fuste, hay resultados en el cambio de crecimiento que también se presentan en valores negativos. Por otro lado, también podríamos cambiar el límite de alerta a 0,0 mm si se desea tener datos más restrictivos.

Sin embargo, cabe resaltar que los resultados marcaron una tendencia constante de un crecimiento del fuste cercano a cero e incluso valores de una contracción del fuste, en las semanas donde se evidenció los problemas de estrés hídrico. Ambas variables, al igual que la tendencia a un flujo de savia muy bajos están muy relacionados entre sí, debido a estas circunstancias de sequía y de calor (Ortuno *et al*, 2006).

Por esto mismo es que Makinen *et al*. (2003), consideró que el uso de dendrómetros automáticos es de un valor "dudoso" para medir el momento de la formación real del xilema, ya que los cambios diarios en el radio del tronco se correlacionaron los cambios en el contenido de agua del mismo. De aquí también renace la importancia de reescalar dichos datos para que se relacionen sobre una base común y comprender mejor la actividad de cambio y los procesos de desarrollo del xilema. Con el enfoque de reescala como lo define Drew y Downes (2009), es posible estimar al menos aproximadamente cuando se formaron porciones de madera radialmente secuenciales. Lo complejo está en definir cuándo sucede la formación de la madera, que un "crecimiento" que en realidad es un estímulo dependiente de las condiciones ambientales, como un ensanchamiento del fuste después de una precipitación, lo que reflejaría los cambios en el contenido de agua en el tallo Makinen *et al*. (2003). Aunque la contribución del tejido del xilema a la contracción y la hinchazón reversibles del fuste del árbol depende de la especie (De Swaef *et al*, 2015).

Como se presentó una situación con una pupa de mariposa que alteró el sensor de crecimiento del VM\_PIE12 (TT96), este dato fue filtrado automáticamente de los datos por contener valores anormales. Se debe recordar que ésta alerta corresponde a valores bajos en los crecimientos radiales.

Se debe tomar en cuenta que, durante la estación de invierno el flujo de savia y el crecimiento radial se acercan muchísimo a valores nulos, así que estas alertas sólo se deben considerar en épocas de crecimiento del árbol (entre mayo y septiembre).

Por último, con el fin de valorar la estabilidad del árbol y detectar los volcamientos

potenciales, las alertas reportan los momentos en que se superó el ángulo (°) permitido en el historial completo. En la Figura 4-34 se muestran estos datos cuando se hizo la calibración y ensayos de los equipos antes de la instalación en campo.

Para esta variable, se recomienda especificar una fecha para el análisis de las alertas. Como alternativa, se podría modificar el código para que a manera de reporte y no como una alerta, se muestre en su lugar, el último ángulo que se obtuvo de los datos.

Table 4: Issues related with the tree angles (°)

TTid	timeStamp	TreeAngle
TV_TT95	2022-03-30 20:00:00	89
TV_TT98	1970-01-01 02:03:05	47
VM_TT92	1970-01-01 02:04:14	65
VM_TT93	2022-03-22 11:00:01	66
VM_TT94	2022-03-22 10:00:00	81
VM_TT96	2022-03-21 18:00:00	54

Figura 4-34. Reporte de alertas para el ángulo del fuste del árbol de *TTRreport.pdf*.

En la actualidad se continúa en mejorar la exactitud de las alertas, especialmente las que tienen el enfoque al estado ecofisiológico del árbol, ya que de esto depende que se puedan tomar medidas a tiempo por parte del propietario, como por ejemplo realizar riegos si es posible cuando el TT+ indique tasas de crecimiento muy bajas, flujo de savia muy bajos o alteraciones mayores en la cantidad de radiación recibida por el espectrómetro (y se verifique en campo su relación con estrés hídrico).

#### **4.5. Integración en sistema de alerta temprana**

Para evaluar la aplicabilidad del empleo IoT y los sensores ecosifiológicos en un sistema de alerta temprana para choperas se ha tenido en cuenta (i) la correspondencia entre la información proporcionada por el sistema y las situaciones de estrés/daños observadas en campo y (ii) la posibilidad real de uso por parte de empresas/administración/propietarios particulares (operatividad).

En primer lugar, los resultados de los análisis de los datos de la densidad del flujo de savia medida por el TT+ cada hora y su comparación con los crecimientos en diámetro medidos en campo semanalmente, con el estado sanitario determinado semanalmente de forma visual, han permitido verificar la validez de esta variable como indicador de alerta temprana de daños en choperas. Además, se han incluido también como indicadores de

posibles daños los crecimientos radiales del fuste por debajo de un umbral y el ángulo de volcamiento del árbol mayor de  $45^\circ$ , como indicador de posibles daños mecánicos. Estos valores son fácilmente modificables en los códigos de R desarrollados en este trabajo y adaptables a las condiciones de la chopera (según la edad y calidad de la estación, el umbral de crecimiento será podrá ser diferente). Se prevé incluir en un futuro otros indicadores de alerta como la radiación recibida por el sensor (indicador de defoliación) o la humedad relativa del fuste, que ya muestran relación con estados de estrés en los datos analizados.

En segundo lugar, la disponibilidad de los dos códigos de R en un repositorio abierto y con explicaciones y anotaciones tan detalladas de cómo emplearlos y cómo modificar los parámetros de entrada y adaptarlos, supone una gran ventaja a la hora de que su uso se pueda generalizar y los usuarios puedan obtener las alertas con la frecuencia que establezcan. En la época de crecimiento mayor de las choperas en la cuenca del Miño-Sil, Duero y del Ebro (abril-septiembre) se recomienda hacer el análisis con una frecuencia mínima semanal. En el caso de que las condiciones meteorológicas sean de calor o frío extremo, se recomienda ejecutar el análisis con una mayor frecuencia (e.g. ola de calor), para poder detectar las anomalías lo antes posible. En el caso de fenómenos extremos de viento o nieve, se recomienda hacer el análisis inmediatamente después, para comprobar posibles derribos/daños. Estos análisis podría hacerlos el interesado o subcontratarlos, obteniendo el reporte de alertas con la frecuencia deseada. Además, una de las líneas de mejora previstas es automatizar el aviso de las alertas, de forma que los análisis se hagan diariamente y los propietarios reciban avisos de las alertas automáticamente en su dispositivo móvil (vía mensaje o e-mail).

Se debe tener en cuenta que, aunque se trata de un sistema low-cost de seguimiento, no es posible, ni recomendable, el sensorizar todos y cada uno de los pies de la plantación. La recomendación sería sensorizar aquellos pies que se puedan considerar “centinela”, bien porque han sufrido daños o estrés en años anteriores y, por lo tanto, pueden ser más susceptibles a nuevos daños, bien porque se encuentran en zonas de la parcela susceptibles a daños por viento o estrés hídrico (nivel freático), o bien porque pertenecen clones especialmente susceptibles a una plaga/enfermedad. Para localizar zonas con daños en años anteriores se recomienda el empleo de series temporales de imágenes de Sentinel-2 MSI, que permiten localizar áreas con valores anómalos de vigor (empleando por ejemplo EO Browser y las series temporales del índice de vegetación NDVI). Se podría emplear también, cuando

esté plenamente operativa, la herramienta PoplarAlert, en desarrollo por parte de investigadores de la Escuela de Ingeniería Agraria y Forestal, que empleando Google Earth Engine y Python como base, permite la detección de anomalías en choperas en el intervalo temporal seleccionado (Álvarez-Taboada *et al.*, 2022). De este modo, se podrían identificar las zonas a sensorizar dentro de una plantación o de un conjunto de plantaciones y localizar ahí un ecosistema con un *TTcloud* con los TT+ necesarios, localizados en los pies “centinela”. En el caso de recibir una alerta en el reporte, se podría analizar con más detalle el resto de los indicadores y valores, acudir a campo para valorar las acciones a realizar para, una vez verificado el problema, mitigarlo y/o evitar su expansión a otras zonas de la plantación. De este modo, la implementación de los TT+ en un sistema de seguimiento de daños y alerta temprana en choperas, tal y como se ha planteado, tendría las siguientes ventajas para sus usuarios: (i) disminuir costes, al no ser necesario la evaluación de daños en campo (gasto económico y de tiempo), (ii) su utilización a escala regional y en grandes extensiones (bajos costes de procesamiento, al emplear los códigos de R), (iii) una fácil identificación de las desviaciones de las condiciones normales relacionadas en mayor medida con la aparición de un daño en la masa forestal, por su alta cobertura espacial y temporal, minimizando así pérdidas económicas y ambientales.

## 5. CONCLUSIONES

El seguimiento y detección de daños/estrés y en tiempo casi real en choperas empleando internet de las cosas (IoT) y los sensores ecofisiológicos TreeTalkers, es operativo y posible utilizando las dos herramientas desarrolladas en este trabajo, que a través del lenguaje de programación R y en siete pasos permiten, para el periodo temporal seleccionado: (i) convertir las diferentes cadenas de datos (números digitales) a variables físicas mensurables relacionadas con el estado fisiológico y sanitario del árbol, (ii) representar gráficamente los datos obtenidos, (iii) generar un informe con los resultados de los análisis y (iv) proporcionar un informe con las alertas de funcionamiento del sistema y con las alertas por valores anómalos las variables seleccionadas como indicadores del estado sanitario del árbol.

Los dos códigos desarrollados en este TFM permiten la automatización del procesamiento de los datos registrados por los sensores, adaptándose al origen de los datos (el propio sensor o el servidor de datos), lo que los hace muy versátiles. Estos códigos tienen la ventaja de que pueden ser aplicados a otros ecosistemas de sensores con otras especies y en otros lugares, puesto que su código está disponible de forma abierta y se acompaña de una descripción detallada en inglés del contenido, que permite la modificación sencilla de los parámetros de entrada y del periodo de análisis requerido.

La calibración del sistema realizada para las plantaciones de chopo se ha mostrado efectiva para (i) seleccionar los valores adecuados de parámetros de cálculo de las variables medidas por los sensores, (ii) identificar varias variables relacionadas con daños y estrés en choperas, y (iii) determinar los valores umbral que definen si se está produciendo una anomalía fisiológica o de tipo mecánico/externo. Se han incluido en el código las siguientes variables generadoras de alertas: la densidad del flujo de savia, el crecimiento radial del fuste y el ángulo de volcamiento del árbol. Se prevé incluir otros indicadores como los cambios en la radiación recibida por el sensor o la humedad relativa del fuste, que han mostrado relación con estados de estrés en los datos analizados. Los valores umbral se pueden modificar en el código y adaptarlos a las condiciones de la chopera en seguimiento.

El sistema desarrollado ha resultado efectivo para la detección temprana de estrés hídrico, empleando la densidad de flujo de savia, previo a su reflejo en la disminución del

crecimiento radial y la pérdida de hojas. Se considera necesario disponer de una serie temporal más larga y con más pies sensorizados para determinar la utilidad de esta variable en el caso de estrés térmico y en el caso de la afección por plagas/enfermedades (escolítidos, pulgón, royas). Para el empleo efectivo del crecimiento radial del fuste como alerta temprana, es necesario mejorar su calibración, así como el pre-procesamiento de los datos.

Para que sea viable económicamente y operativo, el sistema de detección y seguimiento de daños propuesto se basa en sensorizar con los TT+ únicamente los árboles “centinela” (que ya han sufrido daños anteriormente o que pueden considerarse como representativos de una zona de la chopera), que son los que informarían de la alerta temprana a escala de árbol individual, mientras que a escala de rodal se podrían emplear sensores remotos como Sentinel 2 MSI, que con frecuencia semanal proporcionan de forma gratuita del vigor de la vegetación.

Para aumentar la facilidad de uso, se considera idóneo desarrollar en un futuro próximo una versión del código que funciones como un paquete en R o una aplicación interactiva en *Shiny* de R (<https://shiny.rstudio.com/>), para que su utilización sea aún más operativa y permita la toma de decisiones de una forma directa e informada por parte del gestor forestal.

## 6. REFERENCIAS BIBLIOGRÁFICAS

AEMET, 2022a. Valores Climatológicos Normales. León Aeropuerto. *Página web de la Agencia Estatal de Meteorología AEMET*. Disponible en:

<https://web.archive.org/web/20110815015702/http://www.aemet.es/es/elclima/datosclimatologicos/valoresclimatologicos?!=2661&k=cle> (Accedido: 27 de agosto de 2022).

AEMET, 2022b. Valores Climatológicos Normales. León - Ponferrada. *Página web de la Agencia Estatal de Meteorología AEMET*. Disponible en:

<https://web.archive.org/web/20091230145756/http://www.aemet.es/es/elclima/datosclimatologicos/valoresclimatologicos?!=1549&k=cle> (Accedido: 27 de agosto de 2022).

AIEF, Área de Inventarios y Estadísticas Forestales. 2019. Inventario de daños forestales (IDF) en España. Red Europea de Seguimiento de Daños en los Bosques. Nivel I. Resultados del Muestreo de 2019.

Alvarez-Taboada, F., Sánchez-Berasategui, L., Arhrib, Y., Castedo-Dorado, F., Garnica-López, J., Francini, S., D'Amico, G., 2022. An on-line multiscale system based on sentinel 2 imagery and ecophysiological sensors to monitor forest health in poplar plantations. ForestSAT 2022. Berlin 29-03 sept 2022. Disponible en: [https://www.conftool.org/forestsat2022/index.php?page=browseSessions&abstracts=show&form\\_session=40&presentations=show](https://www.conftool.org/forestsat2022/index.php?page=browseSessions&abstracts=show&form_session=40&presentations=show) (Accedido: 6 de septiembre 2022).

Asgharina, S., Leberecht, M., Belevi Marchesini, L., Friess, N., Gianelle, D., Nauss, T., Opgenoorth, L., Yates, J. and Valentini, R., 2022. Towards Continuous Stem Water Content and Sap Flux Density Monitoring: IoT-Based Solution for Detecting Changes in Stem Water Dynamics. *Forests*, 13(7), p.1040.

Asgharina, S., Yates, J. & Valentini, R., 2021a. Converting Raw TreeTalker Data into Physical Units. Version: TT+ 3.0 and later. *DIBAF - Tucsia University*. Nature 4.0 Company.

Asgharina, S.; Yates, J.; Zompanti, R.; Bozzo, F., 2021b. Preparation, testing and installation. TreeTalker. Nature 4.0 Company.

Ayuntamiento de Villamañán, 2022. *Página web del Ayuntamiento de Villamañán*. Disponible en: <http://www.aytovillamanan.es/municipio/> (Accedido: 27 de agosto de 2022).

Becker, P., 1998. Limitations of a compensation heat pulse velocity system at low sap

flow: implications for measurements at night and in shaded trees. *Tree Physiology*, 18(3), pp.177-184. Disponible en: <https://academic.oup.com/treephys/article/18/3/177/1641639?login=false> (Accedido: 27 de agosto de 2022).

Benson, T., 2022. *Página web del Glen Research Center; Aerodinamics*. NASA. Disponible en: <https://www.grc.nasa.gov/www/k-12/VirtualAero/BottleRocket/airplane/short.html> (Accedido: 27 de agosto de 2022).

Buras, A., Rammig, A. and Zang, C.S., 2021. The European Forest Condition Monitor: Using Remotely Sensed Forest Greenness to Identify Hot Spots of Forest Decline. *Frontiers in plant science*, 12.

Burgess, S.S., Adams, M.A. and Bleby, T.M., 2000. Measurement of sap flow in roots of woody plants: a commentary. *Tree Physiology*, 20(13), pp.909-913.

Cailleret, M., Jansen, S., Robert, E.M., Desoto, L., Aakala, T., Antos, J.A., Beikircher, B., Bigler, C., Bugmann, H., Caccianiga, M. and Čada, V., 2017. A synthesis of radial growth patterns preceding tree mortality. *Global change biology*, 23(4), pp.1675-1690.

Castaldi, S., Antonucci, S., Asgharina, S., Battipaglia, G., Belelli Marchesini, L., Cavagna, M., Chini, I., Coccozza, C., Gianelle, D., La Mantia, T., Motisi, A., Niccoli, F., Pacheco Solana, A., Sala, G., Santopuoli, G., Tonon, G., Tognetti, R., Zampedri, R., Zorzi, I., and Valentini, R. 2020. The Italian TREETALKER NETWORK (ITT-Net): continuous large-scale monitoring of tree functional traits and vulnerabilities to climate change, EGU General Assembly 2020. EGU2020-20591. Disponible en: <https://doi.org/10.5194/egusphere-egu2020-20591> (Accedido: 6 de septiembre de 2022).

CHD, Confederación Hidrográfica del Duero, 2020. Beneficios Ambientales y Socioeconómicos de la Populicultura en la Cuenca del Duero. Propuestas del sector para el III Ciclo de Planificación Hidrológica. *AEFCON; COSE; FAFCYL; Junta de Castilla y León; Sociedad Pública de Infraestructura y Medio Ambiente de Castilla y León S.A.* Disponible en: <https://www.chduero.es/documents/20126/1213596/N730.pdf> (Accedido: 6 de septiembre de 2022).

Chopo en Castilla y León, 2021. Página web del Chopo en Castilla y León. La populicultura en Castilla y León. Disponible en: <http://www.populuscyl.es/contenido/la-populicultura-en->



[castilla-leon](#) (Accedido: 6 de septiembre de 2022).

Cui, F., 2020. Deployment and integration of smart sensors with IoT devices detecting fire disasters in huge forest environment. *Computer Communications*, 150, pp.818-827. Disponible en: [https://www.miteco.gob.es/es/biodiversidad/temas/inventarios-nacionales/inventariodedanosforestalesidf2019\\_tcm30-512491.pdf](https://www.miteco.gob.es/es/biodiversidad/temas/inventarios-nacionales/inventariodedanosforestalesidf2019_tcm30-512491.pdf) (Accedido: 6 de septiembre de 2022)

De Swaef, T., De Schepper, V., Vandegheuchte, M.W. and Steppe, K., 2015. Stem diameter variations as a versatile research tool in ecophysiology. *Tree Physiology*, 35(10), pp.1047-1061.

Do, F. and Rocheteau, A., 2002. Influence of natural temperature gradients on measurements of xylem sap flow with thermal dissipation probes. 2. Advantages and calibration of a noncontinuous heating system. *Tree physiology*, 22(9), pp.649-654.

Drew, D.M. and Downes, G.M., 2009. The use of precision dendrometers in research on daily stem size and wood property variation: a review. *Dendrochronologia*, 27(2), pp.159-172.

Ecke, S., Dempewolf, J., Frey, J., Schwaller, A., Endres, E., Klemmt, H.J., Tiede, D. and Seifert, T., 2022. UAV-Based Forest Health Monitoring: A Systematic Review. *Remote Sens.* 2022, 14, 3205.

EpochConverter, 2022. *Página web del Epoch Converter*. Disponible en: <https://espanol.epochconverter.com/semanas/2022> (Accedido: 31 de agosto de 2022).

