

Chapitre 3 : Les logiciels et algorithmique

Logiciel et Algorithmique
Dr. TORCHE YACINE

Table des matières



Objectifs	3
I - C'est quoi un logiciel ?	4
1. Définition	4
2. Les logiciels système	5
3. Les logiciels d'applications	6
3.1. Les utilitaires	6
3.2. Les outils de développement d'applications (Langages de programmation)	7
3.3. Les systèmes de gestion de base de données (SGBD)	7
3.4. Les logiciels applicatifs	8
4. Types de logiciels	9
II - C'est quoi l'algorithmique, l'algorithme ? قيم واوخل	10
1. Définitions	10
1.1. Algorithme	10
1.2. Algorithmique	11
2. Algorithmique et programmation	12
3. Quelle est la différence entre un logiciel et un algorithme	13
3.1. Principe	13
III - L'algorithme au service de la biologie.	15
1. La biologie moléculaire !	15
2. Les algorithmes évolutifs	15
3. Utilisation	16
4. Le logiciel PhyML	16
5. Intérêt	18

Objectifs

A la fin de ce chapitre vous saurez :

1. Ce que c'est qu'un logiciel.
2. Ce que c'est qu'un algorithme.
3. Faire la différence entre un logiciel et un algorithme.
4. et utiliser un algorithme en biologie.

C'est quoi un logiciel ?

Définition	4
Les logiciels système	5
Les logiciels d'applications	6
Types de logiciels	9

1. Définition

Définition

Un logiciel peut être représenté par :

- un ensemble d'instructions ordonnées
- Un objectif précis
- Un langage propre au monde de l'informatique

Complément

Les langages les plus utilisés sont proches du langage courant et sont traduits de façon à ce que chaque instruction soit transmise dans une forme intelligible par la machine.

Un logiciel est un programme exécutable (.exe) qui utilise les ressources matérielles du système informatique, (voir les extensions, chapitre I).

Remarque

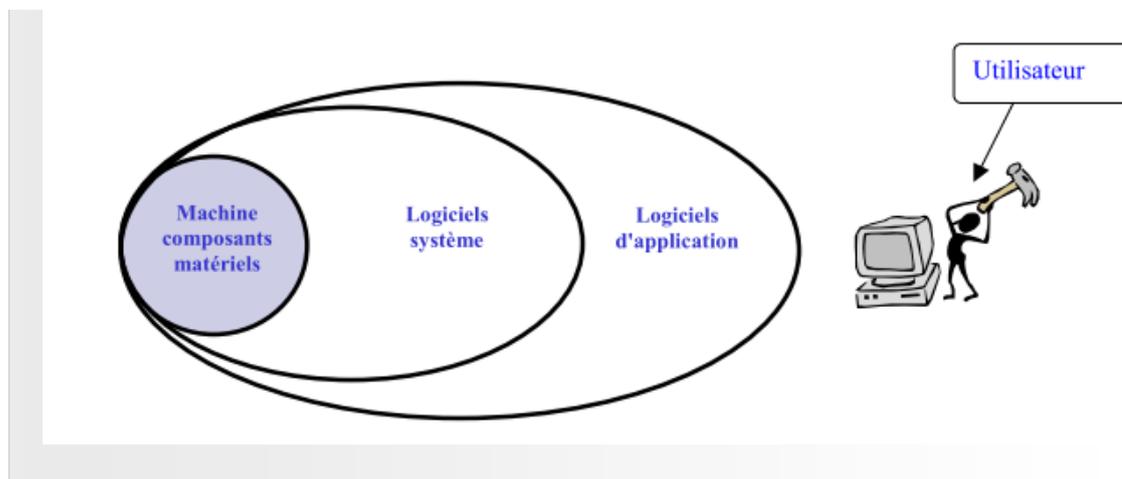
Bien souvent, on oppose le logiciel (*software*) qui reste abstrait au matériel (*hardware*) qui est bien au contraire concret ou visible. Les deux sont indissociables et inter-dépendants.

