Supplementary Materials: Improvement of remote sensing-based assessment of defoliation of Pinus sp. caused by Thaumetopoea pityocampa Denis & Schiffermüller and related environmental drivers in south-eastern Spain.

Javier Pérez-Romero [1], Rafael María Navarro-Cerrillo [1,*], Guillermo Palacios-Rodriguez [1], Cristina Acosta [1] and Francisco Javier Mesas-Carrascosa [2]

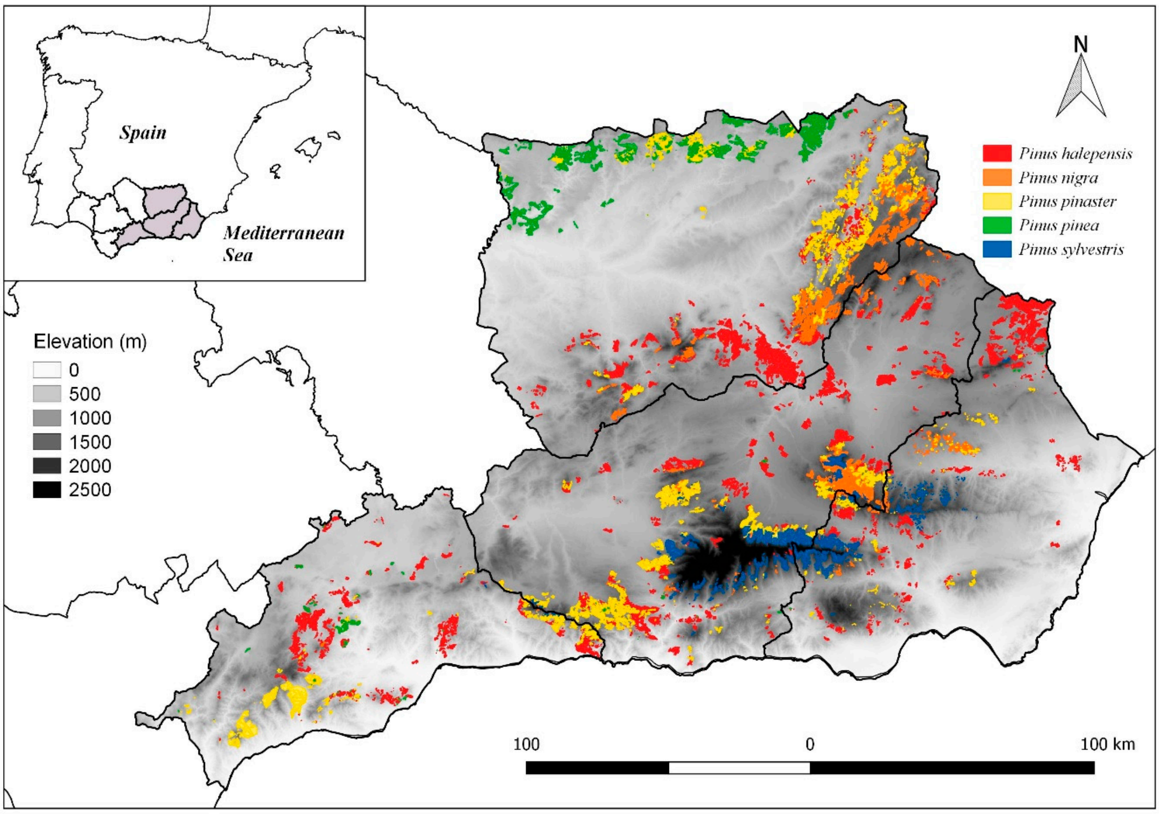

Figure 1. Location of Pinus forests under the annual monitoring campaigns of processionary moth (*Thaumetopoea pityocampa* Denis y Schiffermüller) (1994–2017) using established ground surveys sketch mapping techniques by the Junta de Andalusia Forest Service in Eastern Andalusia (Spain).