FISE, 2022. *Página web del Forest Information System for Europe*. European Commission. Disponible en: <https://forest.jrc.ec.europa.eu/en/> (Accedido: 6 de septiembre de 2022).

Fisher, C., 2011. *Página web de Digi-key Electronics*. Disponible en: <https://www.digikey.it/it/articles/using-an-accelerometer-for-inclination-sensing> (Accedido: 27 de agosto de 2022).

Gaulton, R., Agstner, B., Brauholtz, L., Butler, L., Clements, C., Crowe, A., Jones, G., Kleczkowski, A., McMenemy, P., Pocock, M., Stokes, J., Taylor, M., Village, S., Brown, P. 2022. Multi-scale sensing and citizen science for tree health monitoring in UK treescapes. *School of*

Natural and Environmental Sciences, Newcastle University, United Kingdom; Fera Science Ltd, United Kingdom; Department of Mathematics and Statistics, University of Strathclyde, United Kingdom; The Tree Council, United Kingdom; UK Centre for Ecology and Hydrology, United Kingdom. Live Oral Presentation. Disponible en: [https://www.conftool.org/forestsat2022/index.php?page=browseSessions&abstracts=show&form\\_session=33&presentations=show](https://www.conftool.org/forestsat2022/index.php?page=browseSessions&abstracts=show&form_session=33&presentations=show) (Accedido: 6 de septiembre de 2022).

Granier, A., 1987. Evaluation of transpiration in a Douglas-fir stand by means of sap flow measurements. *Tree physiology*, 3(4), pp.309-320.

IDECyL, 2022. Elevaciones (MDT) de Castilla y León: Servicio WMS. *Página web de la Junta de Castilla y León, Servicio WMS de visualización de mapas*. Disponible en: <https://idecyl.jcyl.es/geonetwork/static/api/records/SPAGOBCYLCITWMSEL> (Accedido: 27 de agosto de 2022).

IFN, Inventario Forestal Nacional. 2021. Cuarto Inventario Forestal Nacional en la C. A. de Castilla y León. Ministerio para la Transición Ecológica y el Reto Demográfico (MITECO). Disponible en: [https://www.miteco.gob.es/es/biodiversidad/temas/inventarios-nacionales/ifn4\\_castillaleon\\_tcm30-543067.pdf](https://www.miteco.gob.es/es/biodiversidad/temas/inventarios-nacionales/ifn4_castillaleon_tcm30-543067.pdf) (Accedido: 6 de septiembre de 2022).

IPPC Secretariat, Gullino, M.L., Albajes, R., Al-Jboory, I., Angelotti, F., Chakraborty, S., Garrett, K.A., Hurley, B.P., Juroszek, P., Makkouk, K. and Pan, X., 2021. *Scientific review of the impact of climate change on plant pests*. FAO on behalf of the IPPC Secretariat.

Isarangkool Na Ayutthaya, S., Do, F.C., Pannengpetch, K., Junjittakarn, J., Maeght, J.L., Rocheteau, A. and Cochard, H., 2010. Transient thermal dissipation method of xylem sap flow measurement: multi-species calibration and field evaluation. *Tree Physiology*, 30(1), pp.139-148. Disponible en: <https://academic.oup.com/treephys/article/30/1/139/1645642?login=true> (Accedido: 27 de agosto de 2022).

James, K.R. and Hallam, C., 2013. Stability of urban trees in high winds. *Arboricultural Journal*, 35(1), pp.28-35.

Kubick, W.R., 2012. Big data, information and meaning. *Applied Clinical Trials*, 21(2), p.26. Disponible en: <https://www.proquest.com/docview/1014265633?pq-origsite=gscholar&fromopenview=true> (Accedido: 29 de agosto de 2022).

Lausch, A., Borg, E., Bumberger, J., Dietrich, P., Heurich, M., Huth, A., Jung, A., Klenke, R., Knapp, S., Mollenhauer, H. and Paasche, H., 2018. Understanding forest health with remote sensing, part III: requirements for a scalable multi-source forest health monitoring network based on data science approaches. *Remote sensing*, 10(7), p.1120.

López-Bernal, Á., Testi, L. and Villalobos, F.J., 2012. Using the compensated heat pulse method to monitor trends in stem water content in standing trees. *Tree physiology*, 32(11), pp.1420-1429.

Magh, R.K., Bonn, B., Grote, R., Burzlaff, T., Pfautsch, S., Rennenberg, H., 2019. Drought superimposes the positive effect of silver fir on water relations of european beech in mature forest stands. *Forests* 10 (10). Disponible en: <https://doi.org/10.3390/f10100897> (Accedido: 3 de septiembre de 2022).

Mäkinen, H., Nöjd, P. and Saranpää, P., 2003. Seasonal changes in stem radius and production of new tracheids in Norway spruce. *Tree Physiology*, 23(14), pp.959-968.

Matasov, V., Belelli Marchesini, L., Yaroslavtsev, A., Sala, G., Fareeva, O., Seregin, I., Castaldi, S., Vasenev, V. and Valentini, R., 2020. IoT monitoring of urban tree ecosystem services: Possibilities and challenges. *Forests*, 11(7), p.775.

Mencuccini, M., Salmon, Y., Mitchell, P., Hölttä, T., Choat, B., Meir, P., O'grady, A., Tissue, D., Zweifel, R., Sevanto, S. and Pfautsch, S., 2017. An empirical method that separates irreversible stem radial growth from bark water content changes in trees: theory and case studies. *Plant, cell & environment*, 40(2), pp.290-303.

Michel, A.K., Prescher, A.K. and Schwärzel, K., 2020. *Forest Condition in Europe: The 2020 Assessment; ICP Forests Technical Report Under the UNECE Convention on Long-range Transboundary Air Pollution (Air Convention)*. Thünen-Institut, Bundesforschungsinstitut für Ländliche Räume, Wald und Fischerei.

Naciones Unidas, 2022. *Página web de las Naciones Unidas*. Agenda 2030 sobre el Desarrollo Sostenible. Objetivos de desarrollo sostenible. Disponible en: <https://www.un.org/sustainabledevelopment/es/objetivos-de-desarrollo-sostenible/> (Accedido: 6 de septiembre de 2022).

Nature 4.0, 2020a. *TT+ User Manual. TreeTalker version TT+, suggested firmware*

TT3\_2\_plus.5\_0h. Nature 4.0 Company.

Nature 4.0, 2020b. *TT-Cloud User Manual*. Nature 4.0 Company.

Nature 4.0, 2022. Página web de Nature 4.0. Disponible en: [www.nature4.org](http://www.nature4.org) (Accedido: 28 de agosto de 2022).

Ortuno, M.F., García-Orellana, Y., Conejero, W., Ruiz-Sánchez, M.C., Alarcón, J.J. and Torrecillas, A., 2006. Stem and leaf water potentials, gas exchange, sap flow, and trunk diameter fluctuations for detecting water stress in lemon trees. *Trees*, 20(1), pp.1-8.

Periódico de España, 2022. *Página web del periódico de España. Julio de 2022, el mes más cálido en España desde hace 60 años*. Consultado en web: [Enlace](#). (Accedido: 29 de agosto de 2022).

Pfautsch, S., 2016. Hydraulic anatomy and function of trees—basics and critical developments. *Current Forestry Reports*, 2(4), pp.236-248.

Phillips, N. and Oren, R., 1998. A comparison of daily representations of canopy conductance based on two conditional time-averaging methods and the dependence of daily conductance on environmental factors. *In Annales des Sciences Forestieres* (Vol. 55, No. 1-2, pp. 217-235). EDP Sciences.

Poyatos, R., Granda, V., Molowny-Horas, R., Mencuccini, M., Steppe, K. and Martínez-Vilalta, J., 2016. SAPFLUXNET: towards a global database of sap flow measurements. *Tree Physiology*, 36(12), pp.1449-1455.

RDocumentation, 2022. *Página web de RDocumentation*. Disponible en: <https://www.rdocumentation.org/packages/prospectr/versions/0.2.5/topics/savitzkyGolay> (Accedido: 13 de agosto, 2022)

Rose, K., Eldridge, S. and Chapin, L., 2015. The internet of things: An overview. *The internet society* (ISOC), 80, pp.1-50.

R-project, 2022. *Página web de R-Project*. CRAN, The Comprehensive R Archive Network. Disponible en: <https://cran.r-project.org/> (Accedido: 30 de agosto de 2022).

SAS Institute, 2022. *Página web del SAS Institute Inc*. Big data, qué es y por qué es importante. Disponible en: [https://www.sas.com/es\\_ar/insights/big-data/what-is-big-data.html](https://www.sas.com/es_ar/insights/big-data/what-is-big-data.html) (Accedido: 29 de agosto de 2022).

Singh, R., Gehlot, A., Akram, S.V., Thakur, A.K., Buddhi, D. and Das, P.K., 2021. Forest 4.0: Digitalization of forest using the Internet of Things (IoT). *Journal of King Saud University-Computer and Information Sciences*.

SUN Lab, 2020. Página web del SUN Lab. Tree Talkers Will monitor Moscow trees. Disponible en: <https://sunlab.rudn.ru/tree-talkers-will-monitor-moscow-trees.html> (Accedido: 6 de septiembre de 2022).

Tomelleri, E., Belelli Marchesini, L., Yaroslavtsev, A., Asgharina, S. and Valentini, R., 2022. Toward a Unified TreeTalker Data Curation Process. *Forests*, 13(6), p.855.

Trumbore, S., Brando, P. and Hartmann, H., 2015. Forest health and global change. *Science*, 349(6250), pp.814-818.

Valentini, R., Belelli Marchesini, L., Gianelle, D., Sala, G., Yaroslavtsev, A., Vasenev, V. and Castaldi, S., 2019. New tree monitoring systems: from Industry 4.0 to Nature 4.0. *Annals of Silvicultural research*, 43(2), pp.84-88. Disponible en: <https://openpub.fmach.it/handle/10449/58628> (Accedido: 27 de agosto de 2022).

Vieira, J., Campelo, F. and Nabais, C., 2022. Environment Controls Seasonal and Daily Cycles of Stem Diameter Variations in Portuguese Oak (*Quercus faginea* Lambert). *Forests*, 13(2), p.170.

Wulder, M.A., Masek, J.G., Cohen, W.B., Loveland, T.R. and Woodcock, C.E., 2012. Opening the archive: How free data has enabled the science and monitoring promise of Landsat. *Remote Sensing of Environment*, 122, pp.2-10.

Zorzi, I., Francini, S., Chirici, G. and Coccozza, C., 2021. The TreeTalkersCheck R package: An automatic daily routine to check physiological traits of trees in the forest. *Ecological Informatics*, 66, p.101433.

Zweifel, R., Sterck, F., Braun, S., Buchmann, N., Eugster, W., Gessler, A., Häni, M., Peters, R.L., Walthert, L., Wilhelm, M. and Ziemińska, K., 2021. Why trees grow at night. *New Phytologist*, 231(6), pp.2174-2185. Disponible en: <https://nph.onlinelibrary.wiley.com/doi/10.1111/nph.17552> (Accedido: 27 de agosto de 2022).

## ANEXOS

### Anexo 1. Tabla de nombres de referencia de las variables de las bases de datos en formato CSV

Nombre de columna (datos del TT+)	Unidad o variable
adc_bandgap	Nivel de batería (d.n.)
adc_vbat	Nivel de batería (d.n.)
batteryLevel	Nivel de batería (mV)
dendrometer	Crecimiento semanal del radio del fuste (mm)
DistGrowth	Distancia del sensor a la corteza (mm)
gain	Ganancia del espectrómetro
growth	Crecimiento (d.n.)
gxm	Ángulo eje x (d.n.)
gxsd	Desviación estándar del ángulo eje x (d.n.)
gym	Ángulo eje y (d.n.)
gysd	Desviación estándar del ángulo eje y (d.n.)
gzm	Ángulo eje z (d.n.)
gzsd	Desviación estándar del ángulo eje z (d.n.)
intTime	Tiempo de integración del espectrómetro
Location	Lugar/ubicación/nombre del TXT
nbits	Número de bits
relativeHumidity	Humedad relativa del aire (%)
RelStemSat	Humedad relativa del fuste (%)
sapFluxDensity	Densidad de flujo de savia (l dm <sup>-2</sup> h <sup>-1</sup> )
stringType	Cadena de tipo de datos: 4D / 55 / 49
stwc	Steam water content (d.n.)
temperature	Temperatura (d.n. / °C)
theat_0	Temperatura calentamiento inicial (d.n.)
theat_Ofilt	Temperatura de referencia inicial (°C)
theat_1	Temperatura calentamiento inicial (d.n.)
theat_1filt	Temperatura calentamiento inicial (°C)
timeStamp	Fecha y hora
TreeMovement	Ángulo del fuste (°)
tref_0	Temperatura de referencia inicial (d.n.)
tref_Ofilt	Temperatura de referencia inicial (°C)
tref_1	Temperatura de referencia inicial (d.n.)
tref_1filt	Temperatura calentamiento inicial (°C)
TTid	Nombre del TT+
TTsn	Número serial del TT+
YYYYww	Año (YYYY) y semana del año (ww)

Banda (d.n)	Banda calibrada	Longitud de onda de la banda espectral (nm)
B1	B610	Banda espectral 610 nm
B2	B680	Banda espectral 680 nm
B3	B730	Banda espectral 730 nm
B4	B760	Banda espectral 760 nm
B5	B810	Banda espectral 810 nm
B6	B860	Banda espectral 860 nm
B7	B450	Banda espectral 450 nm
B8	B500	Banda espectral 500 nm
B9	B550	Banda espectral 550 nm
B10	B570	Banda espectral 570 nm
B11	B600	Banda espectral 600 nm
B12	B650	Banda espectral 650 nm

Nombre columna del TTCloud.csv	Ejemplo	Unidad o variable
cloudId	C0210043	ID del TTCloud
timeStamp	9/1/2022 0:00	Fecha y hora
Location.x	VM	Ubicación / sitio parcela
stringType.x	4B	Tipo cadena de datos 4B
accumulatedRecords	26981	Datos registrados
MCC	214	MCC del operador móvil
MNC	7	MNC del operador móvil
GSMregistration	1	Registro GSM
GSMfield	15	Campo GMS
Battery	4188	Nivel de batería (mV)
firmware	rel.5.1L	Versión del Firmware
Location.y	VM	Ubicación / sitio parcela
stringType.y	4C	Tipo cadena de datos 4C
Sensor1	-68	Señal RSSI del sensor 1
Sensor2	-63	Señal RSSI del sensor 2
Sensor3	-137	Señal RSSI del sensor 3
Sensor4	-56	Señal RSSI del sensor 4
SensorN		Señal RSSI del sensor N
Tipology	sensorSignal	Tipo de alerta
issue	1;3	Identificador del sensor

## Anexo 2. Código A: TT Check from Server.R

### TT\_Check\_from\_Server.R

Isabel Cristina Grisales S.

2022-08-30

TREE TALKERS CHECK FROM SERVER

BY: Isabel C. Grisales Sanchez

E-MAIL: [igriss00@estudiantes.unileon.es](mailto:igriss00@estudiantes.unileon.es)

DATE: August 2022

Developed with R 4.2.1. and RStudio (2022.07.1+554)

```
# Loading required packages
required_packages <- c("signal", "lubridate", "ggplot2", "dplyr",
                      "scales", "reshape2", "knitr", "rmarkdown", "prospe
ctr")
missing_packages <-
  required_packages[!(required_packages %in% installed.packages()[, "Packa
ge"])]
if (length(missing_packages) != 0)
  install.packages(missing_packages, repos = "http://cran.us.r-project.org
")
suppressWarnings(suppressPackageStartupMessages(
  noPrint <- lapply(required_packages, library, character.only = T)
))
```

```
#####
#### PART 1: DOWNLOAD DATABASES -----#####
#####
```

WAYS TO READ AND DOWNLOAD THE DATA:

Two options: (A) indicating the server links and parameters: TT\_Check\_from\_Server.R  
(B) reading \*.txt files that were downloaded directly from TT+: TT\_Check\_from\_TXT.R

```
####
## OPTION A: Indicating the servers Links ----
####
```

Inputs: *TTcloud* server links from Naturetalkers Outputs: One database appended to  
C:/TreeTalkerDB/Raw\_fullDB.csv

```
rm(list=ls())
```

```
# (A.1) Write/modify the following 3 inputs (examples):
```

```

# Server data links for each Cloud
server <- c('http://www.naturetalkers.altervista.org/C0210043/TTCloud.txt'
,
           'http://www.naturetalkers.altervista.org/C0210044/TTCloud.txt'
)
# Give a name to each cloud (site or location name)
site <- c('VM',
          'TV')
# Serial number of each cloud
cloud <- c('C0210043',
           'C0210044')

# (A.2) Run "DownloadTTDB_A" to save the function:

DownloadTTDB_A <- function() {
  # Read input parameters
  inpuParameters <- as.data.frame(cbind(server, site, cloud))
  # Create folder on C:/ for the outputs
  outdir <- file.path("C:", "TreeTalkerDB")
  outfile <- file.path("C:", "TreeTalkerDB", "Raw_fullDB.csv")
  if (file.exists(outfile))
    file.remove(outfile)
  if (isFALSE(dir.exists(outdir))) {
    dir.create(outdir)
  }
  outFileNames <-
    paste0(outdir,
           "/",
           paste0(inpuParameters$cloud, "_", inpuParameters$site),
           ".txt")
  # Download files
  for (i in 1:nrow(inpuParameters)) {
    download.file(inpuParameters$server[i], outFileNames[i])
    file_i <-
      read.table(
        outFileNames[i],
        sep = ";",
        col.names = paste0("V", 1:70),
        fill = T
      )
    file_i <- cbind(file_i, inpuParameters$site[i])
    write.table(
      file_i,
      outfile,
      sep = ";",
      col.names = F,
      append = T
    )
  }
  # Message
  cat("The",
      inpuParameters$server[i],
      "file has been appended to",
      outfile,
      "\n")
}

```



```

}
}

## (A.3) Run the function: ----
DownloadTTDB_A()

## The http://www.naturetalkers.altervista.org/C0210043/TTCloud.txt file h
as been appended to C:/TreeTalkerDB/Raw_fullDB.csv
## The http://www.naturetalkers.altervista.org/C0210044/TTCloud.txt file h
as been appended to C:/TreeTalkerDB/Raw_fullDB.csv

#####
#### PART 2: READ AND SPLIT DATABASES -----#####
#####

ReadServerData <- function() {
  dbname <- "C:/TreeTalkerDB/Raw_fullDB.csv"
  cat("Cleaning the",
      dbname,
      "file and organizing the data into the C:/TreeTalkerDB/Raw_TTDB.csv",
      "and the C:/TreeTalkerDB/Raw_CloudDB.csv files.",
      "\n"
  )
}

# reading servers data
serversData <- read.table(dbname, sep = ";")

# manage V2 column
cutV2 <- strsplit(as.character(serversData$V2), ",")
# select just the second element
ttId <- do.call(c, lapply(cutV2, function(x) {
  x[2]
}))

siteId <- serversData[, ncol(serversData)]

serversData <-
  cbind(siteId, ttId, serversData[, 4:(ncol(serversData) - 1)])

# We have 4 different types of data
dbString <- list("4B", "4C", "4D", "49", "55")
db4B <- serversData[serversData$V4 == "4B", ]
db4C <- serversData[serversData$V4 == "4C", ]
db55 <- serversData[serversData$V4 == "4D" | serversData$V4 == "55", ]
db49 <- serversData[serversData$V4 == "49", ]

## DB cloud
colnames(db4B) <- c(
  "Location",
  "cloudId",
  "stringType",
  "timeStamp",
  "accumulatedRecords",
  "pendingRecords",

