Research note on mapping allergenic tree species distributions in Belgium

Dujardin, Sebastien; Linard, Catherine; DENDONCKER, Nicolas

Publication date:
2017

Document Version
Publisher's PDF, also known as Version of record

BRAIN-be

BELGIAN RESEARCH ACTION THROUGH INTERDISCIPLINARY NETWORKS

BR/154/A1/RespirIT

Research note on mapping allergenic tree species in Belgium

14/04/2017

Updated on 26/02/2021

UNIVERSITÉ DE NAMUR

GÉOGRAPHIE

Dr. Sébastien Dujardin
Pr. Catherine Linard
Pr. Nicolas Dendoncker

Table of contents

1. Introduction

The BRAIN-be/RespirIT research project (contract number: BR/154/A1/RespirIT) aims at exploring and understanding the spatial and temporal effects of plant diversity on respiratory health. By linking the information on the whereabouts and medical condition of individuals to spatially explicit information of plant diversity, air quality and pollen concentrations, it has the ambition to quantitatively, dynamically, and spatially study plant diversity effects on allergic symptom severity. Yet, spatially explicit information on plant diversity is not readily available for Belgium and ready-made methods to link existing land cover maps to proxies of plant diversity are not available. Consequently, the work package 3 of the RespirIT research project seeks to provide spatially explicit proxies for plant diversity by (i) mapping the location of targeted allergenic tree species (task 3.1.1) and (ii) deriving spatially explicit proxies of plant diversity (task 3.1.2).

Within this research note, we report on the investigations accomplished to characterise our study area in terms of plant diversity. First, we provide an overview of the dataset inventory undertaken for mapping allergenic tree species in Belgium. We review the different types of datasets currently available and expose their thematic and spatial resolutions for characterising land use and plant diversity in Belgium. Second, we detail a method for mapping allergenic tree species at the national scale using datasets available within both the Flemish and the Walloon regions. Third, we expose and discuss our main results, including the presence or absence of allergenic trees across Belgium, the spatial distribution of three selected allergenic trees (Alnus, Betula, and Corylus), as well as their relative share in terms of basal area. Finally, we expose the key learnings honed through the mapping of allergenic tree species in Belgium and provide several research outlooks for future research.

2. Dataset inventory

Investigations started with a review of current database available in Belgium allowing for characterizing land use and vegetation cover. We identified three relevant types of georeferenced datasets for studying the spatial effects of plant diversity on respiratory health.

Firstly, several land use maps are available in both Flanders and Wallonia for describing land uses at a high spatial resolution. However, the Flemish and Walloon maps contain different categories of land use. This creates multiple matching issues when considering merging both datasets. Yet, the *IGN Topo Map* provides a comparative advantage as it covers the whole country. The vector map's thematic resolution differentiates coniferous tree species from deciduous tree species.

Secondly, no national-scale biodiversity map exists at this time of writing. While Flanders has completed a "biological valuation map", Brussels and Wallonia are currently in the process of elaborating such a map. Detailed biodiversity inventories exist but need to be georeferenced and mapped. Nonetheless, a useful indicator covering the entire country is available from the LIFEWATCH project. A greenness indicator is available at a 1x1 km raster resolution to characterise plant phenology at high temporal resolution (i.e. across the weekly, monthly, yearly periods).

Thirdly, the most detailed and up-to-date information about tree diversity is contained within forest inventories. Both Flanders and Wallonia have undertaken a region-wide survey characterizing the type of tree species observed within forested areas. These datasets have a thematic resolution differentiating the tree selected allergenic tree species investigated within this research project: namely Alnus, Betula, and Corylus. These are available upon request by the « Agentschap voor Natuur en Bos » and the « Faculté universitaire des Sciences agronomiques de Gembloux » (Unité de Gestion des Ressources forestières et des Milieux naturels), respectively.

Table 1. Dataset inventory summary

	Accuracy	**Coverage**	**Thematic resolution**
Land Use			
IGN Topo Map	1/10.000	Belgium	Coniferous/Discidious
ECOPLAN Land Cover	5x5m	Flanders	Tree species (Betula)
COSW	1/25.000	Wallonia	Coniferous/Discidious
CORINE Land Cover	1/100.000 (250x250m)	Belgium	Coniferous/Discidious
Biodiversity map			
Biological Valuation Map Flanders	1/10.000	Flanders	Habitat types
LIFEWATCH Greeness indicator	1km x 1km	Belgium	Plant phenology / Greenness
Vegetation survey			
Inventaire forestier wallon	Points (1km x 0,5km)	Wallonia	Alnus/Betula/Corylus
Regionale bosinventarisatie	Points (1km x 0,5km)	Flanders	Alnus/Betula/Corylus

3. Allergenic tree species distributions

3.1. Forest inventory data

The most detailed and up-to-date information about tree abundance currently available in Belgium is contained within forest inventories. As shown in **Error! Reference source not found.**, these dataset allow covering forested areas which represent 23% of the total land area in Belgium. Both Flanders (see Afdeling Bos & Groen (2001)): and Wallonia (see Rondeux and Lecomte, 2010, Alderweireld et al., 2015) have undertaken a region-wide survey characterizing the type of tree species observed within forested areas. These dataset have a thematic resolution that differentiates the selected

allergenic tree species investigated within this study, namely Alnus, Betula, and Corylus. We processed each forest inventory separately and then merged both dataset after having checked data consistency.

The main purpose of our analysis was to extract information from the Flemish and Walloon forest inventories about allergenic trees and provide an indicator of their relative abundance within a given tree stand. This was measured by calculating an overall percentage of basal area (i.e. the area of a given section of land that is occupied by the cross-section of tree trunks and stems at the base) for the selected allergenic trees (Alnus, Betula, and Corylus) per sampling plot. In practical terms, we first calculate a specific basal area (in m²/ha) of each allergenic trees species studied. Then, we compute the basal area of all trees observed within the plot. Finally, we calculate a final indicator, which is an overall relative percentage of basal area.

Table 2 Metadata of both forest inventories

	Inventaire forestier wallon	Regionale bosinventarisatie
Number of sampling plots	13.228	2.147
Reference year	1994-2004; 2008-2017	2009-2017
Spatial reference system	Belgian Lambert 72	Belgian Lambert 72
Spatial reference scale	Points (1km x 0,5km)	Points (1km x 0,5km)
Author	Walloon region	INBO

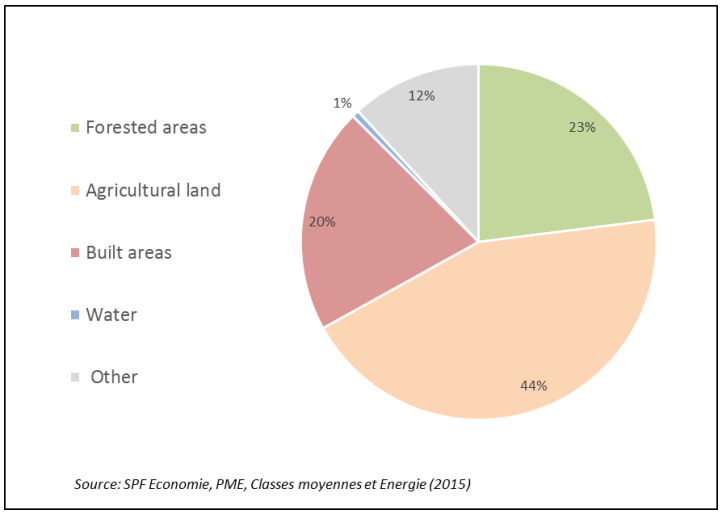


Figure 1. Relative share of land use types in Belgium

3.1.1. Processing the Walloon forest inventory dataset

3.1.1.1. Dataset's content

The Walloon forest inventory is based on 11.080 sampling plots in which trees are surveyed on a regular basis (see Rondeux and Lecomte, 2010, Alderweireld et al., 2015). Since 1994, two survey

cycles were undertaken. The first one took 14 years long. The second one started in 2008 and is still ongoing (to date, 50% is complete). The dataset contains two main tables.

The first table provided within the dataset (sheet "Placettes") details the general characteristics of each plot, including:

- Plot's geographic coordinates
- Date surveyed (ranging from May 2008 to March 2015)
- Overall type of tree stand (e.g. oak grove, pine forest, etc.)
- Structure (coppice, tree stand forest, etc.)
- The *Waleunis* code classification

A second table (sheet "Essence") provides information about the allergenic tree species surveyed within the plot, namely *alnus*, *betula* and *corylus* (respectively alder, birch, and hazel in English). For a given plot, a new line is added each time a different type of tree is observed. Trees are differentiated on the basis of their structure which is either under the form of a coppice (*taillis*) or a stand forest (*futaie*). Stand forests are then sub-divided in two categories of level 1 and 2 (*étage forestier*).

Within the sample table below for instance, the third line corresponds to a *corylus* observed under the form of a coppice. On the same plot (line four), *betula* is also observed under the form of a tree stand of level 2 (high trees).

Table 3. Dataset extract describing the type of trees observed

ign	npl	cy	source	etg_f		ess	essence	pcgha	nha	gha	vha
2856	258	2	tai			20	Noisetier	1			
3178	219	2	fut	1		22	Bouleau	0.073	20.546	2.681	26.207
4956	237	2	tai			20	Noisetier	1	160.132	2.247	15.787
4956	237	2	fut	2		22	Bouleau	0.622	321.403	5.667	44.365

3.1.1.2. Variables available

The dataset "essence" contains different measures describing the presence of allergenic trees. Two of these are of particular interest for the purpose of this study: the type of tree observed (variable "ess" or "essence") and the share of tree species within the tree stand (variable "pcgha").

The variable "ess" allows identifying the type of allergenic tree observed within a plot. It has, however, some limitations as the count only includes trees with a circumference greater than 20 cm.

Besides, a threshold of minimum 20 trees has been set by surveyors for sampling *corylus* (see first line in table 1 above). From this limitation, the variables "nha", "gha", and "vha" are not calculated.

The variable "pcgha" on the opposite is better suited for evaluating the relative presence of allergenic trees. When a tree with a circumference under 20 cm is observed within a plot, the species' relative share within the tree stand is still provided based on an estimation from the surveyor.

Table 4. Variables description

Variables available	Dataset's original definition	English translation
ign, np	*Identification des placettes*	Plot's identification number
cy	*Cycle*	Survey cycle
source	*Type de structure*	Structure type (coppice vs stand forest)
etg_f	*Etage forestier*	Stand forest's level (1 or 2)
ess	*Code de l'* essence	Tree species ID
essence	*Nom de l'* essence	Tree species name
pcgha	*Proportion de l'essence considérée pour un étage et une structure dans le peuplement (% de la surface terrière)*	Share of the tree species considered within the tree stand (% of the basal area)
nha	*Nombre de tiges de l'essence à l'hectare" (n/ha)*	Number of tree species per hectare (n/ha)
gha	*Surface terrière de l'essence à l'hectare (m²/ha)*	Basal area of the tree species per hectare (m²/ha)
vha	*Volume bois fort tige de l'essence à l'hectare (m³/ha)*	Volume of wood per hectare (m³/ha)

Note that the first dataset (sheet "Placettes") contains two entries with geographic coordinates defining the plot location: one is derived from the IGN cartographic grid (which serves as the reference for the sampling), while the other one contains the GPS coordinates from the field. We choose to use the GPS coordinates from the field in prior. When the latter was not available, we used the one from the sampling grid to locate the plot.

3.1.1.3. Aggregation procedures

- *Alnus incana vs Alnus glutinosa*

The datasets' level of detail requires two types of aggregation for the purpose of this study:

- Aggregating observations over the type of Alnus
- Aggregation observations over the type of tree structure.

The dataset differentiates the *Alnus incana* and the *Alnus glutinosa*. When those two types of tree are observed within the same plot, we aggregated these under the general tree category "*Alnus*" in order to simplify the analysis. In total, Alnus is observed within 169 plots. Among these plots, 139 plots contain *alnus incana* or *alnus glutinosa* only, but 30 plots contain a combination of both species. For these cases, all variables were summed up.

- *Coppice vs tree stand*

Likewise, when the same type of allergenic tree was observed under two different forms (e.g. under the form of coppice and high stand forest of level 1), we also summed the two observations into one observation only.

Consequently, these two aggregations simplify the dataset for 285 plots (19.6%) where details about the type of Alnus observed and/or the tree species' structure (coppice, tree stand of level 1 or 2) are not available anymore.

3.1.1.4. Determination of predicted values for non-measured trees

The WFI contains an important number of non-measured trees whose circumference is lower than 20cm. As a result, the basal area in m²/ha is not provided for these observations (see first row in Table 1 for instance). In total, non-measured trees represent 36,7% of the dataset (690 observations). Yet, surveyors provided an estimated overall basal area in percent (variable "pcgha") per tree species and tree structure. Drawing upon this information, we computed an estimated basal area in m²/ha using values from measured trees.

The approach undertaken was two-fold as we considered pure tree stands (pcgha =100%) separately from those containing another tree species (pcgha < 100%). Indeed, a coppice of corylus under a tree stand with of basal area of 50% has not the same meaning than a pure coppice of corylus with a basal area of 100%. The latter's basal area (variable "gha" in m²/ha) may be higher than the former.

Therefore, for trees with a pcgha of 100%, we calculated an average gha per tree specie (Alnus, Betula, and Corylus) and structure (coppice vs tree stand). No values were calculated for corylus under the form of a tree stand as this type of tree structure does not exist. The following mean values ($\bar{x}$) were assigned:

	Coppice		Tree stand	
	N	$\bar{x}$	*N*	$\bar{x}$
Alnus	18	12.2	11	14.3
Betula	66	6.4	57	12.4
Corylus	13	2.6	*NA*	*NA*

Then, for tree species with a pcgha lower than 100%, we built five linear regression models. By establishing a statistical relationship between measured gha and pcgha, we were able to predict and assign a gha value for non-measured observations. As the figure below shows, the correlation between these two variables vary depending on the observations available per tree species (k) and structure (l). The weakest relationship was found for coppices of corylus (plot C; R^2 = 0,513), while the strongest relationship was observed for Alnus under the form of a tree stand (plot D; R^2 = 0,842).

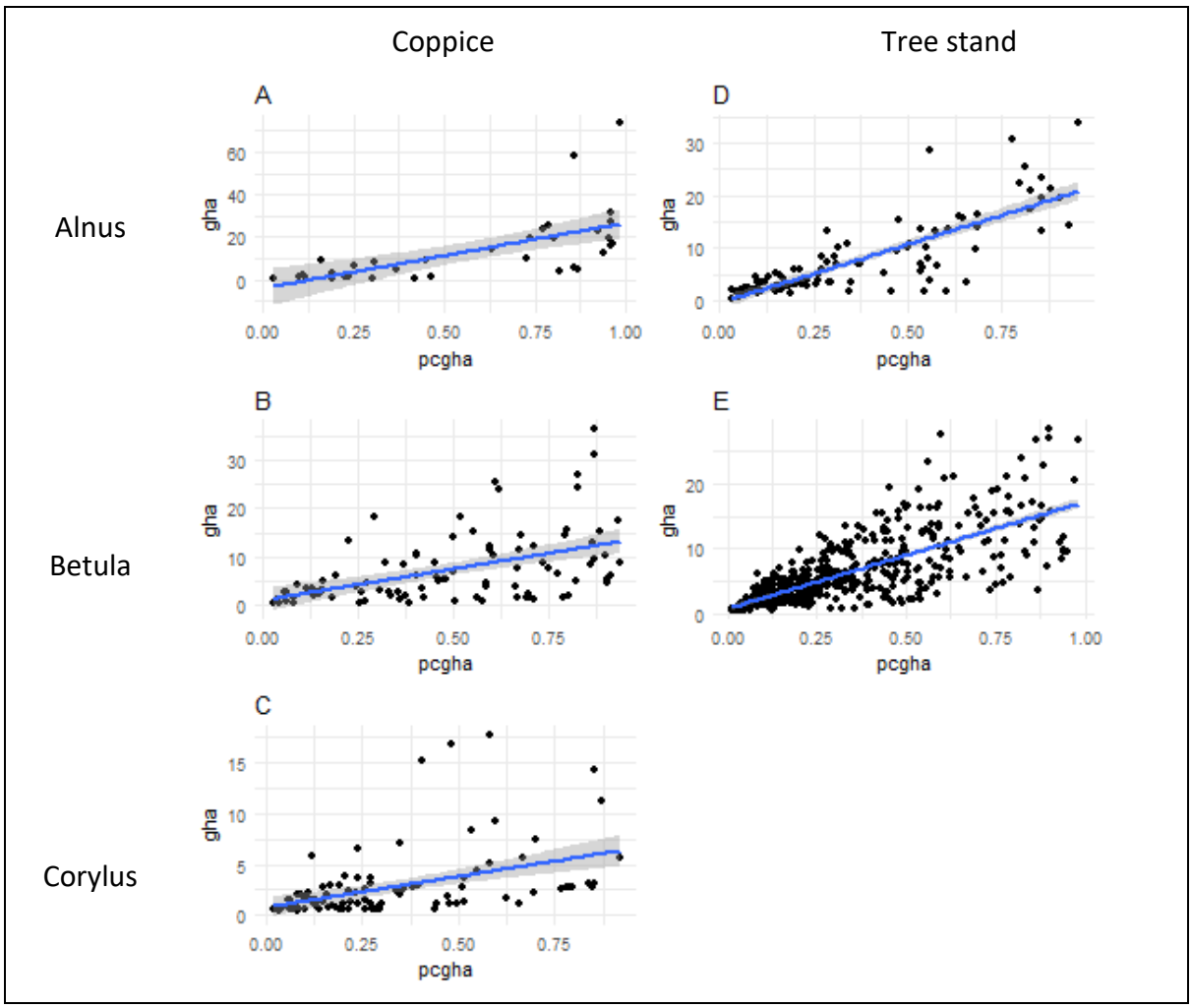
Predicted values were determined as follow:

$$\mathrm{gha_{jk}} = \mathrm{pcgha_{jk}} * \mathrm{E_{jkl}}$$

With: $\mathrm{gha_{jk}}$ = predicted basal area of tree specie k for plot j (m^2/ha)

$\mathrm{pcgha_{jk}}$ = field-estimated basal area of non-measured tree species k (%)

$\mathrm{E_{jkl}}$ = linear regression model estimate for tree specie k and strucure l (m^2/ha)



In total, 1.880 observations from the initial dataset were trimmed down to 1.722 observations after aggregation (8,4% decrease). These 1.722 observations of allergenic trees are found within a total number of 832 sampling plots. When joined to the entire database from the Walloon forest inventory, plots containing at least one allergenic tree represent 7,5% of the dataset.

3.1.2. Processing the Flemish forest inventory dataset

3.1.2.1. Dataset input tables

The Flemish Forest Inventory (FFI) database was provided under the form of an access database. It contains more than 83 tables with information about tree structure, plot types, surveyors, etc. (see Wouters et al., 2008). One table named "Trees_2eBosinv" was selected and exported for processing in R. Table 4 shows an extract of this table, including 9 selected variables used for the purpose of this study. Within the sample table below for instance, the two first lines correspond to two Alnus trees

(Species “10”). On the same plot (line 3 and 4), two Betula are also observed (Species “12”). All trees are alive (status tree = 1).

Table 5. Dataset extract with selected variables

IDPlots	X_m	Y_m	ID	Species	Perimeter_cm	Status_tree	DummyID_A3	DummyID_A3
157138	-4.936	-4.936	8	10	58	1	7	0
157138	0.257	0.257	1	10	30	1	1	0
157138	1.573	1.573	7	12	70	1	6	0
157138	2.887	2.887	6	12	30	1	5	0

Table 5 provides a definition of the key variables used for evaluating trees’ basal area and mapping species’ spatial distribution.

Table 6.Variables description

Variable	Description
IDPlots	Plot’s identification number
X_m	X geographic coordinate
Y_m	Y geographic coordinate
ID	Tree’s ID
Specie	Tree specie’s ID. Number referring to species’ name
Perimeter_cm	Tree perimeter at breast height
Status_tree	Tree status (dead or alive)
CodeCoppice_Individual	Coppice type

3.1.2.2. Data managment and selection

A comparison between the first and second forest inventories was made in order to identify the most relevant dataset to use. In this regard, the table "Trees" (considered as the 1st inventory) and the table "Trees_2eBosinv" (the 2nd forest inventory) were analysed compartively. The table "Trees_2eBosinv" is definitively the most comprehensive dataset. It contains 2.156 plots in total, while the table "Trees" contains 1.721 plots only. Besides, all plots (depite 24 exceptions) from the first inventory are found within the second one (see *Figure 2* below). The survey grid (red tiles in *Figure 2* or ‘rooster’ shapefile) allowed identifying areas were data are not up to date yet. These are plots where field surveys are planned for the year 2018 and 2017.