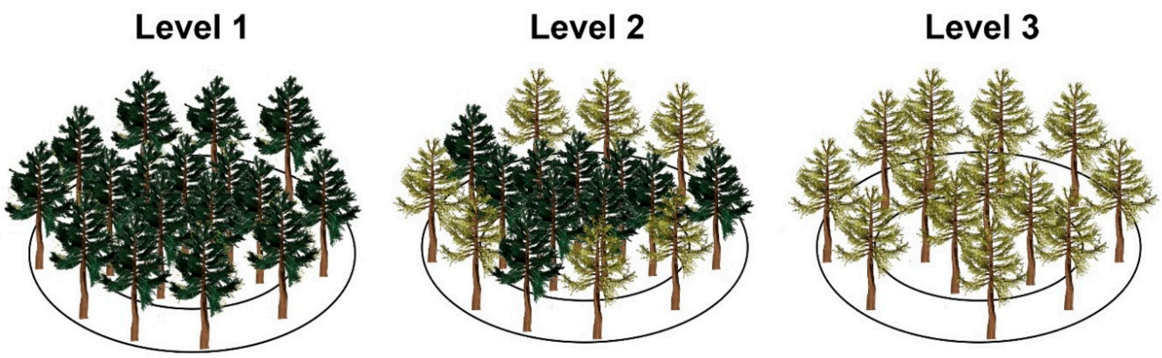

Figure 2. Defoliation grades of processionary moth (*Thaumetopoea pityocampa* Denis y Schiffermüller) using established ground surveys sketch mapping techniques by the Junta de Andalusia Forest Service in Eastern Andalusia (Spain) and area assessment (inside and 60 m-buffer area).

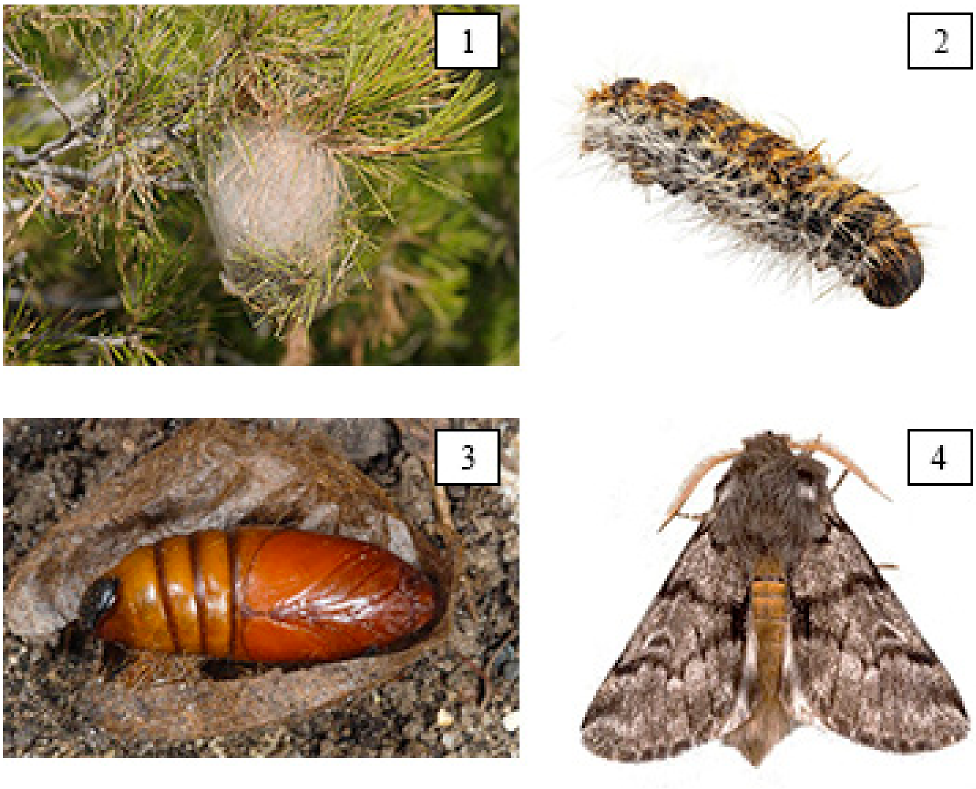

Figure 3. Different stages of the Pine Processionary Moth. (**1**) Nest; (**2**) Larva; (**3**) Pupa; (**4**) Adult. Below the stages the life cycle by *Thaumetopoea pityocampa* is shown.

