# Genetics 2.0

## Généralisation de la bibliothèque Genetics de Capsis4

F. de Coligny - 17.11.2004

# 1 Introduction

La bibliothèque Genetics de Capsis4 a été développée dans le cadre du projet "Couplage de modèles de Flux de Gènes et de modèles de Dynamique Forestière" mené en 2002 et 2003 en réponse à un appel d'offre du Bureau des Ressources Génétiques (coordination P. Dreyfus - INRA, Unité de Recherches Forestières Méditerranéennes).

L'objectif de ce projet était le suivant : "Dans le contexte de la gestion forestière courante, il s'agit par une approche interdisciplinaire, (1) de coupler des modèles génétiques et écologiques (production et dispersion de diaspores, démographie, croissance et compétition), (2) de simuler (plate-forme logicielle) divers modes de gestion, (3) de comparer (validation) à des données expérimentales, sur une sélection de modèles biologiques tempérés et tropicaux, (4) afin d'en tirer de premiers indicateurs de l'impact de la gestion sur les ressources génétiques forestières."

Le cahier des charges pour Genetics a été élaboré en commun par les membres du projet qui comportait notamment des généticiens, des modélisateurs et des informaticiens. Elle a été implémentée par Ingrid Seynave (Seynave & Pichot, 2004) avec un appui particulier de la part de Christian Pichot pour la génétique, de Philippe Dreyfus pour la modélisation et de François de Coligny pour l'informatique.

La bibliothèque propose des outils pour associer des génotypes aux structures de données arbres des modèles de croissance et de dynamique forestière implémentés dans Capsis (Dreyfus, 2004), elle a été utilisée dans les modules :

- VentouG : évolution des pineraies artificielles de l'arrière pays méditerranéen - INRA URFM, Avignon, P. Dreyfus, C. Pichot,
- Selva : forêt guyanaise, Angélique en agrégats dans une matrice d'espèces grises - Cirad forêt, Montpellier, S. Gourlet-Fleury, G. Cornu,
- Alisier : Alisier torminal disséminé dans des peuplements de Chêne sessile - ONF Conservatoire Génétique des Arbres Forestiers, Orléans, S. Oddou-Muratorio,
- Quercus : futaie régulière de Chêne sessile - INRA Laboratoire de Génétique et Amélioration des Arbres Forestiers, Bordeaux, S. Gerber ,
- Lubéron : forêt méditerranéenne, cédraie artificielle - INRA URFM, Avignon, F. Lefèvre, F. Courbet.

Le présent travail a pour objectif de réorganiser la bibliothèque pour permettre son utilisation avec des individus autres que des arbres et de clarifier le rôle de l'espèce des individus dans la bibliothèque (le codage de l'espèce était facultatif, il devient obligatoire). Ces modifications rendront notamment possible l'utilisation de la bibliothèque pour le module Bidasoa qui s'intéresse à la dynamique de populations de truites dans le bassin du fleuve du même nom et éventuellement pour d'autres besoins ultérieurs.

# 2 Quelles modifications

La principale modification a concerné l'introduction d'une interface *Genotypable* pour qualifier l'individu porteur de génotype (par exemple le GeneticTree proposé par la bibliothèque et qui reste en fonction) et d'une deuxième interface *GeneticScene* pour désigner l'objet qui contient les individus génotypables (par exemple AlsStand, l'objet peuplement dans le module Alisier). Ces introductions ont ensuite permis dans toutes les classes de la bibliothèque de remplacer les mentions respectives GeneticTree et GTCStand par Genotypable et GeneticScene, permettant donc d'appliquer toutes les méthodes de la bibliothèque aux objets d'un module qui ne gérerait pas des arbres (mais par exemple des poissons : BidFish et BidStand du module Bidasoa). Un certain nombre de méthodes statiques ou devenues statiques pour l'occasion ont également été déplacées depuis GeneticTree vers GeneticTools pour alléger le code du premier.