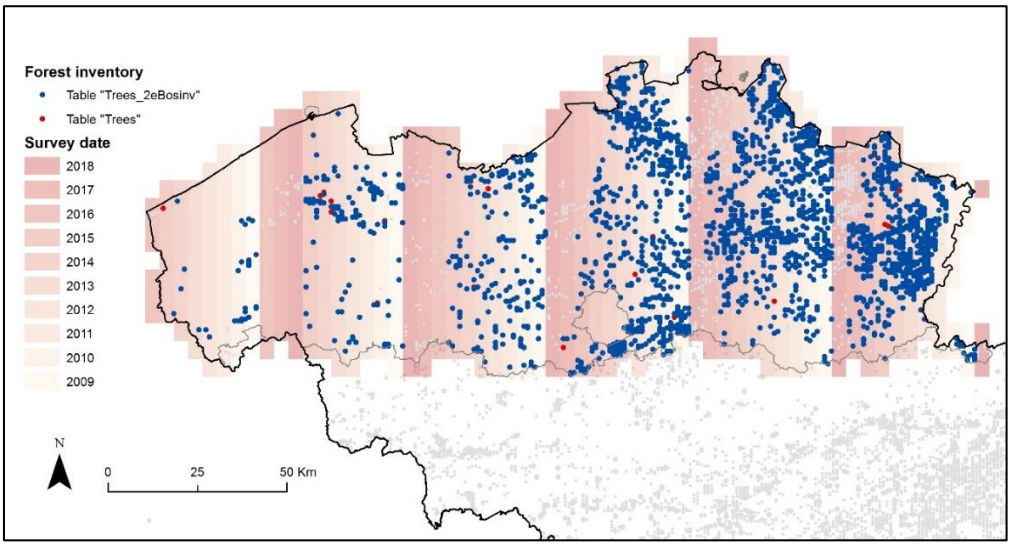


Figure 2. Sampling coverage of the first and second forest inventories

Such data spatial coverage gaps has to crarfully taken into account for the next modelling steps. In **Error! Reference source not found.***Figure 3* below, we illustrate these gaps by showing (i) the plots contained in the FFI database where, according to our calculations, there is either no allergenic tree (red points) or between 0% and 100% of the total basal area (green points). (ii) In yellow, the plots we expect to fall within a forested area (attirubte 'bos'=1 within the orignial 'invb2' shapefile that contains all the sampling plots). Their absence within the IFF dataset can be explained by the fact that either the plot does not exist anymore in the surey or has not been (re)surveyed yet, which explains why it is absent from the FFI database. (iii) The red rectangles consist of tiles from the 'rooster' shapefile (see Figure 3 above) that match with the plots extracted from the FFI database analysis. Finally, (iv) the grey rectangles are the tiles from the 'rooster' shapefile that do not match with the plots extracted from the FFI database analysis. Fortunately, since we received an updated version of the the Flemish forst inventory database in 2019, the yellow points and missing areas in grey are now included in the database and can be considered for predicting the distribution of allergenic tree species (see updated maps in the result sections).

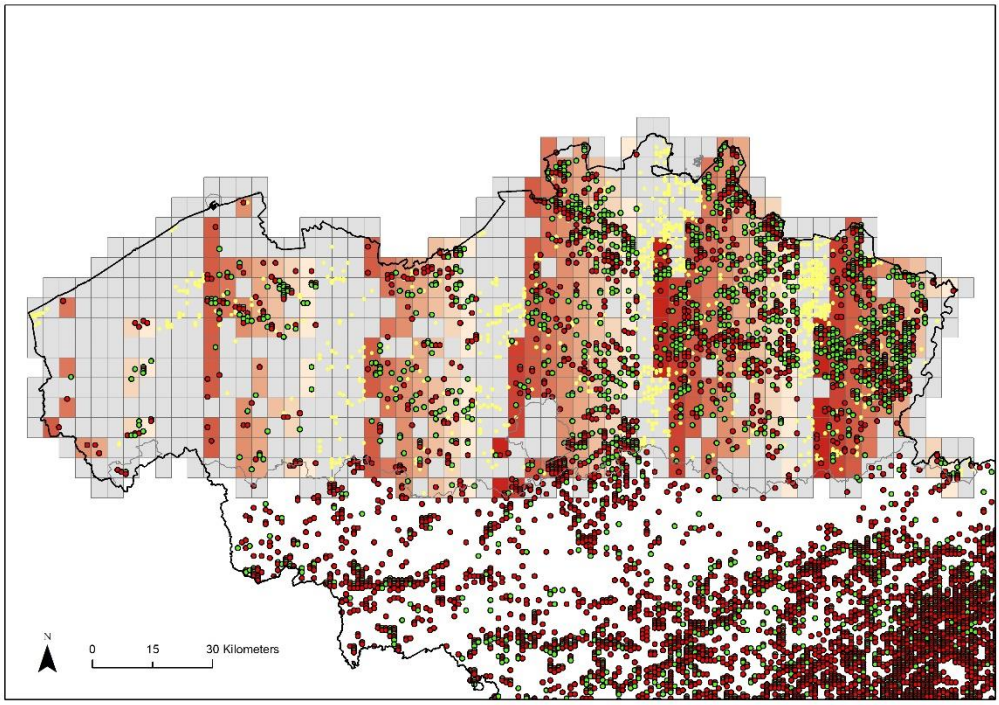


Figure 3. Illustration of data spatial coverage gaps within the Flemish forest inventory.

The dataset selected contains 43.582 observations in total. The first step needed for processing this dataset is to remove all dead trees (2.611 observations) and keep living trees using the variable "Status_tree". Then, a selection is operated to extract 3 allergenic tree species. This selection is based on the variable "species" (see table 6 below).

Table 7. List of allergenic tree species

Species	Dataset's original name	Scientific name
10	Zwarte els	Alnus glutinosa (L.) Gaertn.
11	Zwarte els	Alnus incana (L.) Moench
1470*	Hartbladige els*	Alnus cordata (Loisel.) Duby*
12	Berk	Betula tremula/alba
146*	Zachte berk*	Betula alba L.*
128*	Ruwe berk*	Betula pendula Roth*
14	Hazelaar	Corylus avellana L.

*Not considered for analysis

The FFI contains three additional types of Betula compared to the WFF (species "1470", "146", and "128"). So far, these were not considered for the analysis in order to facilitate the comparison between datasets. Not considering these tree species means deleting 7 observations in total. This low number of Betula alba L. and Betula pendula Roth is due to the lack of specification of tree species by fieldworks who do not specify the type of species when surveying the plots.

Number of observations	
	N
Alnus incana	217
Alnus glutinosa	2251
Betula	4938
Corylus	419

3.1.2.3. Calculation of tree species' basal area

Unlike the WFI, the FFI does not contain a variable describing trees' basal area. Drawing upon Afdeling Bos & Groen (2001, p.29-30), we thus calculated a basal area according to the following third-step procedure.

First, a basal area for each individual tree within a plot is calculated following this equation:

$$g_{ij} = \frac{c_{ij}^{\,2}}{4.\pi.10000}$$

With: g_{ij} = basal area of tree i wihtin the sampling plot j (m^2)

c_{ij} = perimeter of tree i within the sampling plot j (cm), i.e. variable "Tree_perimeter".

Second, a total basal area per hectare is calculated depending on the type of sampling plot (A2, A3 or A4).

$$G_{jk} = \left(\sum_{n2jk} g_{ijk} \times F_2\right) + \left(\sum_{n3jk} g_{ijk} \times F_3\right) + \left(\sum_{n4jk} g_{ijk} \times F_4\right)$$

With: G_{jk} = total basal area of allergenic trees k within the sampling plot j (m^2/ha)

g_{ijk} = basal area of tree i within the sampling plot j (m^2)

$n_{2jk}, n_{3jk}, n_{4jk}$ = number of trees k from plot type A2, A3, and A4 within plot j

F_2, F_3, F_4 = extension factor for plot type A2, A3, and A4

The extension factor used were:

- plot type A1: R1 = 2,25 m → F1 = 628,76
- plot type A2: R2 = 4,5 m → F2 = 157,19

- plot type A3: R3 = 9 m → F3 = 39,30
- plot type A4: R4 = 18 m → F4 = 9,82

Four plot types were identified using variables “perimter” and “Height” based on the following conditions detemrined by Afdeling Bos & Groen (2001, p.29-30):

- plot type A1: “zaailingen” with a total heigh < 2m
- plot type A2: Coppice; trees with perimeter < 22cm and a total height >= 2m
- plot type A3: Trees with perimeter between 22cm and 122cm
- plot type A4: Tree with perimeter > 122cm

Third, after having calculated this basal area for allergenic tree species (G_{jk}), a second basal area (G_{jf}) is caclulated for all tree species observed within the samping plot.

Drawing upon these two values, a relative global basal area in percent is finally measured as follow:

$$Gj = \frac{\mathrm{Gjk}}{\mathrm{Gjf}} * 100$$

With: G_j = relative basal area of allergenic trees k for plot j (%)

G_{jk} = basal area of allergenic tree species (m^2/ha)

G_{jf} = basal area of all tree species (m^2/ha)

3.1.3. Creating a Belgian forest inventory dataset

At the outset, we created a Belgian detaset where both Flemish and Walloon dataset were merged. A shapefile named 'data_sp_BEL_pcGHA_FULL' was built, which contains absolute and relative basal areas values (see table 7 below for variable details). The step-by-step computation method from R scripts can also be found in Appendix under the section named " Belgian Forest Inventry (RespirIT_IF_BEL.R)".

Table 8. Details on output variables

Acronym	Denomination	Description	Unit	Values
ID_BEL	Plot's identification number	In Wallonia a unique ID was built by merging the field 'ign' (ngi-based sampling grid number) and 'npl' (*numéro de placette*) contained in the forest inventory database. In Flanders, the unique ID provided in the database (field 'IDPlots') was kept.	*NA*	*NA*
X	Plot's X coordinate	X geographic coordinate in Belge Lambert 1972 extracted from the forest inventories database	m	Continuous
Y	Plot's Y coordinate	Y geographic coordinate in Belge Lambert 1972 extracted from the forest inventories database	m	Continuous
X_UTM	Plot's Y coordinate in WGS84	Own conversion in WGS 1984 (Web Mercator Auxiliary Sphere) using rgdal's CRS function in R. See https://epsg.io/31370 for transformation details (PROJ.4)	Decimal degrees (°)	Continuous
Y_UTM	Plot's X coordinate	Own conversion in WGS 1984 (Web Mercator Auxiliary Sphere) using rgdal's CRS function in R. See https://epsg.io/31370 for transformation details (PROJ.4)	Decimal degrees (°)	Continuous
Region	Region name	Plot's region. This dataset does not include the Brussels region.	*NA*	Categorical (Wallonia/ Flanders)
gha_ALL	Total basal area of *all* tree species	Total basal area of *all* tree species measured within the sampling plot	m²/ha	Continuous
gha_A	Basal area of *alnus*	Basal area of *alnus* tree species measured within the sampling plot	m²/ha	Continuous
pGHA_A	Relative basal area of *alnus*	*Alnus*' percentage of total basal area	%	Continuous (0-100)
gha_B	Basal area of *betula*	Basal area of *betula* tree species measured within the sampling plot	m²/ha	Continuous
pGHA_B	Relative basal area of *betula*	*Betula*'s percentage of total basal area	%	Continuous (0-100)
gha_C	Basal area of *corylus*	Basal area of *corylus* tree species measured within the sampling plot	m²/ha	Continuous
pGHA_C	Relative basal area of *corylus*	*Corylus*' percentage of total basal area	%	Continuous (0-100)
gha_3	Total basal area of *3 selected* tree species	Total basal area of 3 selected tree species (*Alnus, Betual, Corylus*) measured within the sampling plot	m²/ha	Continuous
pGHA_3	Relative basal area of the *3 selected* tree species	3 selected tree species' percentage of total basal area	%	Continuous (0-100)

Nb: For basal area calculation details in Wallonia and Flanders see p.6 and p.10 respectively. For a discussion on data spatial coverage gaps in Flanders see p.11.

3.1.4. Results

3.1.4.1. Tree structure and tree mix

From the Walloon forest inventory dataset, details about the structure of trees can be provided. The table below shows that *Alnu*s are mainly observed in the first level (highest level) of any stand structure (even-aged stand with one layer, even-aged stand with two layers or uneven-aged stand). *Betula* is also mainly observed in the first level (highest level) of any stand structure. Yet, an important number of trees are also observed under the form of coppice. *Corylus* on the opposite is only observed in stand structures with coppice (i.e. coppice stand, coppice with high stand or high stand with coppice).

Table 9. Number of observations based on the type of tree structure

	Coppice	High stand forest	
		Level 1	Level 2
Alnus incana	13	21	1
Alnus glutinosa	55	109	4
Betula	370	617	46
Corylus	644	0	0

Tree species combine in various ways within each plot. Within the Walloon forest inventory dataset, we identified seven possibilities depending on how allergenic tree combine. In most cases (82%), plots contain one type of tree only. Almost one fifth (17%) of the plots contain a combination of two species. One percent only contains the three species studied.

Table 10. Combination of allergenic tree species in sampling plots

	1	2	3	4	5	6	7
Alnus	X			X	X		X
Betula		X		X		X	X
Corylus			X		X	X	X
TOTAL	*83*	*682*	*433*	*44*	*28*	*169*	*14*
%	*82%*			*17%*			*1%*

3.1.4.2. Presence-absence of allergenic trees

In Wallonia, most forest areas containing the three allergenic tree species studied (alnus, betula, corylus) are located in the southern part of the region. In particular, 90,9% of the sampling plots containing at least one allergenic tree species are located south of the “sillon Sambre et Meuse” (i.e. the former industrial axis running from the city of Mons to Liège through Charleroi and Namur). The highest densities of allergenic tree species can be observed on the northern border of the Ardenne region along both sides of the Meuse river around the French city of Givet. One of the reason explaining such a concentration is the presence of small parcels containing a mix of tree species.

While the "plateau ardennais central" and the "plateau ardennais du nord-est" concentrate large forest areas, these areas are characterized by a low density of allergenic tree species. This may be related to the following factors: (i) soil types are not the best suited for the development of either Alnus, Betula or Corylus, and (ii) an important number of parcels only contain coniferous trees (e.g. "Haut plateau des Fagnes", the German border area) where tree mix is less important. Meanwhile, areas containing the lowest density of allergenic trees in Wallonia are the "plateaux limoneux brabançons et hesbignons" were large parcels of agricultural land are dominant, leaving less space for forested areas.

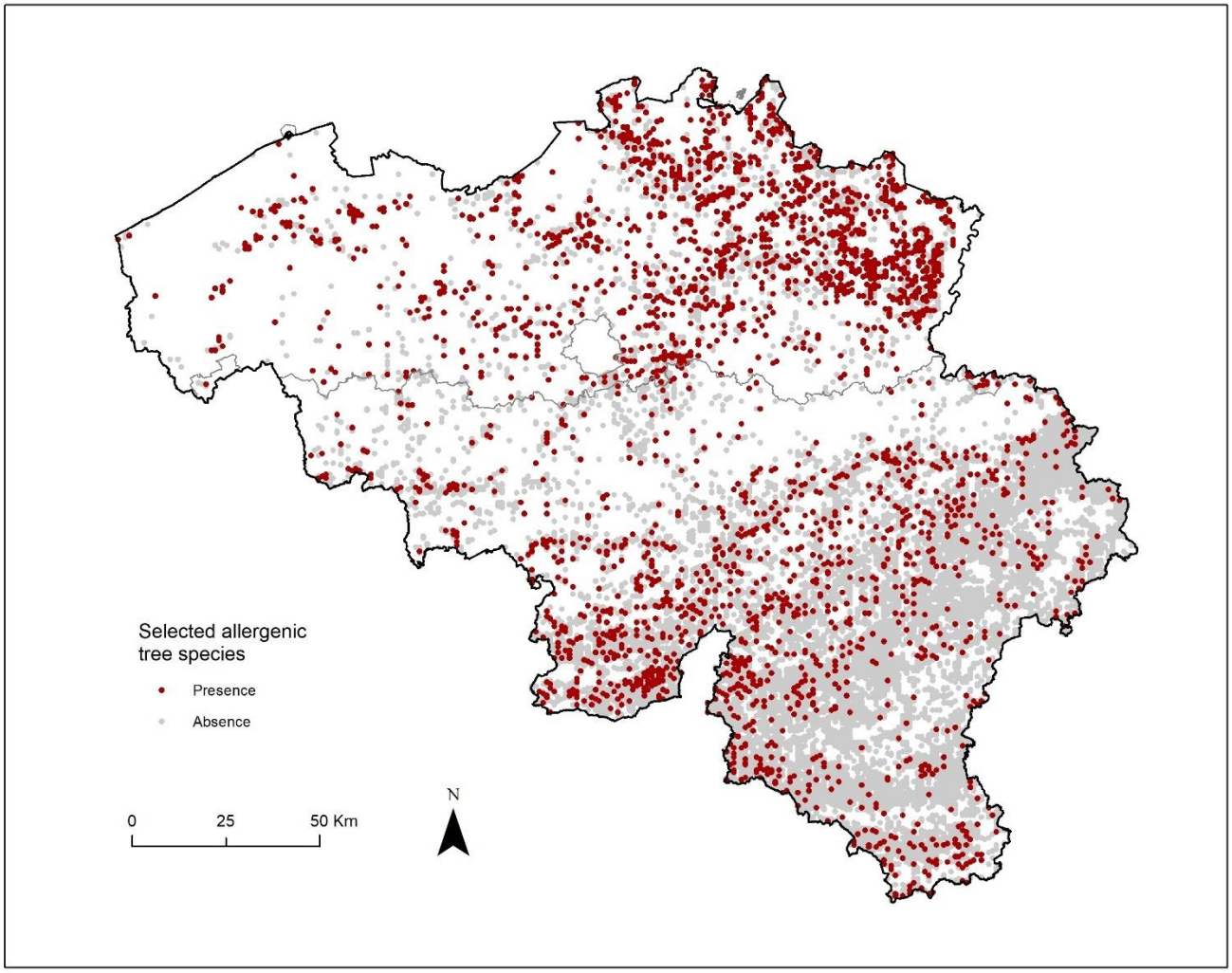


Figure 4. Combined sampling plots from the Walloon and Flemish forest inventories with presence-absence of 3 selected allergenic trees (alnus, betula, corylus)

3.1.4.3. Relative share of allergenic trees

The relative share of allergenic trees was calculated as an important indicator for describing the importance and spatial distribution of allergenic trees throughout Belgium. This variable (named "pcgha") represents the share of allergenic trees within the plot sampled calculated as percentage of the basal area. This percentage was first computed for each tree species individually. Then, all 3 values were summed up to provide an overall percentage.

As the violin plot below shows, *alnus* and *betula* present median values around 20% (red dots). The presence of *corylus* is even lower with median values lower than 5%. Histograms' distributions (grey areas) show that a high proportion of plots contain very low shares of allergenic trees. When considering all tree species together, the average share rises to 15%. Such patterns suggest that allergenic trees species are most often combined with other (non-allergenic) trees species, which may offer a postitive effect by capturing the pollens emitted by the neighbour species.

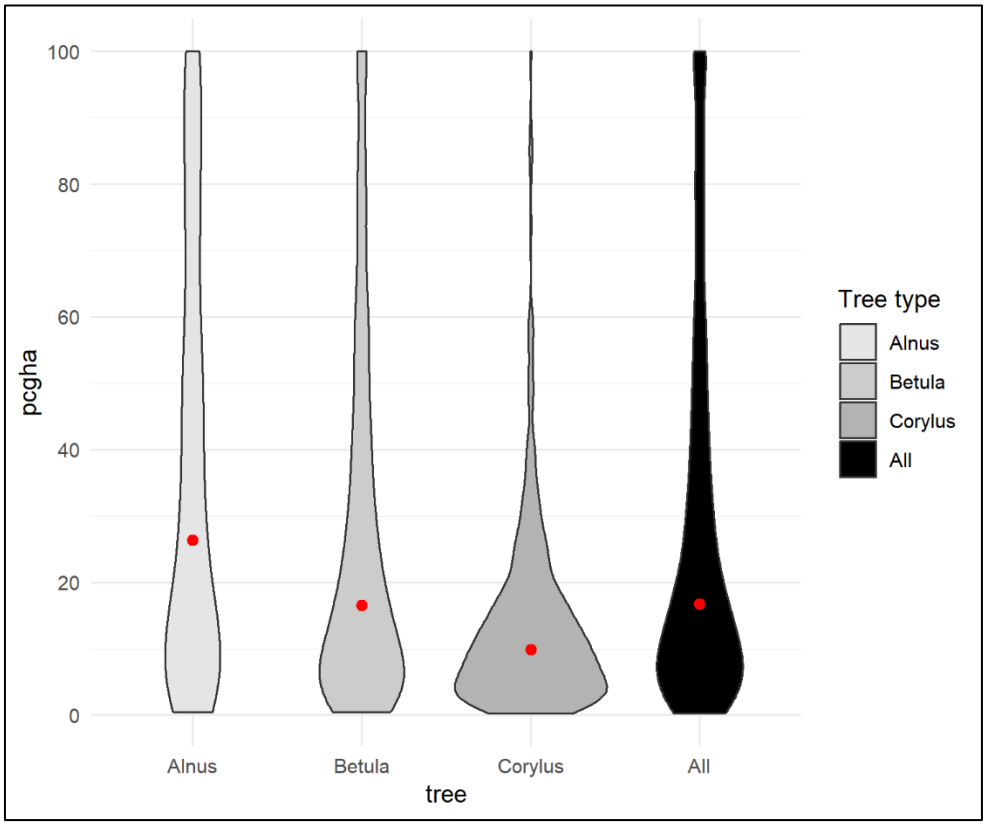


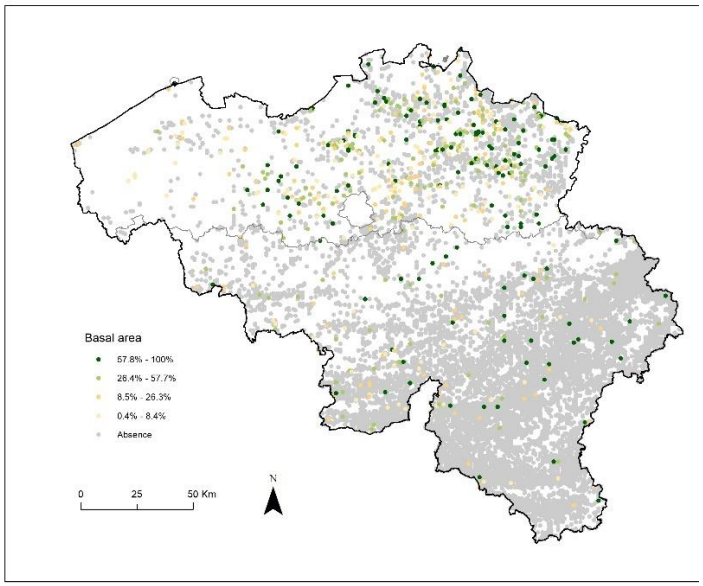
Table 11 below summarises that allergenic trees are present within Belgian forest area in fairly low percentages with average basal areas ranging from 8,7% (Corylus) to 35,7% (Alnus). In total, 1.934 plots out of the 13.228 plots surveyed in the Flemish and the Walloon forest inventories contain at least one allergenic tree.

