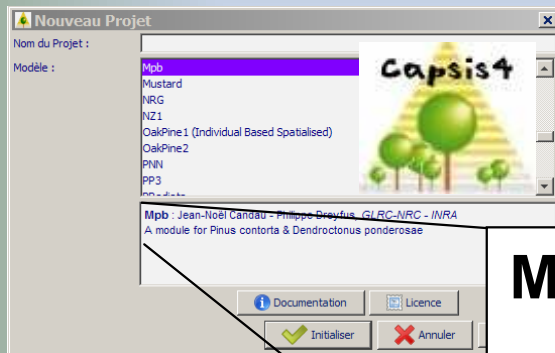


Le module Mpb et la lecture de fichiers « shapefile »



Natural Resources
Canada

Ressources naturelles
Canada



Mpb :

Jean-Noël Candau - Philippe Dreyfus
GLFC-NRC - INRA

A module for *Pinus contorta* & *Dendroctonus ponderosae*



Who's who ?



E-Flora BC: Electronic Atlas
of the Plants of British Columbia

Pinus contorta (lodgepole pine, pin tordu)

Pin formant des peuplements quasi-purs
souvent très denses



Régénération après incendie :



<http://commons.wikimedia.org/wiki/>

Pinus contorta subsp. *latifolia*

Pinus contorta subsp. *contorta*

Pinus contorta subsp. *murrayana*

Dendroctonus ponderosae

Scolyte

ravageur de plusieurs espèces de Pins
en Amérique du Nord

... notamment de *Pinus contorta*
dans le Sud de la Colombie Britannique (Canada)



Jean-Noël Candau

Entomologie forestière

Modélisation des Dynamiques spatio-temporelles

Great Lakes Forestry Centre (Centre de foresterie des Grands Lacs), *Natural Resources Canada*
Sault Ste-Marie, Ontario, Canada

Richard A. Fleming

Analyse des systèmes biologiques

Aménagement des forêts et des paysages, Changements climatiques, Feux de forêt

Canadian Forest Service, Natural Resources Canada,
Sault Ste-Marie, Ontario, Canada

Allan L. Carroll

Entomologie forestière, écologie des populations

Répercussions des changements climatiques sur la dynamique des populations de scolytes

Department of Forest Sciences, Faculty of Forestry,
*University of British Columbia, Vancouver, **BC**, Canada*

Philippe Dreyfus

Dendrométrie, écologie des communautés d'arbres forestiers, autécologie

Modélisation de la croissance et de la dynamique forestière = $f(\text{station, compétition, gestion})$