Un deuxième changement important est l'obligation maintenant pour un module utilisant la bibliothèque de gérer un objet espèce pour les individus, implémentant la nouvelle interface *GenoSpecies* et portant les informations GeneticMap, AlleleDiversity, AlleleEffect et Kinship. Cet objet n'était pas requis dans la première version de Genetics et le GeneticTree comportait des accesseurs sur ces objets pour permettre l'utilisation de Genetics par des modules mono-spécifiques ne gérant pas d'espèce. Dans les faits, les modules utilisateurs géraient tous l'espèce et il a été décidé pour clarifier le code de la bibliothèque d'exiger un objet pour porter les données relatives à l'espèce.

Un troisième travail a consisté à remplacer les interfaces SpeciesDefined et Species par des interfaces plus fonctionnelles *Speciable* et *Species* (même nom pour cette dernière) et utilisées principalement en dehors de la bibliothèque (qui n'a plus besoin de tester l'existence d'un objet espèce maintenant obligatoire).

Enfin, pour faciliter l'adaptation d'un nouveau module à Genetics, un objet *GenotypableImpl* a été construit, contenant toutes les implémentations requises par les méthodes de Genotypable dans l'individu (ou la cohorte : individu en représentant plusieurs avec un effectif - number). Il est ainsi possible de répondre au contrat Genotypable en procédant à des redirections vers les méthodes de GenotypableImpl, comme mis en place dans GeneticTree, dont le code se trouve conséquemment simplifié.

# 3 Nouvelle architecture de la bibliothèque

## *3.1 Schéma général*

Pour les modules forestiers de Capsis (fig. 1a), la classe peuplement doit implémenter l'interface GeneticScene et la classe arbre doit hériter de GeneticTree. La classe espèce doit de plus implémenter l'interface GenoSpecies. Les méthodes d'accès concernant AlleleEffect, KinShip, GeneticMap et AlleleDiversity ont migré de GeneticTree vers GenoSpecies : elles doivent être implémentées dans la classe espèce du module.