Table 11. Summary statistics

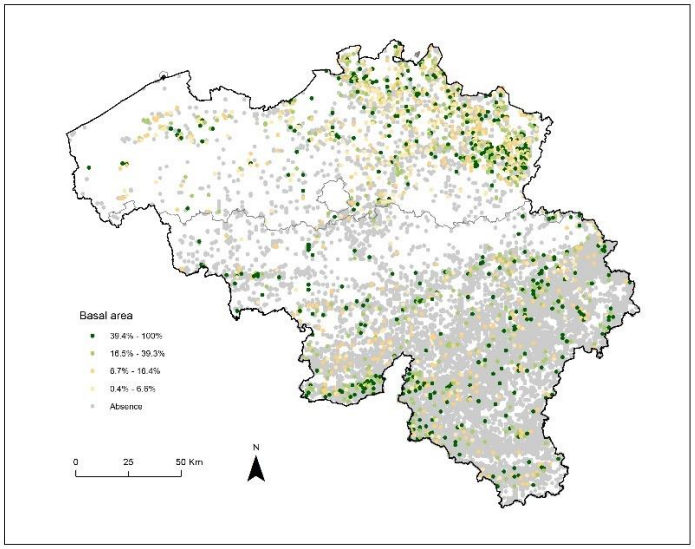
	Sampling plots with/without allergenic trees	Average basal area (%)			
		Alnus	Betula	Corylus	3 selected allergenic trees
Flanders	1.102 / 2.147	33,4	23,1	3,7	27,5
Wallonia	832 / 11.081	41,3	32,3	11,8	29,9
Total	1.934 / 13.228	35,7	27,0	8,7	28,5

3.1.5. Spatial distribution of alnus, betula and corylus

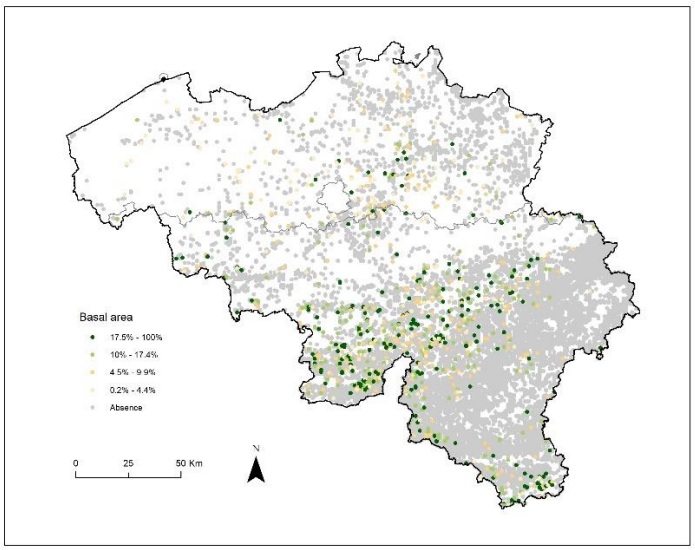
The following maps show the presence and absence of each tree species individually. A four-class categorisation using the quantile method was used to represent differences in the relative presence of the 3 selected allergnic trees. Note that a comparison of categories between species should be avoided as tresholds vary from one specie to another (see values' distribution patterns within the violin plot above). Alnus is the less represented specie throughout the study area. Its spatial distribution presents a scattered pattern with no specific concentration at the regional level. No important densities are found in any of the sub-regions. Betula is more common throughout the country and presents higher densities. At the sub-regional level, it is well represented in the South of the "sillon Sambre et Meuse" and the northeast area of Flanders. The presence of Betula is often associated with major forested areas. The density of corylus is lower than Betula, but higher than Alnus. It is concentrated outside the major forested area. In Wallonia for instance, Betula is located outside the major productive forest areas from the "plateaux ardennais".



Alnus



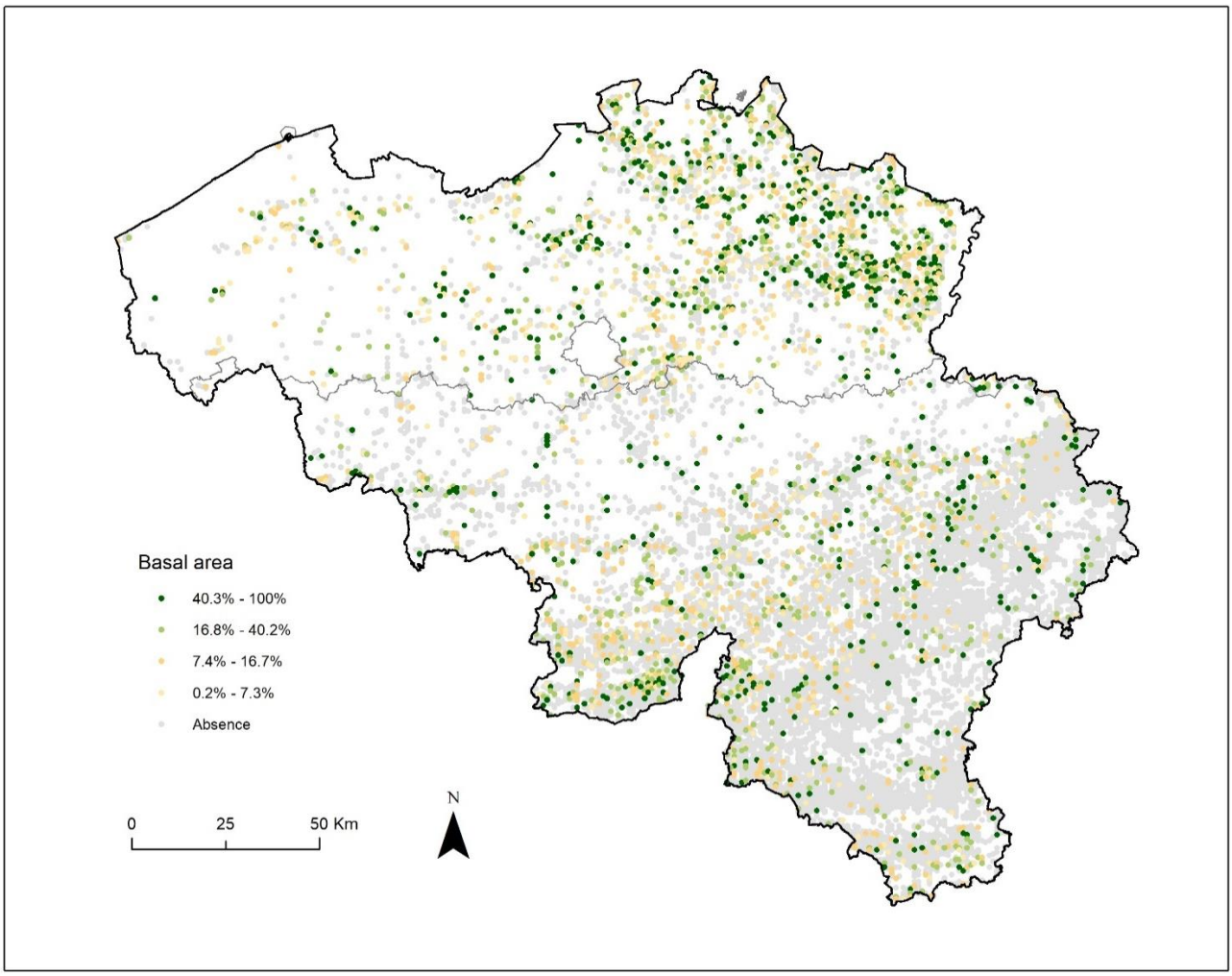
Betula



Corylus

3.1.6. <u>The spatial distribution of 3 selected allergenic tree species in Belgium</u>

The following map pictures the spatial distribution of the percentage of all allergenic trees (variable "pcgha"). No clear pattern emerges as to describe where the highest and lowest percentages are located. The percentage of allergenic trees per plot is heterogeneous throughout the region. This suggests that the relative importance of allergenic trees relies more upon local factors such as soil types for instance.


Basal area
40.3% - 100%
16.8% - 40.2%
7.4% - 16.7%
0.2% - 7.3%
Absence
0
25
50 Km
N

3.2. Non-forest inventory data

Several options were explored for gathering data outside forested areas in this research.

- For the Brussels region, a well detailed vegetation inventory made by Bruxelles Evironnement – IBGE exist. A detailed inventory of presence-only tree species within the public domain is freely available upon request by the organisation but could not be processed in the framework of this project.
- Some project-specific initiatives also exist such as those undertaken during the RespiIT project. For instance, the KULeuven team undertook fieldwork using Durham samplers. Around each sampler, they searched for allergenic tree species in a 2 km radius (12,6km²). They surveyed 15 points during a search effort of 10 minutes with two people (= 0,33 person-hours). The total basal area and birch basal area were measured.
- The Flemish flora database provides a presence-only dataset available for Flanders at a 4km square resolution. The KUL team implemented a Specie Distribution Model (SMD) using Maxent with this data. The low resolution of this dataset did not allow for considering this source as a possible option for our mapping of allergenic tree species.
- Data recrods from Observations.be / Waarnemingen.be is a crowdousourcing platform where volunteers, researchers, and scientists share their plant and animal sightings.

The latter data source was considered as the best candidate for characterizing the presence-absence of allergenic tree species outside forested areas.

3.2.1. Observations.be / Waarnemingen.be

The non-profit organisation "Observation International" provides a website named Observation.org or with many regional aliases, including observations.be/.waarnemingen.be that can be accesed with client applications such as ObsMapp (Android), iObs (iOS) and WinObs (Windows). Builiding upon these platforms, volunteers, researchers and scientists can make field observations and report them at any time. The inclusion of such type of data into a global digital system creates a powerful tool for multiple applications, including conservation, research, policy, experience and education. The main objective of International Observation is "the optimum facilitation of observers in order to make their nature experience even more valuable". Yet, records from such a type of initiatives are opportunistic, less structured and mostly incidental observations. They are often suspected to be information-poor compared to information-rich database from forest inventories (Isaac & Pocock, 2015). However, open citzen data also contain greater number of records gathered (see table 12).

3.2.2. Number of records and frequency

As shown in table 12 and figure 5 below, an important diversity of tree specie records could be accessed freely for this research via the observations.org platform. A specific selection of the three targeted allergenic tree species for the 2008-2018 period could be imlplemented by selecting the most relevant species of Alnus, Betula, and Corylus.

Table 12. List of allergenic tree species

	Species name (latin)	Count
2	Alnus glutinosa	8.200
3	Alnus incana	1.014
1	Alnus cordata*	187
5	Alnus x pubescens (A. glutinosa x incana)	7
6	Alnus x spaethii (A. japonica x subcordata)*	4
4	Alnus spec.	235
8	Betula pendula	7.231
10	Betula pubescens	2.106
7	Betula papyrifera*	2
9	Betula pendula + Betula pubescens	481
12	Betula x aurata (B. pendula x pubescens)	7
11	Betula spec.	1.838
13	Corylus avellana	9.416
14	Corylus colurna*	31
15	Corylus maxima*	16
16	Corylus spec.	4
	TOTAL	30.820

* Not considered for analysis

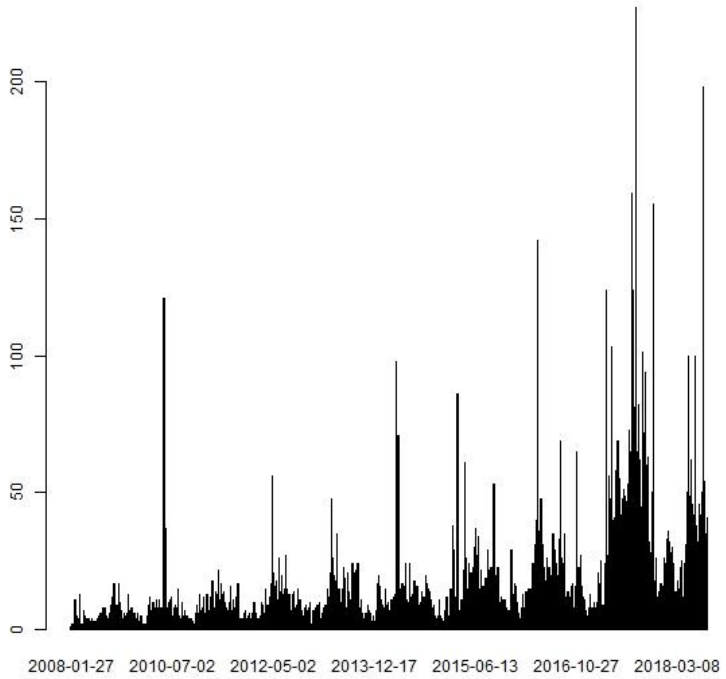


Figure 5. Frequency of opportunistic observations from the observations.org database for the 2008-2018 perdiod.

3.2.3. Spatial footprint

The spatial footprint of opportunistic observations covers the entire country (see Figure 6 below). The location of records extends way beyond forest areas. Many cities and densely populated areas contain an important number of observations from volunteers (see figure 7). Obsevations in city parks, suburban areas, and large infrastrcutures were also reported by participants. A small regional effect can be observed around the 'Pays des collines' – 'Vlaamse Ardennen' border, which may be explained by differences of behaviour between volunteers (differences of interest and/or levels of awareness about the *obsevations.be* initiative), but his effect disappears in the southern periphery of Brussels were observation are continuous from the center the periphery of the city. Higher point density areas also include recreational areas (e.g. forested areas around major cities, touristic valleys) and spaces located along transport infrastructures such as walkways and roads.

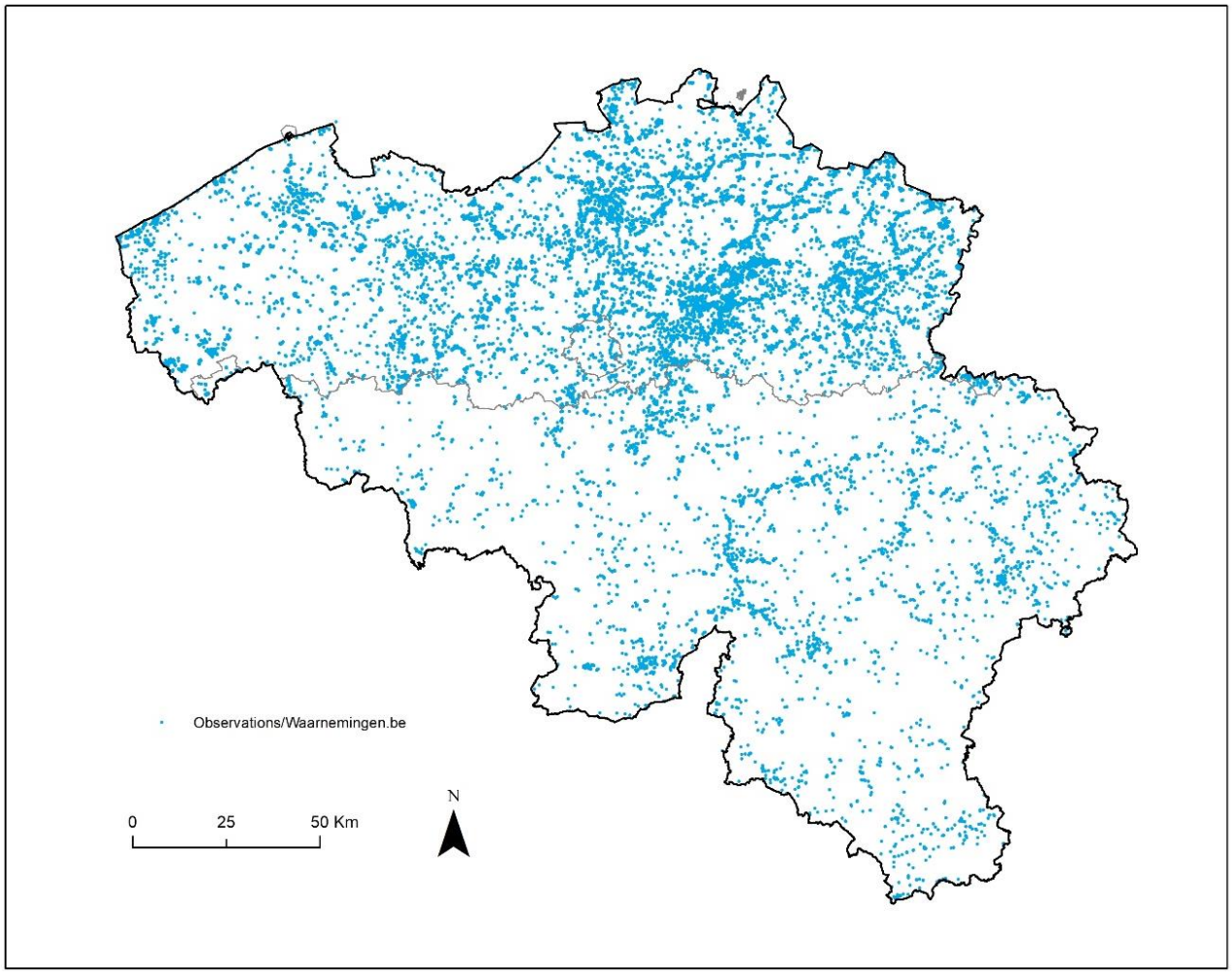


Figure 6. Spatial distribution of presence-only records from the obsevations.org database for the 3 selected allergenic tree speciess (Alnus, Betula, Corylus)

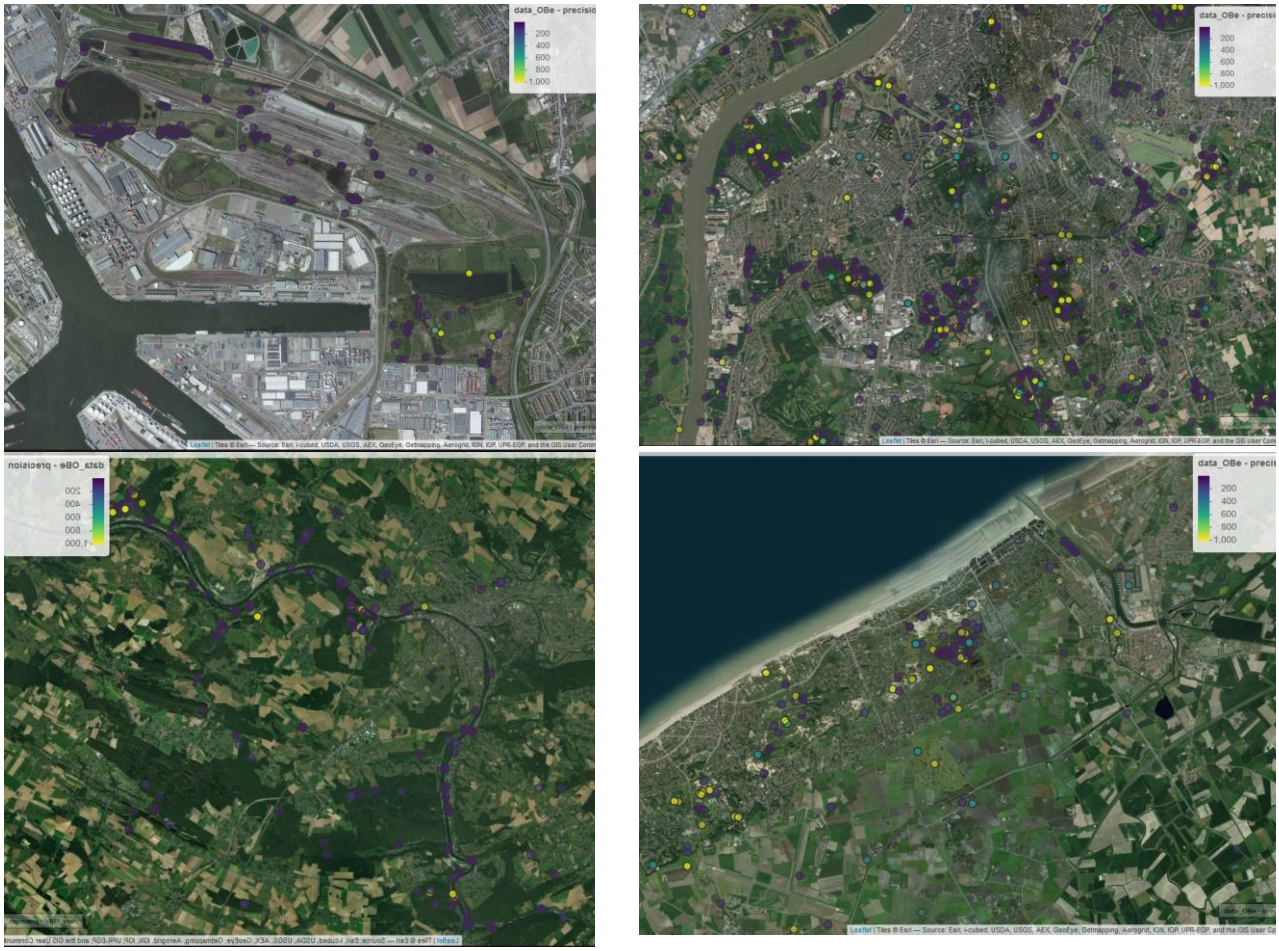


Figure 7. Detailed location of opportunistic observations from the observations.org database in the harbour of Antwerp (upper left), the city outskirst of Antwerp (upper right), the area of Namur city (lower left) and the coastal city of Niewport (lower right). Colours show the level of point occuracy (0-1000m).