INRA - URFM, UR Écologie des Forêts Méditerranéennes, Avignon, France

un
Nouveau
Contexte

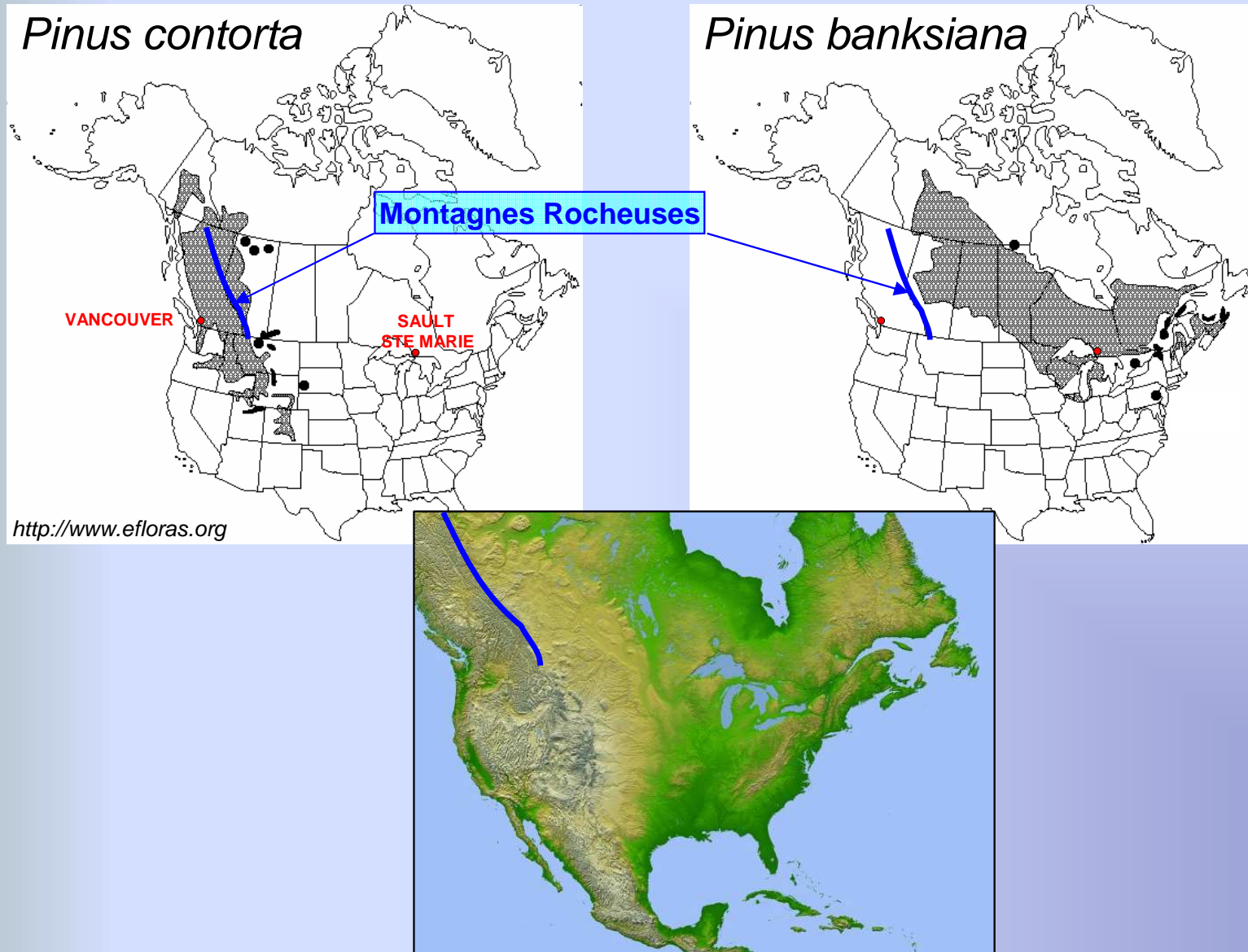
Changements Climatiques

⇒ réchauffement, **hivers doux**

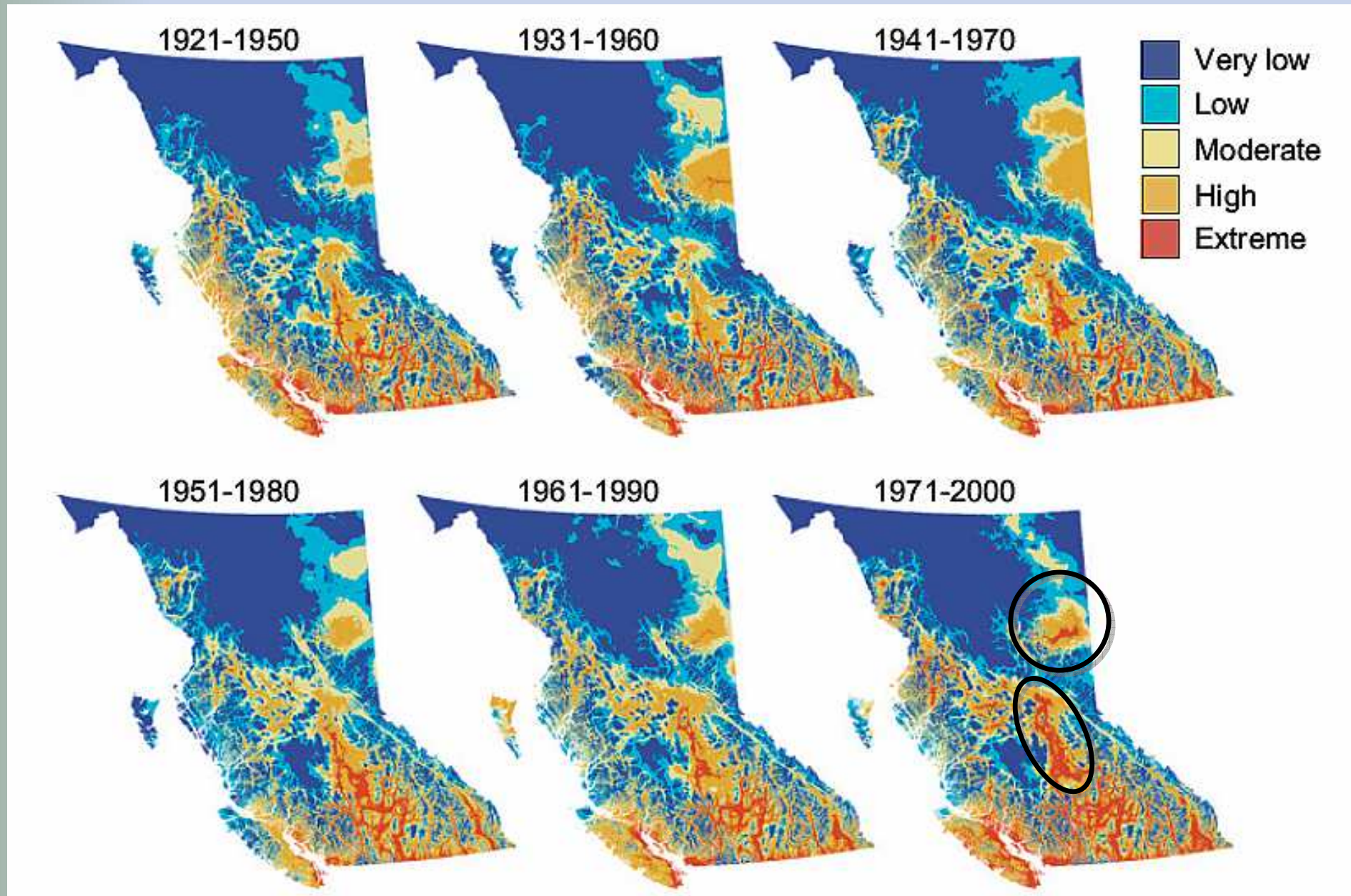
⇒ **barrière** montagneuse (Rocheuses) devient **franchissable** par le MPB

⇒ **risque d'extension vers l'Est** et tout le Canada

notamment aux immenses étendues de ***Pinus banksiana*** (Jack pine, pin gris)

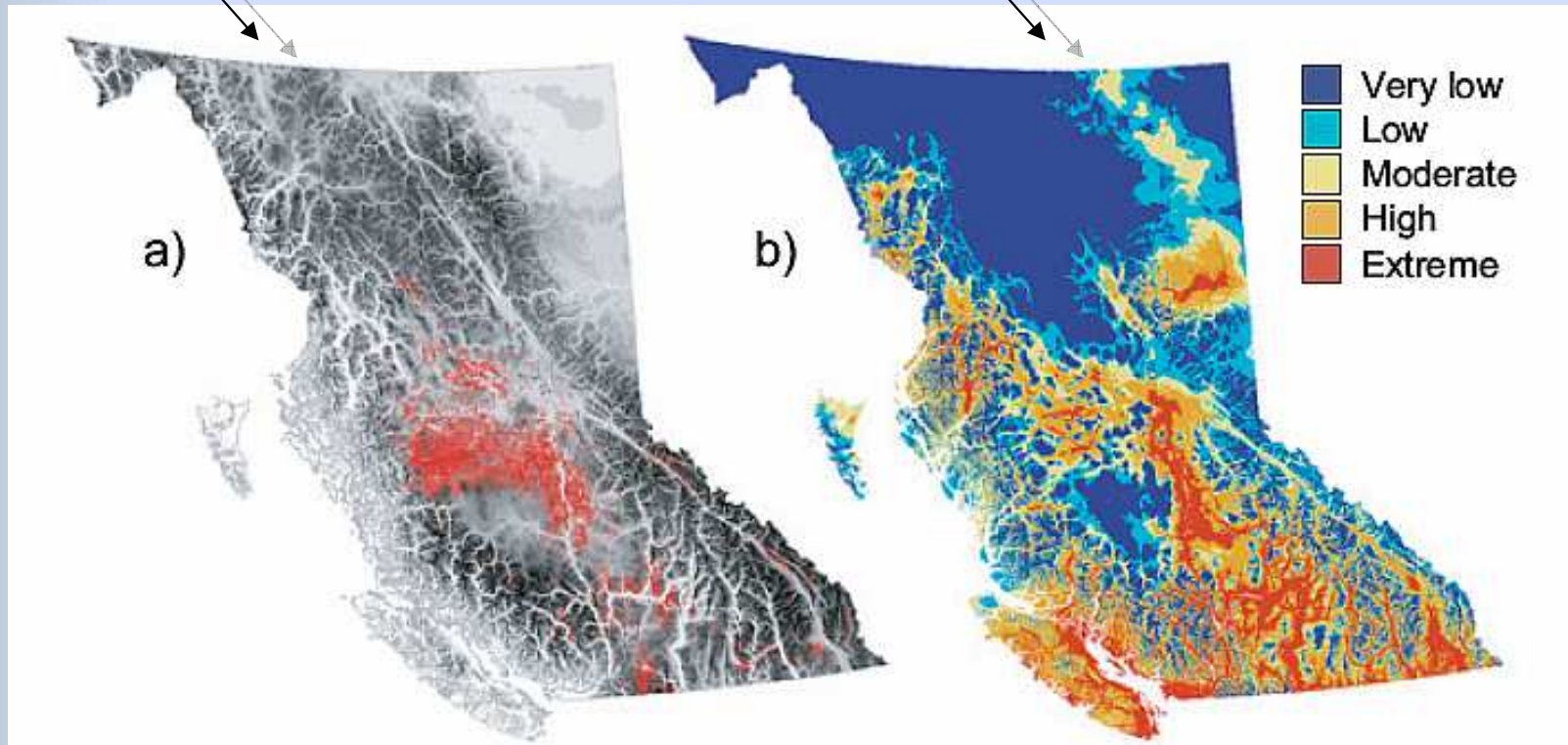


Evolution des classes climatiques +/- favorables au MPB



a) infestations par MPB
de 1998 à 2002

b) classes climatiques
+/- favorables au MPB



Effects of Climate Change on Range Expansion
by the Mountain Pine Beetle in British Columbia
Allan L. Carroll *et al.*, 2003