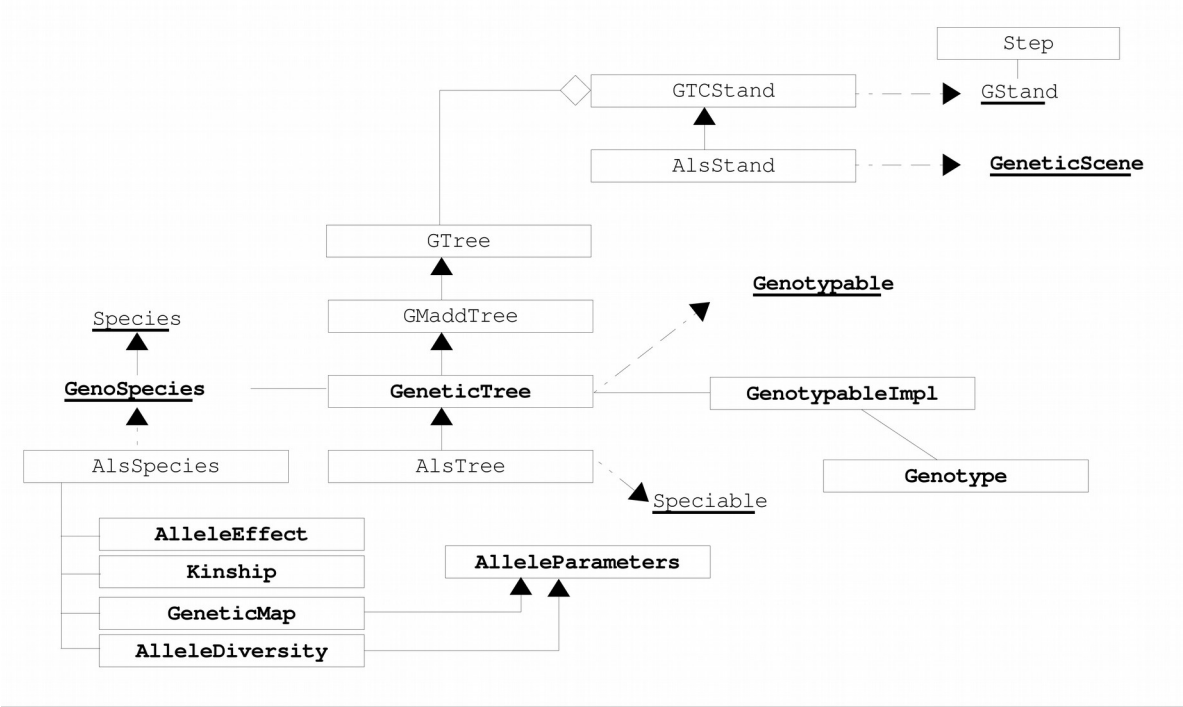


*Fig. 1a. Architecture de Genetics 2.0, cas des modules forestiers, exemple du module Alisier (diagramme de classes UML). En gras, les classes et interfaces de la bibliothèque Genetics*

Pour un module non forestier (fig. 1b), GeneticTree n'est pas utilisée. L'objet portant le génotype implémente directement l'interface Genotypable et implémente les redirections vers l'instance de GenotypableImpl (cf. §3.6)

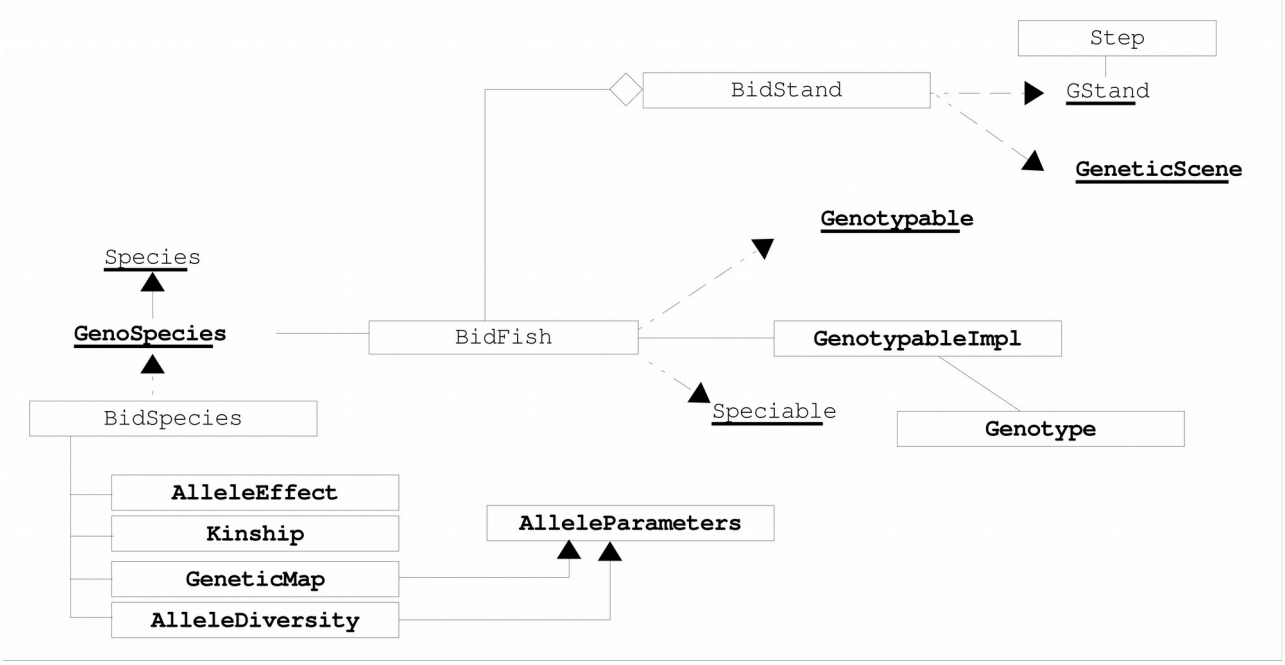


*Fig 1b. Architecture de Genetics 2.0, cas des modules non forestiers, exemple du module Bidasoa*

## 3.2 L'interface Genotypable

Elle décrit toutes les propriétés d'un objet portant un génotype dans Genetics (fig. 2). On y trouve des méthodes d'administration (identifiant, scène ou apparaît l'objet, espèce, référence du GenotypableImpl), d'autres concernant l'hérédité de l'objet (identifiants des père et mère, date de création) et des méthodes liées à la gestion des données génétiques (génotype, paramètres des allèles, valeur phénotypique, consanguinité, etc).