3.2.4. *Complementarity with forest inventories*

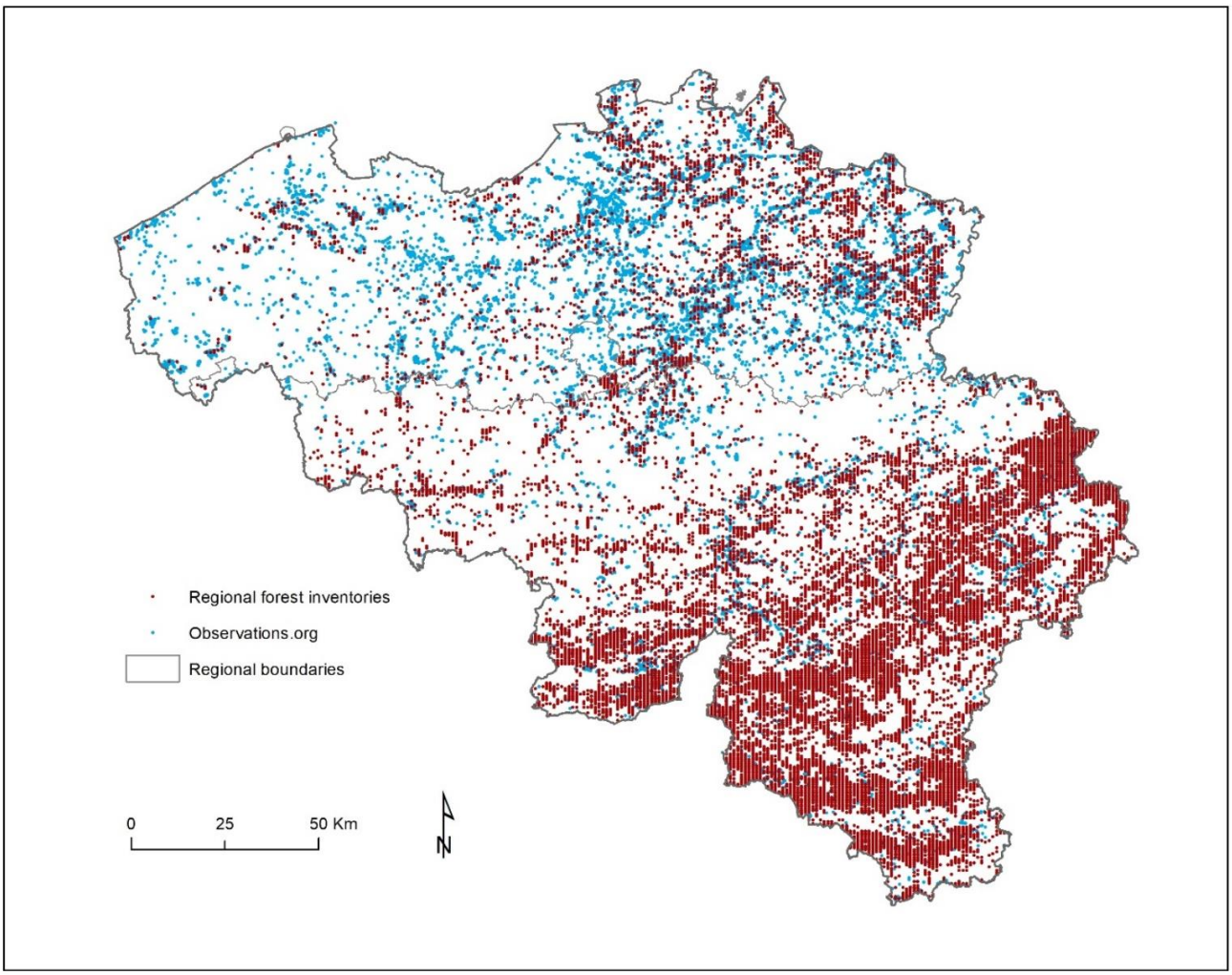


Figure 8. Combined presence-only records from regional forest inventory dataset and obsevations.org database for the 3 selected allergenic tree speciess (alnus, betula, corylus)

4. Conclusions

Forest inventories provide a detailed account of tree species observed within forest areas. The plots sampled offer a high-resolution, up-to-date description of the selected allergenic tree species (*Alnus, Betula*, and *Corylus*) at the scale of a land parcel. This has great potential for ecological modelling because the share of tree species within a tree stand (in percent of the basal area or percent cover) is a key variable for measuring densities of allergenic trees and characteritze its relative abundance of within a forested land parcel. Our results showed that allergenic trees are present within forest areas in relatively low percentages: three-fourth of the sampling plots contain less than 45% of allergenic tree species. Most allergenic trees are observed in the north-eastern part of Flanders and the south of the "sillon Sambre and Meuse". The country's major productive forests do not concentrate the most important share of allergenic trees. The observation records from open citizen science initiatives such as the *observations.org* platform are opportunistic, less structured and mostly incidental observations. While they are often suspected to be information-poor compared to information-rich database from forest inventories, we showed here that they contain a greater number of records gathered at higher time frequency. The spatial footprint of allergenic tree records is much larger than forest inventories and covers many areas outside forested areas. This complementariy should be considered as an opportunity for building comprenhensive database of presence-absence database that can better serve the elaboration of Species Distribution Models (SDMs).

5. Cited references

AFDELING BOS & GROEN 2001. De bosinventarisatie van het Vlaamse Gewest. Resultaten van de eerste inventarisatie 1997-1999. Ministerie van de Vlaamse Gemeenschap.

ALDERWEIRELD, M., BURNAY, F., PITCHUGIN, M. & LECOMTE, H. Inventaire Forestier Wallon-Résultats 1994-2012. 2015. SPW.

ALLOUCHE, O., TSOAR, A. & KADMON, R. 2006. Assessing the accuracy of species distribution models: prevalence, kappa and the true skill statistic (TSS). *Journal of applied ecology,* 43**,** 1223-1232.

ELITH, J., GRAHAM, C. H., ANDERSON, R. P., DUDÍK, M., FERRIER, S., GUISAN, A., HIJMANS, R. J., HUETTMANN, F., LEATHWICK, J. R. & LEHMANN, A. 2006. Novel methods improve prediction of species' distributions from occurrence data. *Ecography,* 29**,** 129-151.

ELITH, J. & LEATHWICK, J. R. 2009. Species distribution models: ecological explanation and prediction across space and time. *Annual review of ecology, evolution, and systematics,* 40**,** 677-697.

GUISAN, A., ZIMMERMANN, N. E., ELITH, J., GRAHAM, C. H., PHILLIPS, S. & PETERSON, A. T. 2007. WHAT MATTERS FOR PREDICTING THE OCCURRENCES OF TREES: TECHNIQUES, DATA, OR SPECIES'CHARACTERISTICS? *Ecological monographs,* 77**,** 615-630.

HEIKKINEN, R. K., MARMION, M. & LUOTO, M. 2012. Does the interpolation accuracy of species distribution models come at the expense of transferability? *Ecography,* 35**,** 276-288.

HILL, L., HECTOR, A., HEMERY, G., SMART, S., TANADINI, M. & BROWN, N. 2017. Abundance distributions for tree species in Great Britain: A two-stage approach to modeling abundance using species distribution modeling and random forest. *Ecology and Evolution,* 7**,** 1043-1056.

HOPKINS, J. J. & KIRBY, K. J. 2007. Ecological change in British broadleaved woodland since 1947. *Ibis,* 149**,** 29-40.

MCINNES, R. N., HEMMING, D., BURGESS, P., LYNDSAY, D., OSBORNE, N. J., SKJØTH, C. A., THOMAS, S. & VARDOULAKIS, S. 2017. Mapping allergenic pollen vegetation in UK to study environmental exposure and human health. *Science of the Total Environment,* 599**,** 483-499.

PEARSON, R. G. & DAWSON, T. P. 2003. Predicting the impacts of climate change on the distribution of species: are bioclimate envelope models useful? *Global ecology and biogeography,* 12**,** 361-371.

PRENTICE, I. C., CRAMER, W., HARRISON, S. P., LEEMANS, R., MONSERUD, R. A. & SOLOMON, A. M. 1992. Special paper: a global biome model based on plant physiology and dominance, soil properties and climate. *Journal of biogeography***,** 117-134.

R CORE TEAM 2015. R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. 2015.

RACKHAM, O. 2008. Ancient woodlands: modern threats. *New Phytologist,* 180**,** 571-586.

RONDEUX, J. & LECOMTE, H. 2010. Inventaire Permanent des Ressources Forestières de Wallonie-Guide méthodologique.

THUILLER, W. 2003. BIOMOD–optimizing predictions of species distributions and projecting potential future shifts under global change. *Global change biology,* 9**,** 1353-1362.

WOUTERS, J., QUATAERT, P., ONKELINX, T. & BAUWENS, D. 2008. Ontwerp en handleiding voor de tweede regionale bosinventarisatie van het Vlaamse Gewest. *Report INBO*.