Table 1. Forestry characteristics of each species from Third National Forest Inventory of Spain.

Jan.	Feb.	Mar.	Apr.	May	Jun.	Jul.	Aug.	Sept.	Oct.	Nov.	Dec.
					Adult	Adult	Adult	Adult			
					Egg	Egg	Egg	Egg	Egg		
Larvae	Larvae							Larvae	Larvae	Larvae	Larvae
		Pupa	Pupa	Pupa	Pupa	Pupa	Pupa				

Table 2. Defoliation grades of *Pinus* forests affected by processionary moth (*Thaumetopoea pityocampa* Denis y Schiffermüller).

Species	Area (ha)	D(Pies/ha)	G (m^2/ha)	V(m^3/ha)	Dm(cm)	FCC(%)
Pinus halepensis	281588	280.88	8.37	33.41	54.25	53.09
Pinus nigra	302471	339.91	9.43	46.34	55.35	42.04
Pinus pinaster	178461	353.94	11.05	46.81	50.81	54.66
Pinus pinea	80866	294.23	7.98	30.65	54.44	32.77
Pinus sylvestris	42262	836.12	18.21	71.24	34.63	48.26

Table 3. Vegetation indices derived from Landsat multispectral bands selected to assess *Pinus* defoliation related to processionary moth .

Grades of defoliation	Description	Grade Reclassified

Level 0	None or some very scattered nests.	Level 1
Level 1	Some nests on the edges of the dough, clear and isolated feet.	
Level 2	Quite a few nests on edges and clear, and some in the centre of the dough.	
Level 3	Partial defoliations on isolated edges and feet and quite a few groups in the centre of the dough.	Level 2
Level 4	Very strong defoliations on isolated edges and feet and partial defoliations in the rest of the stand.	
Level 5	Very strong defoliations throughout the stand	Level 3

Table 4. Climate variables from REDIAM and GEE.

Vegetation Index	Symbols	Equation
Moisture Stress Index	MSI	$\frac{MIR}{NIR}$ (1)
Normalized Difference Infrared Index	NDII	$\frac{(NIR-MIR)}{(NIR+MIR)}$ (2)
Normalized Difference Vegetation Index	NDVI	$\frac{(NIR-RED)}{(NIR+RED)}$ (3)
Ratio Vegetation Index	RVI	$\frac{RED}{NIR}$ (4)

*NIR: Near infrared band; MIR: Mid-infrared band; RED: Red band

Table 4. Climate variables from REDIAM and GEE.

Variable	CODE	UNITS	SOURCE
	Climatic		
Sum of water balances at the end of each month	Bh	mm	REDIAM
Aridity index	Iar		REDIAM
Number of hot days (T-max ≥35 °C)	ndc	Days	REDIAM
Average snow precipitation	snow	mm	REDIAM
Mean temperature warmest month	tmc	ºC	REDIAM
Mean temperature coldest month	tmf	ºC	REDIAM
	Topographic		
Orientation	aspect	Degree	REDIAM
Elevation	Elevation	m	REDIAM
Slope aspect	tp_expo	Degree	REDIAM
Slope steepness	tp_pend	Degree	REDIAM
Radiation in winter	tp_rsd_i	Julian/m^2	REDIAM
Radiation in autumn	tp_rsd_o	Julian/m^2	REDIAM
North to south orientation	tp_su_no	Degree	REDIAM