```

```

    "MCC",
    "MNC",
    "GSMregistration",
    "GSMfield",
    "Battery",
    "firware"
  )
db4B <- db4B[, 1:12]

# remove the columns with empty timeslot otherwise R sees them as a record
db4C$V7 <- NULL

colnames(db4C) <-
  c(
    "Location",
    "cloudId",
    "stringType",
    "timeStamp",
    "TBLlocked",
    paste0("Sensor", 1:64)
  )

# merge cloud different strings (4B and 4C)
cloudDB <- merge(db4B, db4C, by = c('cloudId', 'timeStamp'))
cloudDB <- cloudDB[!duplicated(cloudDB), ]
cloudDB_filename <- file.path("C:", "TreeTalkerDB", "Raw_CloudDB.csv")

write.table(cloudDB,
            cloudDB_filename,
            sep = ";",
            row.names = F)

# DB TT
colnames(db55) <- c(
  "Location",
  "TTsn",
  "stringType",
  "timeStamp",
  "tref_0",
  "theat_0",
  "growth",
  "adc_bandgap",
  "nbits",
  "relativeHumidity",
  "temperature",
  "gzm",
  "gzsd",
  "gym",
  "gysd",
  "gxm",
  "gxsd",
  "tref_1",
  "theat_1",

```

```

    "stwc",
    "adc_vbat"
  )
db55 <- db55[, 1:21]
colnames(db49) <- c("Location",
                  "TTsn",
                  "stringType",
                  "timeStamp",
                  paste0("B", 1:12),
                  "intTime",
                  "gain")

db49 <- db49[, 1:18]

# merge TT different strings (4D and 49)
ttDB <- merge(db55, db49, by = c('TTsn', 'timeStamp'))
ttDB <- ttDB[!duplicated(ttDB), ]

TTsn <- as.character(ttDB$TTsn)
TTid <- paste0(ttDB$Location.x, "_TT", substring(ttDB$TTsn,7))

ttDB <-<- cbind(TTid, ttDB)

ttDB_filename <- file.path("C:", "TreeTalkerDB", "Raw_TTDB.csv")
write.table(ttDB, ttDB_filename, sep = ";", row.names = F)

#return(list(cloudDB, ttDB))
}

# Run the function: ----
ReadServerData()

## Cleaning the C:/TreeTalkerDB/Raw_fullDB.csv file and organizing the data
## into the C:/TreeTalkerDB/Raw_TTDB.csv and the C:/TreeTalkerDB/Raw_CloudDB.csv files.

#####
#### PART 3: CHECK CLOUDS ALERTS -----#####
#####

# Define variables for the function
dbCloud <- cloudDB          # Copy Cloud DB

# Date of interest for clouds, by default yesterday. Format: (YYYY-MM-DD)
whichDay = Sys.Date() - 1

checkCloud <- function(dbCloud, whichDay) {
  cat("Checking the status of clouds on", paste0(whichDay), "\n")
  cat("Check out the outputs on C:/TreeTalkerDB/TTResults as",
      paste0(whichDay, "cloud.csv"), "and", paste0(whichDay, "cloudAlert.csv"), "\n")

  # dbCloudToday
  dbCloud$timeStamp <-

```

```

lubridate::as_datetime(dbCloud$timeStamp) + 7200 # Convert an object to a date-time +2:00

# Filter data by day
dbCloudToday <- as.Date(dbCloud$timeStamp) == whichDay
dbCloudToday <- dbCloud[dbCloudToday,]
# remove the columns with empty timeslots otherwise R recognize them as a record
colNA <- apply(dbCloudToday, 2, function(x) {
  any(x != 0)
})
colNA[is.na(colNA)] <- FALSE
dbCloudToday <- dbCloudToday[, colNA]

# input parameters
minimumGSM <- 9 # condition_1 GSM field level ranges from 0 to 32
maximumNumberOfPendingItems <- 10 # condition_2 Pending data to be sent
minimumBattery <- 4000 # condition_3 Battery level ranges from 3300 to 4100 (mV)
minimumTalkerSignal <- -100 # condition_4 RSSI signal ranges from -137 to -53, -53 excellent signal

# check for factors
for (i in 16:ncol(dbCloudToday)) {
  dbCloudToday[, i] <- as.double(as.character(dbCloudToday[, i]))
}

# verifying the conditions
condition_1 <- dbCloudToday$GSMfield > minimumGSM
condition_2 <-
  dbCloudToday$pendingRecords < maximumNumberOfPendingItems # Empty when all the records are in the server
condition_3 <- dbCloudToday$Battery > minimumBattery
lowSignalSensorsList <- list()

for (i in 1:nrow(dbCloudToday)) {
  Sensors_row_i <-
    as.double(as.character(dbCloudToday[i, 16:ncol(dbCloudToday)]))
  lowSignalSensors <- Sensors_row_i < minimumTalkerSignal
  lowSignalSensors[is.na(lowSignalSensors)] <- FALSE
  whichSensorsLowSignal <-
    (1:(ncol(dbCloudToday) - 14))[lowSignalSensors]
  lowSignalSensorsList[[i]] <- whichSensorsLowSignal
}

# Extract the checking info

# 3.1. GSM field ----
signal <- dbCloudToday[condition_1 == FALSE,]
signal <- signal[, c(1, 2, 3, 9)]

# 3.2. Pending records ----
if (length(condition_2) > 0) {

```

```

    pendingRecords <- dbCloudToday[condition_2 == FALSE,]
    pendingRecords <- pendingRecords[, c(1, 2, 3, 6)]
  } else{
    pendingRecords <- dbCloudToday[FALSE, c(1, 2, 3, 6)]
  }

# 3.3. Battery Level ----
battery <- dbCloudToday[condition_3 == FALSE,]
battery <- cbind(battery[, c(1, 2, 3)], battery$Battery)

# 3.4. RSSI signal ----
lowSignalSensorsList <-
  lapply(lowSignalSensorsList, function(x) {
    paste(x, collapse = ";")
  })
lowSignalSensors <- unlist(lowSignalSensorsList)

lowSignalSensorsDb <- dbCloudToday[lowSignalSensors != "",]
lowSignalSensorsDb <- lowSignalSensorsDb[, c(1, 2, 3)]
lowSignalSensorsDb <-
  cbind(lowSignalSensorsDb, lowSignalSensors[lowSignalSensors != ""])
colnames(lowSignalSensorsDb)[4] <- "whichSensor"

colnames(signal)[4] <- "issue"
colnames(pendingRecords)[4] <- "issue"
colnames(battery)[4] <- "issue"
colnames(lowSignalSensorsDb)[4] <- "issue"
tipology <- c(
  rep("cloudSignal", nrow(signal)),
  rep("pendingRecords", nrow(pendingRecords)),
  rep("battery", nrow(battery)),
  rep("sensorSignal", nrow(lowSignalSensorsDb))
)

out <- rbind(signal, pendingRecords, battery, lowSignalSensorsDb)
out <- cbind(tipology, out$issue, out[, 1:3])
colnames(out)[2] <- "issue"

dbCloudToday <<- dbCloudToday

# Save outputs

out_dir <- file.path("C:", "TreeTalkerDB", "TTResults")
if (dir.exists(out_dir) == F)
  dir.create(out_dir)

write.table(out,
            file.path(out_dir, paste0(whichDay, "_CloudAlert.csv")),
            row.names = FALSE,
            sep = ";")
write.table(dbCloudToday,
            file.path(out_dir, paste0(whichDay, "_Cloud.csv")),
            row.names = FALSE,
            sep = ";")

```

```

    #return(List(dbCloudToday, out))
  }

# Run function:
checkCloud(dbCloud, whichDay)

## Checking the status of clouds on 2022-08-29
## Check out the outputs on C:/TreeTalkerDB/TTResults as 2022-08-29cloud.c
sv and 2022-08-29cloudAlert.csv

```

```

#####
#### PART 4: CHECK TTs -----#####
#####

```

Output: “\_TTDB.csv” in “C:/TreeTalkerDB/TTResults”

```

# Define variables for the function
dbTT <- ttDB # Copy TT's DB from previous step.
            # Filter by TT here if it is necessary

checkTT <- function(dbTT) {
  whichDay <- as.Date(Sys.Date())

  cat("Checking the status of TTs on", paste0(whichDay), "\n")
  cat("Check out the output as the full database on C:/TreeTalkerDB/TTResu
lts as",
      paste0(whichDay, "_TTBD.csv"), "\n")

  # 4.1. TimeStamp -----
  -----
  dbTT$timeStamp <- lubridate::as_datetime(dbTT$timeStamp) + 7200
  #+7200 is the offset GMT+02:00 in Spain

  # 4.2. Battery Level -----
  -----
  batteryLevel <- 2 * 1100 * dbTT$adc_vbat / dbTT$adc_bandgap / 1000
  batteryLevel <- round(batteryLevel, 2)
  dbTT <- cbind(dbTT, batteryLevel)

  # 4.3. Air Temperature -----
  -----
  dbTT$temperature <- dbTT$temperature / 10

  # Clean temperature data
  dbTT$temperature[dbTT$temperature < -20 | dbTT$temperature > 50] <- NA

  # Plot to visualize temperature data
  #plot(dbTT$temperature,
  #      type = "l",
  #      main = "Temperature",
  #      xlab = "Data in order",
  #      ylab = "Temperature (°C)")

```

#### # 4.4. Sap flux density (SFD) -----

*# Method adapted from Tomelleri, et al 2022*

*# From D.N. to temperature*

```
FromDigitalNumberToTemperature = function(x) {
  127.6 - (0.006045 * x) + (1.26E-7 * x ^ 2) - (1.15E-12 * x ^ 3)
}
dbTT$tref_0 <-
  FromDigitalNumberToTemperature(dbTT$tref_0)
dbTT$tref_1 <-
  FromDigitalNumberToTemperature(dbTT$tref_1)
dbTT$theat_0 <-
  FromDigitalNumberToTemperature(dbTT$theat_0)
dbTT$theat_1 <-
  FromDigitalNumberToTemperature(dbTT$theat_1)
```

*# Clean temperature data*

```
dbTT$tref_0[dbTT$tref_0 < -20 | dbTT$tref_0 > 50] <- NA
dbTT$tref_1[dbTT$tref_1 < -20 | dbTT$tref_1 > 50] <- NA
dbTT$theat_0[dbTT$theat_0 < -20 | dbTT$theat_0 > 50] <- NA
dbTT$theat_1[dbTT$theat_1 < -20 | dbTT$theat_1 > 50] <- NA
```

*# Apply a Savitzky-Golay smoothing*

```
library(signal)
```

**## For tref\_0**

```
tfilt <- dbTT$tref_0 # data to be filtered
```

```
ID <- unique(dbTT$TTid)
for (j in 1:length(ID)) {
  ts <- tfilt[dbTT$TTid == ID[j]]
  if (length(ts) < 11) {
    next()
  }
  m = 0
  p = 2
  w = 7
  ts_filt <- savitzkyGolay(ts, m, p, w)
  tfilt[dbTT$TTid == ID[j]] <- ts_filt[1:length(ts)]
}
dbTT$tref_0filt <- tfilt
```

**## For tref\_1**

```
tfilt <- dbTT$tref_1
```

```
ID <- unique(dbTT$TTid)
for (j in 1:length(ID)) {
  ts <- tfilt[dbTT$TTid == ID[j]]
  if (length(ts) < 11) {
    next()
  }
  m = 0
```

```

p = 2
w = 7
ts_filt <- savitzkyGolay(ts, m, p, w)
tfilt[dbTT$TTid == ID[j]] <- ts_filt[1:length(ts)]
}
dbTT$tref_1filt <- tfilt

## For theat_0
tfilt <- dbTT$theat_0

ID <- unique(dbTT$TTid)
for (j in 1:length(ID)) {
  ts <- tfilt[dbTT$TTid == ID[j]]
  if (length(ts) < 11) {
    next()
  }
  m = 0
  p = 2
  w = 7
  ts_filt <- savitzkyGolay(ts, m, p, w)
  tfilt[dbTT$TTid == ID[j]] <- ts_filt[1:length(ts)]
}
dbTT$theat_0filt <- tfilt

## For theat_1
tfilt <- dbTT$theat_1

ID <- unique(dbTT$TTid)
for (j in 1:length(ID)) {
  ts <- tfilt[dbTT$TTid == ID[j]]
  if (length(ts) < 11) {
    next()
  }
  m = 0
  p = 2
  w = 7
  ts_filt <- savitzkyGolay(ts, m, p, w)
  tfilt[dbTT$TTid == ID[j]] <- ts_filt[1:length(ts)]
}
dbTT$theat_1filt <- tfilt

# Plot to visualize one temperature data (tref_0_filt in this case)
#plot(dbTT$tref_0filt,
#  type = "l",
#  main = "tref_0 filtered with Savitzky-Golay",
#  xlab = "Data in order",
#  ylab = "tref_0filt (°C)")

# SFD, method from the manual, SFD (Ldm-2h-1)

# Two options to calculate deltaT:
# delta T1
DeltaT_heat <- dbTT$theat_1filt - dbTT$theat_0filt
# delta T2

```



```

DeltaT_heat2 <- (dbTT$theat_1filt - dbTT$tref_1filt) - (dbTT$theat_0filt
- dbTT$tref_0filt)

# Get the Max value from each TT PER DAY

# create a column by day
dataDay <- paste0(month(dbTT$timeStamp), day(dbTT$timeStamp))

# Create a new df for SFD
TTid <- dbTT$TTid
dbSFD <- as.data.frame(cbind(TTid, dataDay, DeltaT_heat, DeltaT_heat2))

# Aggregate data by TT and by day = obtain the max value per day and TT
delta_temp <- aggregate(DeltaT_heat ~ TTid + dataDay, data = dbSFD, FUN
= max, na.rm = TRUE)

dbSFD <- left_join(dbSFD, delta_temp, by = c ("TTid", "dataDay"))

DeltaTMax <- dbSFD$DeltaT_heat.y

# SFD
sapFluxDensity <-
  4.79 * ((as.double(DeltaTMax) - DeltaT_heat) / DeltaT_heat)

#sapFluxDensity[sapFluxDensity < 0] <- NA

dbTT <- cbind(dbTT, sapFluxDensity)

# Plot for SFD
#plot(dbTT$sapFluxDensity,
# type = "l",
# main = "Sap Flux Density",
# xlab = "Data in order",
# ylab = "SFD (ldm-2h-1)")

# 4.5. Weekly Radial Growth -----
-----

# Packages:
#library(lubridate)
#library(dplyr)

# Conversion range according to the manual (September 2020)
# First data filter for [D.N.]
tgrow <- dbTT$growth
tgrow[tgrow > 85000] <- NA
tgrow[tgrow < 30000] <- NA

# Plot to visualize growth sensor d.n.
#plot(tgrow,
# type = "l",
# main = "Growth sensor d.n.",
# xlab = "Data in order",

```

```

#   ylab = "Growth sensor d.n.")

# Distance from sensor to the bark (cm from the TT+ manual, *10 to mm units)
dbTT$DistGrowth <-
  ((237908.4541 + (-1.1171 * tgrow)) / (199.4330 + tgrow))*10

# Create a time index for the temporal averaging (YYYYww)
time_index <- paste0(year(dbTT$timeStamp), week(dbTT$timeStamp))
dbTT$YYYYww <- time_index

# Method from paper: Tomelleri et al, 2022
ID <- unique(dbTT$TTid)
for (j in 1:length(ID)) {
  myDendro_data_L0 <-
    data.frame(dbTT$TTid, dbTT$timeStamp, dbTT$YYYYww, dbTT$DistGrowth)
  colnames(myDendro_data_L0) <- c("series", "ts", "YYYYww", "dendrometer")
  # Subset dataset for TTids
  myDendro_data_L0 <- myDendro_data_L0 %>% dplyr::filter(series == ID[j])
  if (length(na.omit(myDendro_data_L0$dendrometer))<100){next}

  #remove outliers
  t_05 <- quantile(myDendro_data_L0$dendrometer, p=0.05, na.rm=T)
  t_95 <- quantile(myDendro_data_L0$dendrometer, p=0.95, na.rm=T)
  myDendro_data_L0$dendrometer[myDendro_data_L0$dendrometer<t_05] <- NA
  myDendro_data_L0$dendrometer[myDendro_data_L0$dendrometer>t_95] <- NA

  #create a data.frame with Level 1 data
  myDendro_data_L1 <- myDendro_data_L0

  #temporal averaging
  myDendro_data_weekly <-
    aggregate(dendrometer ~ YYYYww, myDendro_data_L1, median)

  #normalize the radial growth with the initial sensor distance
  myDendro_data_weekly$dendrometer <-
    (max(myDendro_data_weekly$dendrometer) - myDendro_data_weekly$dendrometer)

  #create a data.frame with Level 2 data
  myDendro_data_L2 <- subset(myDendro_data_L1, select = -dendrometer)

  #merge averaged sharp data with the specific TreeTalker data.frame
  myDendro_data_L2 <- merge(myDendro_data_L2, myDendro_data_weekly, all.x=T)

  #insert processed data into the dbTT data.frame
  dbTT$dendrometer[dbTT$TTid == ID[j]] <- myDendro_data_L2$dendrometer
}

# plot weekly radial growth (mm)