1. Appendix: R scripts

1.1. Walloon Forest Inventory (RespirIT_IFW_5.R)

```
#################### SUMMARY ###################################

# PACKAGES
# OVERALL APPRAOCH
# DATA IMPORT
# DATA PROCESSING
# PLOTTING
# FINAL TABLE
# MAPPING

#################### PACKAGES ###################################

library("rJava", lib.loc="~/R/win-library/3.3")
#library("xlsx", lib.loc="~/R/win-library/3.3") #! If activated > conflict with
XLConnect
library("XLConnect", lib.loc="~/R/win-library/3.3")
library("reshape", lib.loc="~/R/win-library/3.3")
library("reshape2", lib.loc="~/R/win-library/3.3")
library("stringi", lib.loc="~/R/win-library/3.3")
library("stringr", lib.loc="~/R/win-library/3.3")
library("shapefiles", lib.loc="~/R/win-library/3.3")
library("plyr", lib.loc="~/R/win-library/3.3")
library("rgdal", lib.loc="~/R/win-library/3.3")
library("vioplot", lib.loc="~/R/win-library/3.3")
library("ggplot2", lib.loc="~/R/win-library/3.3")
#install.packages('ggplot2', repos='http://cran.rstudio.com', type='source')
library("shapefiles", lib.loc="~/R/win-library/3.3")
library("rgdal", lib.loc="~/R/win-library/3.3")

#.rs.restartR()

##  FUNCTIONS ##

# Multiple plot function

multiplot <- function(..., plotlist=NULL, file, cols=1, layout=NULL) {
  library(grid)

  # Make a list from the ... arguments and plotlist
  plots <- c(list(...), plotlist)

  numPlots = length(plots)

  # If layout is NULL, then use 'cols' to determine layout
  if (is.null(layout)) {
    # Make the panel
    # ncol: Number of columns of plots
    # nrow: Number of rows needed, calculated from # of cols
    layout <- matrix(seq(1, cols * ceiling(numPlots/cols)),
                     ncol = cols, nrow = ceiling(numPlots/cols))
  }

  if (numPlots==1) {
    print(plots[[1]])

  } else {
    # Set up the page
    grid.newpage()
    pushViewport(viewport(layout = grid.layout(nrow(layout), ncol(layout))))

    # Make each plot, in the correct location
    for (i in 1:numPlots) {
      # Get the i,j matrix positions of the regions that contain this subplot
      matchidx <- as.data.frame(which(layout == i, arr.ind = TRUE))

      print(plots[[i]], vp = viewport(layout.pos.row = matchidx$row,
                                      layout.pos.col = matchidx$col))
    }
  }
}


#################### OVERALL APPROACH ############################

# Work on plots ("placettes" in french).
# Work on species ("essences" in french).
# Variables creation (1 by 1)
# Operations (splitting, merging, aggregating, etc.)
# Recompilation for export into shapefiles

# Data IMPORT (data_sp, data_pl)
# DATA PROCESSING:
# Plots containing measured trees (A)
# Plots containing 1 non-measured tree or more (B)
# (a) Approach per class
# (b) Approach by linear regression model
# Merging both datasets (A & B)
# Plots containing only 1 tree (C1)
# Plots contraining more than one tree (C2)
# C1_a) Plots with only one "Aulne" (or none)
# C1_b) Plots with more than one "Aulne"
# Re-building "data_spW__C2" (C2_a & C2_b)
# Aggregation over fut & tai for measured trees with more than 1 tree per plot
(C2)
# Re-building "data_spW__C2"
# Re-building "data spW  C1"
# Re-building "data_spW__C"
# Caculating overall GHA values
# a) Alnus, Betual, Corylus
# b) All 3
# PLOTTING
# MAPPING
```

```
#################### DATA IMPORT ###################################

setwd("C:/Users/dujardis/Documents UNamur/20160401 RespirIT")

# Sheet "Placettes"
# Describes the characteristics of each plot
# Tells us about where plots are located and what kind of general species are
found (structure, peuplement).

temp<-loadWorkbook("C:/Users/dujardis/Documents                UNamur/20160401
RespirIT/2_DATA/Inventaire              forestier              Wallon/20160902
IFW/Req_dujardin_010916_VF.xlsx")
data_plW<-readWorksheet(temp, sheet = "Placettes", startRow = 16)

# 1) ID_Pl
# Creation of a unique identifier for each plot

data_plW$ID_pl <- paste(data_plW[,"ign"],data_plW[,"npl"],sep="")

data_plW$ID <- 1:nrow(data_plW) # Unique ID

#colnames(data_plW)

#data <- c(ID_plW)
#data <- data.frame(c(1:nrow(data plW)))
#colnames(data) <- c("ID_pl")

data_plW$ID_X <- 0
data_plW$ID_Y <- 0

# 2) XY coordinates
# To identify the location of each plot,
#  if there are GPS coordinates (x_gps/y_gps variable), then keep the GPS
coordinates.
#  If it is not available (NA), then use the "IGN grid coordinates" (x/y
variable).

#ID_X
for(i in 1:nrow(data_plW)) {
  if( !is.na(data_plW[i,"x_gps"])) {
    data_plW[i,"ID_X"] <- data_plW[i,"x_gps"]
  }
  else {data_plW[i,"ID_X"] <- data_plW[i,"x"]
  }
}
data_plW$ID_X

# ID Y
#data$ID_Y <- 0
for(i in 1:nrow(data_plW)) {
  if( !is.na(data_plW[i,"y_gps"])) {
    data_plW[i,"ID_Y"] <- data_plW[i,"y_gps"]
  }
  else {data plW[i,"ID Y"] <- data plW[i,"y"]
  }
}
```

```
data plW<- data plW[, c(
  "ID",
  "ID_pl", # Reorganizes "ID_pl" as first column
  "ign",
  "npl",
  "cy",
  "tranche",
  "dg_datem",
  "ID_X", # move here
  "ID_Y", # move here
  "x",
  "y",
  "x gps",
  "y_gps",
  "dg_prov",
  "province",
  "dg_rfor_2",
  "region",
  "dg_n2000",
  "stru_pl",
  "structure",
  "peup_pl",
  "peuplement",
  "NHAf",
  "GHAf",
  "VHAf",
  "NHAt",
  "GHAt",
  "VHAt",
  "LIS",
  "lisiere",
  "PHY_OBS1",
  "PHY_OBS2"
)]
```

```
### Sheet "Essences" (tree species)
# Details de different types of tree species found wihtin each plot
# Aulne = Alnus
# Bouleau = Betula
# Noisetier = Corylus

temp<-loadWorkbook("C:/Users/dujardis/Documents                UNamur/20160401
RespirIT/2_DATA/Inventaire              forestier              Wallon/20160902
IFW/Req_dujardin_010916_VF.xlsx")
data_spW <-readWorksheet(temp, sheet = "Essences",startRow = 9)

data_spW$ID_pl <- paste(data_spW[,"ign"],data_spW[,"npl"],sep="")
data_spW$count <- 1 # For SUMs when using the melt/cast function
data_spW$ID <- 1:nrow(data_spW) # Unique ID

#test <- data_spW[(is.na(data_spW$pcgha)),] # One entry has no gha and no pcgha
> useless > to be removed:
data_spW <- data_spW[(!is.na(data_spW$pcgha)),]
```

```
## Reorganise

data_spW <- data_spW[, c(
  "ID",
  "ID_pl", # move here
  "ign",
  "npl",
  "cy",
  "source",
  "etg_f",
  "ess",
  "essence",
  "pcgha",
  "nha",
  "gha",
  "vha",
  "count"
  #  "ess_mix",
  #  "alnus_aggr"
) ]

var_data_spW <- c(
  "ID",
  "ID_pl", # move here
  "ign",
  "npl",
  "cy",
  "source",
  "etg_f",
  "ess",
  "essence",
  "pcgha",
  "nha",
  "gha",
  "vha",
  "count"
  #  "ess_mix",
  #  "alnus_aggr"
)


######################### DATA PROCESSING ############################


## Tree structure (Table)
temp_melt <- melt(data_spW, id = c("ID_pl", "ign", "npl", "cy", "source",
"etg_f", "essence"))
table_spW_tree_str   <-   cast(temp_melt,   essence   ~   source   +   etg_f,
subset=variable=="count", sum)
table_spW_tree_str                                                    <-
table_spW_tree_str[,c("essence","tai_NA","fut_1","fut_2")]
table_spW_tree_str

#Control check:
#temp <- data_spW[data_spW$essence== "Aulne blanc" & data_spW$source== "fut",
]
#temp <- data_spW[data_spW$essence== "Bouleau" & data_spW$source== "tai", ]

## Tree count
# How many different type of allergenic tree specie are recorded per plot?
# > Count the number of duplicate "ID_pl" in "data_spW"
temp <- cast(temp melt, ID pl ~ variable, sum) # Aggregation per plot >
Identification of duplicates
table(temp$count)
316+48+5 # Total number of plot with more than one type of allergenic tree
369/1453*100 # relative share (one fourth)
# Adding this variable the the dataset "data_spW"
data spW$dupl <- 0
data_spW$dupl <- temp$count[match(data_spW$ID_pl, temp$ID_pl)]
# When only one tree type > no aggregation needed. Measures can be taken as it
is
# When more than one > Need for aggregating measures depending on:
# Alnus vs Betula vs Corylus
# tai vs fut 1 vs fut 2


## Assign a number from 1 to 11 identifying the pcgha class
# NB: This variable is not neccesary when using a "regression model approach"
# See work on dataframe "data spW" below

temp <- data_spW
temp$pcgha_class <- 999
for(i in 1:nrow(temp)) {
  if (temp[i,"pcgha"] > 0.99999)  {
    temp[i,"pcgha class"] <- 11 # when pcgha = 1 (the specie is dominant)
  }
  else if (temp[i,"pcgha"] <= 0.99999999 & temp[i,"pcgha"] >= 0.90000000 ){
    temp[i,"pcgha_class"] <- 10
  }
  else if (temp[i,"pcgha"] <= 0.89999999 & temp[i,"pcgha"] >= 0.80000000 ){
    temp[i,"pcgha class"] <- 9
  }
  else if (temp[i,"pcgha"] <= 0.79999999 & temp[i,"pcgha"] >= 0.70000000 ){
    temp[i,"pcgha_class"] <- 8
  }
  else if (temp[i,"pcgha"] <= 0.69999999 & temp[i,"pcgha"] >= 0.60000000 ){
    temp[i,"pcgha_class"] <- 7
  }
  else if (temp[i,"pcgha"] <= 0.59999999 & temp[i,"pcgha"] >= 0.50000000 ){
    temp[i,"pcgha_class"] <- 6
  }
  else if (temp[i,"pcgha"] <= 0.49999999 & temp[i,"pcgha"] >= 0.40000000 ){
    temp[i,"pcgha_class"] <- 5
  }
  else if (temp[i,"pcgha"] <= 0.39999999 & temp[i,"pcgha"] >= 0.30000000 ){
    temp[i,"pcgha_class"] <- 4
  }
  else if (temp[i,"pcgha"] <= 0.29999999 & temp[i,"pcgha"] >= 0.20000000 ){
    temp[i,"pcgha_class"] <- 3
  }
```

```
  else if (temp[i,"pcgha"] <= 0.19999999 & temp[i,"pcgha"] >= 0.10000000 ){
    temp[i,"pcgha class"] <- 2
  }
  else if (temp[i,"pcgha"] <= 0.99999999 & temp[i,"pcgha"] >= 0.00000001 ){
    temp[i,"pcgha_class"] <- 1
  }
  else {
  }
}
#table(temp$pcgha_class)
data_spW <- temp


## Identifying whether plots contain measured trees or not

# Loop over groups
temp <- data_spW
for(grp in unique(temp$ID_pl)) { # Group by 'ID_pl'
  if(any(is.na(temp[temp$ID_pl == grp, 'gha'])) ) { # If a plot contains a 'gha' with no value (NA)
    temp[temp$ID_pl == grp, 'measured'] <- 0 # Then assign 0
  }
  else {temp[temp$ID_pl == grp, 'measured'] <- 1 # Otherwise asign 1
  }
}
data_spW <- temp


## Plots containing measured trees (A)

data_spW__A <- data_spW[data_spW$measured==1,]
nrow(data_spW__A)
#any(is.na(data_spW__A$gha)) # Control to see if no NA values remains
#table(is.na(data_spW__A$pcgha))


## Plots containing 1 non-measured tree or more (B)

data_spW__B <- data_spW[data_spW$measured==0,]
nrow(data_spW__B)
#count(is.na(data_spW__B$gha)) # 690/820 trees not measured
690/1879*100
## Processing plots containing 1 non-measured trees or more (B)

# Idea of assigning an average 'gha' for non-measured trees
# Search for all measured trees within the dataset and determine was is the average 'gha' value
# !Make a difference between 'tai" and 'fut' 1 et 2
# !Make a difference between  tree species (Alnus,Betula,Corylus)
# !Make a difference between  pcgha categories (0.1, 0.2, etc.)
# (!Process obsevations with pcgha = 1 separately)


# (a) Approach per class
# Computing mean values per structure (tai/fut), species (Alnus, Betula, Corylus), and pcgha class (1-10)
temp <- data_spW[!is.na(data_spW$gha) & !is.na(data_spW$pcgha) ,] # Select observations were gha (& pcgha) is known
1189
temp$count <- 1
temp[,"essence"] <- gsub("Aulne glutineux", "Aulne", temp[,"essence"])
temp[,"essence"] <- gsub("Aulne blanc", "Aulne", temp[,"essence"])

temp_melt <- melt(temp, id = c("ID_pl", "ign", "npl", "cy", "source", "etg_f", "essence", "pcgha_class"))

#Tai

#a. Count
temp_aggr <- cast(temp_melt, essence+pcgha_class ~ variable, sum, subset = source=="tai")
count_alnus <- temp_aggr[temp_aggr$essence=="Aulne",9] # Count
count_alnus <- append(count_alnus, mean(count_alnus),5) # ! Alnus, cat. 6 = NA > Assign an average value
count_betula <- temp_aggr[temp_aggr$essence=="Bouleau",9]
count_corylus <- temp_aggr[temp_aggr$essence=="Noisetier",9]
means_count <- data.frame(count_alnus,count_betula, count_corylus)
#b. Mean
temp_aggr <- cast(temp_melt, essence+pcgha_class ~ variable, mean, subset = source=="tai")
mean_alnus <- temp_aggr[temp_aggr$essence=="Aulne",7] # Mean gha
mean_alnus <- append(mean_alnus, mean(mean_alnus),5) # ! Alnus, cat. 6 = NA > Assign an average value value
mean_betula <- temp_aggr[temp_aggr$essence=="Bouleau",7]
mean_corylus <- temp_aggr[temp_aggr$essence=="Noisetier",7]
means tai <- data.frame(mean alnus,mean betula, mean corylus)

plot(means_tai$mean_alnus)
plot(means_tai$mean_betula)
plot(means_tai$mean_corylus)
# No obvious/strong linear correlation
# some means are based upon a single observation...
# ! One pcgha category contains no observation (Alnus/6)

#c. Min
#temp_aggr <- cast(temp_melt, essence+pcgha_class ~ variable, c(sum, mean, min, max), subset = source=="tai")

#Fut

#a. Count
temp_aggr <- cast(temp_melt, essence+pcgha_class ~ variable, sum, subset = source=="fut")
count_alnus <- temp_aggr[temp_aggr$essence=="Aulne",9] # Count
count_betula <- temp_aggr[temp_aggr$essence=="Bouleau",9]
#temp_corylus <- temp_aggr[temp_aggr$essence=="Noisetier",9] # Not applicable (corylus does was not observed under the form of a tree stand)
means_count <- data.frame(count_alnus,count_betula)
#b. Mean
temp_aggr <- cast(temp_melt, essence+pcgha_class ~ variable, mean, subset = source=="fut")
```

```
mean_alnus <- temp_aggr[temp_aggr$essence=="Aulne",7] # Mean gha
mean betula <- temp aggr[temp aggr$essence=="Bouleau",7]
#temp_corylus <- temp_aggr[temp_aggr$essence=="Noisetier",7] # Not applicable
(corylus does was not observed under the form of a tree stand)
means_fut <- data.frame(mean_alnus,mean_betula)

plot(means_fut$mean_alnus)
plot(means fut$mean betula)
#plot(means_mean$temp_corylus)

#c. Min
#temp_aggr <- cast(temp_melt, essence+pcgha_class ~ variable, c(sum, mean, min,
max), subset = source=="tai")

## Assigning 'means' to missing values

temp <- data_spW__B
table(temp$pcgha_class)

#Tai
j <- 1
while(j <= nrow(means_tai)){
  for(i in 1:nrow(temp)) {
    if (temp[i,"pcgha_class"] == rownames(means_tai[j,]) &
        is.na(temp[i,"gha"]) & temp[i,"source"]=="tai" &
        temp[i,"essence"]=="Aulne blanc"){
      temp[i,"gha"] <- means_tai[j,"mean_alnus"]
    }
    else if (temp[i,"pcgha_class"] == rownames(means_tai[j,]) &
             is.na(temp[i,"gha"]) & temp[i,"source"]=="tai" &
             temp[i,"essence"]=="Aulne glutineux"){
      temp[i,"gha"] <- means_tai[j,"mean_alnus"]
    }
    else if (temp[i,"pcgha_class"] == rownames(means_tai[j,]) &
             is.na(temp[i,"gha"])       &       temp[i,"source"]=="tai"       &
temp[i,"essence"]=="Bouleau"){
      temp[i,"gha"] <- means tai[j,"mean betula"]
    }
    else if (temp[i,"pcgha_class"] == rownames(means_tai[j,]) &
             is.na(temp[i,"gha"])       &       temp[i,"source"]=="tai"       &
temp[i,"essence"]=="Noisetier"){
      temp[i,"gha"] <- means_tai[j,"mean_corylus"]
    }
    else {
    }
  }
  j <- j+1
}

#Fut
j <- 1
while(j <= nrow(means_fut)){
  for(i in 1:nrow(temp)) {
    if (temp[i,"pcgha class"] == rownames(means fut[j,]) &
        is.na(temp[i,"gha"]) & temp[i,"source"]=="fut" &
        temp[i,"essence"]=="Aulne blanc" ){
      temp[i,"gha"] <- means_fut[j,"mean_alnus"]
    }
    else if (temp[i,"pcgha_class"] == rownames(means_fut[j,]) &
             is.na(temp[i,"gha"]) & temp[i,"source"]=="fut" &
             temp[i,"essence"]=="Aulne glutineux"){
      temp[i,"gha"] <- means_fut[j,"mean_alnus"]
    }
    else if (temp[i,"pcgha class"] == rownames(means fut[j,]) &
             is.na(temp[i,"gha"])       &       temp[i,"source"]=="fut"       &
temp[i,"essence"]=="Bouleau" ){
      temp[i,"gha"] <- means_fut[j,"mean_betula"]
    }
    else {
    }
  }
  j <- j+1
}

data_spW__B <- temp

# (b) Approach by linear regression model
# Idea of establishing a statistical relationship between pcgha and gha values
based on measured obervations
# Building a linear regression model per structure (tai/fut) and species (Alnus,
Betula, Corylus) allows predicting gha for non measured observations
# !Need to distinguish observations with pcgha = 1 (the specie is dominant) and
where pcgha < 1

# b1 (pcgha < 1): Predicted value provided by the linear regression model
# b2 (pcgha < 1): Average value

temp  <-  data_spW[!is.na(data_spW$gha)  &  !is.na(data_spW$pcgha)  ,]  #  Select
observations were gha (& pcgha) is known
temp[,"essence"] <- gsub("Aulne glutineux", "Aulne", temp[,"essence"]) # All
types of alnus are considered the same
temp[,"essence"] <- gsub("Aulne blanc", "Aulne", temp[,"essence"]) # All types
of alnus are considered the same
temp <- temp[temp$pcgha_class!=11,] # selecting where pcgha < 1
#temp2 <- temp[temp$pcgha_class==1,] # selecting where pcgha = 1

#Tai

tai_alnus <- temp[temp$source=="tai"& temp$essence=="Aulne", ]
pcgha <- tai_alnus$pcgha
gha <- tai_alnus$gha
plot(pcgha, gha, main="Scatterplot",
     xlab="pcgha ", ylab="gha ", pch=19)
cor(gha,pcgha)
#cov(gha,pcgha)
model_tai_alnus  <-  lm(gha~0+pcgha)  #  Regression  through  the  origin  (i.e.
intercept=0)
model_tai_alnus
summary(model_tai_alnus) # R² = 0,632
abline(model tai alnus)
# >The regression equation is gha = 25,48 * pcgha
# Plotting
```

```
df <- data.frame(pcgha,gha)
plot tai alnus <- ggplot(df, aes(x=pcgha, y=gha)) +
  geom_point(shape=19) +    # Use plain circles
  geom_smooth(method=lm) +
  ggtitle("A") +
  theme_minimal()
plot_tai_alnus

tai_betula <- temp[temp$source=="tai"& temp$essence=="Bouleau", ]
pcgha <- tai_betula$pcgha
gha <- tai_betula$gha
plot(pcgha, gha, main="Scatterplot",
     xlab="pcgha ", ylab="gha ", pch=19)
cor(gha,pcgha)
#cov(gha,pcgha)
model_tai_betula <- lm(gha~0+pcgha)
model_tai_betula
summary(model_tai_betula)  # R² = 0,622
abline(model_tai_betula)
# Plotting
df <- data.frame(pcgha,gha)
plot_tai_betula <- ggplot(df, aes(x=pcgha, y=gha)) +
  geom_point(shape=19) +    # Use plain circles
  geom_smooth(method=lm) +
  ggtitle("B") +
  theme_minimal()
plot_tai_betula

tai_corylus <- temp[temp$source=="tai"& temp$essence=="Noisetier", ]
pcgha <- tai_corylus$pcgha
gha <- tai corylus$gha
plot(pcgha, gha, main="Scatterplot",
     xlab="pcgha ", ylab="gha ", pch=19)
cor(gha,pcgha)
#cov(gha,pcgha)
model_tai_corylus <- lm(gha~0+pcgha)
model tai corylus
summary(model_tai_corylus)  # R² = 0,513
abline(model_tai_corylus)
# Plotting
df <- data.frame(pcgha,gha)
plot_tai_corylus <- ggplot(df, aes(x=pcgha, y=gha)) +
  geom_point(shape=19) +    # Use plain circles
  geom_smooth(method=lm) +
  ggtitle("C") +
  theme_minimal()
plot_tai_corylus

#Fut

fut_alnus <- temp[temp$source=="fut"& temp$essence=="Aulne", ]
pcgha <- fut_alnus$pcgha
gha <- fut_alnus$gha
plot(pcgha, gha, main="Scatterplot",
     xlab="pcgha ", ylab="gha ", pch=19)
cor(gha,pcgha)
#cov(gha,pcgha)
model fut alnus <- lm(gha~0+pcgha)
model_fut_alnus
summary(model_fut_alnus)#  R² = 0,842
abline(model_fut_alnus)
# Plotting
df <- data.frame(pcgha,gha)
plot fut alnus <- ggplot(df, aes(x=pcgha, y=gha)) +
  geom_point(shape=19) +    # Use plain circles
  geom_smooth(method=lm) +
  ggtitle("D") +
  theme_minimal()
plot_fut_alnus

fut_betula <- temp[temp$source=="fut"& temp$essence=="Bouleau", ]
pcgha <- fut_betula$pcgha
gha <- fut_betula$gha
plot_fut_betula <- plot(pcgha, gha, main="Scatterplot",
     xlab="pcgha ", ylab="gha ", pch=19)
cor(gha,pcgha)
#cov(gha,pcgha)
model_fut_betula <- lm(gha~0+pcgha)
model_fut_betula
summary(model_fut_betula) # R² = 0,811
abline(model fut betula)
# Plotting
df <- data.frame(pcgha,gha)
plot_fut_betula <- ggplot(df, aes(x=pcgha, y=gha)) +
  geom_point(shape=19) +    # Use plain circles
  geom_smooth(method=lm) +
  ggtitle("E") +
  #theme(panel.background = element_blank())
  #theme(plot.title = element_text(lineheight=.4, face="bold")) +
  theme_minimal()
plot_fut_betula

#predict(model, newdata=data.frame(pcgha=0.5))
#new <- data.frame(pcgha = c(0.1,0.5,0.9))
#predict(model, newdata=new)

# Plotting
mygraph <- multiplot(plot_tai_alnus,
          plot_tai_betula,
          plot_tai_corylus,
          plot_fut_alnus,
          plot_fut_betula, cols=2)
#print(mygraph)
tiff(filename       =       "C:/Users/dujardis/Documents       UNamur/20160401
RespirIT/3_ANALYSIS/Traitements_InventaireForestierWallon/IFW_R_Outputs/multip
lot_tai-fut_Alnus-Betula-Corylus.tiff", width = 500, height = 500, units = "px",
pointsize = 12, bg = "white", res = NA, restoreConsole = TRUE)
multiplot(plot_tai_alnus,
          plot_tai_betula,
          plot tai corylus,
          plot_fut_alnus,
          plot_fut_betula, cols=2)
```

```
dev.off()


## Assigning 'means' to missing values in "data_spW__B"

temp <- data_spW__B

# b1 (pcgha < 1): Predicted value provided by the linear regression model

#Tai

for(i in 1:nrow(temp)) {
    if (temp[i,"pcgha_class"]!=11 &
        is.na(temp[i,"gha"]) &
        temp[i,"source"]=="tai" &
        temp[i,"essence"]=="Aulne blanc"){
      temp[i,"gha"]                  <-                predict(model_tai_alnus,
newdata=data.frame(pcgha=temp[i,"pcgha"]))
    }
    else if (temp[i,"pcgha_class"]!=11 &
        is.na(temp[i,"gha"]) &
        temp[i,"source"]=="tai" &
        temp[i,"essence"]=="Aulne glutineux"){
      temp[i,"gha"]                  <-                predict(model_tai_alnus,
newdata=data.frame(pcgha=temp[i,"pcgha"]))
    }
    else if (temp[i,"pcgha_class"]!=11 &
             is.na(temp[i,"gha"]) &
             temp[i,"source"]=="tai" &
             temp[i,"essence"]=="Bouleau"){
      temp[i,"gha"]                  <-                predict(model tai betula,
newdata=data.frame(pcgha=temp[i,"pcgha"]))
    }
    else if (temp[i,"pcgha_class"]!=11 &
             is.na(temp[i,"gha"]) &
             temp[i,"source"]=="tai" &
             temp[i,"essence"]=="Noisetier"){
      temp[i,"gha"]                  <-                predict(model_tai_corylus,
newdata=data.frame(pcgha=temp[i,"pcgha"]))
    }
    else {
    }
}


#Fut

for(i in 1:nrow(temp)) {
    if (temp[i,"pcgha_class"]!=11 &
        is.na(temp[i,"gha"]) &
        temp[i,"source"]=="fut" &
        temp[i,"essence"]=="Aulne blanc" ){
      temp[i,"gha"]                  <-                predict(model_fut_alnus,
newdata=data.frame(pcgha=temp[i,"pcgha"]))
    }
    else if (temp[i,"pcgha_class"]!=11 &
        is.na(temp[i,"gha"]) &
        temp[i,"source"]=="fut" &
        temp[i,"essence"]=="Aulne glutineux"){
      temp[i,"gha"]                  <-                predict(model_fut_alnus,
newdata=data.frame(pcgha=temp[i,"pcgha"]))
    }
    else if (temp[i,"pcgha_class"]!=11 &
             is.na(temp[i,"gha"]) &