Variable	CODE	UNITS	SOURCE
Slope	Slope	Degree	REDIAM
Tree cover			
Canopy FCC by trees	fcc_tree	%	REDIAM
Species Coniferous	n_esp	Categorical	REDIAM
Terraclimate			
Temperature minimum February	tmin_february	ºC	GEE
Temperature maximum August	tmax_august	ºC	GEE
Precipitation January	precip_january	mm	GEE
Precipitation February	precip_ february	mm	GEE
Precipitation March	precip_march	mm	GEE
Precipitation April	precip_april	mm	GEE
Precipitation May	precip_may	mm	GEE
Precipitation June	precip_june	mm	GEE
Precipitation July	precip_july	mm	GEE
Precipitation August	precip_august	mm	GEE
Precipitation September	precip_september	mm	GEE
Precipitation October	precip_october	mm	GEE
Precipitation November	precip_november	mm	GEE
Precipitation December	precip_december	mm	GEE

Table 5. Area affected annually for each species and grade of defoliation.

	Pinus halepensis			*Pinus nigra*			*Pinus pinaster*			*Pinus pinea*			*Pinus sylvestris*		
	Level 1	**Level 2**	**Level 3**	**Level 1**	**Level 2**	**Level 3**	**Level 1**	**Level 2**	**Level 3**	Level 1	Level 2	Level 3	Level 1	Level 2	Level 3
1994	183975	17825	440	72081	6261	1010	104738	1121		35664	1503		24801	970	
1995	212623	10233		85364	10391		118509	3441		45068	432		32794	380	
1996	205024	4689	158	80405	3938	369	108059	4574		39295	8421		33397		
1997	177378	18372	2937	51284	6940	74	80297	10677	163	41534	5532	366	26731	1507	
1998	181005	44523	2147	68531	27172	617	101057	22232	111	46161	1555		18125	14701	983
1999	217996	10151	384	65434	29969	2790	114314	5573	66	46373	1343		26471	7734	
2000	197516	13706	1177	93220	4211		116575	4176		36365	11048	304	28789	2686	
2001	231223	7114		101660	1652		128742	1812		47121	710		33614	414	
2002	222824	14854	190	93471	7289		115399	18781	830	36938	10281		25218	733	
2003	231240	17292		91229	15065	97	116328	16074	137	52062	3641		28603	7840	
2004	235316	13560	508	98156	10210		123173	17713		47094	9085		15560	11392	759
2005	249765	3688		99204	5781		137836	3082		49600	6553		23033	5384	
2006	253827	7487		104049	1485		150195	4420		44221	12104		33637		
2007	255820	8343		105406	7704		148614	6773		39055	14167	3104	32329		
2008	253864	11877		99112	14765	291	135297	20713		56255	7845		35109	1725	55
2009	256104	12861	308	94671	21368		128702	24474		60743	3519		33987	1393	292
2010	246420	16775	117	79379	35005	94	135347	11066		60286	4412		17151	2627	
2011	271043	5132		114323	2553		151026	5929	111	53380	11318		26233	10400	456
2012	270555	3144		112531	4525	20	151282	5684	459	64833	90		17705	18186	1227
2013	270564	3580		117240			155730	612		64435	112		35275	1843	
2014	264773	11310		116605	635		146277	9937		51020	12262		36413	408	
2015	256511	20324	211	92884	23532	824	120817	33804	1011	42376	18117	2170	34234	2883	
2016	234419	36261	6030	53262	51142	12822	100815	45056	11445	11040	25049	26574	30791	6123	203

Table 6. Analysis of variance of repeated measurements (ANOVA).