```

```
#plot(dbTT$dendrometer,
#   type = "l",
#   main = "Weekly Radial Growth (mm)",
#   xlab = "Data in order",
#   ylab = "Weekly Radial Growth (mm)")
```

```
# 4.6. Tree Movement: angle of oscilation (°) -----
```

```
dbTT$gzm <- as.numeric(dbTT$gzm)
gz <- abs(dbTT$gzm)
gx <- abs(dbTT$gxm)
gy <- abs(dbTT$gym)
```

```
## Yaw angle
```

```
#gxy <- sqrt(gx^2 + gy^2)
#TreeMovement <- (atan2(gxy,gz)*180)/(pi)
```

```
## Pitch angle
```

```
gyz <- sqrt(gy^2 + gz^2)
TreeMovement <- (atan2(gx,gyz)*180)/(pi)
```

```
dbTT <- cbind(dbTT, TreeMovement)
```

```
# 4.7. Spectral Light through the canopy -----
```

```
# To convert the digital numbers (d.n.) into corrected spectral bands
```

```
#Copy bands
```

```
#Near infrared
```

```
dbTT$B610 <- as.numeric(dbTT$B1)
dbTT$B680 <- as.numeric(dbTT$B2)
dbTT$B730 <- as.numeric(dbTT$B3)
dbTT$B760 <- as.numeric(dbTT$B4)
dbTT$B810 <- as.numeric(dbTT$B5)
dbTT$B860 <- as.numeric(dbTT$B6)
```

```
#Visible Light spectrum
```

```
dbTT$B450 <- as.numeric(dbTT$B7)
dbTT$B500 <- as.numeric(dbTT$B8)
dbTT$B550 <- as.numeric(dbTT$B9)
dbTT$B570 <- as.numeric(dbTT$B10)
dbTT$B600 <- as.numeric(dbTT$B11)
dbTT$B650 <- as.numeric(dbTT$B12)
```

```
#Delete high values
```

```
dbTT$B610[dbTT$B610 == 0 | dbTT$B610 > 65000 ] <- NA
dbTT$B680[dbTT$B680 == 0 | dbTT$B680 > 65000 ] <- NA
dbTT$B730[dbTT$B730 == 0 | dbTT$B730 > 65000 ] <- NA
dbTT$B760[dbTT$B760 == 0 | dbTT$B760 > 65000 ] <- NA
dbTT$B810[dbTT$B810 == 0 | dbTT$B810 > 65000 ] <- NA
dbTT$B450[dbTT$B450 == 0 | dbTT$B450 > 65000 ] <- NA
dbTT$B500[dbTT$B500 == 0 | dbTT$B500 > 65000 ] <- NA
```

```
dbTT$B550[dbTT$B550 == 0 | dbTT$B550 > 65000 ] <- NA
dbTT$B570[dbTT$B570 == 0 | dbTT$B570 > 65000 ] <- NA
dbTT$B600[dbTT$B600 == 0 | dbTT$B600 > 65000 ] <- NA
dbTT$B650[dbTT$B650 == 0 | dbTT$B650 > 65000 ] <- NA
```

*#Convert the digital numbers (d.n.) into corrected spectral bands*

```
dbTT$B610 <- -312.45+(1.6699 *dbTT$B610)
dbTT$B680 <- -561.56+(1.5199 *dbTT$B680)
dbTT$B730 <- -1511.2+(1.6209 *dbTT$B730)
dbTT$B760 <- -1012.5+(1.4549 *dbTT$B760)
dbTT$B810 <- 91.58+(0.8414 *dbTT$B810)
dbTT$B860 <- 334.88+(0.531 *dbTT$B860)
dbTT$B450 <- -212.62+(0.4562*dbTT$B450)
dbTT$B500 <- -232.13+(0.6257 *dbTT$B500)
dbTT$B550 <- -842.1+(1.0546 *dbTT$B550)
dbTT$B570 <- -666.72+(1.0462 *dbTT$B570)
dbTT$B600 <- -328.08+(0.8654 *dbTT$B600)
dbTT$B650 <- 202.77+(0.7829 *dbTT$B650)
```

*# 4.8. Stem Water Content -----*

*#' Relative Saturation Index, adapted from Asgharina et al. (2022)*

*# Filter stwc if necessary (capacity 3MHz in this case)*

*# See plot*

```
#plot(dbTT$stwc,
#      type = "l",
#      main = "Stem water content (stwc)",
#      xlab = "Data in order",
#      ylab = "ECf_stem sensor, stwc [d.n.]")
```

*# Get the min value of the stwc sensor from each TT+*

```
stwc_min <-
  aggregate(dbTT$stwc, list(as.character(dbTT$TTid)), min, na.rm=TRUE)
```

*# Ceate a vector with the Min value of stwc per TT+*

```
stwcMin <-
  as.numeric(stwc_min$x)[match(as.character(dbTT$TTid),
                              as.character(stwc_min$Group.1))]
```

*# Obtain temperature (filtered previously) with associated to the Min stwc*

```
Tmin <- dbTT[dbTT$stwc == stwcMin,] # stwc_min$x replaced for stwcMin
```

*stwcT\_Min <-*

```
  as.numeric(Tmin$tref_0filt)[match(as.character(dbTT$TTid),
                                    as.character(Tmin$TTid))]
```

```
ECfsat <- stwcMin
```

```
Tsat <- stwcT_Min
```

*# m, the slope of temperature sensitivity for "water"\**

```

# *in this case for now, it should be the tree species
m = -81

# Expected frequency signal
ECf_adj <- m * (dbTT$tref_0filt - Tsat) + ECfsat

# Relative index of the stem saturation (%)
RelStemSat <- (1-((dbTT$stwc - ECf_adj)/ECf_adj))*100

# Plot for the Relative Stem Saturation (%)
#plot(RelStemSat,
#      type = "l",
#      main = "Relative Stem Saturation",
#      xlab = "Data in order",
#      ylab = "Relative Stem Saturation (%)")

#Join results to the main DF
dbTT <- cbind(dbTT, RelStemSat)

## End of variables calculation -----
-----

# Select columns for results
dbTT_Results <- dbTT[,c(
  "TTid", "TTsn", "timeStamp", "Location.x", "relativeHumidity",
  "temperature", "batteryLevel", "sapFluxDensity",
  "dendrometer", "TreeMovement", "RelStemSat",
  "B610", "B680", "B730", "B760", "B810", "B860",
  "B450", "B500", "B550", "B570", "B600", "B650")]

#' Apply a date filter here if prefer a df filtered by date

# Return as a variable in the global environment
dbTT <<- dbTT # db with some step by step calculations
dbTT_Results <<- dbTT_Results # db with the main results

# Write outputs
out_dir <- file.path("C:", "TreeTalkerDB", "TTResults")
if (dir.exists(out_dir) == F)
  dir.create(out_dir)

write.table(dbTT_Results,
            file.path(out_dir, paste0(whichDay, "_TTBD.csv")),
            row.names = FALSE,
            sep = ";")

#return(list(dbTT))
}

## Run the function: ----
checkTT(dbTT)

```

```
## Checking the status of TTs on 2022-08-30
## Check out the output as the full database on C:/TreeTalkerDB/TTResults
as 2022-08-30_TTBD.csv
```

```
#####
#### PART 5: TTAlert -----
-----#####
#####
```

Creates reports to alert anomalies from TT data

```
# Copy complete data base history
dbTT_alert <- dbTT
```

Activate one of the next 2 parameters options if data results is desired for one day:

```
#Day_alert <- as.Date("2022-08-10") # Day to check the alerts
#Day_alert <- as.Date(Sys.Date()-1) # yesterday
```

And activate next 2 lines to subset data frame by a date (Day\_alert)

```
#data_alert <- as.Date(dbTT_alert$timeStamp) == Day_alert
#dbTT_alert <- dbTT_alert[data_alert,]
```

Entry dates for next lines if data results is desired for a range of dates, format: "YYYY-MM-DD"

```
#Date0 <- as.Date("2022-04-07") # Start
#Date1 <- as.Date("2022-08-17") # End
#dbTT_alert <- dbTT_alert[dbTT_alert$timeStamp >= Date0 & dbTT_alert$timeStamp <= Date1 + 1,]
```

```
TTAlert <- function(dbTT_alert) {
  whichDay <- as.Date(Sys.Date())

  cat("Checking the status of TTs on", paste0(whichDay), "\n")

  # Input parameters, when these conditions are TRUE values are reported as alerts

  minimumBattery <- 3.5 # Battery level ranges should be > 3.5 (mV) to work property
  maximumAngle <- 45 # Risk of a falling tree (angle)
  minimumSFD <- 1.0 # Low weekly sap flux density (ldm-2h-1)
  minimumGrowth <- 0.1 # Low weekly radial growth (mm)

  # 5.1. Low battery level alert ----

  batteryAlert <- dbTT_alert[dbTT_alert$batteryLevel <= minimumBattery, ]
  batteryAlert <- batteryAlert[, c("TTid", "timeStamp", "batteryLevel")]

  if (nrow(batteryAlert) > 0) {
```

```

batteryAlert2 <- aggregate(timeStamp ~ TTid, data= batteryAlert, FUN=m
ax)

batteryAlert2 <- left_join(batteryAlert2, batteryAlert,
                           by = c("TTid", "timeStamp"))

batteryAlert <- batteryAlert2
}

# 5.2. Tree movement alert ----

angleAlert <- dbTT_alert[dbTT_alert$TreeMovement >= maximumAngle, ]
angleAlert <- angleAlert[, c("TTid", "timeStamp", "TreeMovement")]

if (nrow(angleAlert) > 0) {
  angleAlert <-
    aggregate(angleAlert[, 2:3], list(as.character(angleAlert$TTid)), mi
n)
  colnames(angleAlert)[1] <- "TTid"
  angleAlert$TreeMovement <- round(angleAlert$TreeMovement)
  colnames(angleAlert)[3] <- "TreeAngle"
}

# 5.3. Sap Flux Density (SFD) alert ----

sap_alert <- dbTT_alert[, c("TTid", "timeStamp", "YYYYww", "sapFluxDensi
ty")]
sap_alert$sapFluxDensity <- round(sap_alert$sapFluxDensity, 4)

# Maximum weekly SFD
WeekMaxSFD <- aggregate(sapFluxDensity ~ TTid + YYYYww, data = sap_alert
, FUN = max, na.rm = TRUE)

# Alert df
SFDAlert <- WeekMaxSFD[WeekMaxSFD$sapFluxDensity <= minimumSFD, ]

if (nrow(SFDAlert) > 0) {
  SFDAlert <- left_join(SFDAlert, sap_alert, by = c ("TTid", "YYYYww", "s
apFluxDensity"))
}

# 5.4. Radial growth alert ----

growth_alert <- dbTT_alert[, c("TTid", "timeStamp", "YYYYww", "dendromet
er")]
growth_alert$dendrometer <- round(growth_alert$dendrometer, 2)

# Maximum weekly growth
WeekMaxGrowth <- aggregate(dendrometer ~ TTid + YYYYww, data = growth_al
ert, FUN = max, na.rm = TRUE)

```

```

# Package
#Library(dplyr)
WeekMaxGrowth <- WeekMaxGrowth %>%
  group_by(TTid) %>%
  mutate(deltaGrowth = dendrometer - lag(dendrometer, default = 0))

# Alert df
growthAlert <- WeekMaxGrowth[WeekMaxGrowth$deltaGrowth <= minimumGrowth,
]

# Include week after installation of TTs on trees, otherwise full report
growthAlert <- growthAlert[growthAlert$YYYYww >= 202216, ]

# Sort df by TT, not by time
if (nrow(SFDAlert) > 0) {
  growthAlert <- growthAlert[order(growthAlert$TTid),]
}

# Save outputs ----

out_dir <- file.path("C:", "TreeTalkerDB", "TTResults")
if (dir.exists(out_dir) == F)
  dir.create(out_dir)

write.table(batteryAlert,
            file.path(out_dir, paste0(whichDay, "_BatteryTTAlert.csv")),
            row.names = FALSE,
            sep = ";")

write.table(angleAlert,
            file.path(out_dir, paste0(whichDay, "_AngleTTAlert.csv")),
            row.names = FALSE,
            sep = ";")

write.table(SFDAlert,
            file.path(out_dir, paste0(whichDay, "_SapFDTTAlert.csv")),
            row.names = FALSE,
            sep = ";")

write.table(growthAlert,
            file.path(out_dir, paste0(whichDay, "_GrowthTTAlert.csv")),
            row.names = FALSE,
            sep = ";")

batteryAlert <<- batteryAlert
angleAlert <<- angleAlert
SFDAlert <<- SFDAlert
growthAlert <<- growthAlert
}

# Run function:
TTAlert(dbTT_alert)
## Checking the status of TTs on 2022-08-30

```



```
#####
#### PART 6: PLOTS -----
-----#####
#####
```

*# Define db for the function*

Graphics for complete data (leave by default)

```
TTdb_today <- dbTT # Copy TT's DB from previous step
```

Filter by one TT here if it is necessary. Examples:

```
#unique(dbTT$TTsn)
#TTdb_today <- dbTT[dbTT$TTsn == "621B0393",]
```

Two or more TTs at the same time

```
#TTdb_today <- dbTT[dbTT$TTsn == "621B0396" | dbTT$TTsn == "621B0397",]
```

*# Entry dates for graphics, format: "YYYY-MM-DD"*

```
Date0 <- as.Date("2022-05-01") # Start
```

```
Date1 <- as.Date("2022-05-08") # End
```

plotting is recommended for 1 week or 2 for all variables, disable line that starts with 'scale\_x' (\*\*) on each variable when plotting > 2 weeks. If the plotting fails, be sure that the timestamp from TTs are on time.

```
Plot <- function(TTdb_today, Date0, Date1) {

  # Path for outputs
  outPathReport <-
    file.path(file.path("C:", "TreeTalkerDB"), "TTPlots")
  if (dir.exists(outPathReport) == F)
    dir.create(outPathReport)

  # Message
  cat("Calculating and saving some graphics in", outPathReport, "\n")

  # Path for outputs 2 (plots)
  TTdirectory <- file.path("C:/", "TreeTalkerDB")
  outPath <-
    file.path(TTdirectory, "TTPlots", paste0(Date0, "_to_", Date1, "_plots"))
  if (dir.exists(outPath) == F)
    dir.create(outPath)

  # data reading
  whichDay = Sys.Date()

  # Subset data frame with dates of interest
  data_new <-
    TTdb_today[TTdb_today$timeStamp >= Date0 & TTdb_today$timeStamp <= Date1 + 1,]
```

```
# Packages
#library(ggplot2)
#library(lubridate)
#library(scales)
#library(dplyr)
#library(reshape2)

theme_set(theme_bw())

# 6.1. Air Temperature and air humidity -----

ggplot(data_new, aes(x=timeStamp)) +
  geom_line(aes(y=temperature, col=TTid), size=0.8) +
  labs(title="Air Temperature",
        subtitle = paste0("From: ", Date0, " to ", Date1),
        y="Temperature [°C]",
        x="Date") +
  scale_x_datetime(date_breaks = "12 hours", labels= date_format("%b %d-%H:%M")) + #**
  theme(axis.text.x = element_text(angle = 25, vjust=1.0, hjust=1.0, size = 8))

ggsave(file.path(outPath, "AirTemperature.PNG"),
        width = 8, height = 5)

ggplot(data_new, aes(x=timeStamp)) +
  geom_line(aes(y=relativeHumidity, col=TTid), size=0.8) +
  labs(title="Air Humidity",
        subtitle = paste0("From: ", Date0, " to ", Date1),
        y="Relative humidity (%)",
        x="Date") +
  scale_x_datetime(date_breaks = "12 hours", labels= date_format("%b %d-%H:%M")) + #**
  theme(axis.text.x = element_text(angle = 25, vjust=1.0, hjust=1.0, size = 8))

ggsave(file.path(outPath, "AirHumidity.PNG"),
        width = 8, height = 5)

# 6.2. Battery Level -----

ggplot(data_new, aes(x=timeStamp)) +
  geom_line(aes(y=batteryLevel, col=TTid), size=0.8) +
  geom_hline(yintercept = 3.5, color="red", size=1.0) +
  labs(title="TT's Battery Voltage",
        subtitle = paste0("From: ", Date0, " to ", Date1),
        y="Battery Voltage [mV]",
        x="Date") +
  scale_x_datetime(date_breaks = "12 hours", labels= date_format("%b %d-%H:%M")) + #**
  theme(axis.text.x = element_text(angle = 25, vjust=1.0, hjust=1.0, size = 8))
```

```

ggsave(file.path(outPath, "BatteryTT.PNG"),
        width = 8, height = 5)

# 6.3. Sap flux density (SFD) -----

ggplot(data_new, aes(x=timeStamp)) +
  geom_line(aes(y=sapFluxDensity, col=TTid), size=0.8) +
  labs(title="Sap Flux Density",
        subtitle = paste0("From: ", Date0, " to ", Date1),
        y="SFD [ldm-2h-1]",
        x="Date") +
  scale_x_datetime(date_breaks = "12 hours", labels= date_format("%b %d-
%H:%M")) + #**
  theme(axis.text.x = element_text(angle = 25, vjust=1.0, hjust=1.0, siz
e = 8))

ggsave(file.path(outPath, "sapFluxDensity.PNG"),
        width = 8, height = 5)

# 6.4. Radial growth -----

ggplot(data_new, aes(x=timeStamp)) +
  geom_line(aes(y=dendrometer, col=TTid), size=0.8) +
  labs(title="Weekly Radial Growth",
        subtitle = paste0("From: ", Date0, " to ", Date1),
        y="Radial Growth [mm]",
        x="Date") +
  scale_x_datetime(date_breaks = "1 week", labels= date_format("%b %d-%H
:%M")) + #**
  theme(axis.text.x = element_text(angle = 25, vjust=1.0, hjust=1.0, siz
e = 8))

ggsave(file.path(outPath, "RadialGrowth.PNG"),
        width = 8, height = 5)

# 6.5. Stem water content -----

ggplot(data_new, aes(x=timeStamp)) +
  geom_line(aes(y=RelStemSat, col=TTid), size=0.8) +
  labs(title="Relative Stem Saturation",
        subtitle = paste0("From: ", Date0, " to ", Date1),
        y="Relative stem saturation (%)",
        x="Date") +
  scale_x_datetime(date_breaks = "12 hours", labels= date_format("%b %d-
%H:%M")) + #**
  theme(axis.text.x = element_text(angle = 25, vjust=1.0, hjust=1.0, siz
e = 8))

ggsave(file.path(outPath, "RelStemSaturation.PNG"),
        width = 8, height = 5)

# 6.6. Light trough the canopy, spectral bands -----

# Select data only from 10AM to 2PM

```

```

data_new$hour <- paste0(hour(data_new$timeStamp))
dbspectral <- filter(data_new, hour > 10, hour <= 14 )

#Library(dplyr)
datalist = list() #initialize the list
ID <- unique(dbspectral$TTid)
for (j in 1:length(ID)) {

  # Subset dataset for TTids
  spectrom_L0 <- dbspectral %>% dplyr::filter(TTid == ID[j])

  if (length(spectrom_L0$timeStamp)<10){next}

  #aggregate to daily
  spectrom_L0$Day <- floor_date(spectrom_L0$timeStamp, "day")

  spectrom_L1 <- spectrom_L0 %>%
    group_by(Day) %>%
    summarize(L450 = median(B450, na.rm = TRUE),
              L500 = median(B500, na.rm = TRUE),
              L550 = median(B550, na.rm = TRUE),
              L570 = median(B570, na.rm = TRUE),
              L600 = median(B600, na.rm = TRUE),
              L650 = median(B650, na.rm = TRUE),
              L610 = median(B610, na.rm = TRUE),
              L680 = median(B680, na.rm = TRUE),
              L730 = median(B730, na.rm = TRUE),
              L760 = median(B760, na.rm = TRUE),
              L810 = median(B810, na.rm = TRUE),
              L860 = median(B860, na.rm = TRUE)
    )
  spectrom_L1$TTid <- rep(ID[j], length(spectrom_L1$Day))

  datalist[[j]] <- spectrom_L1 # add it to your list
}

# Average of data between hours, per day
spectrom_L1_all <- dplyr::bind_rows(datalist)

# Average between days, bands and TTid
spectrom_L2_all <- spectrom_L1_all %>%
  group_by(TTid) %>%
  summarise_at(vars("L450", "L500", "L550", "L570", "L600",
                   "L650", "L610", "L680", "L730", "L760",
                   "L810", "L860"), mean)

# Change column names for wavelength
Wavelength <- c(450, 500, 550, 570, 600, 650, 610, 680, 730, 760, 810, 860)
colnames(spectrom_L2_all) <- c("TTid", Wavelength)

# Reshape df for plot
#Library(reshape2)
spectral <- melt(spectrom_L2_all, id.vars = "TTid")

```

```

ggplot(spectral, aes(x=variable)) +
  geom_line(aes(y=value, col=TTid, group=TTid), size=0.8) +
  labs(title="Daily average of spectral bands data",
        subtitle = paste0("From: ", Date0, " to ", Date1),
        y="Calibrated spectral bands",
        x="Wavelength (nm)")

ggsave(file.path(outPath, "SpectralBands.PNG"),
        width = 8, height = 5)
}

## Run the function: ----
suppressWarnings(Plot(TTdb_today, Date0, Date1))

## Calculating and saving some graphics in C:/TreeTalkerDB/TTPlots

#####
###
# PART 7: REPORT WITH RMARKDOWN -----
#####
###

```

Input the folder path with the "TTReport.Rmd" file downloaded to create the report with RMarkdown (TTReport.pdf)

```

render_path <- file.path("C:/Users/Isabel/Downloads", "TTReport.Rmd")

# Run next lines
suppressWarnings(
  rmarkdown::render(
    render_path,
    output_format = c("html_document", "pdf_document")[i],
    output_file = paste0("Report", c(".html", ".pdf"))[i],
    output_dir = file.path("C:", "TreeTalkerDB")
  )
)

##
## Output created: C:/TreeTalkerDB/Report.pdf

```

END OF SCRIPT

### Anexo 3. Código B: TT\_Check\_from\_TXT.R

## TT\_Check\_from\_TXT.R

Isabel Cristina Grisales S.