Fondamental

Les programmes sont des instructions écrites dans des fichiers que l'ordinateur est capable d'exécuter.

On en distingue 2 sortes principales :

- Les systèmes d'exploitations (OS , Operating System) ou logiciel système.
- Les logiciels (applications, programmes exécutables, vous pouvez trouver plusieurs appellations pour le même mot)



2. Les logiciels système

Définition

Les *logiciels système* sont des types de programmes informatiques conçus pour faire *fonctionner* un ordinateur et des applications. Si l'on visualise le système informatique comme un modèle à plusieurs couches, le logiciel système est l'interface entre le matériel et les applications utilisatrices.

Fondamental

Les principales fonctions des *logiciels système* sont :

- Se placent entre le matériel et les autres logiciels.
- Permettent aux logiciels d'application une indépendance vis à vis du matériel
- Réalisent les fonctions de base du système (impression, visualisation, écriture mémoire ...)
- Traitent les demandes d'accès au matériel des logiciels d'application

Remarque

Sans eux, chaque *logiciel application* serait écrit spécifiquement pour chaque configuration matérielle.



Exemple

- les systèmes d'exploitation S.E ou O.S (Operating System), largement discutés dans le

 **Remarque**

Un certain nombre de logiciels utilitaires sont généralement livrés avec les systèmes d'exploitation. Cependant, ces programmes ne sont pas considérés comme faisant partie du système d'exploitation. Bien que les logiciels utilitaires livrés avec les systèmes d'exploitation soient de plus en plus complets et sophistiqués, les utilisateurs installent souvent des logiciels utilitaires tiers en remplacement ou en complément de ceux fournis avec le système d'exploitation, comme *les antivirus* et les utilitaires de compression de fichiers *WinRar*.

 **Exemple**

- Les outils d'administration de systèmes partage et gestion des ressources
 - Ex : Norton Navigator, Partition magic...
- Les utilitaires de sauvegarde et de restauration
 - Ex : Drive Image, Active Backup...
- Les antivirus ...
- Les logiciels de compactage ou de compression de fichiers
 - Ex : Winzip, 7zip...
- Etc...

3.2. Les outils de développement d'applications (Langages de programmation)

 **Définition**

Souvent considérés comme des utilitaires, ils permettent la production d'applications. Ils sont classés selon leur degré d'évolution de 1ère génération à 4ème génération.

Un langage de programmation est une notation conventionnelle destinée à formuler des algorithmes et produire des programmes informatiques qui les appliquent. D'une manière similaire à une langue naturelle, un langage de programmation est composé d'un alphabet, d'un vocabulaire, de règles de grammaire et de significations.

 **Exemple**

Ex : Delphi, Windev, C++, Java, Power Builder...

 **Complément**

Vous pouvez approfondir vos connaissances sur la programmation en consultant le fichier PDF (cf. res. pdf) et le site internet *Apprenez à programmer en C !*

3.3. Les systèmes de gestion de base de données (SGBD)

 **Définition**

En anglais *DBMS* pour *database management system*.

Ils permettent l'accès et la gestion des bases de données. Certains peuvent permettre l'écriture d'application.

Un SGBD permet de décrire, manipuler et interroger les données d'une Base de Données. Il est chargé de tous les problèmes liés aux accès concurrents, à la sauvegarde et la restauration des données. Il doit de plus veiller au contrôle, à l'intégrité et la sécurité des données.

Complément

Les systèmes de gestion de base de données sont des logiciels universels, indépendants de l'usage qui est fait des bases de données². Ils sont utilisés pour de nombreuses applications informatiques, notamment les guichets automatiques bancaires, les logiciels de réservation, les bibliothèques numériques, les logiciels d'inventaire, les progiciels de gestion intégrés ou la plupart des blogs et sites web.