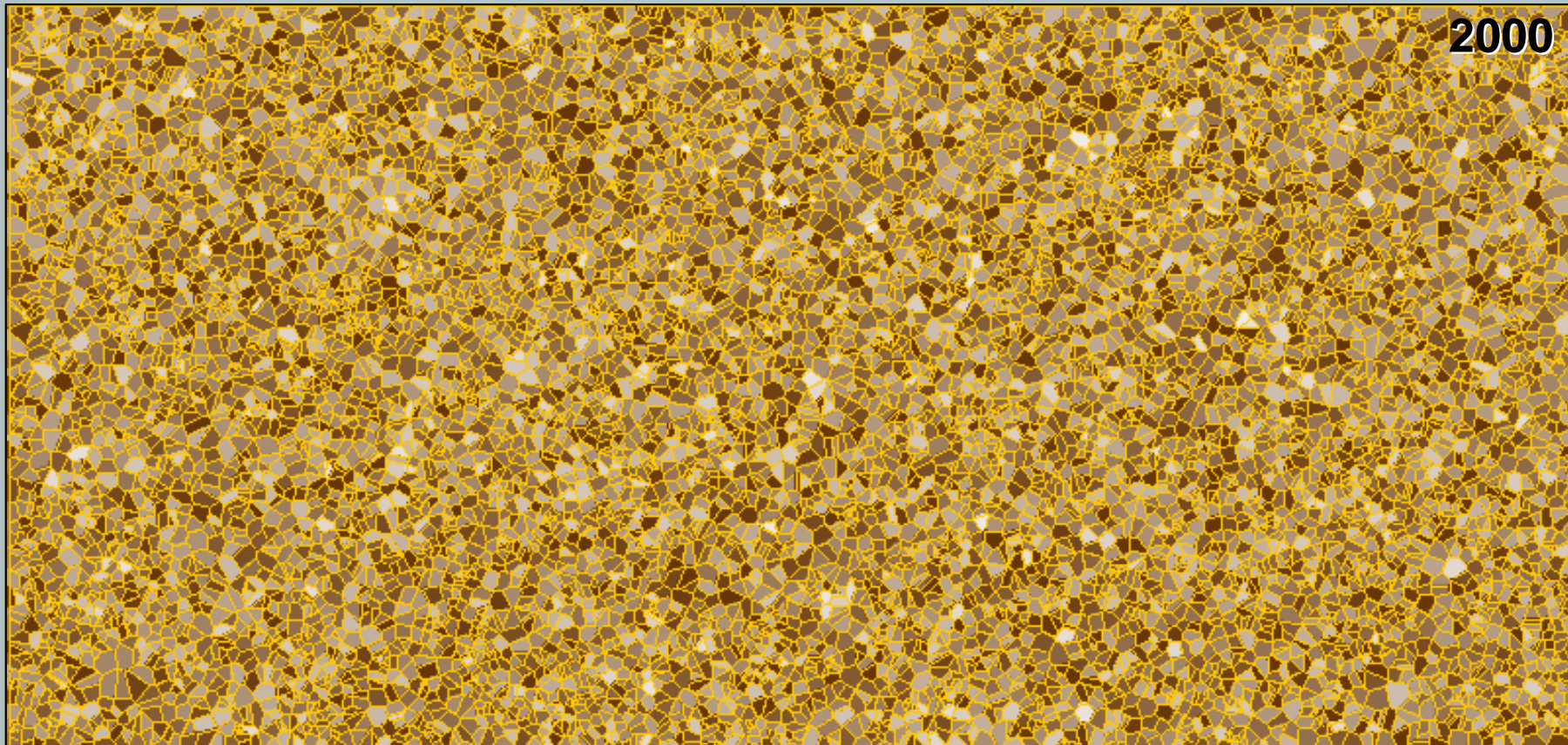
*“ If the current warming trend continues, this pest will expand its range in Alberta. Jack pine (Pinus banksiana Lamb.) is a potential beetle host in Alberta. In northern Alberta, lodgepole and jack pine overlap in distribution and hybridize. If the mountain pine beetle successfully colonizes hybrid lodgepole-jack pine and pure jack pine forests,
Canada will face a major ecological, social and economical disaster.”*

The Mountain Pine Beetle: Scope of the Problem
and Key Issues in Alberta
Hideji Ono, 2004

*Alberta Sustainable Resource Development,
Public Lands and Forests Division,
Edmonton, Alberta*

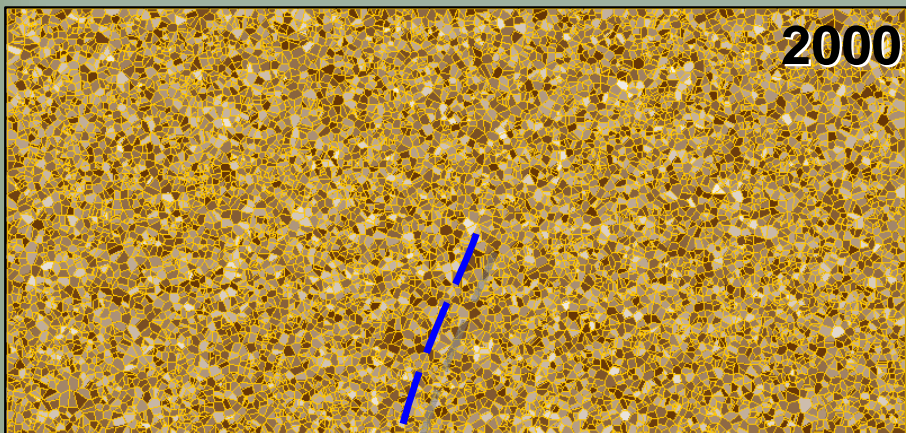
***Exemple de
Simulation
avec le module MPB***

- **Carte de peuplements fictive** ($\approx 10\,000$ parcelles)

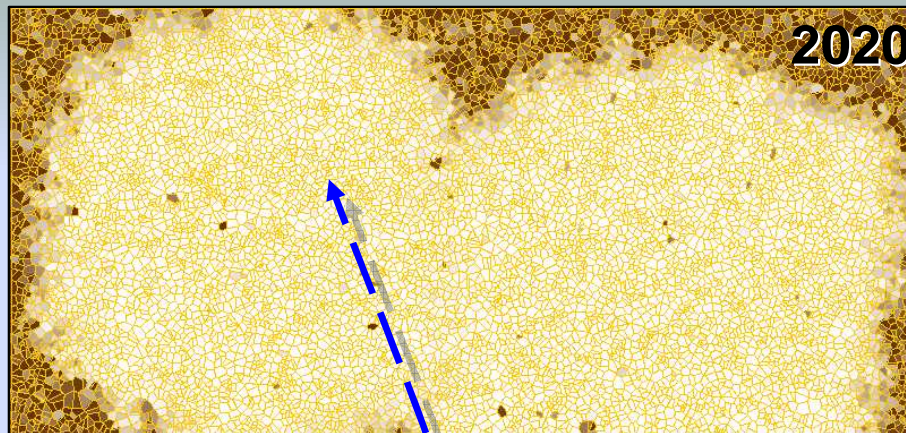


- Information initiale par peuplement, +/- déterminée par la disponibilité en conditions réelles :
« **Site Index** », **N pins /ha**, **Hauteur dominante**
- **3 peuplements** (tirés au hasard) sont **infestés** par le MPB
- Simulation sur **20 ans**, avec un pas annuel