2022-08-30

TREE TALKERS CHECK FROM TXT

BY: Isabel C. Grisales Sanchez

E-MAIL: [igriss00@estudiantes.unileon.es](mailto:igriss00@estudiantes.unileon.es)

DATE: August, 2022

Developed with R 4.2.1. and RStudio (2022.07.1+554)

```
# Loading required packages
required_packages <- c("signal", "lubridate", "ggplot2", "dplyr",
                      "scales", "reshape2", "knitr", "rmarkdown", "prospe
ctr")
missing_packages <-
  required_packages[!(required_packages %in% installed.packages()[, "Packa
ge"])]
if (length(missing_packages) != 0)
  install.packages(missing_packages, repos = "http://cran.us.r-project.org
")
suppressWarnings(suppressPackageStartupMessages(
  noPrint <- lapply(required_packages, library, character.only = T)
))
```

```
#####
#### PART 1: DOWNLOAD DATABASES -----#####
#####
```

WAYS TO READ AND DOWNLOAD THE DATA:

Two options: (A) indicating the server links and parameters: TT\_Check\_from\_Server.R

(B) reading \*.txt files that were downloaded directly from TT+: TT\_Check\_from\_TXT.R

```
####
## OPTION B: Having the data on *.txt files from TT+ -----
####
```

Inputs: TXT files downloaded directly from TT+ Outputs: One database appended to  
C:/TreeTalkerDB/Raw\_fullDB.csv

```
rm(list=ls())
```

```
# (B.1) Write/modify the input directory, folder with TXT files from TTs (
example):
```

```
setwd("C:/Users/Isabel/Documents/TFM - Tree Talkers/Códigos de R/DatosTFM")
)
```

*# (B.2) Run "DownloadTTDB\_B" to save the function:*

```
DownloadTTDB_B <- function() {
  # Read input parameters
  filelist <- list.files(pattern = ".*.txt")
  # Path to new folder and files
  outdir <- file.path("C:", "TreeTalkerDB")
  outfile <- file.path("C:", "TreeTalkerDB", "Raw_fullDB.csv")
  if (file.exists(outfile))
    file.remove(outfile)
  if (isFALSE(dir.exists(outdir))) {
    dir.create(outdir)
  }
  FileNames <- paste0(getwd(), "/", filelist)
  for (i in 1:length(FileNames)) {
    file_i <-
      read.table(
        FileNames[i],
        sep = ";",
        col.names = paste0("V", 1:70),
        fill = T
      )
    file_i <- cbind(file_i, filelist[i])
    write.table(
      file_i,
      outfile,
      sep = ";",
      col.names = F,
      append = T
    )
    cat("The file",
        filelist[i],
        "has been appended to",
        outfile,
        "\n")
  }
}
```

*## (B.3) Run the function: ----*

```
DownloadTTDB_B()
```

```
## The file VM_PIE15.txt has been appended to C:/TreeTalkerDB/Raw_fullDB.csv
## The file VM_PIE20.txt has been appended to C:/TreeTalkerDB/Raw_fullDB.csv
## The file VM_PIE22.txt has been appended to C:/TreeTalkerDB/Raw_fullDB.csv
```

```
#####
#### PART 2: READ AND SPLIT DATABASES -----#####
#####

ReadTXTData <- function() {
  dbname <- "C:/TreeTalkerDB/Raw_fullDB.csv"
  cat("Cleaning the",
      dbname,
      "file and organizing the data into the C:/TreeTalkerDB/ttDB.csv file."
  ,
    "\n"
  )

  # reading servers data
  serversData <- read.table(dbname, sep = ";")

  # manage V2 column
  cutV2 <- strsplit(as.character(serversData$V2), ",")

  # select just the second element
  ttId <- do.call(c, lapply(cutV2, function(x) {
    x[1]
  })))

  siteId <- serversData[, ncol(serversData)]

  serversData <-
    cbind(siteId, ttId, serversData[, 4:(ncol(serversData) - 1)])

  # We have 2 different types of data from TT+ (4D and 55 are the same)
  dbString <- list("4D", "49", "55")
  db55 <- serversData[serversData$V4 == "4D" | serversData$V4 == "55", ]
  db49 <- serversData[serversData$V4 == "49", ]

  # DB TT
  colnames(db55) <- c(
    "Location",
    "TTsn",
    "stringType",
    "timeStamp",
    "tref_0",
    "theat_0",
    "growth",
    "adc_bandgap",
    "nbits",
    "relativeHumidity",
    "temperature",
    "gzm",
    "gzsd",
    "gym",
    "gysd",
    "gxm",
    "gxsd",
    "tref_1",
  )
}
```



```

    "theat_1",
    "stwc",
    "adc_vbat"
  )
  db55 <- db55[, 1:21]

  colnames(db49) <- c("Location",
                    "TTsn",
                    "stringType",
                    "timeStamp",
                    paste0("B", 1:12),
                    "intTime",
                    "gain")

  db49 <- db49[, 1:18]

  # merge TT different strings (4D/55 and 49)
  ttDB <- cbind(db55, db49)
  ttDB <- ttDB[,c(2,4,1,3,5:22,24,26:39)]

  # add the TTid to the db
  TTsn <- as.character(ttDB$TTsn)
  TTid <- substr(ttDB$Location,1,nchar(ttDB$Location)-4)

  ttDB <-< cbind(TTid, ttDB)

  ttDB_filename <- file.path("C:", "TreeTalkerDB", "Raw_TTDB.csv")
  write.table(ttDB, ttDB_filename, sep = ";", row.names = F)

  #return(List(ttDB))
}

# Run the function: ----
ReadTXTData()

## Cleaning the C:/TreeTalkerDB/Raw_fullDB.csv file and organizing the data
into the C:/TreeTalkerDB/ttDB.csv file.

#####
#### PART 3: CHECK CLOUDS ALERTS -----#####
#####

# There are no TTClouds in these txt files

#####
#### PART 4: CHECK TTs -----#####
#####

```

Output: "\_TTDB.csv" in "C:/TreeTalkerDB/TTResults"

```

# Define variables for the function
dbTT <- ttDB # Copy TT's DB from previous step.
            # Filter by TTid here or at the end if it is necessary

```

```

checkTT <- function(dbTT) {
  whichDay <- as.Date(Sys.Date())

  cat("Checking the status of TTs on", paste0(whichDay), "\n")
  cat("Check out the output as the full database in C:/TreeTalkerDB/TTResults as",
      paste0(whichDay, "_TTDB.csv"), "\n")

  # 4.1. TimeStamp -----
  dbTT$timeStamp <- lubridate::as_datetime(dbTT$timeStamp) +7200
  #+7200 is the offset GMT+02:00 in Spain

  # 4.2. Battery Level -----
  batteryLevel <- 2 * 1100 * dbTT$adc_vbat / dbTT$adc_bandgap / 1000
  batteryLevel <- round(batteryLevel, 2)
  dbTT <- cbind(dbTT, batteryLevel)

  # 4.3. Air Temperature -----
  dbTT$temperature <- dbTT$temperature / 10

  # clean temperature data
  dbTT$temperature[dbTT$temperature < -20 | dbTT$temperature > 50] <- NA

  # Plot to visualize temperature data
  #plot(dbTT$temperature,
  #     type = "l",
  #     main = "Temperature",
  #     xlab = "Data in order",
  #     ylab = "Temperature (°C)")

  # 4.4. Sap flux density (SFD) -----
  # Method adapted from Tomelleri, et al 2022

  # From D.N. to temperature
  FromDigitalNumberToTemperature = function(x) {
    127.6 - (0.006045 * x) + (1.26E-7 * x ^ 2) - (1.15E-12 * x ^ 3)
  }
  dbTT$tref_0 <-
    FromDigitalNumberToTemperature(dbTT$tref_0)
  dbTT$tref_1 <-
    FromDigitalNumberToTemperature(dbTT$tref_1)
  dbTT$theat_0 <-
    FromDigitalNumberToTemperature(dbTT$theat_0)
  dbTT$theat_1 <-
    FromDigitalNumberToTemperature(dbTT$theat_1)

  # clean temperature data
  dbTT$tref_0[dbTT$tref_0 < -20 | dbTT$tref_0 > 50] <- NA
  dbTT$tref_1[dbTT$tref_1 < -20 | dbTT$tref_1 > 50] <- NA

```

```
dbTT$theat_0[dbTT$theat_0 < -20 | dbTT$theat_0 > 50] <- NA
dbTT$theat_1[dbTT$theat_1 < -20 | dbTT$theat_1 > 50] <- NA
```

```
# Apply a Savitzky-Golay smoothing
#library(signal)
```

```
## For tref_0
```

```
tfilt <- dbTT$tref_0 # data to be filtered
```

```
ID <- unique(dbTT$TTid)
for (j in 1:length(ID)) {
  ts <- tfilt[dbTT$TTid == ID[j]]
  if (length(ts) < 11) {
    next()
  }
  p = 2
  n = 7
  m = 0
  ts_filt <- savitzkyGolay(ts, m, p, n)
  tfilt[dbTT$TTid == ID[j]] <- ts_filt[1:length(ts)]
}
dbTT$tref_0filt <- tfilt
```

```
## For tref_1
```

```
tfilt <- dbTT$tref_1
```

```
ID <- unique(dbTT$TTid)
for (j in 1:length(ID)) {
  ts <- tfilt[dbTT$TTid == ID[j]]
  if (length(ts) < 11) {
    next()
  }
  p = 2
  n = 7
  m = 0
  ts_filt <- savitzkyGolay(ts, m, p, n)
  tfilt[dbTT$TTid == ID[j]] <- ts_filt[1:length(ts)]
}
dbTT$tref_1filt <- tfilt
```

```
## For theat_0
```

```
tfilt <- dbTT$theat_0
```

```
ID <- unique(dbTT$TTid)
for (j in 1:length(ID)) {
  ts <- tfilt[dbTT$TTid == ID[j]]
  if (length(ts) < 11) {
    next()
  }
  p = 2
  n = 7
  m = 0
  ts_filt <- savitzkyGolay(ts, m, p, n)
  tfilt[dbTT$TTid == ID[j]] <- ts_filt[1:length(ts)]
}
```

```

}
dbTT$theat_0filt <- tfilt

## For theat_1
tfilt <- dbTT$theat_1

ID <- unique(dbTT$TTid)
for (j in 1:length(ID)) {
  ts <- tfilt[dbTT$TTid == ID[j]]
  if (length(ts) < 11) {
    next()
  }
  p = 2
  n = 7
  m = 0
  ts_filt <- savitzkyGolay(ts, m, p, n)
  tfilt[dbTT$TTid == ID[j]] <- ts_filt[1:length(ts)]
}
dbTT$theat_1filt <- tfilt

# Plot to visualize one temperature data (tref_0_filt in this case)
#plot(dbTT$tref_0filt,
#  type = "l",
#  main = "tref_0 filtered with Savitzky-Golay",
#  xlab = "Data in order",
#  ylab = "tref_0filt (°C)")

# SFD, method from the manual, SFD (Ldm-2h-1)

# Two options to calculate deltaT:
# delta T1
DeltaT_heat <- dbTT$theat_1filt - dbTT$theat_0filt
# delta T2
DeltaT_heat2 <- (dbTT$theat_1filt - dbTT$tref_1filt) - (dbTT$theat_0filt
- dbTT$tref_0filt)

# Get the Max value from each TT PER DAY

# create a column by day
dataDay <- paste0(month(dbTT$timeStamp), day(dbTT$timeStamp))

# Create a new df for SFD
TTid <- dbTT$TTid
dbSFD <- as.data.frame(cbind(TTid, dataDay, DeltaT_heat, DeltaT_heat2))

# Aggregate data by TT and by day = obtain the max value per day and TT
delta_temp <- aggregate(DeltaT_heat ~ TTid + dataDay, data = dbSFD, FUN
= max, na.rm = TRUE)

dbSFD <- left_join(dbSFD, delta_temp, by = c ("TTid", "dataDay"))

DeltaTMax <- dbSFD$DeltaT_heat.y

# SFD

```

```

sapFluxDensity <-
  4.79 * ((as.double(DeltaTMax) - DeltaT_heat) / DeltaT_heat)

#sapFluxDensity[sapFluxDensity < 0] <- NA

dbTT <- cbind(dbTT, sapFluxDensity)

# Plot for SFD
#plot(dbTT$sapFluxDensity,
# type = "l",
# main = "Sap Flux Density",
# xlab = "Data in order",
# ylab = "SFD (Ldm-2h-1)")

# 4.5. Weekly Radial Growth -----
-----

# Packages:
#library(Lubridate)
#library(dplyr)

# Conversion range according to the manual (September 2020)
# First data filter for [D.N.]
tgrow <- dbTT$growth
tgrow[tgrow > 85000] <- NA
tgrow[tgrow < 30000] <- NA

# Plot to visualize growth sensor d.n.
#plot(tgrow,
# type = "l",
# main = "Growth sensor d.n.",
# xlab = "Data in order",
# ylab = "Growth sensor d.n.")

# Distance from sensor to the bark (cm from the TT+ manual, *10 to mm un
its)
dbTT$DistGrowth <-
  ((237908.4541 + (-1.1171 * tgrow)) / (199.4330 + tgrow))*10

# Create a time index for the temporal averaging (YYYYww)
time_index <- paste0(year(dbTT$timeStamp), week(dbTT$timeStamp))
dbTT$YYYYww <- time_index

# Method from paper: Tomelleri et al, 2022
ID <- unique(dbTT$TTid)
for (j in 1:length(ID)) {
  myDendro_data_L0 <-
    data.frame(dbTT$TTid, dbTT$timeStamp, dbTT$YYYYww, dbTT$DistGrowth)
  colnames(myDendro_data_L0) <- c("series", "ts", "YYYYww", "dendrometer
")

# Subset dataset for TTids
myDendro_data_L0 <- myDendro_data_L0 %>% dplyr::filter(series == ID[j])

```

```

)
  if (length(na.omit(myDendro_data_L0$dendrometer))<100){next}

  #remove outliers
  t_05 <- quantile(myDendro_data_L0$dendrometer, p=0.05, na.rm=T)
  t_95 <- quantile(myDendro_data_L0$dendrometer, p=0.95, na.rm=T)
  myDendro_data_L0$dendrometer[myDendro_data_L0$dendrometer<t_05] <- NA
  myDendro_data_L0$dendrometer[myDendro_data_L0$dendrometer>t_95] <- NA

  #create a data.frame with Level 1 data
  myDendro_data_L1 <- myDendro_data_L0

  #temporal averaging
  myDendro_data_weekly <-
    aggregate(dendrometer ~ YYYYww, myDendro_data_L1, median)

  #normalize the radial growth with the initial sensor distance
  myDendro_data_weekly$dendrometer <-
    (max(myDendro_data_weekly$dendrometer) - myDendro_data_weekly$dendro
meter)

  #create a data.frame with Level 2 data
  myDendro_data_L2 <- subset(myDendro_data_L1, select = -dendrometer)

  #merge averaged sharp data with the specific TreeTalker data.frame
  myDendro_data_L2 <- merge(myDendro_data_L2, myDendro_data_weekly, all.
x=T)

  #insert processed data into the dbTT data.frame
  dbTT$dendrometer[dbTT$TTid == ID[j]] <- myDendro_data_L2$dendrometer
}

# plot weekly radial growth (mm)
#plot(dbTT$dendrometer,
#  type = "l",
#  main = "Weekly Radial Growth (mm)",
#  xlab = "Data in order",
#  ylab = "Weekly Radial Growth (mm)")

# 4.6. Tree Movement: angle of oscilation (°) -----
-----

dbTT$gzm <- as.numeric(dbTT$gzm)
gz <- abs(dbTT$gzm)
gx <- abs(dbTT$gxm)
gy <- abs(dbTT$gym)

## Yaw angle
#gxy <- sqrt(gx^2 + gy^2)
#TreeMovement <- (atan2(gxy,gz)*180)/(pi)

## Pitch angle
gyz <- sqrt(gy^2 + gz^2)