Exemple

Il existe de nombreux systèmes de gestion de base de données. En 2008, *Oracle* détenait près de la moitié du marché des SGBD avec *MySQL* et *Oracle Database*. Vient ensuite *IBM* avec près de 20 %, laissant peu de place pour les autres acteurs.

Remarque

Les SGBD sont souvent utilisés par d'autres logiciels ainsi que les administrateurs ou les développeurs. Ils peuvent être sous forme de composant logiciel, de serveur, de logiciel applicatif ou d'environnement de programmation.

3.4. Les logiciels applicatifs

Définition

En informatique, le terme application désigne à la fois l'activité d'un utilisateur susceptible d'être automatisée et le logiciel qui automatise cette activité (logiciel applicatif).

Selon le grand dictionnaire terminologique, Un logiciel applicatif (ou logiciel d'application) est un ensemble de programmes informatiques qui servent à aider un utilisateur à faire un certain travail. Les termes logiciel applicatif, ou application, ou applicatif (utilisé comme substantif dans le langage courant) peuvent également être employés.

Applications qui par définition sont automatisées par informatique

- La bureautique : création, expédition et réception de documents écrits. En font partie la publication assistée par ordinateur (PAO) et le traitement de texte.
- La conception assistée par ordinateur (CAO) : calcul, dessin, simulation de produits avant leur réalisation. En font partie le dessin assisté par ordinateur (DAO) et la synthèse d'image 3D.
- La gestion de données : collecte, suivi, organisation et contrôle d'informations.
- L'émulation : imitation d'un logiciel ou d'un matériel informatique à l'aide d'un autre logiciel.

- Le groupware : travail de groupe dans un réseau informatique.
- Le courrier électronique et la voix sur réseau IP.
- Le jeu vidéo.
- Le calcul numérique : calculs mathématiques par informatique.

Applications qui peuvent être automatisées par informatique

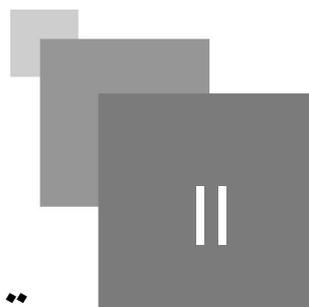
- La visioconférence, la photographie en particulier la photographie numérique, la vidéo et la création de diaporamas.
- La comptabilité, la billetterie et les statistiques.

4. Types de logiciels

Quelque soit les logiciels, on peut également les classer en 4 grands types :

- Les logiciels commerciaux vendus dans le commerce
- Les Shareware mis à disposition gratuitement pour essais ou tests et qui doivent être payés pour une utilisation régulière.
- Les Freeware mis à disposition gratuitement et qui peuvent être utilisés librement sans aucune contribution
- Les logiciels libres qui sont fournis normalement avec leur code source et qui peuvent être modifiés librement. (pour certains, la licence GPL définit les conditions)

C'est quoi l'algorithmique, l'algorithme ? ةي مزراوخل ا



Définitions	10
Algorithmique et programmation	12
Quelle est la différence entre un logiciel et un algorithme	13

1. Définitions

Algorithme	10
Algorithmique	11

1.1. Algorithme

Définition

Un algorithme est une suite ordonnée d'instructions qui indique la démarche à suivre pour résoudre une série de problèmes équivalents.

Le mot algorithme est dérivé du nom du mathématicien *Al Khwarizmi* qui a vécu au 9ème siècle, et qui était membre de l'académie des sciences à Bagdad.

Exemple : Trouver son chemin !

Extrait d'un dialogue entre un touriste égaré et un autochtone.

- Pourriez-vous m'indiquer le chemin de la gare, s'il vous plaît ?
- Oui bien sur : vous allez tout droit jusqu'au prochain carrefour, vous prenez à gauche au carrefour et ensuite la troisième à droite, et vous verrez la gare juste en face de vous.
- Merci.