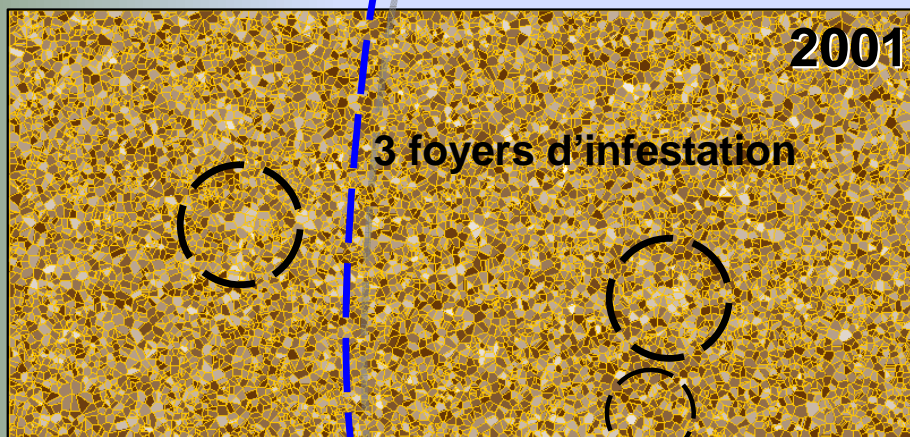
2000



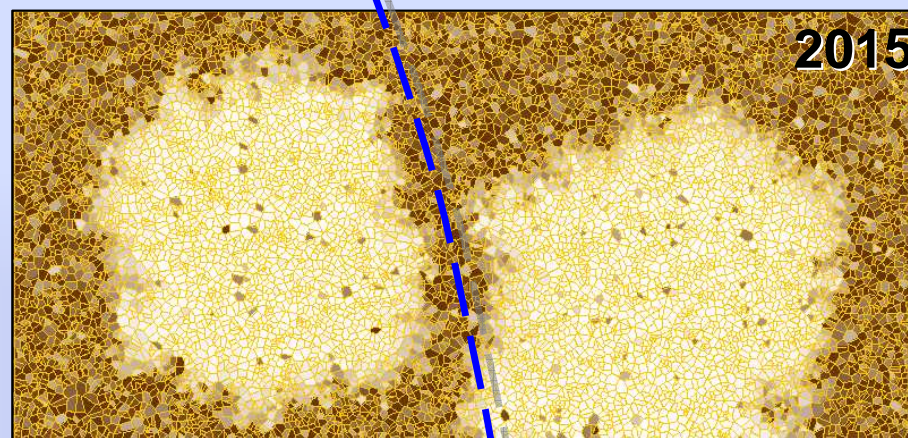
2020



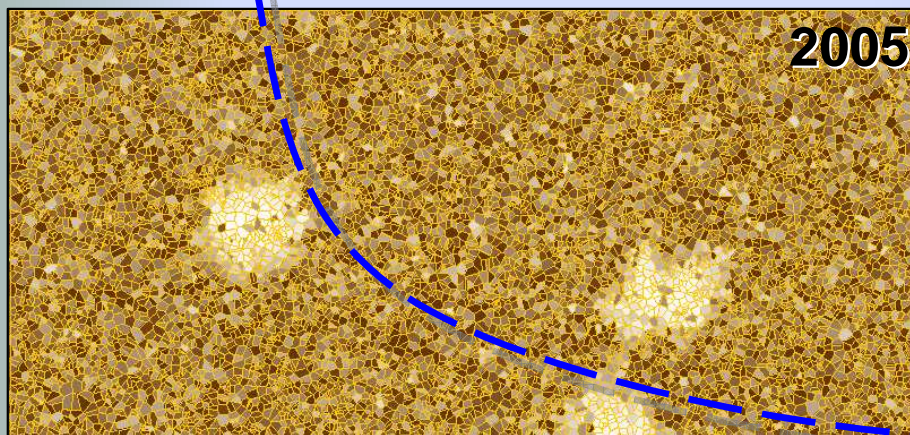
2001



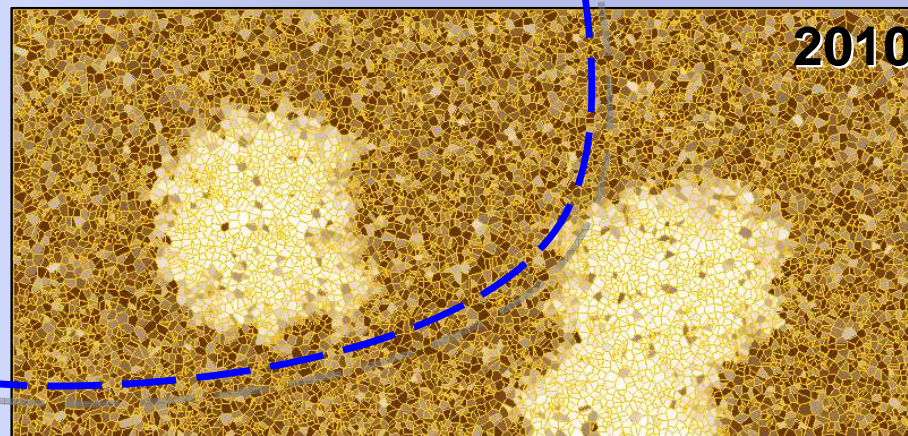
2015



2005



2010



Structure et Fonctionnement du module

● « **terrain** » en **cellules polygonales** : parcelles, peuplements

● Chaque peuplement est représenté par 1 arbre à effectif (« numberable »)

● **Modèle de croissance du Pin** : de type « peuplement » \approx « arbre moyen »

Le modèle met à jour la **hauteur**, l'**âge** (dédit de SI et Hdom), l'**effectif** et calcule le **diamètre** (moyen, noté QMD), la **surface terrière** (notée BA) qui correspondent à cet arbre moyen en fonction :

- des conditions stationnelles (SI)
- éventuellement de la gestion sylvicole
- ... et des exploitations
- des attaques de MPB

● **Contagion du MPB** : à chaque pas annuel,
niveau d'infestation d'un peuplement
= Σ (niveaux d'infestation des peuplements voisins) dans un rayon fixé

provisoire ...

● **Mortalité** : probabilité annuelle de mort = 0.8 (*i.e.* effectif réduit de 80%) si :
- niveau d'infestation > 0
- et diamètre (QMD) > 15 cm

provisoire ...

Études en cours : QUESTIONS , OBJECTIFS

- Comment les **propriétés du paysage** interagissent avec la **dynamique** des populations de **MPB** et à quelles échelles ?
- Comment ces interactions pourraient jouer dans le contexte de l'invasion des écosystèmes boréaux ?
- Évaluer la **sensibilité** actuelle (à l'ouest des Rocheuses) et **à venir** (tout le Canada) des paysages aux pullulations de MPB ?