		Level 1	Level 2	Level 3	*P*
INSIDE	MSI	0.763 ± 0.22	0.783 ± 0.23	0.869 ± 0.19	<0.001***
	NDII	0.194 ± 0.13	0.188 ± 0.14	0.127 ± 0.12	<0.001***
	NDVI	0.414 ± 0.18	0.392 ± 0.19	0.457 ± 0.18	<0.001***
	RVI	2.929 ± 1.39	2.84 ± 1.47	3.26 ± 1.42	0.124
BUFFER	MSI	0.787 ± 0.21	0.795 ± 0.22	0.883 ± 0.18	<0.001***
	NDII	0.178 ± 0.13	0.171 ± 0.13	0.114 ± 0.10	<0.001***
	NDVI	0.375 ± 0.17	0.346 ± 0.19	0.311 ± 0.22	<0.001***
	RVI	2.574 ± 1.13	2.432 ± 1.22	2.134 ± 1.45	<0.001***

Table 7. Values of Random Forest with different variables from Vsurf function. Average model coefficients obtained from the likeliest regression models (AICc < 4), estimated using robust sandwich estimator of the covariance matrix. Variables selection for each model according to the order of relevance of the VSURF (Figure 5).

Models	Mean of squared residuals	% Var explained	RMSE	R2	S	Nº Variables
Models 1	0.007	35.7	0.08	0.62	0.58	2
Models 2	0.005	56.8	0.07	0.75	0.70	3
Models 3*	0.004	62.6	0.06	0.79	0.74	4
Models 4	0.004	64.8	0.06	0.80	0.76	5
Models 5	0.004	66.0	0.06	0.80	0.76	6
Models 6	0.004	66.4	0.06	0.81	0.77	7
Models 7	0.004	66.6	0.06	0.81	0.77	8
Models 8	0.004	66.7	0.06	0.81	0.77	9
Models 9	0.004	66.7	0.06	0.81	0.77	10
Models 10	0.004	67.0	0.06	0.81	0.77	11
Models 11	0.004	68.0	0.06	0.82	0.78	12
Models 12	0.004	68.6	0.06	0.82	0.78	13
Models 13	0.004	68.9	0.06	0.82	0.79	14
Models 14	0.003	69.5	0.06	0.83	0.79	15

*Selected model according to VSURF.

Computer code: javascript code

```
///    1 LOAD POLYGONS

var Stands = ee.FeatureCollection('ft:1Jm4AY5QOY1AhAIZ17wFBDFmxO7H9b4cyD6duKsI4');

///    2 UPLOAD IMAGE COLLECTION

//Landsat5

var collection1 = ee.ImageCollection('LANDSAT/LT05/C01/T1_SR')

.filterBounds(Stands)

.filterDate('1992-01-01','2016-12-31');

//Landsat7

var collection11 = ee.ImageCollection('LANDSAT/LE07/C01/T1_SR')

.filterBounds(Stands)

.filterDate('1992-01-01','2016-12-31');
```

```
//Landsat8
var collection111 = ee.ImageCollection('LANDSAT/LC08/C01/T1_SR')
.filterBounds(Stands)
.filterDate('1992-01-01','2016-12-31');
///   3 EXPORT LIST
Export.table.toDrive({
  collection: collection1,
  description:'Landsat5_LIST',
  fileFormat: 'txt'
});
Export.table.toDrive({
  collection: collection11,
  description: 'Landsat7_LIST',
  fileFormat: 'txt'
});
Export.table.toDrive({
  collection: collection111,
  description: 'Landsat8_LIST',
  fileFormat: 'txt'
});
```