Dans ce dialogue, la réponse de l'autochtone est la description d'une suite ordonnée d'instructions (allez tout droit, prenez à gauche, prenez la troisième à droite) qui manipulent des données (carrefour, rues) pour réaliser la tâche désirée (aller à la gare). Ici encore, chacun a déjà été confronté à ce genre de situation et donc, consciemment ou non, a déjà construit un algorithme dans sa tête (ie. définir la suite d'instructions pour réaliser une tâche). Mais quand on définit un algorithme, celui-ci ne doit contenir que des instructions compréhensibles par celui qui devra l'exécuter (des humains dans les 2 exemples précédents).

Fondamental : A quoi sert un algorithme ?

- À décrire les étapes de résolution d'un problème :
 - de façon structurée et compacte.
 - à partir d'opérations de base.
 - indépendamment d'un langage de programmation.
- "étapes" aussi appelées "pas de l'algorithme"
- Les *données* du problème en *entrée*.
- Le *résultat* de sa résolution en *sortie*.

1.2. Algorithmique

Définition

L'algorithmique est la science des algorithmes.

L'algorithmique s'intéresse à l'art de construire des algorithmes ainsi qu'à caractériser leur validité, leur robustesse, leur réutilisabilité, leur complexité ou leur efficacité.

Définition : Validité d'un algorithme

La validité d'un algorithme est son aptitude à réaliser exactement la tâche pour laquelle il a été conçu.

Si l'on reprend l'exemple (trouver son chemin) de l'algorithme de recherche du chemin de la gare, l'étude de sa validité consiste à s'assurer qu'on arrive effectivement à la gare en exécutant scrupuleusement les instructions dans l'ordre annoncé.

Définition : Robustesse d'un algorithme

La robustesse d'un algorithme est son aptitude à se protéger de conditions anormales d'utilisation.

Dans l'exemple, la question de la robustesse de l'algorithme se pose par exemple si le chemin proposé a été pensé pour un piéton, alors que le « touriste égaré » est en voiture et que la « troisième à droite » est en sens interdit.

Définition : La ré-utilisabilité d'un algorithme

La ré-utilisabilité d'un algorithme est son aptitude à être réutilisé pour résoudre des tâches équivalentes à celle pour laquelle il a été conçu.

L'algorithme de recherche du chemin de la gare est-il réutilisable tel quel pour se rendre à la mairie ? A priori non, sauf si la mairie est juste à côté de la gare.

Définition : La complexité d'un algorithme

La complexité d'un algorithme est le nombre d'instructions élémentaires à exécuter pour réaliser la tâche pour laquelle il a été conçu.

Si le « touriste égaré » est un piéton, la complexité de l'algorithme de recherche de chemin peut se

compter en nombre de pas pour arriver à la gare.

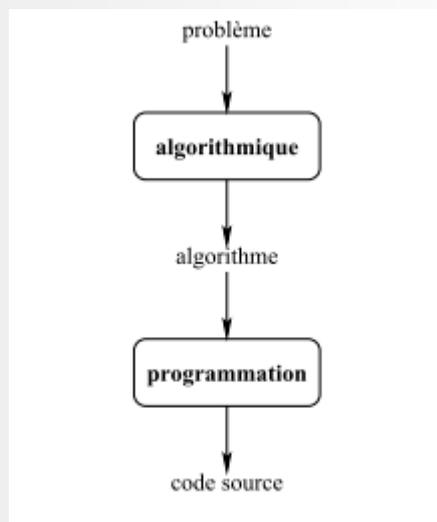
Définition : L'efficacité d'un algorithme

L'efficacité d'un algorithme est son aptitude à utiliser de manière optimale les ressources du matériel qui l'exécute.

N'existerait-il pas un raccourci piétonnier pour arriver plus vite à la gare ?