Études en cours : MODÈLE

Modèle spatialement explicite à l'échelle du paysage à 3 composantes (sous-modèles) :

1

- sous-modèle de **dynamique forestière** :
... de type « peuplement »

2

- sous-modèle de **dynamique de population du MPB** :
« modèle d'éruption »

3

- sous-modèle spatial :
dispersion à +/- longue distance **du MPB**

Modèle de croissance :

fondé sur l'approche

« Stand Density Management Diagram »

$$H = f(\text{Site index}, \hat{\text{Âge}})$$

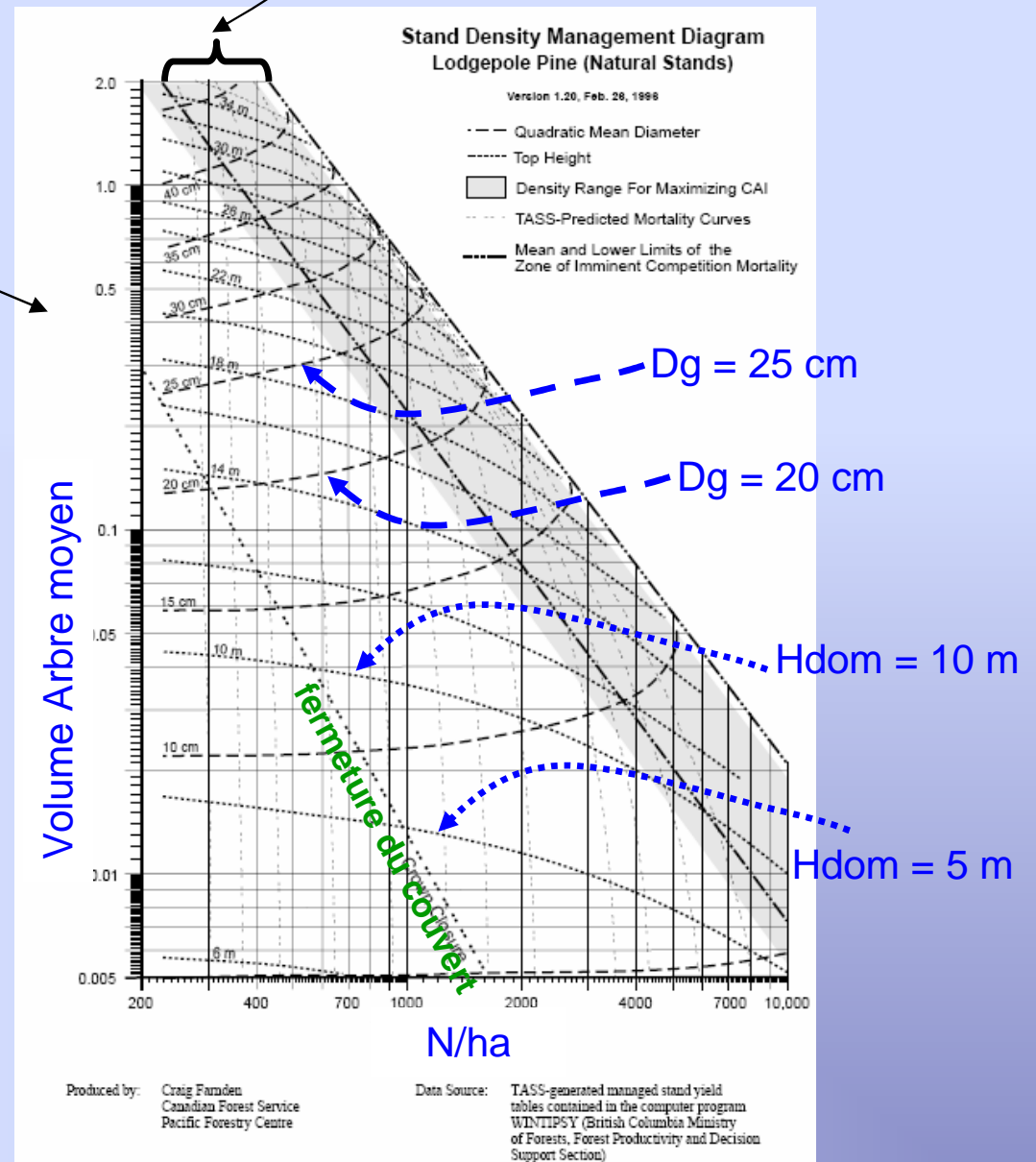
$$N = f(H_{\text{dom}}, N_{\text{ha initial}})$$

$$Q = f(H_{\text{dom}}, N_{\text{ha}})$$

D'abord pour *Pinus contorta* ...

Actuellement,
effort principalement
sur *Pinus banksiana*

zone de mortalité
par concurrence



Études en cours : **RÉSULTATS**

- Le sous-modèle de dynamique forestière permet de décrire la manière dont **une population de MPB de faible densité peut franchir les limitations induites par divers facteurs et « exploser »** dans un peuplement de *Pinus contorta* (ou de *P. banksiana*).
- La croissance d'une population de MPB est non-linéaire.
Quand la densité de MPB est basse, les défenses des arbres, la concurrence d'autres espèces et le manque d'habitat approprié limitent fortement son évolution.
Quand la densité de MPB est forte, les défenses d'un arbre individuel ne suffisent plus (1000 MPB tuent un arbre adulte).
- Effet de l'**âge** (indirect ?) :
 - peuplements plus susceptibles > 80 ans
 - après 160 ans, le phloème est devenu trop mince pour une attaque massive de MPB.
- Autres propriétés du peuplement affectent sa capacité à résister :
diamètre moyen (QMD = D_g , seuil à 15 cm), **densité** (N/ha)

Import/Export Shapefile

Fichiers d'**entrée** dans le module MPB :

- texte
- ou **shapefile**, issu d'un SIG

Sortie, fichiers d'export :

- texte (csv), intégrant l'étape en cours et toutes les précédentes
- ou **shapefile**, pour la seule étape en cours, lisible par un SIG

Shapefile = un format de fichier SIG, de couche « **vectorielle** ».

Initialement développé par ESRI (logiciels commerciaux), **devenu un standard** (échange entre logiciels SIG).

4 fichiers associés (**même nom, extensions différentes**) :

- **fichier.shp** : **géométrie des objets (pour MPB : polygones = peuplements)**

- **fichier.dbf** : **données attributaires** relatives à ces objets (pour MPB : *SI, N/ha, QMD, G/ha, MPBlevel ...*)

- **fichier.shx** : stocke l'index de la géométrie

- **fichier.prj** : information sur le système de coordonnées géographiques (ellipsoïde, projection, etc.)

(en exportation, les 3 fichiers .shp, .shx et .prj, inchangés, sont simplement copiés)

Appui sur la bibliothèque GeoTools en Java ... et sur Samuel D-K !

GeoTools The Open Source **Java GIS Toolkit**

GeoTools is an open source Java library that provides tools for geospatial data.

<http://geotools.org/>


```
public Collection createRecordSet (String fileName) {
```

```
    shapeRecords = new ArrayList ();
```

```
    FeatureSource fsource;
```

```
    try {
```

```
        URL shapeURL = new File(fileName).toURI().toURL();
```

```
        ShapefileDataStore datastore = new ShapefileDataStore(shapeURL);
```

```
        String name = datastore.getTypeNames()[0];
```

```
        fsource = datastore.getFeatureSource(name);
```

```
    } catch (Exception e) {
```

```
        //err("Error while reading shape.");
```

```
        return null;
```

```
    }
```

```
    FeatureCollection features;
```

```
    try {
```

```
        features = fsource.getFeatures();
```

```
    } catch (Exception e) {
```

```
        //err("Error while reading shape.");
```

```
        return null;
```

```
    }
```

```
    Iterator iterator = features.iterator();
```

```
    try {
```

```
        while( iterator.hasNext() ){
```

```
            SimpleFeatureImpl feature = (SimpleFeatureImpl) iterator.next();
```

```
            ShapefileCompartmentRecord sfaur  
                = new ShapefileCompartmentRecord ();
```

... puis on lit la valeur de chaque attribut :

```
            sfaur.Compld = (int) ((Double) feature.getAttribute("COMPID")).doubleValue();
```

```
            sfaur.CompName = (String) feature.getAttribute("COMPNAME");
```

```
            sfaur.SI = (Double) feature.getAttribute("SI");
```

```
            sfaur.topHeight = (Double) feature.getAttribute("TOPHEIGHT");
```

```
            sfaur.stemsPerHa = (int) ((Double) feature.getAttribute("STEMSPERHA")).doubleValue();
```

Constituer une collection d'enregistrements correspondant aux cellules-peuplements (les MpbComp ↔ PolygonalCell) ...