             temp[i,"source"]=="fut" &
             temp[i,"essence"]=="Bouleau" ){
      temp[i,"gha"]                  <-                predict(model_fut_betula,
newdata=data.frame(pcgha=temp[i,"pcgha"]))
    }
    else {
    }
}

# b2 (pcgha = 1): Average value

#Tai

for(i in 1:nrow(temp)) {
    if (temp[i,"pcgha_class"]==11 &
        is.na(temp[i,"gha"]) &
        temp[i,"source"]=="tai" &
        temp[i,"essence"]=="Aulne blanc"){
      temp[i,"gha"] <- means_tai[11,"mean_alnus"]
    }
    else if (temp[i,"pcgha_class"]==11 &
             is.na(temp[i,"gha"]) &
             temp[i,"source"]=="tai" &
             temp[i,"essence"]=="Aulne glutineux"){
      temp[i,"gha"] <- means_tai[11,"mean_alnus"]
    }
    else if (temp[i,"pcgha_class"]==11 &
             is.na(temp[i,"gha"]) &
             temp[i,"source"]=="tai" &
             temp[i,"essence"]=="Bouleau"){
      temp[i,"gha"] <- means_tai[11,"mean_betula"]
    }
    else if (temp[i,"pcgha_class"]==11 &
             is.na(temp[i,"gha"]) &
             temp[i,"source"]=="tai" &
             temp[i,"essence"]=="Noisetier"){
      temp[i,"gha"] <- means_tai[11,"mean_corylus"]
    }
    else {
    }
}


#Fut

for(i in 1:nrow(temp)) {
    if (temp[i,"pcgha_class"]==11 &
        is.na(temp[i,"gha"]) &
```

```
        temp[i,"source"]=="fut" &
        temp[i,"essence"]=="Aulne blanc" ){
      temp[i,"gha"] <- means_fut[11,"mean_alnus"]
    }
    else if (temp[i,"pcgha_class"]==11 &
             is.na(temp[i,"gha"]) &
             temp[i,"source"]=="fut" &
             temp[i,"essence"]=="Aulne glutineux"){
      temp[i,"gha"] <- means_fut[11,"mean_alnus"]
    }
    else if (temp[i,"pcgha_class"]==11 &
             is.na(temp[i,"gha"]) &
             temp[i,"source"]=="fut" &
             temp[i,"essence"]=="Bouleau" ){
      temp[i,"gha"] <- means_fut[11,"mean_betula"]
    }
    else {
    }
}

data_spW__B <- temp
plot(data_spW__B$pcgha, data_spW__B$gha, main="Scatterplot",
     xlab="pcgha ", ylab="gha ", pch=19)
abline(lm(data_spW__B$gha~data_spW__B$pcgha),  col="red")  #  regression  line
(y~x)

# NB: The same should be done for 'nha' and 'vha' values...

# Once missing values are assigned, the aggregation procedure can be implemented
over A & B

## Merging both datasets (A & B)

data_spW__C <- rbind(data_spW__A, data_spW__B)


##  Processing plots containing only 1 tree (C1)

data_spW__C1 <- data_spW__A[data_spW__A$dupl==1,]

# Replacement of "Aulne blanc" & "Aulne glutineux" by "Aulne"
data_spW__C1[,"essence"]      <-      gsub("Aulne      glutineux",      "Aulne",
data_spW__C1[,"essence"])
data_spW__C1[,"essence"]       <-       gsub("Aulne       blanc",       "Aulne",
data_spW__C1[,"essence"])
# tag here when chnage was made under the name of aulne_dupl?

# No aggregation needed over tai
# No aggregation needed over fut

##  Processing plots contraining more than one tree (C2)

data_spW__C2 <- data_spW__A[data_spW__A$dupl>1,]

# Aggregation needed over:
# a) essence: "Aulne blanc"/"Aulne glutineux"
# b) tai
# c) fut 1/2

# Identify where two types of alnus are observerd within the same plot
temp <- data_spW__C2
for(grp in unique(temp$ID_pl)) { # Group by 'ID_pl'
  if(any(temp[temp$ID_pl == grp, 'essence'] == "Aulne blanc") &
     any(temp[temp$ID_pl == grp, 'essence'] == "Aulne glutineux")) { # If a plot
contains both "Aulne blanc" AND "Aulne glutineux"
    temp[temp$ID_pl == grp, 'dupl_aulne'] <- 1 # Then assign 1
  }
  else {temp[temp$ID_pl == grp, 'dupl_aulne'] <- 0 # Otherwise asign 0
  }
}
table(temp$dupl_aulne) # 5 plots where both "Aulne glutuneux" and "Aulne blanc"
are found together
data_spW__C2 <- temp

# Replacement of "Aulne blanc" & "Aulne glutineux" by "Aulne"
data_spW__C2[,"essence"]      <-      gsub("Aulne      glutineux",      "Aulne",
data_spW__C2[,"essence"])
data_spW__C2[,"essence"]       <-       gsub("Aulne       blanc",       "Aulne",
data_spW__C2[,"essence"])

# C1 a) Where only one "Aulne" (or none) is found wihtin the plot
data_spW__C2_a <- data_spW__C2[data_spW__C2$dupl_aulne ==0,]

# C1_b) Where more than one "Aulne" is found within the plot
data_spW__C2_b <- data_spW__C2[data_spW__C2$dupl_aulne ==1,]

# Identify where more than one "taillis" is found within the same plot
temp <- data_spW__C2_b
for(grp in unique(temp$ID_pl)) { # Group by 'ID_pl'
  if(sum(str_count(temp[temp$ID_pl == grp, 'source'],'tai')) > 1) { # If more
than one "tai" is found in "source"
    temp[temp$ID_pl == grp, 'dupl_tai'] <- sum(str_count(temp[temp$ID_pl ==
grp, 'source'],'tai')) # Then count
  }
  else {temp[temp$ID_pl == grp, 'dupl_tai'] <- 0 # Otherwise asign 0
  }
}
data_spW__C2_b <- temp

# Identify where more than one "futaie" is found within the same plot
temp <- data_spW__C2_b
for(grp in unique(temp$ID_pl)) { # Group by 'ID_pl'
  if(sum(str_count(temp[temp$ID_pl == grp, 'source'],'fut')) > 1) { # If more
than one "tai" is found in "source"
    temp[temp$ID_pl == grp, 'dupl_fut'] <- sum(str_count(temp[temp$ID_pl ==
grp, 'source'],'fut')) # Then assign the count
  }
  else {temp[temp$ID_pl == grp, 'dupl_fut'] <- 0 # Otherwise asign 0
  }
}
data_spW__C2_b <- temp
```

```
# Aggregation over "fut"

# temp <- data_spW__C2_b[data_spW__C2_b$source == "fut",] # use your condition!
> dupl_fut > 0
# temp_melt <- melt(temp, id = c("ID_pl", "ign", "npl", "cy", "source", "etg_f",
"essence"))
#  temp_aggr_fut  <-  cast(temp_melt,  ID_pl+source+essence  ~  variable,  sum)  #
Aggregation per plot > Identification of duplicates
# ! cast function loses the important attributes of "source" and "essence".
#                          temp_aggr_fut$dupl_aulne                              <-
data_spW__C2_b$dupl_aulne[match(temp_aggr_fut$ID_pl, data_spW__C2_b$ID_pl)]
# temp_aggr_fut$dupl_fut <- data_spW__C2_b$dupl_fut[match(temp_aggr_fut$ID_pl,
data_spW__C2_b$ID_pl)]

# Aggregation over "tai"
# temp <- data_spW__C2_b[data_spW__C2_b$source == "tai",]
# temp_melt <- melt(temp, id = c("ID_pl", "ign", "npl", "cy", "source", "etg_f",
"essence"))
# temp_aggr_tai <- cast(temp_melt, ID_pl+source+essence ~ variable, sum) #
# ! cast function loses the important attributes of "source" and "essence".
#                          temp_aggr_fut$dupl_aulne                              <-
data_spW__C2_b$dupl_aulne[match(temp_aggr_tai$ID_pl, data_spW__C2_b$ID_pl)]
# temp_aggr_fut$dupl_tai <- data_spW__C2_b$dupl_tai[match(temp_aggr_tai$ID_pl,
data_spW__C2_b$ID_pl)]
# temp aggr <- rbind(temp aggr tai,temp aggr fut)

# Aggregation
temp <- data_spW__C2_b
temp_melt <- melt(temp, id = c("ID_pl", "ign", "npl", "cy", "source", "etg_f",
"essence"))
temp aggr <- cast(temp melt, ID pl+source+essence ~ variable, sum)

# Re-building "data_spW__C2_b"
# temp_aggr <- rbind(temp_aggr_tai,temp_aggr_fut)
data_spW__C2_b <- temp_aggr
data_spW__C2_b$ign <- 0
data spW  C2 b$npl <- 0
data_spW__C2_b$cy <- 0
data_spW__C2_b$etg_f <- 0

var_data_spW <- c(
  "ID",
  "ID_pl",
  "ign",
  "npl",
  "cy",
  "source",
  "etg f",
  "ess",
  "essence",
  "pcgha",
  "nha",
  "gha",
  "vha",
  "count",
  "dupl",
  "pcgha_class",
  "measured",
  "dupl_aulne",
  "dupl_tai",
  "dupl_fut"
)
data_spW__C2_b <- data_spW__C2_b[, var_data_spW] # Re-ordering columns

# Re-building "data_spW__C2"
#colnames(data_spW__C2_a)
#colnames(data_spW__C2_b)
data_spW__C2_a$dupl_tai <- 0
data_spW__C2_a$dupl_fut <- 0
data spW  C2 <- rbind(data spW  C2 a, data spW  C2 b)


## Aggregation over fut & tai for measured trees with more than 1 tree per plot
(C2)

# Identify where more than one "taillis" is found within the same plot
temp <- data_spW__C2
for(grp in unique(temp$ID_pl)) { # Group by 'ID_pl'
  if(sum(str_count(temp[temp$ID_pl == grp, 'source'],'tai')) > 1) { # If more
than one "tai" is found in "source"
    temp[temp$ID pl  ==  grp,  'dupl tai'] <-  sum(str count(temp[temp$ID pl  ==
grp, 'source'],'tai')) # Then the count
  }
  else {temp[temp$ID_pl == grp, 'dupl_tai'] <- 0 # Otherwise asign 0
  }
}
data spW  C2 <- temp

# Identify where more than one "futaie" is found within the same plot
temp <- data_spW__C2
for(grp in unique(temp$ID_pl)) { # Group by 'ID_pl'
  if(sum(str_count(temp[temp$ID_pl == grp, 'source'],'fut')) > 1) { # If more
than one "tai" is found in "source"
    temp[temp$ID_pl  ==  grp,  'dupl_fut'] <-  sum(str_count(temp[temp$ID_pl  ==
grp, 'source'],'fut')) # Then assign the count
  }
  else {temp[temp$ID_pl == grp, 'dupl_fut'] <- 0 # Otherwise asign 0
  }
}
data_spW__C2 <- temp

# Identify where more than one "Alnus" is found within the same plot
temp <- data_spW__C2
for(grp in unique(temp$ID pl)) { # Group by 'ID pl'
  if(sum(str_count(temp[temp$ID_pl == grp, 'essence'],'Aulne')) > 1) { # If more
than one "tai" is found in "source"
    temp[temp$ID_pl == grp, 'dupl_Alnus'] <- sum(str_count(temp[temp$ID_pl ==
grp, 'essence'],'Aulne')) # Then the count
  }
  else {temp[temp$ID pl == grp, 'dupl Alnus'] <- 0 # Otherwise asign 0
  }
}
```

```
data_spW__C2 <- temp
# Identify where more than one "Betula" is found within the same plot
temp <- data_spW__C2
for(grp in unique(temp$ID_pl)) { # Group by 'ID_pl'
  if(sum(str_count(temp[temp$ID_pl == grp, 'essence'],'Bouleau')) > 1) { # If
more than one "tai" is found in "source"
    temp[temp$ID_pl == grp, 'dupl_Betula'] <- sum(str_count(temp[temp$ID_pl ==
grp, 'essence'],'Bouleau')) # Then the count
  }
  else {temp[temp$ID_pl == grp, 'dupl_Betula'] <- 0 # Otherwise asign 0
  }
}
data_spW__C2 <- temp
# Identify where more than one "Corylus" is found within the same plot
temp <- data_spW__C2
for(grp in unique(temp$ID_pl)) { # Group by 'ID_pl'
  if(sum(str_count(temp[temp$ID_pl == grp, 'essence'],'Noisetier')) > 1) { # If
more than one "tai" is found in "source"
    temp[temp$ID_pl == grp, 'dupl_Corylus'] <- sum(str_count(temp[temp$ID_pl ==
grp, 'essence'],'Noisetier')) # Then the count
  }
  else {temp[temp$ID_pl == grp, 'dupl_Corylus'] <- 0 # Otherwise asign 0
  }
}
data spW  C2 <- temp


# Aggregation
temp <- data_spW__C2
temp_melt <- melt(temp, id = c("ID_pl", "ign", "npl", "cy", "source", "etg_f",
"essence"))
#temp_aggr  <-  cast(temp_melt,  ID_pl+source+essence  ~  variable,  sum)  #
Aggregation per plot > Identification of duplicates
temp_aggr <- cast(temp_melt, ID_pl+essence ~ variable, sum) # Aggregation per
plot > Identification of duplicates
# ! cast function loses and aggregates important attributes of "source" and
"essence".
temp_aggr$"count"<-                  data_spW__C2$count[match(temp_aggr$ID_pl,
data_spW__C2$ID_pl)]
temp_aggr$"dupl"<-                    data_spW__C2$dupl[match(temp_aggr$ID_pl,
data_spW__C2$ID_pl)]
temp_aggr$"measured"<-             data_spW__C2$measured[match(temp_aggr$ID_pl,
data_spW__C2$ID_pl)]
temp_aggr$"dupl_aulne"<-         data_spW__C2$dupl_aulne[match(temp_aggr$ID_pl,
data_spW__C2$ID_pl)]
temp_aggr$"dupl_tai"<-             data_spW__C2$dupl_tai[match(temp_aggr$ID_pl,
data_spW__C2$ID_pl)]
temp aggr$"dupl fut"<-             data spW  C2$dupl fut[match(temp aggr$ID pl,
data_spW__C2$ID_pl)]
temp_aggr$"dupl_Alnus"<-         data_spW__C2$dupl_Alnus[match(temp_aggr$ID_pl,
data_spW__C2$ID_pl)]
temp_aggr$"dupl_Betula"<-       data_spW__C2$dupl_Betula[match(temp_aggr$ID_pl,
data_spW__C2$ID_pl)]
temp aggr$"dupl Corylus"<-     data spW  C2$dupl Corylus[match(temp aggr$ID pl,
data_spW__C2$ID_pl)]
```

```
# Re-building "data_spW__C"

# Re-building "data_spW__C2"
# temp_aggr <- rbind(temp_aggr_tai,temp_aggr_fut)
data_spW__C2 <- temp_aggr
data_spW__C2$ign <- 0
data_spW__C2$npl <- 0
data spW  C2$cy <- 0
data_spW__C2$source <- 0
data_spW__C2$etg_f <- 0

var_data_spW <- c(
  "ID",
  "ID pl",
  "ign",
  "npl",
  "cy",
  "source",
  "etg_f",
  "ess",
  "essence",
  "pcgha",
  "nha",
  "gha",
  "vha",
  "count",
  "dupl",
  "pcgha_class",
  "measured",
  "dupl_aulne",
  "dupl tai",
  "dupl_fut",
  "dupl_Alnus",
  "dupl_Betula",
  "dupl_Corylus"
)
data spW  C2 <- data spW  C2[, var data spW] # Re-ordering columns

# Re-building C1
data_spW__C1$dupl_aulne <- 0
data_spW__C1$dupl_tai <- 0
data_spW__C1$dupl_fut <- 0
data_spW__C1$dupl_Alnus <- 0
data_spW__C1$dupl_Betula <- 0
data_spW__C1$dupl_Corylus <- 0

data_spW__C2 <- data_spW__C2[, var_data_spW] # Re-ordering columns (cf. C2)

# Re-building C
data_spW__C <- rbind(data_spW__C1, data_spW__C2)


## COMPUTING OVERALL GHA VALUES ###


## Creating D
```

```r
data_spW__D <- data_spW__C[, 1:13] # Only keeping the essential variables

# Assigning plots' overall GAH values from "data_plW"
# Two overal values (fut/tai) exist
# Total nha, gha and vha

#data_spW__D$NHAf <- 0
data spW  D$GHAf <- 0
#data_spW__D$VHAf <- 0
#data_spW__D$NHAt <- 0
data_spW__D$GHAt <- 0
#data_spW__D$VHAt <- 0

for(i in 1:nrow(data spW  D)) {
  if (data_spW__D[i,"source"]=="fut") {
    data_spW__D$GHAf         <-         data_plW$GHAf[match(data_spW__D$ID_pl,
data_plW$ID_pl)]
  }
  else if (data_spW__D[i,"source"]=="tai") {
    data_spW__D$GHAt         <-         data_plW$GHAt[match(data_spW__D$ID_pl,
data_plW$ID_pl)]
  }
  else {
  }
}

data_spW__D$GHAf[is.na(data_spW__D$GHAf)] <- 0 # Assigning 0 for NAs
data_spW__D$GHAt[is.na(data_spW__D$GHAt)] <- 0 # Assigning 0 for NAs
data_spW__D$GHA_TOT <- data_spW__D$GHAf + data_spW__D$GHAt

data spW ALL <- data spW  D


## a) Alnus, Betula, and Corylus
#

data spW Alnus <- data spW  D[data spW  D$essence == "Aulne",]
data_spW_Betula <- data_spW__D[data_spW__D$essence == "Bouleau",]
data_spW_Corylus <- data_spW__D[data_spW__D$essence == "Noisetier",]

# Computing pcGHA (basal area in %)
data_spW_Alnus$pcGHA_Alnus <- data_spW_Alnus$gha/data_spW_Alnus$GHA_TOT*100
data_spW_Betula$pcGHA_Betula
data_spW_Betula$gha/data_spW_Betula$GHA_TOT*100
data_spW_Corylus$pcGHA_Corylus
data_spW_Corylus$gha/data_spW_Corylus$GHA_TOT*100

# Descriptive statistics
summary(data_spW_Alnus)

# Plot
hist(data_spW_Alnus$pcGHA_Alnus, nclass = 100)
hist(data_spW_Betula$pcGHA_Betula, nclass = 100)
hist(data spW Corylus$pcGHA Corylus, nclass = 100)

## b) All 3 species
# Requires aggregation over "ID pl", but "essence" is dropped this time

temp <- data_spW__D
temp_melt <- melt(temp, id = c("ID_pl", "ign", "npl", "cy", "source", "etg_f",
"essence"))
#temp_aggr  <-  cast(temp_melt,  ID_pl+source+essence  ~  variable,  sum)  #
Aggregation per plot > Identification of duplicates
temp_aggr <- cast(temp_melt, ID_pl ~ variable, sum)
data_spW_ALL3 <- temp_aggr

nrow(table(data_spW_ALL3$ID_pl))

# Computing pcGHA
data_spW_ALL3$pcGHA_ALL3 <- data_spW_ALL3$gha/data_spW_ALL3$GHA_TOT*100

# Plot
hist(data_spW_ALL3$pcGHA_ALL3, nclass = 100)


################################                                    PLOTTING
#######################################

## Violin plot
# With Alnus, Betula, Corylus, and ALL3

temp1 <-data.frame(data_spW_Alnus$pcGHA_Alnus)
temp1$tree <- c("Alnus")
colnames(temp1)[colnames(temp1) == "data_spW_Alnus.pcGHA_Alnus"] <- "pcgha"
temp2 <- data.frame(data_spW_Betula$pcGHA_Betula)
temp2$tree <- "Betula"
colnames(temp2)[colnames(temp2) == "data_spW_Betula.pcGHA_Betula"] <- "pcgha"
temp3 <- data.frame(data_spW_Corylus$pcGHA_Corylus)
temp3$tree <- "Corylus"
colnames(temp3)[colnames(temp3) == "data_spW_Corylus.pcGHA_Corylus"] <- "pcgha"
temp4 <- data.frame(data_spW_ALL3$pcGHA_ALL)
temp4$tree <- "All"
colnames(temp4)[colnames(temp4) == "data_spW_ALL3.pcGHA_ALL"] <- "pcgha"

mean(temp1$pcgha)
mean(temp2$pcgha)
mean(temp3$pcgha)
<- mean(temp4$pcgha)

<- temp_plot <- rbind(temp4, temp1, temp2, temp3)

temp_plot$tree <- factor(temp_plot$tree, levels=c("Alnus", "Betula", "Corylus",
"All"))

colour_type <- c(Alnus = "grey",
                 Betula = "grey",
                 Corylus ="grey",
                 All = "black")

mygraph <- ggplot(temp_plot, aes(x=tree, y=pcgha)) +
  geom_violin(aes(fill = factor(tree)), scale = "area") +
```

```
  stat_summary(fun.y=median, geom="point", size=2, color="red") +
  # geom boxplot(width=0.05) +
  scale_fill_manual(values= colour_type, name="Tree type") +
  scale_y_continuous(breaks=seq(0, 100, 20)) +
  guides(fill=guide_legend(reverse=FALSE)) +
  theme_minimal()
print(mygraph)
ggsave(mygraph,                                file="data spW Violing-Plot.tiff",
path="C:/Users/dujardis/Documents                                UNamur/20160401
RespirIT/3_ANALYSIS/Traitements_InventaireForestierWallon/IFW_R_Outputs/",
width=6, height=5, dpi=250)


############# FINAL TABLE #################
#  Merging  "ALL3   -  Alnus  -  Betula  -  Corylus"  with  all  records  from
"Pts_forestiers" (all sampling plots from the WFI)

shp      <-      readOGR(dsn="C:/Users/dujardis/Documents      UNamur/20160401
RespirIT/2_DATA/Inventaire    forestier    Wallon/20160606    IFW/Pts_for_CY1",
layer="Pts_forestiers_CY1",    stringsAsFactors=FALSE,    use_iconv    =    TRUE,
encoding="UTF-8")
data_spW_FULL <- data.frame(shp@data)
data_spW_FULL$ID_pl <- paste(data_spW_FULL$IGN, data_spW_FULL$NPL,sep="")
data_spW_FULL <- data_spW_FULL[,c("ID_pl","X","Y")]
data spW FULL                          <-                          merge(data spW FULL,
data_spW_Alnus[,c("ID_pl","gha","pcGHA_Alnus")], by="ID_pl", all=TRUE)
colnames(data_spW_FULL)[colnames(data_spW_FULL) == "gha"] <- "gha_alnus"
data_spW_FULL                          <-                          merge(data_spW_FULL,
data_spW_Betula[,c("ID_pl","gha","pcGHA_Betula")], by="ID_pl", all=TRUE)
colnames(data_spW_FULL)[colnames(data_spW_FULL) == "gha"] <- "gha_betula"
data spW FULL                          <-                          merge(data spW FULL,
data_spW_Corylus[,c("ID_pl","gha","pcGHA_Corylus")], by="ID_pl", all=TRUE)
colnames(data_spW_FULL)[colnames(data_spW_FULL) == "gha"] <- "gha_corylus"
data_spW_FULL                          <-                          merge(data_spW_FULL,
data_spW_ALL3[,c("ID_pl","gha","pcGHA_ALL3",      "GHA_TOT")],      by="ID_pl",
all=TRUE)
colnames(data spW FULL)[colnames(data spW FULL) == "gha"] <- "gha all3"
data_spW_FULL[is.na(data_spW_FULL )] <- 0 # replace NA values
#colnames(data_spW_FULL)
data_spW_FULL <- data_spW_FULL[c("ID_pl",
                "X",
                "Y",
                "GHA_TOT",
                "gha_alnus",
                "pcGHA_Alnus",
                "gha_betula",
                "pcGHA_Betula",
                "gha corylus",
                "pcGHA_Corylus",
                "gha_all3",
                "pcGHA_ALL3")] # Reorder by column name


write.xlsx(data_spW_FULL,      "C:/Users/dujardis/Documents      UNamur/20160401
RespirIT/3 ANALYSIS/Traitements InventaireForestierWallon/IFW R Outputs/data s
pW_FULL.xlsx")
```

```
# FULL = all plots sampled (even those without allergenic trees)
# ALL = all with trees sampled separately
# ALL3 = All3 species together (same column)
# Alnus = Alnus only
# Betula = //
# Corylus = //


################################                                    MAPPING
#########################################

### Building the output shapefile

# shp info
ogrInfo("C:/Users/dujardis/Documents                              UNamur/20160401
RespirIT/3_ANALYSIS/Traitements_InventaireForestierWallon",
"Pts_forestiers_CY1_ID_pl")

# import shp
shp      <-      readOGR(dsn="C:/Users/dujardis/Documents      UNamur/20160401
RespirIT/3_ANALYSIS/Traitements_InventaireForestierWallon",
layer="Pts_forestiers_CY1_ID_pl", stringsAsFactors=FALSE, encoding="UTF-8")



# Draw points shp
#points(shp, axes=TRUE, border="gray")

# projection info
#print(proj4string(shp))

# Joint attributes

# 0) ALL

# a) ALL3
shp data spW ALL3 <- shp
#shp_data_spW_ALL3$ID_pl            <-            paste(shp_data_spW_ALL3$IGN,
shp_data_spW_ALL3$NPL,sep="")  #  creating  the  same  "ID_pl"  for  joining  the
shapefile
shp_data_spW_ALL3@data  <-  join(shp_data_spW_ALL3@data,  data_spW_ALL3,  by  =
"ID_pl", type = "left")
#shp_data_spW_ALL3@data  <-  join(shp_data_spW_ALL3@data,  data_spW_ALL3,  by  =
"ID_pl", type = "full")

# Check for the join whether both "ID_Pl" match
temp1 <- data.frame(shp_data_spW_ALL3@data)
temp2 <- data spW ALL3
temp1$check <- 0
j <- 1
while(j <= nrow(temp1)){
  if ( any( temp2$ID_pl == temp1[j,"ID_pl"])) {
    temp1[j,"check"] <- 1
  }
  j <- j+1
}
```

```
table(temp1$check)
nrow(temp2)
821-832
# ! 11 plots from the dataset have not match with the shapefile

writeOGR(shp_data_spW_ALL3,    "C:/Users/dujardis/Documents    UNamur/20160401
RespirIT/3_ANALYSIS/Traitements_InventaireForestierWallon/IFW_R_Outputs",
"data spW ALL3", driver="ESRI Shapefile", layer options= c(encoding= "UTF-8"),
overwrite_layer=T)

#b) Alnus
shp_data_spW_Alnus <- shp
shp_data_spW_Alnus$ID_pl           <-           paste(shp_data_spW_Alnus$IGN,
shp data spW Alnus$NPL,sep="") # creating the same "ID pl" for joining the
shapefile
shp_data_spW_Alnus@data <- join(shp_data_spW_Alnus@data, data_spW_Alnus, by =
"ID_pl", type = "left")

writeOGR(shp_data_spW_Alnus,    "C:/Users/dujardis/Documents    UNamur/20160401
RespirIT/3_ANALYSIS/Traitements_InventaireForestierWallon/IFW_R_Outputs",
"data_spW_Alnus", driver="ESRI Shapefile", layer_options= c(encoding= "UTF-8"),
overwrite_layer=T)

#c) Betula
shp data spW Betula <- shp
shp_data_spW_Betula$ID_pl          <-          paste(shp_data_spW_Betula$IGN,
shp_data_spW_Betula$NPL,sep="") # creating the same "ID_pl" for joining the
shapefile
shp_data_spW_Betula@data <- join(shp_data_spW_Betula@data, data_spW_Betula, by
= "ID_pl", type = "left")

writeOGR(shp_data_spW_Betula,   "C:/Users/dujardis/Documents    UNamur/20160401
RespirIT/3_ANALYSIS/Traitements_InventaireForestierWallon/IFW_R_Outputs",
"data_spW_Betula", driver="ESRI Shapefile", layer_options= c(encoding= "UTF-
8"), overwrite_layer=T)

#d) Corylus
shp_data_spW_Corylus <- shp
shp_data_spW_Corylus$ID_pl         <-         paste(shp_data_spW_Corylus$IGN,
shp_data_spW_Corylus$NPL,sep="") # creating the same "ID_pl" for joining the
shapefile
shp_data_spW_Corylus@data <- join(shp_data_spW_Corylus@data, data_spW_Corylus,
by = "ID_pl", type = "left")

writeOGR(shp_data_spW_Corylus,   "C:/Users/dujardis/Documents   UNamur/20160401
RespirIT/3_ANALYSIS/Traitements_InventaireForestierWallon/IFW_R_Outputs",
"data_spW_Corylus", driver="ESRI Shapefile", layer_options= c(encoding= "UTF-
8"), overwrite_layer=T)
```