La classe GeneticTree, superclasse des classes arbres pour les modules forestiers utilisant Genetics, implémente l'interface Genotypable. Dans le code des méthodes de la bibliothèque, il est fait référence systématiquement à Genotypable au lieu de GeneticTree pour autoriser l'utilisation d'objets portant un génotype et n'héritant pas de GeneticTree.

```
public interface Genotypable  {
       public int getId ();
       public GeneticScene getScene ();
       public GenotypableImpl getImpl ();
       public GenoSpecies getGenoSpecies ();
       public boolean isGenotyped ();
       public boolean isMultiGenotyped ();
       public int getMId ();
       public int getPId ();
       public int getCreationDate ();
       public int getNumber ();
       public AlleleParameters getAlleleParameters ();
       public MultiGenotype getMultiGenotype ();
       public void setMultiGenotype (MultiGenotype mg);
       public Map getPhenoValue ();
       public void setPhenoValue (Map m);
       public void setIndividualGenotype (IndividualGenotype ig);
       public Genotype getGenotype ();
       public double getConsanguinity ();
       public void setConsanguinity (double f);
       public double getGlobalConsanguinity ();
       public void setGlobalConsanguinity (double f);
       public double getGeneticValue (String caractereName);
       public double getFixedEnvironmentalValue (String parameterName);
       public double getPhenoValue (String caractereName);
}
```

*Fig 2. L'interface Genotypable*

## 3.3 La classe GeneticTree

Dans GeneticTree, les données génétiques qui ne changent pas dans le temps pour un même arbre (identifiants des père et mère, génotype s'il est individuel, etc) ont migré de la classe interne *Immutable* qui a disparu vers la classe *GenotypableImpl*. En effet, les occurrences des arbres portant un même identifiant dans l'historique de simulation ont toutes une référence vers le même objet GenotypableImpl. En revanche, les données susceptibles de changer au cours de la simulation sont référencées dans GeneticTree (multiGenotype, phenoValue).

Les méthodes exigées dans GeneticTree par l'interface Genotypable sont toutes redirigées (fig. 3) vers des méthodes de même nom dans l'objet GenotypableImpl associé (la référence s'appelle *geeImpl*), ce qui permet d'éviter la duplication de leur code.

```
       public int getMId () {return geeImpl.getMId ();}
       public int getPId () {return geeImpl.getPId ();}
       public int getCreationDate () {return geeImpl.getCreationDate ();}
       ...
       public Genotype getGenotype () {
            return geeImpl.getGenotype (this);
       }
       public double getConsanguinity () {
            return geeImpl.getConsanguinity (this);
       }
       ...
```

*Fig 3. Redirections dans GeneticTree vers les implémentations de GenotypableImpl*

Certains accesseurs font exception (fig. 4) et sont implémentés directement dans GeneticTree parce qu'ils font référence à des variables d'instances déclarées dans cette classe. C'est le cas de l'accesseur getImpl () qui renvoie la référence de l'objet GenotypableImpl associé, de getNumber () qui renvoie l'effectif de l'objet : 1 si c'est un individu, un nombre si c'est une cohorte, getMultiGenotype (), méthode outil utilisée dans la bibliothèque quand on veut accéder directement