*... on charge la collection de « polygones SIG » d'après le nom du **fichier .shp** indiqué par l'utilisateur ...*

Itération sur ces « polygones SIG »

« polygone SIG » suivant ...

initialisation d'un nouveau « record » pour une cellule-peuplement (une MpbComp) correspondant à ce « polygone SIG »

nom de la colonne dans le .dbf

nécessaire
pour les calculs
de contagion/dispersion
et pour la visualisation
en carte dans Capsis

```
public Collection createRecordSet (String fileName) {
```

```
...
```

chargement collection de « polygones SIG »

```
Iterator iterator = features.iterator();
```

```
try {
```

```
while( iterator.hasNext() ) {
```

récupération des attributs

```
...
```

```
MultiPolygon geom = (MultiPolygon) feature.getDefaultGeometry();
```

```
GeometryCollection geomcoll = (GeometryCollection) geom;
```

```
Coordinate[] coords = geom.getCoordinates();
```

```
Collection vertices = new ArrayList ();
```

puis Récupération de la **Géométrie**
de chaque « cellule-peuplement »

on traduit la **géométrie**
d'un « polygone SIG »

en **liste de « vertex » (x, y)**

```
if(Double.isNaN(coords[0].z)) { // polygones en 2D
```

```
Vertex2d vertex2 = null;
```

```
for (int i = 0; i < geom.getNumPoints(); i++) {
```

```
vertex2 = new Vertex2d (Math.floor (coords[i].x), Math.floor (coords[i].y));
```

```
vertices.add(vertex2);
```

```
}
```

```
// add the first vertex again at the end
```

```
vertex2 = new Vertex2d (Math.floor (coords[0].x), Math.floor (coords[0].y));
```

```
vertices.add(vertex2);
```

```
} else { // polygones en 3D
```

```
... idem pour le cas où ce serait des vertex 3D (altitude z)
```

```
}
```

```
sfaur.vertices = vertices;
```

```
shapeRecords.add (sfaur);
```

ajout de la géométrie (liste de vertex)
à cet enregistrement « cellule-peuplement »

ajout dans la collection d'enregistrements de « cellules-peuplements »

```
}
```

```
finally {
```

```
features.close( iterator );
```

```
}
```

```
return shapeRecords;
```

```
}
```

à la construction du MpbPlot,
chaque enregistrement servira à construire une MpbComp

```
public void createRecordSet(GScene sc) throws Exception {
```

```
    MpbStand stand = (MpbStand) sc;
```

```
    MpbPlot plot = (MpbPlot) stand.getPlot ();
```

*D'abord, on lit la **collection** de « **polygones SIG** »
et on en fait une **copie***

```
////////// IMPORT CODE
```

```
// Load source shapefile and read its contents
```

```
// in order to catch its ShapefileDataStore and its schema (see readDataStore.getSchema() below)
```

```
URL shpFileURL = (new File(stand.getSourceName ())).toURI().toURL();
```

```
ShapefileDataStore readDataStore = new ShapefileDataStore(shpFileURL);
```

```
FeatureCollection readfeatcoll = readDataStore.getFeatureSource().getFeatures();
```

```
// Create a new collection of "readfeat"s ...
```

```
FeatureCollection<SimpleFeatureType, SimpleFeature> outfeatcoll = FeatureCollections.newCollection();
```

```
// ... in which the original features (from readfeatcoll) are copied (to outfeatcoll) one at a time
```

```
Iterator iterator = readfeatcoll.iterator();
```

```
try {
```

```
    while (iterator.hasNext()) {
```

```
        // Extract readfeat
```

```
        SimpleFeature readfeat = (SimpleFeature) iterator.next();
```

```
        outfeatcoll.add(readfeat);
```

```
    }
```

```
} finally {
```

```
    readfeatcoll.close(iterator);
```

```
}
```

On est prêt à faire l'exportation proprement dite ...

```
////////// EXPORT CODE
```

```
...
```


EXPORT CODE

1. DEFINING A NEW FEATURE TYPE :

// Build a new feature type from the previous (add 3 columns : QMD, BA, MPBLEVEL)

// i.e. the new feature type is derived from the old feature type :

// - existing attributes are kept (there value will be updated, eventually)

// - additional attributes are added (which value will be set further)

// in order to build the new feature type, we need a tool : a SimpleFeatureTypeBuilder

SimpleFeatureTypeBuilder typeBuilder = new SimpleFeatureTypeBuilder();

// returns the FeatureType for the file being read. N.B.: a ShapefileDataStore contains only one Type.

SimpleFeatureType oldFeatureType = readDataStore.getSchema();

// the name of the new readfeat type is the same as the name of the old readfeat type :

typeBuilder.setName(oldFeatureType.getName());

// ... same coordinate reference system also :

typeBuilder.setCRS(oldFeatureType.getCoordinateReferenceSystem());

// typeBuilder gets attribute descriptors from oldFeatureType

typeBuilder.setAttributes(oldFeatureType.getAttributeDescriptors()); // setAttributes directly sets the list of attributes
("setAttributeDescriptors" should be a better name ?)

// add new columns, giving (name, type)

typeBuilder.add("QMD", Double.class);

typeBuilder.add("BA", Double.class);

typeBuilder.add("MPBLEVEL", Integer.class);

// everything is now configured : now, build the new feature type

SimpleFeatureType outFeatureType = typeBuilder.buildFeatureType();

On définit la liste des attributs de chaque « polygone SIG » ...

... ceux qui existent déjà ...

... et ceux qu'on veut ajouter

**//// 2. GIVING A VALUE TO EACH ATTRIBUTE in each new feature
(according to type of each attribute in the new feature type) :**

On prépare les 4 fichiers ...

```
// Create the output shapefile
String tempExportFileName = CapsisSettings.getDir("tmp");
tempExportFileName += File.separator;
String tempExportFileNameWOextension = tempExportFileName + "temp";
String tempSHPFileName = tempExportFileNameWOextension + ".shp";
String tempSHXFileName = tempExportFileNameWOextension + ".shx";
String tempDBFFFileName = tempExportFileNameWOextension + ".dbf";
String tempPRJFileName = tempExportFileNameWOextension + ".prj";
```

```
tempSHPFile = new File(tempSHPFileName);
tempSHXFile = new File(tempSHXFileName);
tempDBFFFile = new File(tempDBFFFileName);
tempPRJFile = new File(tempPRJFileName);
```

*... en version temporaire
(la méthode **save** de MpbExportShapefile.java
permettra à l'utilisateur de leur donner un nom
définitif)*

```
URL outUrl = tempSHPFile.toURI().toURL();
ShapefileDataStore outDataStore = new ShapefileDataStore(outUrl);
```

```
// Set the FeatureType of this DataStore
outDataStore.createSchema(outFeatureType);
```