1.2. Flemish Forest Inventory (RespirIT_IFF_4.R)

```
##################### SUMMARY ###################################

# PACKAGES
# OVERALL APPROACH
# DATA IMPORT
# DATA PROCESSING
# PLOTTING
# FINAL TABLE
# MAPPING

##################### PACKAGES ##################################

#.rs.restartR()

options(java.parameters = "- Xmx1024m") # to overcome the "OUtOfMemoryError
(Java)" occuring when loading xlsx files with too icj column entries (GC overhead
limit exceeded)
library("rJava", lib.loc="~/R/win-library/3.3")
#library("xlsx", lib.loc="~/R/win-library/3.3") #! Conflicting with XLConnect
when loaded
library("XLConnect", lib.loc="~/R/win-library/3.3")
library("reshape", lib.loc="~/R/win-library/3.3")
library("reshape2", lib.loc="~/R/win-library/3.3")
library("stringi", lib.loc="~/R/win-library/3.3")
library("stringr", lib.loc="~/R/win-library/3.3")
library("shapefiles", lib.loc="~/R/win-library/3.3")
library("plyr", lib.loc="~/R/win-library/3.3")
library("rgdal", lib.loc="~/R/win-library/3.3")
library("vioplot", lib.loc="~/R/win-library/3.3")
library("ggplot2", lib.loc="~/R/win-library/3.3")
library("shapefiles", lib.loc="~/R/win-library/3.3")
library("rgdal", lib.loc="~/R/win-library/3.3")
library(xlsx)


##################### OVERALL APPROACH ###########################

# Importing datasets: "Plots" and "Trees_2eBosinv"
# Selecting and sorting observations
# Computing a basal area (gha) for the 3 selected allergenic trees
# Computing a global basal area (GHA) for all trees
# Comparing these two by computing an overall relative basal area (pcGHA)

# Main tables generated
#data_spF_ALL # All tree species within the dataset
#data_spF_ALL3 # All 3 allergenic tree species
#data_spF_Alnus # Alnus
#data spF Betula # Betula
#data_spF_Corylus # Corylus

##################### DATA IMPORT ################################

setwd("C:/Users/dujardis/Documents UNamur/20160401 RespirIT")

## Importing sheet "Trees_2eBosinv"
```

```
data_import_sp<-loadWorkbook("C:/Users/dujardis/Documents      UNamur/20160401 RespirIT/3 ANALYSIS/Traitements InventaireForestierFlamand/IFF DatabaseExtracts/Trees_2eBosinv_EmptyFieldsRemoved.xlsx")
data_import_sp  <-readWorksheet(data_import_sp,  sheet  =  "Trees_2eBosinv", startRow = 1)

#colnames(data_import_sp)

data_spF <- data_import_sp

data_spF <- data_spF[,c(
  "IDPlots", # Plot's ID
  "X_m",
  "Y_m",
  "ID", # Specie's ID
  "Species",
  #"Species_Scientific", #empty
  "DBH_mm", # Diameter at Breast Height
  "Height_m",
  "Perimeter_cm", # Tree perimeter
  "Status_tree", # Dead (2) or alive (1)
  "CodeCoppice_Individual" # 'Individuele boom' (10) or 'Hakhoutstoof' (20)
  #"DummyID_A3",
  #"DummyID_A4"
)] # Variables selected so far...


## Importing sheet "Plots"

data_import_pl<-loadWorkbook("C:/Users/dujardis/Documents      UNamur/20160401 RespirIT/3 ANALYSIS/Traitements InventaireForestierFlamand/IFF DatabaseExtracts/Plots.xlsx")
data_import_pl <- readWorksheet(data_import_pl, sheet = "Plots", startRow = 1)

#colnames(data_plF)

data_plF <- data_import_pl
data_plF <- data_plF[,c(
  "ID",
  "PlotSize",
  "PlotShape",
  "Area_m2",
  "Plot_type"
)] # Variables selected so far...

table(data_plF$PlotSize)
# = Total number of plots within the 2nd inventory



#################### DATA PROCESSING ###################################

# TOTAL Nbr of observations =
nrow(data_spF)

# Check Rows with NAs
nrow(data_spF[is.na(data_spF$ID),] )
#data_spF <- data_spF[!is.na(data_spF$ID),] # Cleaning up rows with NAs

# Nbr of dead trees =
nrow(data_spF[data_spF$Status_tree == 2,])
# Pecentage =
nrow(data_spF[data_spF$Status_tree == 2,])/nrow(data_spF)*100

data_spF <- data_spF[!data_spF$Status_tree == 2,] # Removing dead trees

# "data_spF" = dataset with all tree species

# Creating a dataset with 3 selected tree species (Alnus, Betula & Corylus)

data_spF__D <- data_spF
data_spF__D <- data_spF__D[data_spF__D$Species == 10 | # Alnus glutinosa (L.) Gaertn.
                           data_spF__D$Species == 11 | # Alnus incana (L.) Moench
                          #data_spF__D$Species == 1470 | # Alnus cordata (Loisel.) Duby
                           data_spF__D$Species == 12 | # Betual tremula/alba
                          #data_spF__D$Species == 146 | # Betula alba L.
                          #data_spF__D$Species == 128 | # Betula pendula Roth
                           data_spF__D$Species == 14 , ] # Corylus avellana L.

#< Replacing figures by names
data_spF__D[,"Species"] <- gsub(10, "Aulne", data_spF__D[,"Species"]) #another option
data_spF__D[,"Species"] <- gsub(11, "Aulne", data_spF__D[,"Species"]) #another option
data_spF__D[,"Species"]  <-  gsub(12,  "Bouleau",  data_spF__D[,"Species"]) #another option
data_spF__D[,"Species"]  <-  gsub(14,  "Noisetier",  data_spF__D[,"Species"]) #another option

nrow(data_spF__D)

# -> 7.855 observations if keeping all types of Alnus/Betula/Corylus
# -> 7.848 observations if keeping only Alnus glutinosa, Alnus incana, Betula tremula/alba, and Corylus avellana
# 7 obsevations are omitted > Let's not bother with them


## Tree Structure ##


data_spF_melt  <-  melt(data_spF__D,  id  =  c("IDPlots",  "X_m",  "Y_m",  "ID", "Species", "Status_tree", "CodeCoppice_Individual"))
temp_aggr <- cast(data_spF_melt, Species ~ variable)
# "Aggregation requires fun.aggregate: length used as default" > usual if "sum" not defined (calculates freq. by default)
#table_spF_tree_str <- temp_aggr[,c("essence","tai_NA","fut_1","fut_2")]
temp_aggr <- temp_aggr[,1:2 ]
colnames(temp_aggr) <- c("species","Count")
table_spF_tree_str <- temp_aggr
table_spF_tree_str
# ! No information about trees' structure (coppice vs tree stand)
```

```
## Allergenic tree species' basal area ##

## Based on Afdeling Bos & Groen (2001), p29-30,
# Calculating species' basal area (gha) in m2/ha requires:

# Tree perimeter (cm)
# Plots' type (only A3 & A4 ?)
# Extension factor(*):
# For A3 plots, R3=9m, F3=39.30
# For A4 plots, R4=18m, F4=9.82

# * Allows extrapolating the basal area to the hectare
#  !  Walloon  Forest  Inventory:  F4=9,80  for  trees  greater  than  120cm circumference...


# Testing variable consistancy for "DummyID_A3" and "DummyID_A4"
#summary(data_spF__D) # variables summary
#table(data_spF__D$DummyID_A3) # values summary (0 to 71 ?)
#table(data_spF__D$DummyID_A4) # values summary (0 to 16 ?)
#table(is.na(data_spF__D$DummyID_A3)) # Nbr of NAs
#table(is.na(data_spF__D$DummyID_A4)) # Nbr of NAs
#nrow(data spF  D[is.na(data spF  D$DummyID A3)&is.na(data spF  D$DummyID A4),]) # When both A3 & A4 contain NAs
#1039/7855*100
# For 13,2% of the sample, we have no information about the plot type (A3 or A4)
# These entries must be removed for caclulating species' basal area

# Keeping entries where Perimeter is available
# i.e. Removing 1 entry...
nrow(data_spF__D[is.na(data_spF__D$Perimeter_cm),])
data_spF__D <-data_spF__D[ !is.na(data_spF__D$Perimeter_cm),]
#nrow(data_spF__D[is.na(data_spF__D$Height_m),])
#View(data spF  D[is.na(data spF  D$Height m),])

# Assiging the nbr 3 for A3 plots
# Assigning the nnbr 4 for A4 plots
# Assining the nbr 0 for lines with both A3 & A4 == 0

data_spF__D$Plot_type <- 99
for(i in 1:nrow(data_spF__D)) {
  if(#data_spF__D[i,"Height_m"]>= 2 &
           data_spF__D[i,"Perimeter_cm"]< 22)  {
    data_spF__D[i,"Plot_type"]  <- 2
  }
  else if (#data_spF__D[i,"Height_m"]>= 2 &
           data_spF__D[i,"Perimeter_cm"]>= 22 &
           data_spF__D[i,"Perimeter_cm"]< 122)  {
    data_spF__D[i,"Plot_type"]  <- 3
  }
  else if (#data spF  D[i,"Height m"]>= 2 &
           data_spF__D[i,"Perimeter_cm"]>= 122)  {
    data_spF__D[i,"Plot_type"]  <- 4
  }
  else {
  }
}

table(data_spF__D$Plot_type)

#temp <- data spF  D[data spF  D$Plot type==0,]
#1774/7855*100
#temp           <-           data_spF__D[data_spF__D$Plot_type==0           & data_spF__D$Species=="Noisetier",]
#337/1774*100
# For 22.6% of the sample, no plot type assigned (!)
# Many (19%) are corylus
# Total uncomplete entries for working with the "DummyID_A3" and "DummyID_A4" variables
#1774+1039
#(2813)/7855*100
# > loosing 35,8% > Too much > To be addressed later on (!)

#data_spF__D <- data_spF__D[!data_spF__D$Plot_type==0,] # Carrying on with the 5.035 entries remaining...

## Computing the the basal area ('gha')
data spF  D$gha <- (data spF  D$Perimeter cm^2)/(4*pi*10000)

#Assigning the extension factor
for(i in 1:nrow(data_spF__D)) {
  if(data_spF__D[i,"Plot_type"]== 4 )  {
    data_spF__D[i,"gha"]  <- data_spF__D[i,"gha"] * 9.82
  }
  else if (data_spF__D[i,"Plot_type"]== 3 )  {
    data_spF__D[i,"gha"]  <- data_spF__D[i,"gha"] * 39.30
  }
  else if (data_spF__D[i,"Plot_type"]== 2 )  {
    data_spF__D[i,"gha"]  <- data_spF__D[i,"gha"] * 157.19
  }
    else if (data_spF__D[i,"Plot_type"]== 1 )  {
    data_spF__D[i,"gha"]  <- data_spF__D[i,"gha"] * 628.76
  }
  else { data_spF__D[i,"gha"]  <- NA
  }
}
#summary(data_spF__D)

# Aggregating 'gha' per plot
data_spF_melt  <-  melt(data_spF__D,  id  =  c("IDPlots",  "X_m",  "Y_m",  "ID", "Species", "Status tree", "CodeCoppice Individual"))

# ALL3
data_spF_ALL3 <- cast(data_spF_melt, IDPlots ~ variable, sum) # Summing each tree's 'gha' per plot

# Alnus, Betula & Corylus
data_spF_Alnus  <-  cast(data_spF_melt,  IDPlots  ~  variable,  sum,  subset  =  Species=="Aulne" )
```

```
data_spF_Betula <- cast(data_spF_melt, IDPlots ~ variable, sum, subset =
Species=="Bouleau")
data_spF_Corylus <- cast(data_spF_melt, IDPlots ~ variable, sum, subset =
Species=="Noisetier")



## All tree species' basal area ##

# Testing variable consistancy for "DummyID_A3" and "DummyID_A4"
# summary(data_spF) # variables summary
# table(data_spF$DummyID_A3) # values summary (0 to 71 ?)
# table(data_spF$DummyID_A4) # values summary (0 to 16 ?)
# table(is.na(data spF$DummyID A3)) # Nbr of NAs
# table(is.na(data_spF$DummyID_A4)) # Nbr of NAs
#   nrow(data_spF[is.na(data_spF$DummyID_A3)&is.na(data_spF$DummyID_A4),])    #
When both A3 & A4 contain NAs

# Keeping entries where Perimeter is available
# i.e. Removing 2 entries...
nrow(data_spF[is.na(data_spF$Perimeter_cm),])
data_spF <-data_spF[ !is.na(data_spF$Perimeter_cm),]
#nrow(data_spF__D[is.na(data_spF__D$Height_m),])
#View(data_spF__D[is.na(data_spF__D$Height_m),])

# Removing lines When both A3 & A4 contain NAs
#          data_spF          <-data_spF[!is.na(data_spF$DummyID_A3)          &
!is.na(data_spF$DummyID_A4),]

# Assiging the nbr 3 for A3 plots
# Assigning the nnbr 4 for A4 plots
# Assining the nbr 0 for lines with both A3 & A4 == 0

data_spF$Plot_type <- 99
for(i in 1:nrow(data_spF)) {
  if(#data_spF__D[i,"Height_m"]>= 2 &
    data spF[i,"Perimeter cm"]< 22)  {
    data_spF[i,"Plot_type"]  <- 2
  }
  else if (#data_spF__D[i,"Height_m"]>= 2 &
    data_spF[i,"Perimeter_cm"]>= 22 &
    data_spF[i,"Perimeter_cm"]< 122)  {
    data_spF[i,"Plot_type"]  <- 3
  }
  else if (#data_spF__D[i,"Height_m"]>= 2 &
    data_spF[i,"Perimeter_cm"]>= 122)  {
    data_spF[i,"Plot_type"]  <- 4
  }
  else {
  }
}
table(data_spF$Plot_type)
#temp <- data_spF[data_spF$Plot_type==0,]

data_spF <- data_spF[!data_spF$Plot_type==0,] # Carrying on without the entrie
where plot_type = 0


## Computing the basal area ('gha')
data_spF$gha_all <- (data_spF$Perimeter_cm^2)/(4*pi*10000)

#Assigning the extension factor
for(i in 1:nrow(data_spF)) {
  if(data_spF[i,"Plot_type"]== 4 )  {
    data spF[i,"gha all"]  <- data spF[i,"gha all"] * 9.82
  }
  else if (data_spF[i,"Plot_type"]== 3 )  {
    data_spF[i,"gha_all"]  <- data_spF[i,"gha_all"] * 39.30
  }
  else if (data_spF[i,"Plot_type"]== 2 )  {
    data spF[i,"gha all"]  <- data spF[i,"gha all"] * 157.19
  }
  else if (data_spF[i,"Plot_type"]== 1 )  {
    data_spF[i,"gha_all"]  <- data_spF[i,"gha_all"] * 628.76
  }
  else { data_spF[i,"gha_all"]  <- NA
  }
}
#table(is.na(data_spF$gha_all))

# Aggregating
data spF melt <- melt(data spF, id = c("IDPlots", "X m", "Y m", "ID", "Species",
"Status_tree", "CodeCoppice_Individual"))
data_spF_ALL <- cast(data_spF_melt, IDPlots ~ variable, sum) # Aggregating 'gha'
per plot

#Cleaning
#data spF ALL$DBH mm <- NULL
#data_spF_ALL$Height_m <- NULL
#data_spF_ALL$Perimeter_cm <- NULL
#data_spF_ALL$Plot_type <- NULL

#Adding XY coordinates
#data spF ALL$X
#data_spF_ALL$Y


## Calculating the relative basal area in % ##

# ALL
data_spF_ALL3$gha_TOT <- 0
data_spF_ALL3$gha_TOT   <-   data_spF_ALL$gha_all[match(data_spF_ALL3$IDPlots,
data_spF_ALL$IDPlots)]
data_spF_ALL3$pcGHA_ALL3 <- data_spF_ALL3$gha/data_spF_ALL3$gha_TOT*100
hist(data spF ALL3$pcGHA ALL3)

# Alnus

data_spF_Alnus$gha_TOT <- 0
data_spF_Alnus$gha_TOT<-    data_spF_ALL$gha_all[match(data_spF_Alnus$IDPlots,
data spF ALL$IDPlots)]
data_spF_Alnus$pcGHA_Alnus <- data_spF_Alnus$gha/data_spF_Alnus$gha_TOT*100
hist(data_spF_Alnus$pcGHA_Alnus)
```

```
# Betula

data_spF_Betula$gha_TOT <- 0
data_spF_Betula$gha_TOT<-  data_spF_ALL$gha_all[match(data_spF_Betula$IDPlots,
data_spF_ALL$IDPlots)]
data_spF_Betula$pcGHA_Betula
data spF Betula$gha/data spF Betula$gha TOT*100
hist(data_spF_Betula$pcGHA_Betula)

# Corylus

data_spF_Corylus$gha_TOT <- 0
data spF Corylus$gha TOT<-
data_spF_ALL$gha_all[match(data_spF_Corylus$IDPlots, data_spF_ALL$IDPlots)]
data_spF_Corylus$pcGHA_Corylus
data_spF_Corylus$gha/data_spF_Corylus$gha_TOT*100
hist(data_spF_Corylus$pcGHA_Corylus)


#################### PLOTTING ############################

## Violin plot
# With Alnus, Betula, Corylus, and ALL3

temp1 <-data.frame(data_spF_Alnus$pcGHA_Alnus)
temp1$tree <- c("Alnus")
colnames(temp1)[colnames(temp1) == "data_spF_Alnus.pcGHA_Alnus"] <- "pcgha"
temp2 <- data.frame(data_spF_Betula$pcGHA_Betula)
temp2$tree <- "Betula"
colnames(temp2)[colnames(temp2) == "data spF Betula.pcGHA Betula"] <- "pcgha"
temp3 <- data.frame(data_spF_Corylus$pcGHA_Corylus)
temp3$tree <- "Corylus"
colnames(temp3)[colnames(temp3) == "data_spF_Corylus.pcGHA_Corylus"] <- "pcgha"
temp4 <- data.frame(data_spF_ALL3$pcGHA_ALL)
temp4$tree <- "All"
colnames(temp4)[colnames(temp4) == "data spF ALL3.pcGHA ALL"] <- "pcgha"

mean(temp1$pcgha)
mean(temp2$pcgha)
mean(temp3$pcgha)
mean(temp4$pcgha)

temp_plot <- rbind(temp4, temp1, temp2, temp3)

temp_plot$tree <- factor(temp_plot$tree, levels=c("Alnus", "Betula", "Corylus",
"All"))

colour_type <- c(Alnus = "grey",
                 Betula = "grey",
                 Corylus ="grey",
                 All = "black")

mygraph <- ggplot(temp plot, aes(x=tree, y=pcgha)) +
  geom_violin(aes(fill = factor(tree)), scale = "area") +
  stat_summary(fun.y=median, geom="point", size=2, color="red") +
  # geom_boxplot(width=0.05) +
  scale fill manual(values= colour type, name="Tree type") +
  scale_y_continuous(breaks=seq(0, 100, 20)) +
  guides(fill=guide_legend(reverse=FALSE)) +
  theme_minimal()
print(mygraph)
ggsave(mygraph,                                    file="data_spF_Violing-Plot.tiff",
path="C:/Users/dujardis/Documents                                  UNamur/20160401
RespirIT/3_ANALYSIS/Traitements_InventaireForestierFlamand/IFF_R_Outputs/",
width=6, height=5, dpi=250)


############# FINAL TABLE #################

data_spF_FULL                          <-                          merge(data_spF_ALL,
data_spF_Alnus[,c("IDPlots","gha","pcGHA_Alnus")], by="IDPlots", all=TRUE)   #
merge by row names (by=0 or by="row.names")
colnames(data_spF_FULL)[colnames(data_spF_FULL) == "gha"] <- "gha_alnus"
data_spF_FULL                          <-                          merge(data_spF_FULL,
data_spF_Betula[,c("IDPlots","gha","pcGHA_Betula")], by="IDPlots", all=TRUE)  #
merge by row names (by=0 or by="row.names")
colnames(data_spF_FULL)[colnames(data_spF_FULL) == "gha"] <- "gha_betula"
data_spF_FULL                          <-                          merge(data_spF_FULL,
data spF Corylus[,c("IDPlots","gha","pcGHA Corylus")], by="IDPlots", all=TRUE)
# merge by row names (by=0 or by="row.names")
colnames(data_spF_FULL)[colnames(data_spF_FULL) == "gha"] <- "gha_corylus"
data_spF_FULL                          <-                          merge(data_spF_FULL,
data_spF_ALL3[,c("IDPlots","gha","pcGHA_ALL3")],  by="IDPlots",  all=TRUE)    #
merge by row names (by=0 or by="row.names")
colnames(data spF FULL)[colnames(data spF FULL) == "gha"] <- "gha all3"
data_spF_FULL[is.na(data_spF_FULL )] <- 0                  # replace NA values

write.xlsx(data_spF_FULL,     "C:/Users/dujardis/Documents     UNamur/20160401
RespirIT/3_ANALYSIS/Traitements_InventaireForestierFlamand/IFF_R_Outputs/data_
spF_FULL.xlsx")

#################### MAPPING ############################


### Building the output shapefile

# shp info
ogrInfo("C:/Users/dujardis/Documents                              UNamur/20160401
RespirIT/3_ANALYSIS/Traitements_InventaireForestierFlamand", "invb2")
#ogrInfo("C:/Users/dujardis/Documents                             UNamur/20160401
RespirIT/2_DATA/Inventaire forestier Flamand/From L. Goveare - LNE Vlanderen
20161121/invb2", "invb2")

# import shp
shp      <-      readOGR(dsn="C:/Users/dujardis/Documents      UNamur/20160401
RespirIT/3_ANALYSIS/Traitements_InventaireForestierFlamand",     layer="invb2",
stringsAsFactors=FALSE, encoding="UTF-8")
#shp      <-      readOGR(dsn="C:/Users/dujardis/Documents      UNamur/20160401
RespirIT/2_DATA/Inventaire forestier Flamand/From L. Goveare - LNE Vlanderen
20161121/invb2", layer="invb2", stringsAsFactors=FALSE, encoding="UTF-8")
```

```
#shp      <-      readOGR(dsn="C:/Users/dujardis/Documents      UNamur/20160401
RespirIT/2_DATA/Inventaire    forestier    Wallon/20160606    IFW/Pts_for_CY1",
layer="Pts_forestiers_CY1",    stringsAsFactors=FALSE,    use_iconv    =    TRUE,
encoding="UTF-8")
#test <- data.frame(shp@data)

# Draw points shp
#points(shp, axes=TRUE, border="gray")

# projection info
#print(proj4string(shp))

# Join attributes

# 0) FULL
shp_data_spF_FULL <- shp
#shp_data_spF_ALL3$PLOTNR           <-           paste(shp_data_spF_ALL3$IGN,
shp_data_spF_ALL3$NPL,sep="") # creating the same "PLOTNR" for joining the
shapefile
colnames(data_spF_FULL)[colnames(data_spF_FULL) == "IDPlots"] <- "PLOTNR" #
Renaming the variable describing plots' IDs to match with the shapefile
shp_data_spF_FULL@data <- join(shp_data_spF_FULL@data, data_spF_FULL, by =
"PLOTNR", type = "left")

writeOGR(shp_data_spF_FULL,    "C:/Users/dujardis/Documents    UNamur/20160401
RespirIT/3_ANALYSIS/Traitements_InventaireForestierFlamand/IFF_R_Outputs",
"data_spF_FULL", driver="ESRI Shapefile", layer_options= c(encoding= "UTF-8"),
overwrite_layer=T)

# a) ALL
shp_data_spF_ALL3 <- shp
#shp_data_spF_ALL3$PLOTNR           <-           paste(shp_data_spF_ALL3$IGN,
shp_data_spF_ALL3$NPL,sep="") # creating the same "PLOTNR" for joining the
shapefile
colnames(data_spF_ALL3)[colnames(data_spF_ALL3) == "IDPlots"] <- "PLOTNR" #
Renaming the variable describing plots' IDs to match with the shapefile
#class(data_spF_ALL3$PLOTNR) <- "integer" # changes the class of a selected
column of a df

shp_data_spF_ALL3@data <- join(shp_data_spF_ALL3@data, data_spF_ALL3, by =
"PLOTNR", type = "left")

writeOGR(shp_data_spF_ALL3,    "C:/Users/dujardis/Documents    UNamur/20160401
RespirIT/3_ANALYSIS/Traitements_InventaireForestierFlamand/IFF_R_Outputs",
"data_spF_ALL3", driver="ESRI Shapefile", layer_options= c(encoding= "UTF-8"),
overwrite_layer=T)

# Check for the join whether both "ID_Pl" match
temp1 <- data.frame(shp_data_spF_ALL3@data)
temp2 <- data_spF_ALL3
temp1$check <- 0
j <- 1
while(j <= nrow(temp1)){
  if ( any( temp2$PLOTNR == temp1[j,"PLOTNR"])) {
    temp1[j,"check"] <- 1
  }
  j <- j+1
}
table(temp1$check)
nrow(temp1)
nrow(temp2)
# Clear: all IDs from the table match with shapfile's IDs

#b) Alnus
shp_data_spF_Alnus <- shp
#shp_data_spF_Alnus$PLOTNR          <-          paste(shp_data_spF_Alnus$IGN,
shp_data_spF_Alnus$NPL,sep="") # creating the same "PLOTNR" for joining the
shapefile
colnames(data spF Alnus)[colnames(data spF Alnus) == "IDPlots"] <- "PLOTNR" #
Renaming the variable describing plots' IDs to match with the shapefile
shp_data_spF_Alnus@data <- join(shp_data_spF_Alnus@data, data_spF_Alnus, by =
"PLOTNR", type = "left")

writeOGR(shp_data_spF_Alnus,    "C:/Users/dujardis/Documents    UNamur/20160401
RespirIT/3_ANALYSIS/Traitements_InventaireForestierFlamand/IFF_R_Outputs",
"data_spF_Alnus", driver="ESRI Shapefile", layer_options= c(encoding= "UTF-8"),
overwrite_layer=T)

#c) Betula
shp data spF Betula <- shp
#shp_data_spF_Betula$PLOTNR          <-          paste(shp_data_spF_Betula$IGN,
shp_data_spF_Betula$NPL,sep="") # creating the same "PLOTNR" for joining the
shapefile
colnames(data_spF_Betula)[colnames(data_spF_Betula) == "IDPlots"] <- "PLOTNR" #
Renaming the variable describing plots' IDs to match with the shapefile
shp data spF Betula@data <- join(shp data spF Betula@data, data spF Betula, by
= "PLOTNR", type = "left")

writeOGR(shp_data_spF_Betula,    "C:/Users/dujardis/Documents    UNamur/20160401
RespirIT/3_ANALYSIS/Traitements_InventaireForestierFlamand/IFF_R_Outputs",
"data_spF_Betula", driver="ESRI Shapefile", layer_options= c(encoding= "UTF-
8"), overwrite layer=T)

#d) Corylus
shp_data_spF_Corylus <- shp
#shp_data_spF_Corylus$PLOTNR          <-          paste(shp_data_spF_Corylus$IGN,
shp_data_spF_Corylus$NPL,sep="") # creating the same "PLOTNR" for joining the
shapefile
colnames(data_spF_Corylus)[colnames(data_spF_Corylus) == "IDPlots"] <- "PLOTNR"
# Renaming the variable describing plots' IDs to match with the shapefile
shp_data_spF_Corylus@data <- join(shp_data_spF_Corylus@data, data_spF_Corylus,
by = "PLOTNR", type = "left")

writeOGR(shp_data_spF_Corylus,    "C:/Users/dujardis/Documents    UNamur/20160401
RespirIT/3_ANALYSIS/Traitements_InventaireForestierFlamand/IFF_R_Outputs",
"data_spF_Corylus", driver="ESRI Shapefile", layer_options= c(encoding= "UTF-
8"), overwrite_layer=T)
```