Fondamental

L'algorithmique permet ainsi de passer d'un problème à résoudre à un algorithme qui décrit la démarche de résolution du problème. La programmation a alors pour rôle de traduire cet algorithme dans un langage « compréhensible » par l'ordinateur afin qu'il puisse exécuter l'algorithme automatiquement.



Du problème au code source

2. Algorithmique et programmation

Étapes de conception d'un programme informatique

- 1 – identifier le problème : quelle(s) donnée(s), quel(s) résultat(s) ?
- 2 – organiser les actions : écrire l'algorithme (pseudo-code, organigramme)
 - réfléchir aux informations à manipuler
 - analyser le problème et le décomposer éventuellement en sous-problèmes
 - rendre l'algorithme compréhensible et efficace
 - penser à l'utilisateur
- 3 – Traduire cet algorithme en langage de programmation
- 4 – Compiler le programme pour qu'il puisse être exécutable



Complément

Vous pouvez vous approfondir sur les algorithmes en consultant *ces vidéos* en français, et en arabe (*vidéos en arabe*)

3. Quelle est la différence entre un logiciel et un algorithme

Principe

13

3.1. Principe



Fondamental

Le différence entre un algorithme et un programme est souvent une question de niveau de détail. Un algorithme est souvent exprimé avec une notation indépendante de tout langage de programmation alors qu'un programme est écrit dans un langage de programmation particulier.



Fondamental

Donc un algorithme est *un concept*, tandis qu'un programme est *l'implémentation* de ce concept. Ainsi, un programme est entièrement dépendant de la machine qui va l'exécuter, contrairement à un algorithme.



Complément

Une autre différence entre algorithme et programme est que l'exécution d'un algorithme doit toujours *se terminer avec un résultat*, alors que celle d'un programme peut conduire à une boucle infinie (ne jamais s'arrêter).



Remarque

Un algorithme est donc une méthode pour résoudre un problème particulier dont on est sûr qu'elle trouve toujours une réponse en un temps d'exécution fini.



Exemple

Lisez attentivement !

Exemple : Pour déterminer si un entier n est *premier* (à savoir qu'il ne contient pas de facteur autre que 1 et n), l'algorithme suivant peut être utilisé :

Pour chaque entier i , $2 \leq i < n$, vérifier si i est un facteur de n (en divisant n par i puis en vérifiant si le résultat est entier). Si c'est le cas, arrêter avec la réponse "non". Si aucune valeur i n'est facteur de n , alors arrêter avec la réponse "oui".

Il y a dans cette définition (informelle) d'algorithme un problème majeur. Prenons l'exemple des nombres premiers. Pourquoi peut-on supposer que l'on sait comment diviser n par i et vérifier que le résultat est un entier ?

Ne faudrait-il pas avoir un algorithme pour ça aussi ?

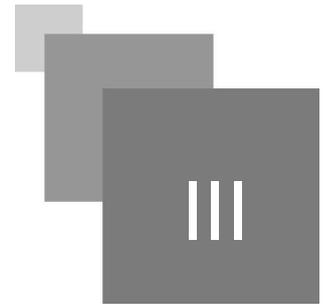
Et si jamais on trouve un algorithme pour cette division, ne faudrait-il pas trouver un algorithme pour chaque étape de celle-ci ? Et ainsi de suite. Ou, au contraire, pourquoi ne peut-on pas simplement supposer que "vérifier si un entier est premier" est une opération élémentaire ?

Pourquoi est-il nécessaire de trouver un algorithme pour cette opération ?

En d'autres termes : comment savoir si une opération est *élémentaire* (et ne nécessite par conséquent aucun algorithme) ?

La réponse est qu'une opération est élémentaire lorsque un ordinateur peut l'exécuter *très rapidement*, en réalité en un nombre relativement faible de cycles d'horloge.

L'algorithme au service de la biologie.