*Un peu de mécanique, dont **initialisation du
lecteur de collection de « polygones SIG » ...***

```
// Need another one tool : a feature writer
FeatureWriter outFeatureWriter = outDataStore.getFeatureWriter(outDataStore.getTypeNames()[0],
Transaction.AUTO_COMMIT);
```

// For each line (=each readfeat) in the outfeatcoll, set the value of each attributes

iterator = outfeatcoll.iterator();

try {

List<Object> att = null; // list of objects that will be "translated" to a feature

int count = 0;

while (iterator.hasNext()) {

// Extract readfeat

SimpleFeature readfeat = (SimpleFeature) iterator.next();

// Get the next, empty readfeat from the writer

SimpleFeature outfeat = (SimpleFeature) outFeatureWriter.next();

// first, add attributes that were already present in the read shapefile : ... they will be updated further, if necessary

att = readfeat.getAttributes(); // Nota Bene : getAttributes returns an Object []

int id = (int) ((Double) readfeat.getAttribute("COMPID")).doubleValue();

Iterator enucomp = plot.getCellsAtLevel1 ().iterator ();

MpbComp comp = null;

while (enucomp.hasNext ()) {

comp = (MpbComp) enucomp.next ();

if(comp.getCompId() == id) {

break;

}

// ADD VALUES for the NEW ATTRIBUTES

// ... in the same order as they were declared

att.add(0.1d * (int) (10d * comp.getDg ())); // QMD

att.add(comp.getGha ()); // BA

att.add(comp.getMpbLevel ()); // MPBLEVEL

outfeat.setAttributes (att); // the list of objects is "translated" to a feature

// UPDATE VALUE for some attributes

// COMPID, COMPNAME, SI are not changed

// Only TOPHEIGHT and STEMSPERHA

outfeat.setAttribute("TOPHEIGHT", comp.getHmax ());

outfeat.setAttribute("STEMSPERHA", comp.getNha ());

outFeatureWriter.write();

}

outFeatureWriter.close();

} finally {

readfeatcoll.close(iterator);

}

Itération sur la collection de « polygones SIG »

Itération sur les cellules du « plot »

pour trouver la « cellule-peuplement » (une MpbComp) correspondant au « polygone SIG »

On renseigne les 3 nouveaux attributs en prenant la valeur de ladite « cellule-peuplement » (une MpbComp)

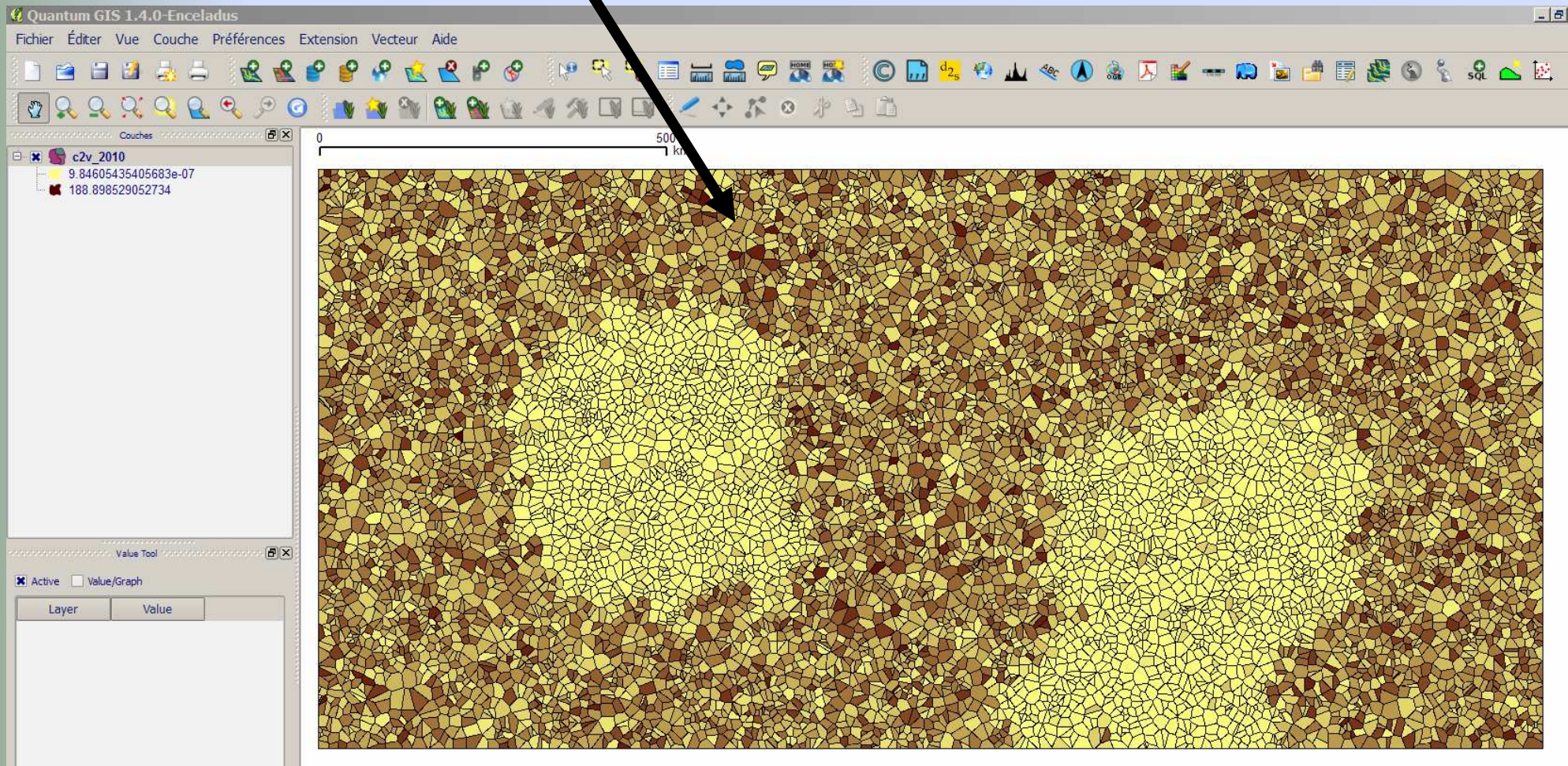
On met à jour certains attributs déjà présents

La boucle terminée, on ferme le « lecteur de collection de polygones SIG »

... le shapefile exporté peut être chargé dans un SIG :