à l'instance d'un multiGenotype (la méthode normale d'accès à un Génotype est getGenotype (), qui renvoie une instance de IndividualGenotype ou MultiGenotype selon les cas), setMultiGenotype () et setIndividualGenotype (), ainsi que les accesseurs get/setPhenoValue ().

```
public GenotypableImpl getImpl () {return geeImpl;}
public abstract int getNumber ();
public MultiGenotype getMultiGenotype () {return multiGenotype;}
public void setMultiGenotype (MultiGenotype mg) {
     multiGenotype = mg;
     if (mg != null) {  // fc - 16.11.2004
          geeImpl.genotype = null;
     }
}
public void setIndividualGenotype (IndividualGenotype ig) {
     geeImpl.genotype = ig;
     if (ig != null) {  // fc - 16.11.2004
          multiGenotype = null;
     }
}
public Map getPhenoValue () {return phenoValue;}
public void setPhenoValue (Map m) {phenoValue = m;}
```

*Fig 4. Méthodes implémentées dans GeneticTree*

## 3.4 L'interface GeneticScene

Elle décrit toutes les méthodes requises par Genetics au niveau de l'objet qui contient la liste de Genotypable, qui était jusqu'à présent forcément une instance de GTCStand. A présent, toute classe implémentant GeneticScene (fig. 5) peut contenir les objets Genotypable, par exemple la classe BidStand du module Bidasoa qui contient des listes de poissons.

```
public interface GeneticScene  {
     public Genotypable getGenotypable (int id);
     public Collection getGenotypables ();
     public Step getStep ();
     public int getDate ();
     public boolean isInitialScene ();
}
```

*Fig 5. L'interface GeneticScene*

On reconnaît certaines méthodes des instances de GStand (qui peuvent être associées à un Step de l'historique) : getStep () et getDate (). Les méthodes getGenotypable (id) et getGenotypables () généralisent les méthodes getTree (id) et getTrees (), alors que la méthode isInitialScene () reprend isInitialStand () qui signifie qu'une scène est accrochée sous l'étape racine (fig. 6). Ces nouvelles méthodes ne font aucune référence explicite à des arbres ou des peuplements d'arbres. Leurs implémentations dans les classes peuplements des modules forestiers peuvent rediriger vers les implémentations existantes pour les arbres.

Il est important de noter que la méthode getGenotypables () peut renvoyer une collection contenant des objets instances de Genotypable et d'autres non. La bibliothèque sait gérer de tels mélanges qui ont été prévus dans le cahier des charges.

```
        public Genotypable getGenotypable (int id) {
             return (Genotypable) getTree (id);     // for genetics2_0
        }
        public Collection getGenotypables () {return getTrees ();}
        public boolean isInitialScene () {return isInitialStand ();}
        ...
```

*Fig 6. Implémentation des méthodes génériques de GeneticScene. Exemple : getGenotypable (int id) dans AlsStand est redirigée vers la méthode getTree (int id)*

## 3.5 L'interface GenoSpecies

Elle doit être implémentée par l'objet espèce du module considéré (fig. 7). L'objet en question devra fournir des accesseurs sur les structures de données génétiques relatives à l'espèce : effet des allèles, apparentement, carte génétique et diversité allélique.

L'interface GenoSpecies hérite de l'interface Species du package capsis.util, qui exige des méthodes pour récupérer un code (valeur entière) et un nom pour l'espèce.

```
public interface GenoSpecies extends Species {
        public int getValue (); // each species must have a different value
        public String getName ();    // Species interface
        public GeneticMap getGeneticMap ();
        public void setGeneticMap (GeneticMap gm);
        public AlleleDiversity getAlleleDiversity ();
        public void setAlleleDiversity (AlleleDiversity ad);
        public AlleleEffect getAlleleEffect ();
        public void setAlleleEffect (AlleleEffect ae);
        public Kinship getKinship ();
        public void setKinship (Kinship a);
}
```

*Fig 7. L'interface GenoSpecies*

## 3.6 La classe GenotypableImpl

Elle fournit des implémentations pour les méthodes exigées par Genotypable. Les classes implémentant Genotypable peuvent rediriger leurs méthodes vers les méthodes proposées, qui sont généralement des méthodes déplacées depuis la classe GeneticTree de Genetics 1.0.