La biologie moléculaire !	15
Les algorithmes évolutifs	15
Utilisation	16
Le logiciel PhyML	16
Intérêt	18

1. La biologie moléculaire !

Définition

Discipline consacrée à l'étude des molécules porteuses du message héréditaire (ADN, ARN), de leur structure, synthèse, altérations (mutations).

Définition

Le séquençage de l'ADN consiste à déterminer l'ordre d'enchaînement des nucléotides (T,A,C et G) pour un fragment d'ADN donné.

Problématique

Pour comprendre les liens de parenté et l'évolution des espèces vivantes, les chercheurs peuvent comparer leurs génomes, stockés sous la forme de millions de séquences d'ADN dans des banques de données. Une tâche parfois insurmontable, en effet on peut avoir des centaines voir des milliers de séquences d'ADN et de protéines, et c'est extrêmement difficile voir impossible d'avoir des informations de nos séquences, c'est là qu'intervient les algorithmes et les logiciels.

2. Les algorithmes évolutifs

Définition

Les algorithmes évolutionnistes ou algorithmes évolutionnaires (evolutionary algorithms en anglais), sont une famille d'algorithmes dont le principe s'inspire de la théorie de l'évolution pour résoudre des problèmes divers. Ce sont donc des méthodes de calcul bioinspirées. L'idée est de faire évoluer un ensemble de solutions à un problème donné, dans l'optique de trouver les meilleurs résultats. Ce sont des algorithmes dits stochastiques, car ils utilisent itérativement des processus aléatoires.

La grande majorité de ces méthodes sont utilisées pour résoudre des problèmes d'optimisation, elles



sont en cela des métaheuristiques, bien que le cadre général ne soit pas nécessairement dédié aux algorithmes d'optimisation au sens strict. On les classe également parmi les méthodes d'intelligence computationnelle.

Conseil

Vous pouvez approfondir vos connaissances sur les algorithmes génétiques, évolutionnistes en consultant *ce site 1* et *2* (en anglais, vous pouvez voir la version traduite en arabe dans la rubrique langue de Wikipédia).

3. Utilisation

Vous êtes biologiste et vous voulez savoir quel est le degré de parenté et l'histoire évolutive entre les différentes espèces végétaux que vous êtes en train d'étudier.

Méthode

La méthode la plus fiable, celle qui ne vous induira pas en erreur par des ressemblances morphologiques superficielles, c'est *la phylogénie moléculaire* : comparer le génome de vos plantes.

Pour ce faire, vous disposez désormais de banques de données regroupant des millions de séquences d'ADN correspondant à des dizaines de milliers d'espèces vivantes. Mais, sans outils adaptés pour les comparer, elles resteront une suite muette de A, T, C et G.

Complément

C'est là qu'interviennent Olivier Gascuel et Stéphane Guindon, chercheurs au Laboratoire d'informatique, de robotique et de microélectronique de Montpellier (Lirmm).

Ces chercheurs ont mis au point en 2003 un puissant algorithme. Il permet d'estimer les relations évolutives liant un ensemble d'organismes à partir de la comparaison de leur ADN ou bien des acides aminés de leurs protéines.

Fondamental

On comprend pourquoi : cet algorithme permet de réduire drastiquement les temps de calcul par rapport aux méthodes précédentes. De plus, il peut être utilisé avec des données bien plus complexes et longues. Personne n'osait aborder des problèmes qui mettaient en jeu 100, 200 ou 500 espèces et des séquences longues de plusieurs milliers, voire dizaines de milliers de lettres. Après des jours et des jours de calcul on devait tout arrêter sans avoir obtenu de résultat. Leur algorithme permet de traiter ces données. »

4. Le logiciel PhyML

Fondamental

Sur la base de cet algorithme, les chercheurs ont développé un logiciel baptisé *PhyML*. Il est en accès

libre sur Internet, comme c'est l'usage dans le domaine.