```

```
TreeMovement <- (atan2(gx,gyz)*180)/(pi)
```

```
dbTT <- cbind(dbTT, TreeMovement)
```

```
# 4.7. Spectral Light through the canopy -----
```

```
# To convert the digital numbers (d.n.) into corrected spectral bands
```

```
#Copy bands
```

```
#Near infrared
```

```
dbTT$B610 <- as.numeric(dbTT$B1)
```

```
dbTT$B680 <- as.numeric(dbTT$B2)
```

```
dbTT$B730 <- as.numeric(dbTT$B3)
```

```
dbTT$B760 <- as.numeric(dbTT$B4)
```

```
dbTT$B810 <- as.numeric(dbTT$B5)
```

```
dbTT$B860 <- as.numeric(dbTT$B6)
```

```
#Visible Light spectrum
```

```
dbTT$B450 <- as.numeric(dbTT$B7)
```

```
dbTT$B500 <- as.numeric(dbTT$B8)
```

```
dbTT$B550 <- as.numeric(dbTT$B9)
```

```
dbTT$B570 <- as.numeric(dbTT$B10)
```

```
dbTT$B600 <- as.numeric(dbTT$B11)
```

```
dbTT$B650 <- as.numeric(dbTT$B12)
```

```
#Delete high values
```

```
dbTT$B610[dbTT$B610 == 0 | dbTT$B610 > 65000 ] <- NA
```

```
dbTT$B680[dbTT$B680 == 0 | dbTT$B680 > 65000 ] <- NA
```

```
dbTT$B730[dbTT$B730 == 0 | dbTT$B730 > 65000 ] <- NA
```

```
dbTT$B760[dbTT$B760 == 0 | dbTT$B760 > 65000 ] <- NA
```

```
dbTT$B810[dbTT$B810 == 0 | dbTT$B810 > 65000 ] <- NA
```

```
dbTT$B450[dbTT$B450 == 0 | dbTT$B450 > 65000 ] <- NA
```

```
dbTT$B500[dbTT$B500 == 0 | dbTT$B500 > 65000 ] <- NA
```

```
dbTT$B550[dbTT$B550 == 0 | dbTT$B550 > 65000 ] <- NA
```

```
dbTT$B570[dbTT$B570 == 0 | dbTT$B570 > 65000 ] <- NA
```

```
dbTT$B600[dbTT$B600 == 0 | dbTT$B600 > 65000 ] <- NA
```

```
dbTT$B650[dbTT$B650 == 0 | dbTT$B650 > 65000 ] <- NA
```

```
#Convert the digital numbers (d.n.) into corrected spectral bands
```

```
dbTT$B610 <- -312.45+(1.6699 *dbTT$B610)
```

```
dbTT$B680 <- -561.56+(1.5199 *dbTT$B680)
```

```
dbTT$B730 <- -1511.2+(1.6209 *dbTT$B730)
```

```
dbTT$B760 <- -1012.5+(1.4549 *dbTT$B760)
```

```
dbTT$B810 <- 91.58+(0.8414 *dbTT$B810)
```

```
dbTT$B860 <- 334.88+(0.531 *dbTT$B860)
```

```
dbTT$B450 <- -212.62+(0.4562*dbTT$B450)
```

```
dbTT$B500 <- -232.13+(0.6257 *dbTT$B500)
```

```
dbTT$B550 <- -842.1+(1.0546 *dbTT$B550)
```

```
dbTT$B570 <- -666.72+(1.0462 *dbTT$B570)
```

```
dbTT$B600 <- -328.08+(0.8654 *dbTT$B600)
```

```
dbTT$B650 <- 202.77+(0.7829 *dbTT$B650)
```

```
# 4.8. Stem Water Content -----
```

```

-----

#' Relative Saturation Index, adapted from Asgharinia et al. (2022)

# Filter stwc if necessary (capacity 3MHz in this case)
# See plot
#plot(dbTT$stwc,
#      type = "l",
#      main = "Stem water content (stwc)",
#      xlab = "Data in order",
#      ylab = "ECf_stem sensor, stwc [d.n.]")

# Get the min value of the stwc sensor from each TT+
stwc_min <-
  aggregate(dbTT$stwc, list(as.character(dbTT$TTid)), min)

# Ceate a vector with the Min value of stwc per TT+
stwcMin <-
  as.numeric(stwc_min$x)[match(as.character(dbTT$TTid),
                              as.character(stwc_min$Group.1))]

# Obtain temperature (filtered previously) with associated to the Min st
wc
Tmin <- dbTT[dbTT$stwc == stwcMin,] # stwc_min$x replaced for stwcMin

stwcT_Min <-
  as.numeric(Tmin$tref_0filt)[match(as.character(dbTT$TTid),
                                    as.character(Tmin$TTid))]

ECfsat <- stwcMin
Tsat <- stwcT_Min

# m, the slope of temperature sensitivity for "water"*
# *in this case for now, it should be the tree species
m = -81

# Expected frequency signal
ECf_adj <- m * (dbTT$tref_0filt - Tsat) + ECfsat

# Relative index of the stem saturation (%)
RelStemSat <- (1-((dbTT$stwc - ECf_adj)/ECf_adj))*100

# Plot for the Relative Stem Saturation (%)
#plot(RelStemSat,
#      type = "l",
#      main = "Relative Stem Saturation",
#      xlab = "Data in order",
#      ylab = "Relative Stem Saturation (%)")

#Join results to the main DF
dbTT <- cbind(dbTT, RelStemSat)

## End of variables calculation -----

```



```

# Select columns for results
dbTT_Results <- dbTT[,c(
  "TTid", "TTsn", "timeStamp", "Location", "relativeHumidity",
  "temperature", "batteryLevel", "sapFluxDensity",
  "dendrometer", "TreeMovement", "RelStemSat",
  "B610", "B680", "B730", "B760", "B810", "B860",
  "B450", "B500", "B550", "B570", "B600", "B650")]

#' Apply a date filter here if prefer a df filtered by date

# Return as a variable in the global environment
dbTT <- dbTT # db with some step by step calculations
dbTT_Results <- dbTT_Results # db with the main results

# Write outputs
out_dir <- file.path("C:", "TreeTalkerDB", "TTResults")
if (dir.exists(out_dir) == F)
  dir.create(out_dir)

write.table(dbTT_Results,
  file.path(out_dir, paste0(whichDay, "_TTDB.csv")),
  row.names = FALSE,
  sep = ";")

#return(list(dbTT))
}

## Run the function: ----
checkTT(dbTT)

## Checking the status of TTs on 2022-08-30
## Check out the output as the full database in C:/TreeTalkerDB/TTResults
as 2022-08-30_TTDB.csv

#####
#### PART 5: TTAlert -----
-----#####
#####

```

Creates reports to alert anomalies from TT data

```

# Copy complete data base history
dbTT_alert <- dbTT

```

Activate one of the next 2 parameters options if data results is desired for one day:

```

#Day_alert <- as.Date("2022-08-10") # Day to check the alerts
#Day_alert <- as.Date(Sys.Date()-1) # yesterday

```

And activate next 2 lines to subset data frame by a date (Day\_alert)

```

#data_alert <- as.Date(dbTT_alert$timeStamp) == Day_alert
#dbTT_alert <- dbTT_alert[data_alert,]

```

Entry dates for next lines if data results is desired for a range of dates, format: "YYYY-MM-DD"

```
#Date0 <- as.Date("2022-04-07") # Start
#Date1 <- as.Date("2022-08-17") # End
#dbTT_alert <- dbTT_alert[dbTT_alert$timeStamp >= Date0 & dbTT_alert$timeStamp <= Date1 + 1,]

TTAlert <- function(dbTT_alert) {
  whichDay <- as.Date(Sys.Date())

  cat("Checking the status of TTs on", paste0(whichDay), "\n")

  # Input parameters, when these conditions are TRUE values are reported as alerts

  minimumBattery <- 3.5      # Battery level ranges should be > 3.5 (mV) to work properly
  maximumAngle <- 45        # Risk of a falling tree (angle)
  minimumSFD <- 1.0         # Low weekly sap flux density (ldm-2h-1)
  minimumGrowth <- 0.1      # Low weekly radial growth (mm)

  # 5.1. Low battery level alert ----

  batteryAlert <- dbTT_alert[dbTT_alert$batteryLevel <= minimumBattery, ]
  batteryAlert <- batteryAlert[, c("TTid", "timeStamp", "batteryLevel")]

  if (nrow(batteryAlert) > 0) {
    batteryAlert2 <- aggregate(timeStamp ~ TTid, data= batteryAlert, FUN=max)
    batteryAlert2 <- left_join(batteryAlert2, batteryAlert,
                              by = c("TTid", "timeStamp"))
    batteryAlert <- batteryAlert2
  }

  # 5.2. Tree movement alert ----

  angleAlert <- dbTT_alert[dbTT_alert$TreeMovement >= maximumAngle, ]
  angleAlert <- angleAlert[, c("TTid", "timeStamp", "TreeMovement")]

  if (nrow(angleAlert) > 0) {
    angleAlert <-
      aggregate(angleAlert[, 2:3], list(as.character(angleAlert$TTid)), min)
    colnames(angleAlert)[1] <- "TTid"
    angleAlert$TreeMovement <- round(angleAlert$TreeMovement)
    colnames(angleAlert)[3] <- "TreeAngle"
  }
}
```

```

# 5.3. Sap Flux Density (SFD) alert ----

sap_alert <- dbTT_alert[, c("TTid", "timeStamp", "YYYYww", "sapFluxDensity")]
sap_alert$sapFluxDensity <- round(sap_alert$sapFluxDensity, 4)

# Maximum weekly SFD
WeekMaxSFD <- aggregate(sapFluxDensity ~ TTid + YYYYww, data = sap_alert,
  FUN = max, na.rm = TRUE)

# Alert df
SFDAlert <- WeekMaxSFD[WeekMaxSFD$sapFluxDensity <= minimumSFD, ]

if (nrow(SFDAlert) > 0) {
  SFDAlert <- left_join(SFDAlert, sap_alert, by = c("TTid", "YYYYww", "sapFluxDensity"))
}

# 5.4. Radial growth alert ----

growth_alert <- dbTT_alert[, c("TTid", "timeStamp", "YYYYww", "dendrometer")]
growth_alert$dendrometer <- round(growth_alert$dendrometer, 2)

# Maximum weekly growth
WeekMaxGrowth <- aggregate(dendrometer ~ TTid + YYYYww, data = growth_alert,
  FUN = max, na.rm = TRUE)

# Package
#Library(dplyr)
WeekMaxGrowth <- WeekMaxGrowth %>%
  group_by(TTid) %>%
  mutate(deltaGrowth = dendrometer - lag(dendrometer, default = 0))

# Alert df
growthAlert <- WeekMaxGrowth[WeekMaxGrowth$deltaGrowth <= minimumGrowth,
]

# Include week after installation of TTs on trees, otherwise full report
growthAlert <- growthAlert[growthAlert$YYYYww >= 202216, ]

# Sort df by TT, not by time
if (nrow(SFDAlert) > 0) {
  growthAlert <- growthAlert[order(growthAlert$TTid),]
}

# Save outputs ----

out_dir <- file.path("C:", "TreeTalkerDB", "TTResults")
if (dir.exists(out_dir) == F)

```

```

dir.create(out_dir)

write.table(batteryAlert,
            file.path(out_dir, paste0(whichDay, "_BatteryTTAlert.csv")),
            row.names = FALSE,
            sep = ";")

write.table(angleAlert,
            file.path(out_dir, paste0(whichDay, "_AngleTTAlert.csv")),
            row.names = FALSE,
            sep = ";")

write.table(SFDAlert,
            file.path(out_dir, paste0(whichDay, "_SapFDTTAlert.csv")),
            row.names = FALSE,
            sep = ";")

write.table(growthAlert,
            file.path(out_dir, paste0(whichDay, "_GrowthTTAlert.csv")),
            row.names = FALSE,
            sep = ";")

batteryAlert <- batteryAlert
angleAlert <- angleAlert
SFDAlert <- SFDAlert
growthAlert <- growthAlert
}

# Run function:
TTAlert(dbTT_alert)

## Checking the status of TTs on 2022-08-30

#####
#### PART 6: Plots -----
-----#####
#####

# Define db for the function

```

Graphics for complete data (leave by default)

```
TTdb_today <- dbTT # Copy TT's DB from previous step
```

Filter by one TT here if it is necessary. Examples:

```
#unique(dbTT$TTsn)
#TTdb_today <- dbTT[dbTT$TTsn == "621B0393",]
```

Two or more TTs at the same time

```
#TTdb_today <- dbTT[dbTT$TTsn == "621B0396" | dbTT$TTsn == "621B0397",]
# Entry dates for graphics, format: "YYYY-MM-DD"
```

```
Date0 <- as.Date("2022-05-01") # Start
Date1 <- as.Date("2022-05-07") # End
```

plotting is recommended for 1 week or 2 for all variables, disable line that starts with 'scale\_x' (\*\*) on each variable when plotting > 2 weeks If the plotting fails, be sure that the timestamp from TTs are on time.

```
Plot <- function(TTdb_today, Date0, Date1) {

  # Path for outputs
  outPathReport <-
    file.path(file.path("C:", "TreeTalkerDB"), "TTPlots")
  if (dir.exists(outPathReport) == F)
    dir.create(outPathReport)

  # Message
  cat("Calculating and saving some graphics in", outPathReport, "\n")

  # Path for outputs 2 (plots)
  TTdirectory <- file.path("C:/", "TreeTalkerDB")
  outPath <-
    file.path(TTdirectory, "TTPlots", paste0(Date0, "_to_", Date1, "_plots"))
  if (dir.exists(outPath) == F)
    dir.create(outPath)

  # data reading
  whichDay = Sys.Date()

  # Subset data frame with dates of interest
  data_new <-
    TTdb_today[TTdb_today$timeStamp >= Date0 & TTdb_today$timeStamp <= Date1 + 1,]

  # Packages
  #library(ggplot2)
  #library(lubridate)
  #library(scales)
  #library(dplyr)
  #library(reshape2)

  theme_set(theme_bw())

  # 6.1. Air Temperature and air humidity -----

  ggplot(data_new, aes(x=timeStamp)) +
    geom_line(aes(y=temperature, col=TTid), size=0.8) +
    labs(title="Air Temperature",
         subtitle = paste0("From: ", Date0, " to ", Date1),
         y="Temperature [°C]",
         x="Date") +
    scale_x_datetime(date_breaks = "12 hours", labels= date_format("%b %d-%H:%M")) + #**
    theme(axis.text.x = element_text(angle = 25, vjust=1.0, hjust=1.0, siz
```

```
e = 8))

ggsave(file.path(outPath, "AirTemperature.PNG"),
        width = 8, height = 5)

ggplot(data_new, aes(x=timeStamp)) +
  geom_line(aes(y=relativeHumidity, col=TTid), size=0.8) +
  labs(title="Air Humidity",
        subtitle = paste0("From: ", Date0, " to ", Date1),
        y="Relative humidity (%)",
        x="Date") +
  scale_x_datetime(date_breaks = "12 hours", labels= date_format("%b %d-%H:%M")) + #**
  theme(axis.text.x = element_text(angle = 25, vjust=1.0, hjust=1.0, size = 8))
e = 8))

ggsave(file.path(outPath, "AirHumidity.PNG"),
        width = 8, height = 5)

# 6.2. Battery Level -----

ggplot(data_new, aes(x=timeStamp)) +
  geom_line(aes(y=batteryLevel, col=TTid), size=0.8) +
  geom_hline(yintercept = 3.5, color="red", size=1.0) +
  labs(title="TT's Battery Voltage",
        subtitle = paste0("From: ", Date0, " to ", Date1),
        y="Battery Voltage [mV]",
        x="Date") +
  scale_x_datetime(date_breaks = "12 hours", labels= date_format("%b %d-%H:%M")) + #**
  theme(axis.text.x = element_text(angle = 25, vjust=1.0, hjust=1.0, size = 8))
e = 8))

ggsave(file.path(outPath, "BatteryTT.PNG"),
        width = 8, height = 5)

# 6.3. Sap flux density (SFD) -----

ggplot(data_new, aes(x=timeStamp)) +
  geom_line(aes(y=sapFluxDensity, col=TTid), size=0.8) +
  labs(title="Sap Flux Density",
        subtitle = paste0("From: ", Date0, " to ", Date1),
        y="SFD [ldm-2h-1]",
        x="Date") +
  scale_x_datetime(date_breaks = "12 hours", labels= date_format("%b %d-%H:%M")) + #**
  theme(axis.text.x = element_text(angle = 25, vjust=1.0, hjust=1.0, size = 8))
e = 8))

ggsave(file.path(outPath, "sapFluxDensity.PNG"),
        width = 8, height = 5)

# 6.4. Radial growth -----
```

```

ggplot(data_new, aes(x=timeStamp)) +
  geom_line(aes(y=dendrometer, col=TTid), size=0.8) +
  labs(title="Weekly Radial Growth",
        subtitle = paste0("From: ", Date0, " to ", Date1),
        y="Radial Growth [mm]",
        x="Date") +
  scale_x_datetime(date_breaks = "1 week", labels= date_format("%b %d-%H
:%M")) + ###
  theme(axis.text.x = element_text(angle = 25, vjust=1.0, hjust=1.0, siz
e = 8))

ggsave(file.path(outPath, "RadialGrowth.PNG"),
        width = 8, height = 5)

# 6.5. Stem water content -----

ggplot(data_new, aes(x=timeStamp)) +
  geom_line(aes(y=RelStemSat, col=TTid), size=0.8) +
  labs(title="Relative Stem Saturation",
        subtitle = paste0("From: ", Date0, " to ", Date1),
        y="Relative stem saturation (%)",
        x="Date") +
  scale_x_datetime(date_breaks = "12 hours", labels= date_format("%b %d-
%H:%M")) + ###
  theme(axis.text.x = element_text(angle = 25, vjust=1.0, hjust=1.0, siz
e = 8))

ggsave(file.path(outPath, "RelStemSaturation.PNG"),
        width = 8, height = 5)

# 6.6. Light trough the canopy, spectral bands -----

# Select data only from 10AM to 2PM
data_new$hour <- paste0(hour(data_new$timeStamp))
dbspectral <- filter(data_new, hour > 10, hour <= 14 )

#Library(dplyr)
datalist = list() #initialize the list
ID <- unique(dbspectral$TTid)
for (j in 1:length(ID)) {

  # Subset dataset for TTids
  spectrom_L0 <- dbspectral %>% dplyr::filter(TTid == ID[j])

  if (length(spectrom_L0$timeStamp)<10){next}

  #aggregate to daily
  spectrom_L0$Day <- floor_date(spectrom_L0$timeStamp, "day")

  spectrom_L1 <- spectrom_L0 %>%
    group_by(Day) %>%
    summarize(L450 = median(B450, na.rm = TRUE),
              L500 = median(B500, na.rm = TRUE),
              L550 = median(B550, na.rm = TRUE),

```

```

    L570 = median(B570, na.rm = TRUE),
    L600 = median(B600, na.rm = TRUE),
    L650 = median(B650, na.rm = TRUE),
    L610 = median(B610, na.rm = TRUE),
    L680 = median(B680, na.rm = TRUE),
    L730 = median(B730, na.rm = TRUE),
    L760 = median(B760, na.rm = TRUE),
    L810 = median(B810, na.rm = TRUE),
    L860 = median(B860, na.rm = TRUE)
  )
  spectrom_L1$TTid <- rep(ID[j], length(spectrom_L1$Day))

  datalist[[j]] <- spectrom_L1 # add it to your list
}

# Average of data between hours, per day
spectrom_L1_all <- dplyr::bind_rows(datalist)

# Average between days, bands and TTid
spectrom_L2_all <- spectrom_L1_all %>%
  group_by(TTid) %>%
  summarise_at(vars("L450", "L500", "L550", "L570", "L600",
                    "L650", "L610", "L680", "L730", "L760",
                    "L810", "L860"), mean)

# Change column names for wavelength
Wavelength <- c(450, 500, 550, 570, 600, 650, 610, 680, 730, 760, 810, 8
60)
colnames(spectrom_L2_all) <- c("TTid", Wavelength)

# Reshape df for plot
#library(reshape2)
spectral <- melt(spectrom_L2_all, id.vars = "TTid")

ggplot(spectral, aes(x=variable)) +
  geom_line(aes(y=value, col=TTid, group=TTid), size=0.8) +
  labs(title="Daily average of spectral bands data",
       subtitle = paste0("From: ", Date0, " to ", Date1),
       y="Calibrated spectral bands",
       x="Wavelength (nm)")

ggsave(file.path(outPath, "SpectralBands.PNG"),
        width = 8, height = 5)
}

## Run the function: ----
suppressWarnings(Plot(TTdb_today, Date0, Date1))

## Calculating and saving some graphics in C:/TreeTalkerDB/TTPlots

```



```
#####  
###  
# PART 7: REPORT WITH RMARKDOWN -----  
#####  
###
```

Input the folder path with the "TTReport.Rmd" file downloaded to create the report with RMarkdown (TTReport.pdf)

```
render_path <- file.path("C:/Users/Isabel/Downloads", "TTReport.Rmd")  
  
# Run next lines  
suppressWarnings(  
  rmarkdown::render(  
    render_path,  
    output_format = c("html_document", "pdf_document")[i],  
    output_file = paste0("Report", c(".html", ".pdf"))[i],  
    output_dir = file.path("C:", "TreeTalkerDB")  
  )  
)  
  
##  
## Output created: C:/TreeTalkerDB/Report.pdf
```

END OF SCRIPT

## Anexo 4. Código RMarkdown para TTReport (*TTReport.Rmd*)

### TT\_Report.Rmd

Isabel Cristina Grisales S.