1.3. Belgian Forest Inventry (RespirIT_IF_BEL.R)

```
#################### SUMMARY ###################################

# PACKAGES
# OVERALL APPRAOCH
# DATA IMPORT
# DATA PROCESSING
# PLOTTING
# FINAL TABLE
# MAPPING

#################### PACKAGES ###################################

library("rJava", lib.loc="~/R/win-library/3.3")
library("xlsx", lib.loc="~/R/win-library/3.3") #! If activated > conflict with
XLConnect
library("XLConnect", lib.loc="~/R/win-library/3.3")
library("reshape", lib.loc="~/R/win-library/3.3")
library("reshape2", lib.loc="~/R/win-library/3.3")
library("stringi", lib.loc="~/R/win-library/3.3")
library("stringr", lib.loc="~/R/win-library/3.3")
library("shapefiles", lib.loc="~/R/win-library/3.3")
library("plyr", lib.loc="~/R/win-library/3.3")
library("rgdal", lib.loc="~/R/win-library/3.3")
library("vioplot", lib.loc="~/R/win-library/3.3")
library("ggplot2", lib.loc="~/R/win-library/3.3")
#install.packages('ggplot2', repos='http://cran.rstudio.com', type='source')
library("shapefiles", lib.loc="~/R/win-library/3.3")
library("rgdal", lib.loc="~/R/win-library/3.3")

#.rs.restartR()

##  FUNCTIONS ##

# Multiple plot function

multiplot <- function(..., plotlist=NULL, file, cols=1, layout=NULL) {
  library(grid)

  # Make a list from the ... arguments and plotlist
  plots <- c(list(...), plotlist)

  numPlots = length(plots)

  # If layout is NULL, then use 'cols' to determine layout
  if (is.null(layout)) {
    # Make the panel
    # ncol: Number of columns of plots
    # nrow: Number of rows needed, calculated from # of cols
    layout <- matrix(seq(1, cols * ceiling(numPlots/cols)),
                     ncol = cols, nrow = ceiling(numPlots/cols))
  }

  if (numPlots==1) {
    print(plots[[1]])

  } else {
    # Set up the page
    grid.newpage()
    pushViewport(viewport(layout = grid.layout(nrow(layout), ncol(layout))))

    # Make each plot, in the correct location
    for (i in 1:numPlots) {
      # Get the i,j matrix positions of the regions that contain this subplot
      matchidx <- as.data.frame(which(layout == i, arr.ind = TRUE))

      print(plots[[i]], vp = viewport(layout.pos.row = matchidx$row,
                                      layout.pos.col = matchidx$col))
    }
  }
}


############# FINAL TABLE #################

## FULL

temp1  <- data_spW_FULL
temp1$Region <- c("Wallonia")
colnames(temp1)[colnames(temp1) == "ID_pl"] <- "ID_BEL"
temp1$X <- NULL
temp1$Y <- NULL
colnames(temp1)[colnames(temp1) == "GHA_TOT"] <- "gha_ALL"

temp2  <- data spF FULL
temp2$Region <- c("Flanders")
colnames(temp2)[colnames(temp2) == "PLOTNR"] <- "ID_BEL"
temp2$DBH_mm <- NULL
temp2$Height_m <- NULL
temp2$Perimeter_cm <- NULL
temp2$Plot type <- NULL
colnames(temp2)[colnames(temp2) == "gha_all"] <- "gha_ALL"

data_sp_BEL_FULL <- rbind(temp1, temp2)
data_sp_BEL_FULL <- data_sp_BEL_FULL[,c(
"ID_BEL",
"Region",
"gha_ALL",
"gha_alnus",
"pcGHA_Alnus",
"gha_betula",
"pcGHA Betula",
"gha_corylus",
"pcGHA_Corylus",
"gha_all3",
"pcGHA_ALL3")]

write.xlsx(data sp BEL FULL,    "C:/Users/dujardis/Documents    UNamur/20160401
RespirIT/3_ANALYSIS/Traitements_InventaireForestier/IF_R_Outputs/data_sp_BEL_F
ULL.xlsx")
```

```
## All 3

# Select
#temp1 <- data_spW_ALL3[,c("ID_pl","pcGHA_ALL3")]
#temp2 <- data_spF_ALL3[,c("PLOTNR","pcGHA_ALL3")]
temp1 <- data_spW_FULL[,c("ID_pl","pcGHA_ALL3")]
temp2 <- data spF FULL[,c("PLOTNR","pcGHA ALL3")]
# Rename
colnames(temp1) <- c("ID_BEL","pcGHA_ALL3")
colnames(temp2) <- c("ID_BEL","pcGHA_ALL3")
# Merge
data_sp_BEL_ALL3 <- rbind(temp1, temp2)

#write.xlsx(data_sp_BEL_ALL3,   "C:/Users/dujardis/Documents   UNamur/20160401
RespirIT/3_ANALYSIS/Traitements_InventaireForestier/IF_R_Outputs/data_sp_BEL_A
LL3.xlsx")

## Alnyus
temp1 <- data_spW_FULL[,c("ID_pl","pcGHA_Alnus")]
temp2 <- data_spF_FULL[,c("PLOTNR","pcGHA_Alnus")]
colnames(temp1) <- c("ID_BEL","pcGHA_Alnus")
colnames(temp2) <- c("ID_BEL","pcGHA_Alnus")
data_sp_BEL_Alnus <- rbind(temp1, temp2)

## Betula
temp1 <- data_spW_FULL[,c("ID_pl","pcGHA_Betula")]
temp2 <- data_spF_FULL[,c("PLOTNR","pcGHA_Betula")]
colnames(temp1) <- c("ID_BEL","pcGHA_Betula")
colnames(temp2) <- c("ID_BEL","pcGHA_Betula")
data sp BEL Betula <- rbind(temp1, temp2)

## Corylus
temp1 <- data_spW_FULL[,c("ID_pl","pcGHA_Corylus")]
temp2 <- data_spF_FULL[,c("PLOTNR","pcGHA_Corylus")]
colnames(temp1) <- c("ID_BEL","pcGHA_Corylus")
colnames(temp2) <- c("ID BEL","pcGHA Corylus")
data_sp_BEL_Corylus <- rbind(temp1, temp2)

## SUMMARY TABLE ##


summary(data_sp_BEL_FULL)
table(data_sp_BEL_FULL$Region)

temp <- data_sp_BEL_FULL[!data_sp_BEL_FULL$pcGHA_ALL3==0,]
summary(temp)

mean(temp$pcGHA_ALL3)
mean(temp$pcGHA_ALL3)
mean(temp$pcGHA_ALL3)
mean(temp$pcGHA_ALL3)

temp melt <- melt(temp, id = c("ID BEL", "Region"))
table_sp_BEL_FULL <- cast(temp_melt, Region ~ variable, mean)
table_sp_BEL_FULL

table sp BEL FULL <- cast(temp melt,  ~ variable, mean)

##################### PLOTTING ############################

## Violin plot
# With Alnus, Betula, Corylus, and ALL3

temp1 <-data.frame(data_sp_BEL_Alnus$pcGHA_Alnus)
temp1$tree <- c("Alnus")
colnames(temp1)[colnames(temp1) == "data_sp_BEL_Alnus.pcGHA_Alnus"] <- "pcgha"
temp1 <- temp1[!(temp1$pcgha==0), ]

temp2 <- data.frame(data sp BEL Betula$pcGHA Betula)
temp2$tree <- "Betula"
colnames(temp2)[colnames(temp2)   ==   "data_sp_BEL_Betula.pcGHA_Betula"]   <-
"pcgha"
temp2 <- temp2[!(temp2$pcgha==0), ]

temp3 <- data.frame(data_sp_BEL_Corylus$pcGHA_Corylus)
temp3$tree <- "Corylus"
colnames(temp3)[colnames(temp3)   ==   "data_sp_BEL_Corylus.pcGHA_Corylus"]   <-
"pcgha"
temp3 <- temp3[!(temp3$pcgha==0), ]

temp4 <- data.frame(data_sp_BEL_ALL3$pcGHA_ALL3)
temp4$tree <- "All"
colnames(temp4)[colnames(temp4) == "data_sp_BEL_ALL3.pcGHA_ALL3"] <- "pcgha"
temp4 <- temp4[!(temp4$pcgha==0), ]

mean(temp1$pcgha)
mean(temp2$pcgha)
mean(temp3$pcgha)
mean(temp4$pcgha)

temp_plot <- rbind(temp4, temp1, temp2, temp3)

temp_plot$tree <- factor(temp_plot$tree, levels=c("Alnus", "Betula", "Corylus",
"All"))

colour_type <- c(Alnus = "grey90",
                 Betula = "grey80",
                 Corylus ="grey70",
                 All = "black")

mygraph <- ggplot(temp_plot, aes(x=tree, y=pcgha)) +
  geom_violin(aes(fill = factor(tree)), scale = "area") +
  stat_summary(fun.y=median, geom="point", size=2, color="red") +
  # geom_boxplot(width=0.05) +
  scale_fill_manual(values= colour_type, name="Tree type") +
  scale_y_continuous(breaks=seq(0, 100, 20)) +
  guides(fill=guide_legend(reverse=FALSE)) +
  theme_minimal()
print(mygraph)
ggsave(mygraph,                          file="data_sp_BEL_Violing-Plot.tiff",
path="C:/Users/dujardis/Documents                          UNamur/20160401
```

```
RespirIT/3_ANALYSIS/Traitements_InventaireForestier/IF_R_Outputs/",    width=6,
height=5, dpi=250)


######################## MAPPING ###################################


# import shp
shp      <-      readOGR(dsn="C:/Users/dujardis/Documents      UNamur/20160401
RespirIT/3_ANALYSIS/Traitements_InventaireForestier", layer="IF_BEL_Plots_ID",
stringsAsFactors=FALSE, encoding="UTF-8")

# a) ALL
shp data sp BEL ALL3 <- shp
#colnames(data_sp_BEL_ALL3)[colnames(data_sp_BEL_ALL3)    ==    "IDPlots"]    <-
"ID_BEL" # Renaming the variable describing plots' IDs to match with the
shapefile
shp_data_sp_BEL_ALL3@data <- join(shp_data_sp_BEL_ALL3@data, data_sp_BEL_ALL3,
by = "ID_BEL", type = "left")

writeOGR(shp_data_sp_BEL_ALL3,   "C:/Users/dujardis/Documents   UNamur/20160401
RespirIT/3_ANALYSIS/Traitements_InventaireForestier/IF_R_Outputs",
"data_sp_BEL_pcGHA_ALL3", driver="ESRI Shapefile", layer_options= c(encoding=
"UTF-8"), overwrite_layer=T)

# b) Alnus
shp_data_sp_BEL_Alnus <- shp
shp_data_sp_BEL_Alnus@data         <-         join(shp_data_sp_BEL_Alnus@data,
data_sp_BEL_Alnus, by = "ID_BEL", type = "left")

writeOGR(shp data sp BEL Alnus,  "C:/Users/dujardis/Documents  UNamur/20160401
RespirIT/3_ANALYSIS/Traitements_InventaireForestier/IF_R_Outputs",
"data_sp_BEL_pcGHA_Alnus", driver="ESRI Shapefile", layer_options= c(encoding=
"UTF-8"), overwrite_layer=T)

# c) Betula
shp data sp BEL Betula <- shp
shp_data_sp_BEL_Betula@data        <-        join(shp_data_sp_BEL_Betula@data,
data_sp_BEL_Betula, by = "ID_BEL", type = "left")

writeOGR(shp_data_sp_BEL_Betula,  "C:/Users/dujardis/Documents  UNamur/20160401
RespirIT/3_ANALYSIS/Traitements_InventaireForestier/IF_R_Outputs",
"data sp BEL pcGHA Betula", driver="ESRI Shapefile", layer options= c(encoding=
"UTF-8"), overwrite_layer=T)

# d) Corylus
shp_data_sp_BEL_Corylus <- shp
shp_data_sp_BEL_Corylus@data       <-       join(shp_data_sp_BEL_Corylus@data,
data_sp_BEL_Corylus, by = "ID_BEL", type = "left")

writeOGR(shp_data_sp_BEL_Corylus, "C:/Users/dujardis/Documents UNamur/20160401
RespirIT/3_ANALYSIS/Traitements_InventaireForestier/IF_R_Outputs",
"data_sp_BEL_pcGHA_Corylus",     driver="ESRI     Shapefile",     layer_options=
c(encoding= "UTF-8"), overwrite_layer=T)
```