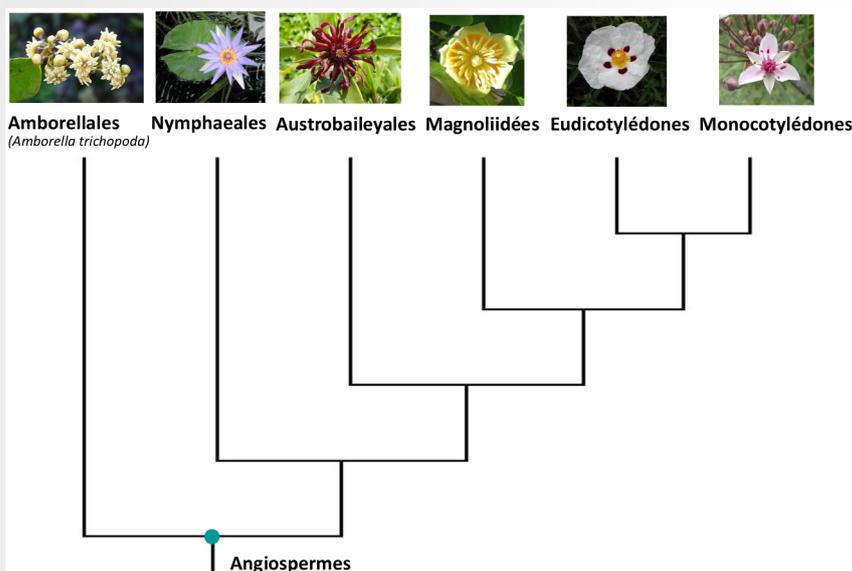
Méthode

Les chercheurs du monde entier profitent de sa simplicité d'utilisation : ils n'ont plus qu'à entrer leurs données et à attendre que les huit processeurs de l'ordinateur du Lirmm s'en occupent.

Ils reçoivent les résultats par e-mail quelques heures ou minutes plus tard. « Le serveur tourne à plein régime : il y a trois ou quatre utilisateurs en permanence.

Fondamental

- Le principe de base de l'algorithme est celui du maximum de vraisemblance, concept majeur en statistique. Il consiste ici à définir un modèle hypothétique qui décrive à la fois le degré de parenté entre les espèces étudiées et les mutations qui ont pu se produire à partir de la séquence originale postulée.
- Ensuite, il calcule la probabilité que les données correspondent à l'hypothèse de départ. Le processus est réitéré, modèle après modèle, jusqu'à ce que cette probabilité soit maximale. Le modèle correspondant constitue la réponse de l'algorithme.
- Ce tâtonnement demandait des temps de calculs colossaux jusqu'à l'arrivée de PhyML, qui est en fait une approximation plus que satisfaisante de ce principe. C'est ainsi qu'on voit les espèces former un arbre où chaque nœud correspond à un ancêtre commun et où la longueur des branches représente le temps pendant lequel les espèces ont évolué.



Arbre phylogénétique simplifié des Angiospermes, d'après APG IV

Remarque

Grâce à PhyML, on a pu reconstruire l'arbre phylogénétique des primates à partir de séquences d'ADN représentant plus de 900 000 paires de bases. Chaque nœud correspond à un ancêtre commun, et la longueur des branches, au temps d'évolution.

5. Intérêt

La phylogénétique a provoqué de véritables petites révolutions chez les biologistes systématiciens. Ainsi, on s'est aperçu que dans le monde végétal, les caractères morphologiques comme la forme des feuilles ou la couleur des fleurs étaient de très mauvais indicateurs pour regrouper les plantes en ordres, familles et genres. En revanche, évaluer leur degré de parenté au niveau du génome est maintenant une méthode très sûre.



Complément

Vous pouvez trouver plusieurs logiciels et algorithmes conçus spécialement pour un problème de biologie donné, en troisième année et dans le module Bioinformatique on va utiliser le logiciel *MEGA software* que vous pouvez d'ores et déjà télécharger et essayer de vous familiariser avec.