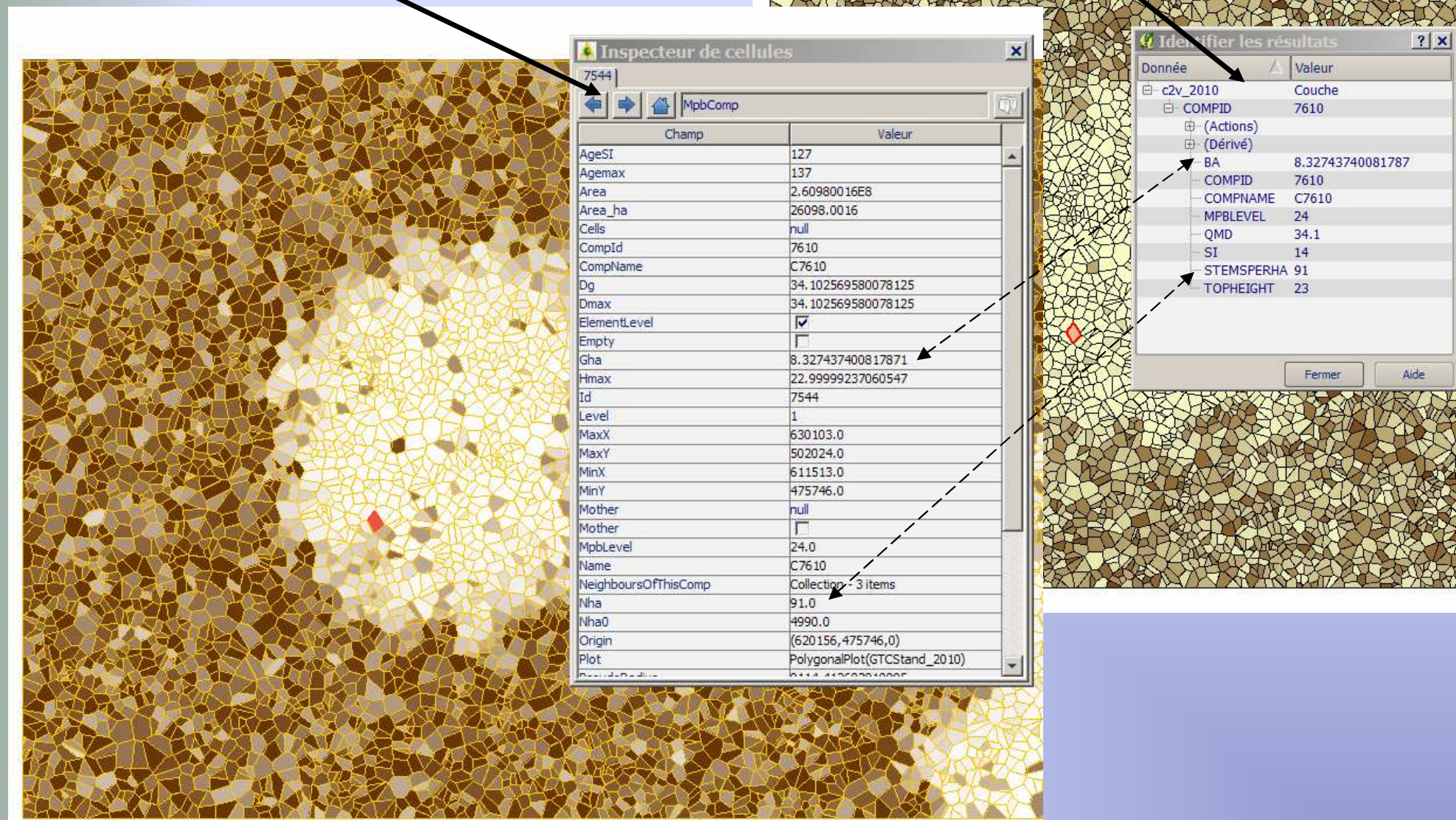
SVMpb dans Capsis4,
module Mpb :

Couche vectorielle dans QGIS / GRASS
(ou logiciels commerciaux) :



Attributs 2010 du peuplement 7610
lus dans Capsis4, module Mpb :

Attributs 2010 du peuplement 7610
lus dans QGIS / GRASS :



Attributs 2010 du peuplement 7610
lus directement dans le .dbf du shapefile exporté :

1	A	B	C	D	E	F	G	H	I
1	COMPID	COMPNAME	SI	TOPHEIGHT	STEMSPERHA	QMD	BA	MPBLEVEL	
7270	7606	C7606	18.0	29.0	2931	25.1		146.0	0
7271	7607	C7607	9.0	14.0	2674	18.1		69.0	0
7272	7608	C7608	19.0	29.0	0	32.2		0.0	203579404
7273	7609	C7609	23.0	36.0	813	38.0		92.6	0
7274	7610	C7610	14.0	23.0	91	34.1		8.3	24
7275	7611	C7611	17.0	28.0	0	32.9		0.0	100926361
7276	7612	C7612	15.0	25.0	3	37.9		0.3	67469
7277	7613	C7613	6.0	12.0	0	19.8		0.0	539111680

Microsoft Excel - Classeur1.xls															
Fichier Edition Affichage Insertion Format Outils Données Fenêtre ?															
Collage spécial... Filtre élaboré... 200%															
Arial 8 Remplacer...															
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
5	# Stand date : 2010														

Attributs **de 2000 à 2010** du peuplement 7610
lus dans l'**export au format .csv** depuis Capsis4, module Mpb:

#	Key	Date	CompId	CompNa	XCenter	YCenter	ZCenter	Aream2	SI	Nha0	Agemax	Hmax	Dmax	Nha	Gha	mpbLevel
17	COMP	2000	7610	C7610	621462	483762	0	2.61E+08	14	4990	127	20.0	29.1	499	33.2	0
18	COMP	2001	7610	C7610	621462	483762	0	2.61E+08	14	4990	128	20.3	29.6	494	34.0	0
19	COMP	2002	7610	C7610	621462	483762	0	2.61E+08	14	4990	129	20.6	30.1	489	34.8	0
20	COMP	2003	7610	C7610	621462	483762	0	2.61E+08	14	4990	130	20.9	30.6	484	35.6	0
21	COMP	2004	7610	C7610	621462	483762	0	2.61E+08	14	4990	131	21.2	31.1	479	36.4	0
22	COMP	2005	7610	C7610	621462	483762	0	2.61E+08	14	4990	132	21.5	31.6	474	37.2	0
23	COMP	2006	7610	C7610	621462	483762	0	2.61E+08	14	4990	133	21.8	32.1	469	38.0	0
24	COMP	2007	7610	C7610	621462	483762	0	2.61E+08	14	4990	134	22.1	32.6	465	38.8	0
25	COMP	2008	7610	C7610	621462	483762	0	2.61E+08	14	4990	135	22.4	33.1	460	39.6	0
26	COMP	2009	7610	C7610	621462	483762	0	2.61E+08	14	4990	136	22.7	33.6	455	40.5	0
27	COMP	2010	7610	C7610	621462	483762	0	2.61E+08	14	4990	137	23.0	34.1	91	8.3	24

GeoTools The Open Source Java GIS Toolkit

GeoTools is an open source Java library that provides tools for geospatial data.

<http://geotools.org/>

Téléchargement des éléments nécessaires :

c:\capsis4>**ant install_geotools**
(by Samuel Dufour, via [ivy:retrieve])

Fichiers .jar mis dans **capsis4/ext** :

Puis spécifier dans capsis4/**build.xml**

(utilisé par ant.bat)

que le module requiert « geotools » :

par ex. :

```
<exclude name="ventoux/**" unless="geotools"/>  
  <exclude name="mpb/**" unless="geotools"/>
```

etc. (idem pour d'autres lignes dans build.xml)

gt-main-2.6-M2.jar
gt-shapefile-2.6-M2.jar
gt-epsg-hsql-2.6-M2.jar
gt-api-2.6-M2.jar
jts-1.10.jar
jdom-1.0.jar
commons-beanutils-1.7.0.jar
gt-referencing-2.6-M2.jar
vecmath-1.3.2.jar
commons-pool-1.3.jar
gt-metadata-2.6-M2.jar
geoapi-2.3-M1.jar
geoapi-pending-2.3-M1.jar
jsr-275-1.0-beta-2.jar
commons-logging-1.1.1.jar
hsqldb-1.8.0.7.jar