2022-08-30

```
---  
title: "TreeTalkers Check ULE"  
author: "Isabel Grisales"  
date: "`r Sys.Date()`"  
output:  
  pdf_document: default  
  html_document:  
    df_print: paged  
  word_document: default  
---
```

```
```\{=tex}  
\setlength\parindent{0pt}  
\setlength{\parskip}{0pt}  
```\{r setup, include=F, echo = F, fig.align="center", results="asis"}  
knitr::opts_chunk\$set(warning = FALSE, echo = F)  
```\
```

#### **# 1. Summary of TT Alerts**

This information is showing the alerts coming from the TreeTalkers data.

Note: The meaning of "YYYYww" is the year and week of the year from data that were assessed periodically every week.

```
```\{r 1, message=FALSE, results="asis"}  
  
library(ggplot2)  
whichDay <- as.Date(Sys.Date())  
  
#input_path <- file.path("C:", "out", "1")  
input_files_csv <- list.files("C:/TreeTalkerDB/TTResults", full.names = T,  
pattern = ".csv")  
  
# Read input files  
lista_files_read <- lapply(input_files_csv, read.table, sep = ";", header  
= T)  
  
cloudDB <- lista_files_read[[1]]  
cloudAlert <- lista_files_read[[2]] #  
angleAlert <- lista_files_read[[3]] #  
batteryAlert <- lista_files_read[[4]] #
```

```

growthAlert <- lista_files_read[[5]] #
SFDAAlert <- lista_files_read[[6]] #
FullDB <- lista_files_read[[7]]

# TTAAlerts
if(nrow(batteryAlert)==0){
cat("No issues related with the batteries where automatically identified")
}else{
cat("The number of battery issues are", nrow(batteryAlert))
knitr::kable(batteryAlert, caption = "Issues related with batteries (mV)")
}

cat(" \n")
cat(" \n")
cat(" \n")

if(nrow(SFDAAlert)==0){
cat("No issues related with the sap flux density where automatically
identified")
}else{
cat("The number of sap flux density issues are", nrow(SFDAAlert))
knitr::kable(SFDAAlert, caption = "Issues related with sap flux density (l
d m-2 h-1)")
}

cat(" \n")
cat(" \n")
cat(" \n")

if(nrow(growthAlert)==0){
cat("No issues related with the weekly radial growth where automatically
identified")
}else{
cat("The number of weekly radial growth issues are", nrow(growthAlert))
knitr::kable(growthAlert, caption = "Issues related with the weekly radial
growth (mm)")
}

cat(" \n")
cat(" \n")
cat(" \n")

if(nrow(angleAlert)==0){
cat("No issues related with the tree angles where automatically
identified")
}else{
cat("The number of tree angles issues are", nrow(angleAlert))
knitr::kable(angleAlert, caption = "Issues related with the tree angles
(°)")
}

cat(" \n")
cat(" \n")
cat(" \n")

```

```
...
```

## # 2. Summary of TTCloud Alerts

This information is showing the alerts coming from the TreeTalkers data.

Note: Column "issue" reports how many and which sensors are affected.

```
```{r 2, message=FALSE, results="asis"}

## Clouds Alerts
if(nrow(cloudAlert)==0){
cat("No issues related with the TTCloud where automatically identified")
}else{
cat("The number of TTCloud issues are", nrow(cloudAlert))
knitr::kable(cloudAlert, caption = "Issues related with TTClouds")
}

cat(" \n")
cat(" \n")
cat(" \n")

...

\newpage
```

## # 3. TT graphs

The graphs are important to understand if the sensors are working properly and to detect the trend of the TreeTalkers batteries, air temperature, air humidity, weekly radial growth, sap flux density and the spectral bands per each tree.

### ## 3.1. Temperature

```
```{r Temperature, echo=FALSE, fig.show='asis', out.width='100%',
fig.align='center', out.extra=NULL, fig.cap = "Air temperature"}

Plots_dir_list <- list.dirs("C:/TreeTalkerDB/TTplots", recursive = FALSE)

temperaturePlots <- list.files(Plots_dir_list, pattern = "Temperature",
full.names = TRUE)

knitr::include_graphics(temperaturePlots)

...

\newpage
```

### ## 3.2. Humidity

```
```{r Humidity, echo=FALSE, fig.show='asis', out.width='100%',
fig.align='center', out.extra=NULL, fig.cap = "Air Humidity"}
```

```
HumidityPlots <- list.files(Plots_dir_list, pattern = "Humidity",
full.names = TRUE)

knitr::include_graphics(HumidityPlots)
```

\newpage

## 3.3. Stem radial growth variation

```{r RadialGrowth, cho=FALSE, out.width='100%', fig.show='asis',
fig.align='center', out.extra=NULL, fig.cap = "Stem radial growth
variation"}```

growthPlots <- list.files(Plots_dir_list, pattern = "RadialGrowth",
full.names = TRUE)

knitr::include_graphics(growthPlots)
```

\newpage
```

### **## 3.4. Sap flux Density**

```
```{r Sapflow, echo=FALSE, out.width='100%', fig.show='asis',
fig.align='center', out.extra=NULL, fig.cap = "Sap flux Density"}```

SapfluxDensityPlots <- list.files(Plots_dir_list, pattern =
"sapFluxDensity", full.names = TRUE)

knitr::include_graphics(SapfluxDensityPlots)
```

\newpage
```

### **## 3.5. Spectral light bands through the canopy**

```
```{r SpectralBands, echo=FALSE, out.width='100%', fig.show='asis',
fig.align='center', out.extra=NULL, fig.cap = "Spectral light bands through
the canopy"}```

spectrometerPlots <- list.files(Plots_dir_list, pattern = "SpectralBands",
full.names = TRUE)

knitr::include_graphics(spectrometerPlots)
```

\newpage
```

### **## 3.6. Stem Water Content**

```
```{r RelStemSaturation, echo=FALSE, out.width='100%', fig.show='asis',
fig.align='center', out.extra=NULL, fig.cap = "Stem Water Content"}```

stwcPlots <- list.files(Plots_dir_list, pattern = "RelStemSaturation",
full.names = TRUE)
```

```
knitr::include_graphics(stwcPlots)
```

```
```\n
```

```
\\newpage
```

### ## 3.7. *TreeTalkers Battery Level*

```
```{r Battery, echo=FALSE, out.width='100%', fig.show='asis',  
fig.align='center', out.extra=NULL, fig.cap = "TreeTalkers Battery Level"}
```

```
batteryPlots <- list.files(Plots_dir_list, pattern = "Battery", full.names  
= TRUE)
```

```
knitr::include_graphics(file.path(batteryPlots))
```

```
```\n
```

### **Anexo 5. Reporte (TTReport.pdf)**

# TreeTalkers Check ULE

Isabel Grisales

2022-09-04

## 1. Summary of TT Alerts

This information is showing the alerts coming from the TreeTalkers data.

Note: The meaning of “YYYYww” is the year and week of the year from data that were assessed periodically every week.

The number of battery issues are 4

Table 1: Issues related with batteries (mV)

TTid	timeStamp	batteryLevel
TV_TT95	2022-07-15 18:00:00	2.99
VM_TT92	2022-07-20 17:00:00	3.02
VM_TT93	2022-07-17 18:00:00	2.97
VM_TT94	2022-07-19 19:00:00	3.07

The number of sap flux density issues are 7

Table 2: Issues related with sap flux density (l d m<sup>-2</sup> h<sup>-1</sup>)

TTid	YYYYww	sapFluxDensity	timeStamp
TV_TT98	202212	0.0000	2022-03-22 16:00:00
VM_TT96	202212	0.0550	2022-03-21 22:00:00
VM_TT92	202227	0.6327	2022-07-02 16:00:00
VM_TT92	202228	0.7803	2022-07-13 19:00:00
TV_TT98	202229	0.7331	2022-07-21 18:00:00
VM_TT93	202229	0.7697	2022-07-21 19:00:00
VM_TT93	202230	0.8637	2022-07-25 18:00:01

The number of weekly radial growth issues are 52

Table 3: Issues related with the weekly radial growth (mm)

TTid	YYYYww	dendrometer	deltaGrowth
TV_TT95	202216	0.05	-0.06
TV_TT95	202217	0.00	-0.05
TV_TT95	202218	0.00	0.00
TV_TT95	202225	4.45	0.01
TV_TT95	202226	4.46	0.01
TV_TT95	202229	6.11	0.03
TV_TT95	202230	6.11	0.00

TTid	YYYYww	dendrometer	deltaGrowth
TV_TT95	202233	6.89	-0.01
TV_TT95	202234	6.89	0.00
TV_TT95	202235	6.89	0.00
TV_TT95	202236	6.89	0.00
TV_TT98	202229	3.88	-0.12
TV_TT98	202230	3.88	0.00
TV_TT98	202231	1.70	-2.18
TV_TT98	202232	0.49	-1.21
TV_TT98	202233	0.49	0.00
TV_TT98	202234	0.25	-0.24
TV_TT98	202235	0.23	-0.02
TV_TT98	202236	0.11	-0.12
VM_TT92	202216	0.12	0.00
VM_TT92	202217	0.15	0.03
VM_TT92	202218	0.21	0.06
VM_TT92	202224	2.97	0.04
VM_TT92	202226	3.61	0.00
VM_TT92	202227	3.63	0.02
VM_TT92	202228	3.63	0.00
VM_TT92	202229	3.63	0.00
VM_TT92	202230	3.61	-0.02
VM_TT92	202231	3.62	0.01
VM_TT93	202217	6.06	-0.14
VM_TT93	202218	3.14	-2.92
VM_TT93	202219	2.99	-0.15
VM_TT93	202220	3.00	0.01
VM_TT93	202221	2.30	-0.70
VM_TT93	202222	2.19	-0.11
VM_TT93	202224	0.00	-5.84
VM_TT93	202228	3.53	-2.53
VM_TT93	202229	2.18	-1.35
VM_TT93	202230	2.16	-0.02
VM_TT93	202231	1.87	-0.29
VM_TT94	202216	2.75	0.00
VM_TT94	202219	3.43	-0.01
VM_TT94	202220	2.82	-0.61
VM_TT94	202222	2.65	-0.81
VM_TT94	202224	3.51	0.00
VM_TT94	202225	1.06	-2.45
VM_TT94	202226	0.99	-0.07
VM_TT94	202227	0.06	-0.93
VM_TT94	202228	0.00	-0.06
VM_TT94	202230	1.89	-0.69
VM_TT94	202231	1.78	-0.11
VM_TT94	202232	1.06	-0.72

The number of tree angles issues are 6



Table 4: Issues related with the tree angles (°)

TTid	timeStamp	TreeAngle
TV_TT95	2022-03-30 20:00:00	89
TV_TT98	1970-01-01 02:03:05	47
VM_TT92	1970-01-01 02:04:14	65
VM_TT93	2022-03-22 11:00:01	66
VM_TT94	2022-03-22 10:00:00	81
VM_TT96	2022-03-21 18:00:00	54

## 2. Summary of TTCloud Alerts

This information is showing the alerts coming from the TreeTalkers data. Note: Column “issue” reports how many and which sensors are affected.

The number of TTCloud issues are 24

Table 5: Issues related with TTClouds

tipology	issue	cloudId	timeStamp	Location.x
sensorSignal	1;3	C0210043	2022-09-03 00:00:00	VM
sensorSignal	1;3	C0210043	2022-09-03 01:00:00	VM
sensorSignal	1;3	C0210043	2022-09-03 02:00:00	VM
sensorSignal	1;3	C0210043	2022-09-03 03:00:00	VM
sensorSignal	1;3	C0210043	2022-09-03 04:00:00	VM
sensorSignal	1;3	C0210043	2022-09-03 05:00:00	VM
sensorSignal	1;3	C0210043	2022-09-03 06:00:00	VM
sensorSignal	1;3	C0210043	2022-09-03 07:00:00	VM
sensorSignal	1;3	C0210043	2022-09-03 08:00:00	VM
sensorSignal	1;3	C0210043	2022-09-03 09:00:00	VM
sensorSignal	1;3	C0210043	2022-09-03 10:00:00	VM
sensorSignal	1;3	C0210043	2022-09-03 11:00:00	VM
sensorSignal	1;3	C0210043	2022-09-03 12:00:00	VM
sensorSignal	1;3	C0210043	2022-09-03 13:00:00	VM
sensorSignal	1;3	C0210043	2022-09-03 14:00:00	VM
sensorSignal	1;3	C0210043	2022-09-03 15:00:00	VM
sensorSignal	1;3	C0210043	2022-09-03 16:00:00	VM
sensorSignal	1;3	C0210043	2022-09-03 17:00:00	VM
sensorSignal	1;3	C0210043	2022-09-03 18:00:00	VM
sensorSignal	1;3	C0210043	2022-09-03 19:00:00	VM
sensorSignal	1;3	C0210043	2022-09-03 20:00:00	VM
sensorSignal	1;3	C0210043	2022-09-03 21:00:00	VM
sensorSignal	1;3	C0210043	2022-09-03 22:00:00	VM
sensorSignal	1;3	C0210043	2022-09-03 23:00:00	VM

### 3. TT graphs

The graphs are important to understand if the sensors are working properly and to detect the trend of the TreeTalkers batteries, air temperature, air humidity, weekly radial growth, sap flux density and the spectral bands per each tree.