### 3.6.1 Variables d'instances

GenotypableImpl contient les variables d'instance (fig. 8) du Genotypable qui ne changent pas au cours de la simulation. Ces variables étaient précédemment dans une classe interne nommée Immutable de GeneticTree qui a maintenant disparu. Les différentes instances d'un même Genotypable au cours de la simulation (par exemple les instances de l'arbre 12 en 2000, 2001,... 2010) ont tous une référence vers la même instance de GenotypableImpl qui contient leur identifiant, les identifiants de ses père et mère, sa date de création, etc.

```
public class GenotypableImpl {

	public Genotype genotype;
	public int mId;
	public int pId;
	public int creationDate;
	public double consanguinity;
	public double globalConsanguinity;
	public Map genoValue;
	public Map fixedEnvironmentalValue;
	...
```

*Fig 8. Les variables d'instance de GenotypableImpl*

## 3.6.2 Construction et méthodes

Les individus (ou cohortes) gérés dans les modules Capsis comportent généralement deux constructeurs bien particuliers (fig. 9 et 10) : un pour la création de la première instance d'un individu dans la simulation (par exemple à la lecture d'un fichier d'inventaire en début de simulation ou par régénération) et un autre pour la création d'une autre instance du même individu à une étape de simulation donnée (par exemple un arbre qui a grandi un an après). Ces deux constructeurs doivent invoquer des méthodes particulières de GenotypableImpl comme indiqué figures 9 et 10.

```
	/**  Constructor for new logical GeneticTree.
	*/
	public GeneticTree (    int           id,
	                        GStand        stand,
	                        int           age,
	                        double        height,
	                        double        dbh,
	                        boolean       marked,
	                        double        x,
	                        double        y,
	                        double        z,
	                        Genotype  genotype,
	                        int           mId,
	                        int           pId,
	                        int           creationDate
	            ) throws Exception {

	     super (id, stand, age, height, dbh, marked, x, y, z);

	     geeImpl = new GenotypableImpl ();
	     geeImpl.init (this, genotype, mId, pId, creationDate,
	               -1, -1, null, null);
	}
```

*Fig 9. Création de l'instance de GenotypableImpl à la construction de la première instance d'un GeneticTree*

```
        /**  Constructor for new instance of existing logical GeneticTree.
        */
        public GeneticTree (    GeneticTree   modelTree,
                                GStand        stand,
                                int           age,
                                double        height,
                                double        dbh,
                                boolean       marked) {
             super ((GMaddTree) modelTree, stand, age, height, dbh, marked);

             geeImpl = modelTree.getImpl ();
             geeImpl.update (this);
        }
```

*Fig 10. Liaison de l'instance de GenotypableImpl existante à une nouvelle instance de GeneticTree créée pendant la simulation à une date ultérieure*

Les autres méthodes de GenotypableImpl ont été déplacées depuis GeneticTree et pour certaines, reçoivent un paramètre supplémentaire : la référence au Genotypable qui invoque la méthode et qui peut être nécessaire au traitement. Dans la plupart des cas, les références au Génotypable ont été nommées gee pour ressembler aux anciennes références à tree. Il est rappelé que les mentions à tree.set/getGeneticMap () (ainsi que AlleleEffect, AlleleDiversity et Kinship) ont été remplacées par des mentions gee.getGenoSpecies ().set/getGeneticMap () (et de manière analogue pour les autres propriétés) pour se conformer à la nouvelle architecture autour de l'espèce décrite figure 1a.

Un accesseur getGeneticsVersion () a été ajouté dans le Genotypable pour renvoyer la version de la bibliothèque. Son implémentation se trouve également dans GenotypableImpl, elle renvoie actuellement "2.0".

# 4 Références

Seynave I., Pichot C., 2004, Bibliothèque Genetics de Capsis, Documentation

Dreyfus P., 2004, Couplage de modèles de flux de gènes et de modèles de dynamique forestière, Rapport Final.