Computer code: Python code

```
import ee
ee.Initialize()
def getQABits(image, start, end, mascara):
    # Compute the bits we need to extract.
    pattern = 0
    for i in range(start,end+1):
        pattern += 2**i
    # Return a single band image of the extracted QA bits, giving the    band a new name.
    return image.select([0], [mascara]).bitwiseAnd(start).rightShift(start)
#A function to mask out cloudy pixels.
```

```
def maskQuality(image):

    # Select the QA band.

    QA = image.select('pixel_qa')

    # Get the internal_cloud_algorithm_flag bit.

    sombra = getQABits(QA,3,3,'cloud_shadow')

    nubes = getQABits(QA,5,5,'cloud')

    #  var cloud_confidence = getQABits(QA,6,7,  'cloud_confidence')

    cirrus_detected = getQABits(QA,9,9,'cirrus_detected')

    #var cirrus_detected2 = getQABits(QA,8,8,  'cirrus_detected2')

    #Return an image masking out cloudy areas.

    return image.updateMask(sombra.eq(0)).updateMask(nubes.eq(0)) #.updateMask(cirrus_detected.eq(0)))

KML = ee.FeatureCollection('ft:1uKfaqMx1JH_Q4KfsoJ_RCc_kKB9z_OvfjVkA7rxZ'); #Stands

Contorno = ee.FeatureCollection('ft:1MFU9TOsYu1IGyJjZzPc3bSK2SBfVehxFtmSwTU1S'); #Perimeter

poligono = [[-6.30615234375,36.29741818650811],[-1.669921875,36.29741818650811], [-1.669921875,38.634036452919226], [-
6.328125,38.59970036588818],[-6.30615234375,36.29741818650811]];

archivo_texto='C:\engine\Landsat8_LIST.txt'

landsat ='LC08'

archivo=open(archivo_texto ,'r+')

linea=archivo.readline()

while linea!='':

    #image access

    linea=linea.replace('\n',"")

    print (linea)

    nombre_imagen='LANDSAT/'+landsat+'/C01/T1_SR/'+linea

    imagen = ee.Image(nombre_imagen)

    QA = imagen.select(['pixel_qa'])

    sombra = getQABits(QA, 3, 3, 'cloud_shadow')

    nube = getQABits(QA, 5, 5, 'cloud')

    print(imagen)

    # get info metadata

    informacion = imagen.getInfo()

    propiedades = informacion['properties']
```

```
for elemento in propiedades:
    print (elemento)
print (propiedades['SOLAR_ZENITH_ANGLE'])
valores_nube = [352, 368, 416, 432, 480, 864, 880, 928, 944, 992, 328, 392, 840, 904, 1350]
# scored = ee.Algorithms.Landsat.simpleCloudScore(imagen_TOA);
mascara = imagen.select(['pixel_qa']).neq(322);
imagen_mascara = imagen.updateMask(mascara)
# Calculo del NDVI Landsat 8
RVI = imagen_mascara.expression(
    '((B5) / (B4))', {
        'B5': imagen.select('B5'),
        'B4': imagen.select('B4'),
    }).rename('RVI');
RVI_mascara = RVI
for valor in valores_nube:
    mascara = imagen.select(['pixel_qa']).neq(valor)
    RVI_mascara = RVI_mascara.updateMask(mascara)
mascara=RVI_mascara.select(['RVI']).gte(0)
RVI_mascara = RVI_mascara.updateMask(mascara)
mascara=RVI_mascara.lte(20)
RVI_mascara = RVI_mascara.updateMask(mascara)
#task = ee.batch.Export.image.toDrive(
    #image=NDVI_mascara,
    #region=poligono,
    #scale=30,
    #description=linea
#);
#task.start()
RVI_mean = RVI_mascara.reduceRegions(
    collection=KML,
    reducer=ee.Reducer.mean(),
```

```
        scale=30
    );
    RVI_sd = RVI_mascara.reduceRegions(
        collection=KML,
        reducer=ee.Reducer.stdDev(),
        scale=30
    );
    print (imagen.getInfo()['id']);
    task = ee.batch.Export.table.toDrive(
        collection=RVI_mean,
        description=linea,
        driveFolder='RVI_L8',
        fileFormat='CSV'
    );
    task.start()
    #task = ee.batch.Export.table.toDrive(
        #collection=NDVI_sd,
        #description=linea,
        #fileFormat='KML'
    #);
    #task.start()
    linea = archivo.readline()
archivo.close()
print('fin')
```