#### 3.1. Temperature

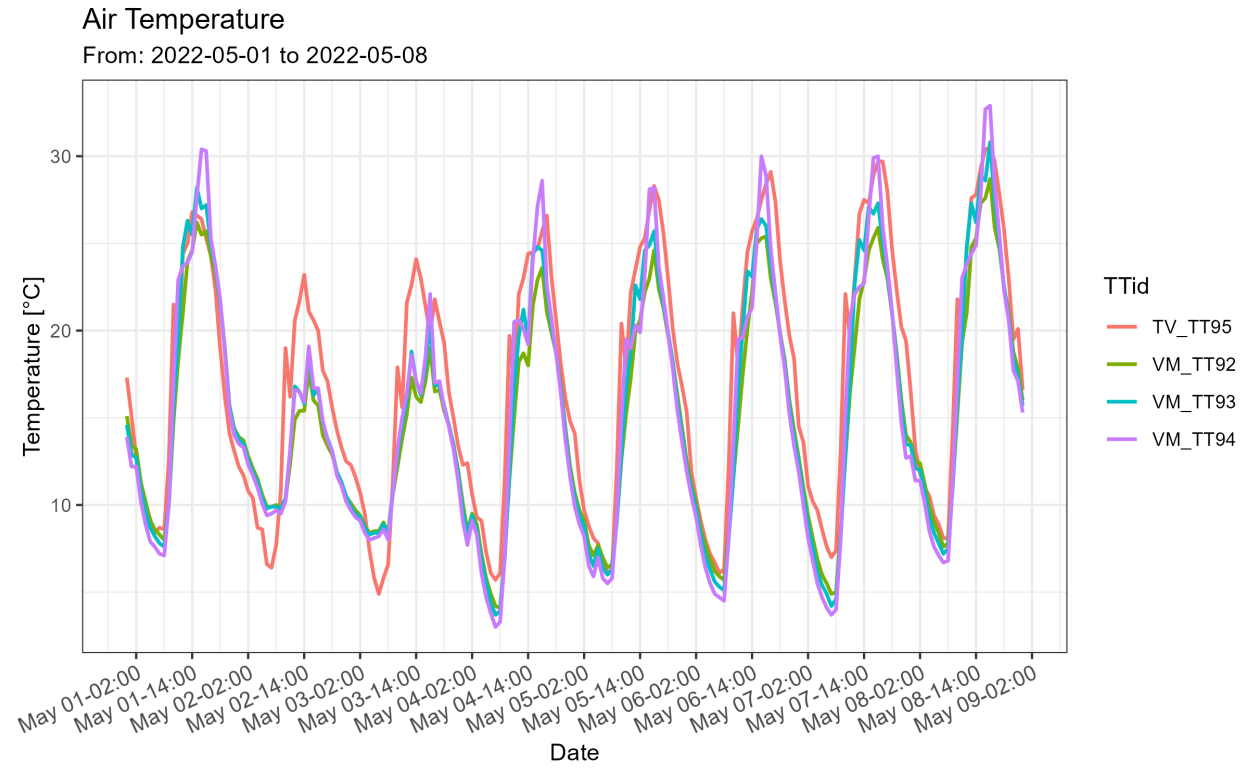


Figure 1: Air temperature

### 3.2. Humidity

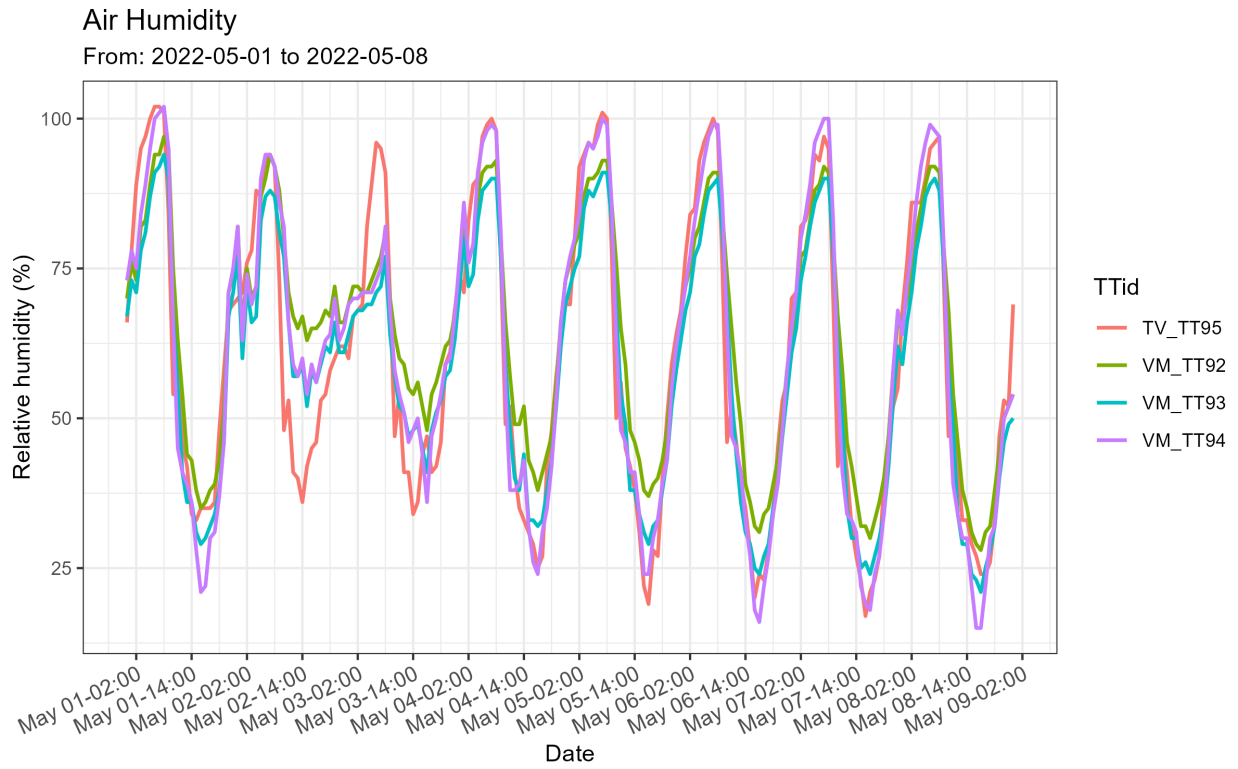


Figure 2: Air Humidity

### 3.3. Stem radial growth variation

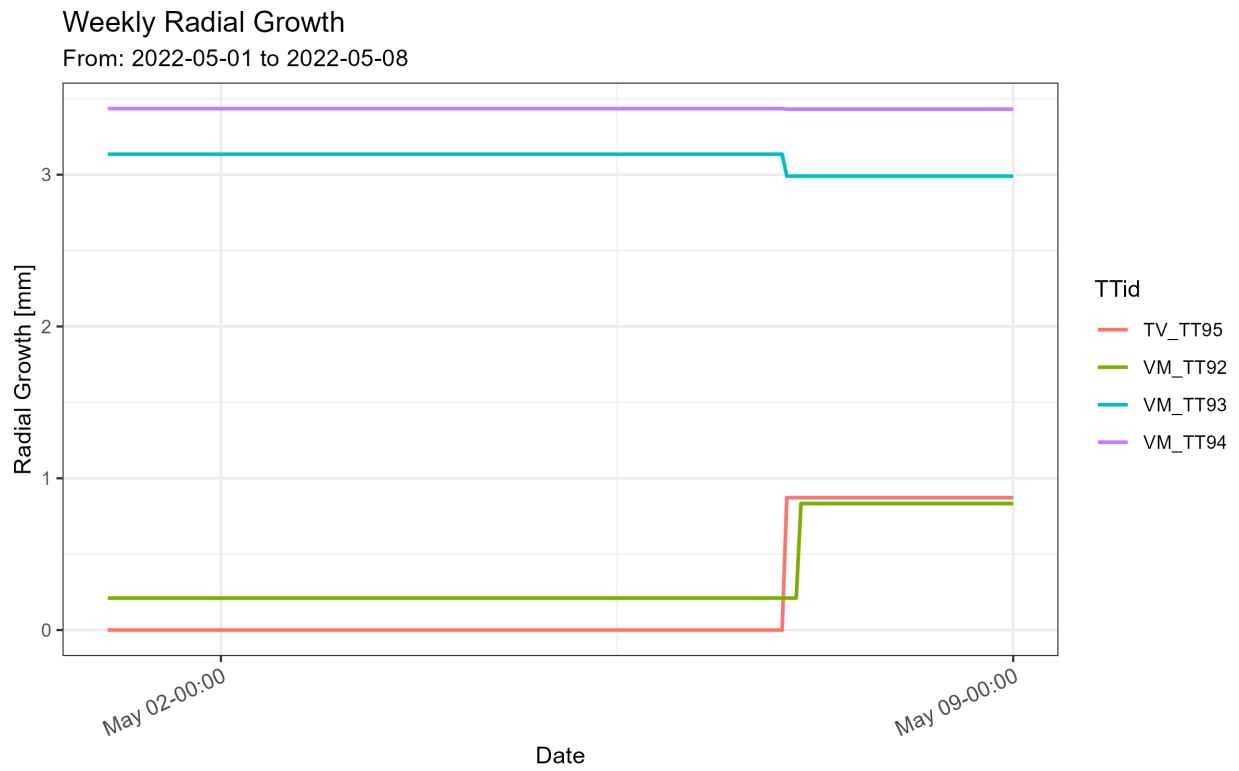


Figure 3: Stem radial growth variation

### 3.4. Sap flux Density

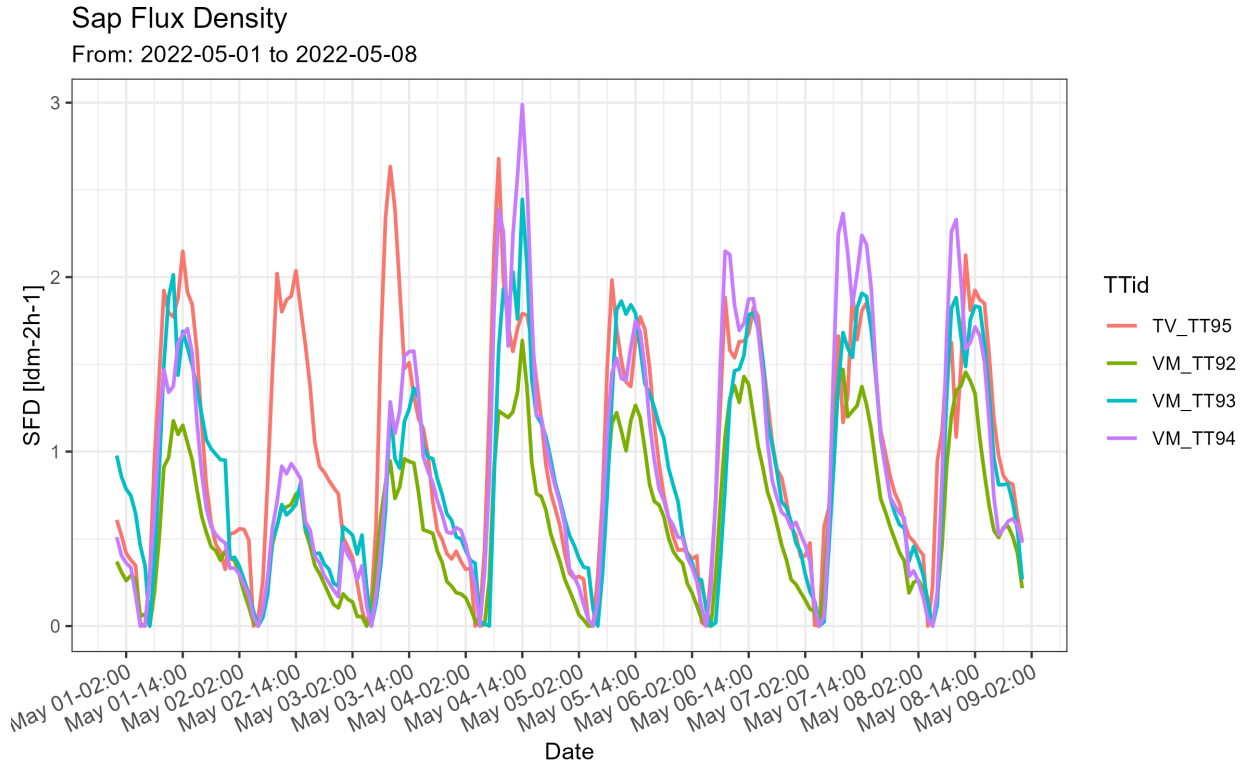


Figure 4: Sap flux Density

### 3.5. Spectral light bands through the canopy

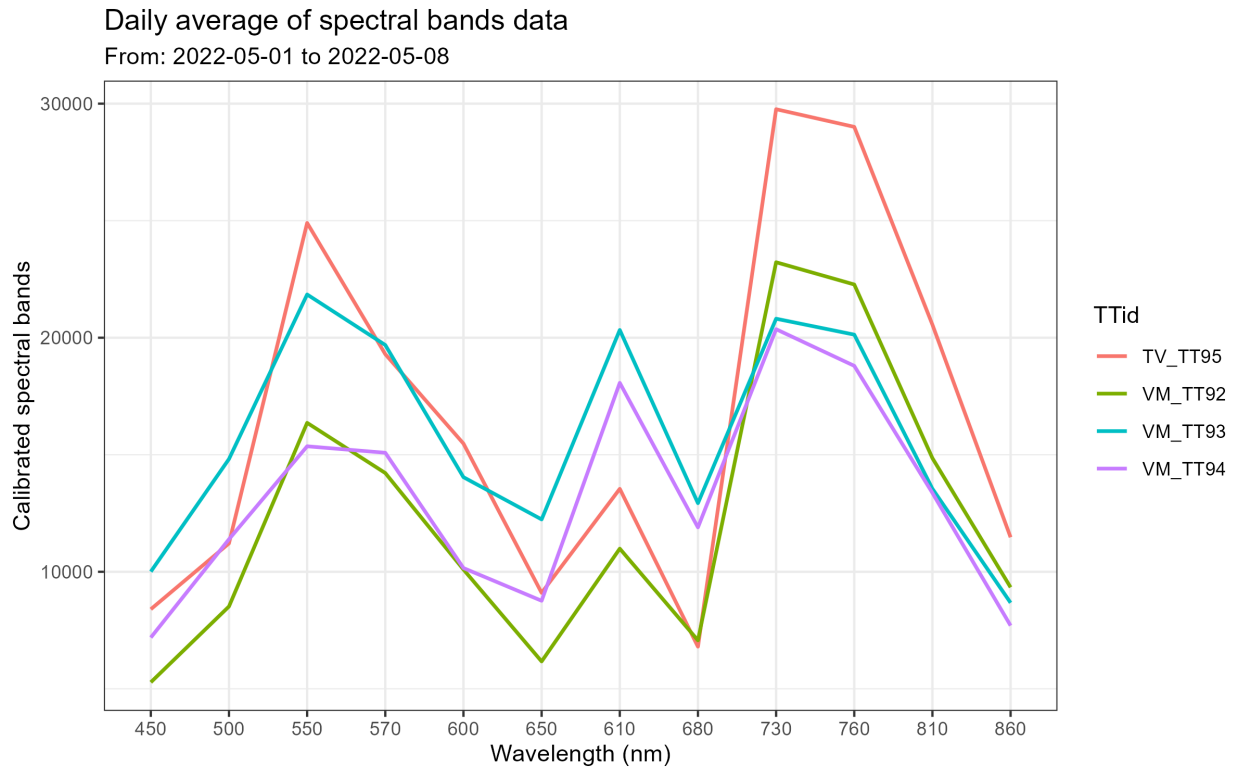


Figure 5: Spectral light bands through the canopy

### 3.6. Stem Water Content

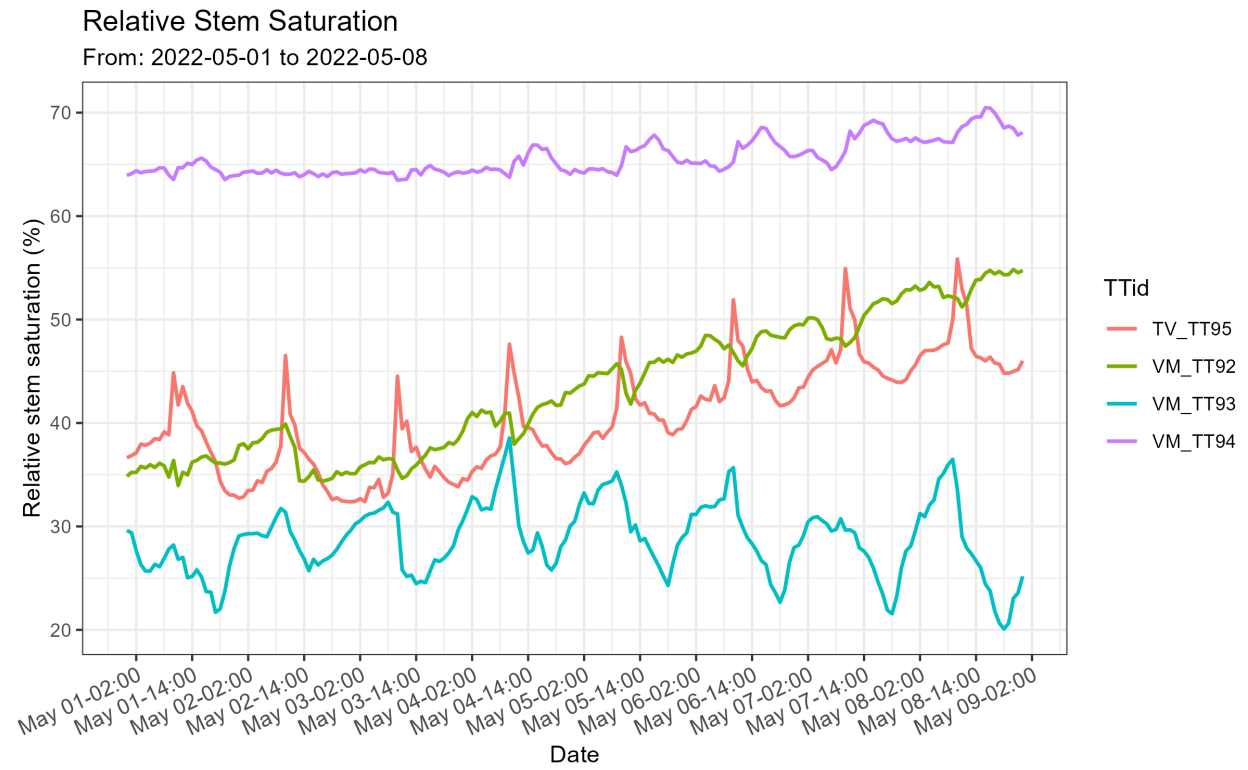


Figure 6: Stem Water Content

### 3.7. TreeTalkers Battery Level

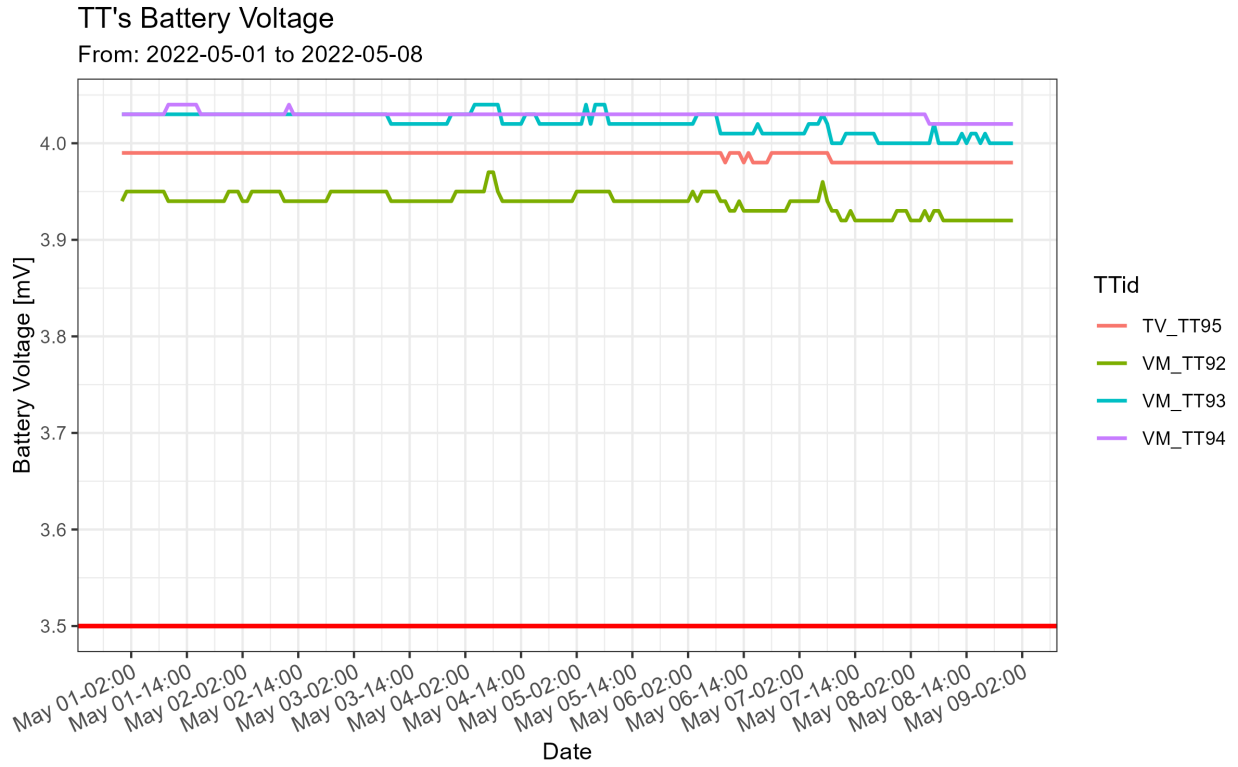


Figure 7: TreeTalkers Battery Level