

# TANAGRA : un logiciel gratuit pour l'enseignement et la recherche

Ricco Rakotomalala

ERIC – Université Lumière Lyon 2  
5, av Mendès France  
69676 Bron  
rakotoma@univ-lyon2.fr  
<http://eric.univ-lyon2.fr/~ricco>

**Résumé.** TANAGRA est un logiciel « open source » librement accessible sur le web, il tente de concilier deux types d'utilisation. D'une part, en proposant une interface suffisamment conviviale, il est accessible aux utilisateurs non-spécialistes qui veulent effectuer des études sur des données réelles. D'autre part, en définissant une architecture simplifiée à l'extrême, les efforts de développement portent sur l'essentiel, à savoir la mise au point et l'intégration d'algorithmes de fouille de données, les chercheurs peuvent ainsi mener des expérimentations sur les méthodes. Dans cet article, nous présentons les principales fonctionnalités du logiciel en essayant de le positionner sur l'échiquier des (très) nombreux logiciels diffusés actuellement.

## 1 Introduction

TANAGRA est un logiciel gratuit de DATA MINING destiné à l'enseignement et à la recherche, diffusé sur internet (<http://eric.univ-lyon2.fr/~ricco/tanagra>). Il implémente une série de méthodes de fouilles de données issues du domaine de la statistique exploratoire, de l'apprentissage automatique et des bases de données. Sa principale originalité est qu'il tente de concilier une utilisation « néophyte » et « experte ».

Son premier objectif est d'offrir aux étudiants et aux experts d'autres domaines (médecine, bio-informatique, marketing, etc.) une plate-forme facile d'accès, respectant les standards des logiciels actuels, notamment en matière d'interface et de mode de fonctionnement, il doit être possible d'utiliser le logiciel pour mener des études sur des données réelles. Le second objectif est de proposer aux chercheurs une architecture leur facilitant l'implémentation des techniques qu'ils veulent étudier, de comparer les performances de ces algorithmes. TANAGRA se comporte alors plus comme une plate-forme d'expérimentation qui leur permettrait d'aller à l'essentiel en leur épargnant toute la partie ingrate de la programmation de ce type d'outil, notamment la gestion des données. Point très important à nos yeux, la disponibilité du code source est un gage de crédibilité scientifique, elle assure la reproductibilité des expérimentations publiées par d'autres chercheurs et, surtout, elle permet la comparaison et la vérification des implémentations.

TANAGRA n'intègre pas en revanche tout ce qui fait la puissance des outils commerciaux : multiplicité des sources de données, accès direct aux entrepôts de données et autres datamarts, interactivité des traitements avec des outils de visualisation sophistiqués. Ces outils, aussi séduisants et utiles soient-ils dans le cadre d'études sur des données réelles,

imposent des standards de développement autrement plus élaborés avec une forte proportion du code source destiné à la gestion des données et de l'interface. Nous avons voulu justement nous éloigner autant que possible de cet écueil en définissant sciemment une architecture simplifiée afin que le rapport code de calcul sur code de gestion soit le plus élevé possible tout en préservant un minimum d'ergonomie.

Dans cet article, nous présentons dans la section 2 l'architecture du logiciel et ses principales fonctionnalités. Dans la section suivante, nous tenterons de positionner TANAGRA face aux très nombreux outils de fouille de données existants. Enfin, dans la 4<sup>ème</sup> et dernière section, nous concluons en évoquant les enjeux de la diffusion du logiciel.

## 2 Fonctionnement et principales fonctionnalités

### 2.1 Organisation des traitements

TANAGRA s'inscrit dans le paradigme actuel de la filière ou diagramme de traitements : les séquences d'opérations appliquées sur les données sont visualisées à l'aide d'un graphe.

Chaque nœud représente un opérateur de fouille de données, soit de modélisation, soit de transformation, il est donc susceptible de produire de nouvelles données (les projections sur un axe factoriel par exemple). Nous le désignons également sous le terme de composant en référence au vocabulaire utilisé dans les outils de programmation visuelle. L'arête reliant deux nœuds représente le flux des données vers l'opérateur suivant. Ce mode de représentation qui est le standard actuel des logiciels de fouille de données autorise, par rapport aux logiciels pilotés par menus, la définition d'enchaînement d'opérations sur les données, tout en affranchissant l'utilisateur, par rapport aux outils fonctionnant avec un langage de script, l'apprentissage d'un langage de programmation. Dans TANAGRA, seul la représentation arborescente est autorisée, la source de données à traiter est unique.

La fenêtre principale du logiciel est subdivisée en trois grandes zones (Figure 1) : (a) en dessous la série des composants disponibles, ils sont regroupés en catégories ; (b) sur la gauche, le diagramme de traitements, représentant l'analyse courante ; (c) dans le cadre de droite, l'affichage des résultats consécutifs à l'exécution de l'opérateur sélectionné. Il est bien sûr possible de sauvegarder, soit sous un format binaire, soit sous la forme d'un fichier texte, la séquence d'instructions – le programme en quelque sorte -- définie par un utilisateur. Seuls le programme est sauvegardé, les résultats ne le sont pas. Le format texte permet à un utilisateur avancé de le manipuler directement afin de définir un nouveau diagramme de traitements.

### 2.2 Accès aux données

Enjeu très important s'il en fut, l'accès aux données a été réellement simplifié. En effet, seuls les fichiers texte avec séparateurs tabulation sont acceptés, les données manquantes ne sont pas gérées. Lors de l'importation, les données sont automatiquement recodées, deux types de variables sont reconnus : les variables continues, codées en flottant simple précision (4 octets par valeur), et les variables discrètes où 255 modalités sont acceptées (1 octet par valeur). Après recodage, l'ensemble des données est chargé en mémoire centrale, il est dès lors aisé de calculer les capacités théoriques du logiciel en fonction de la mémoire disponible.

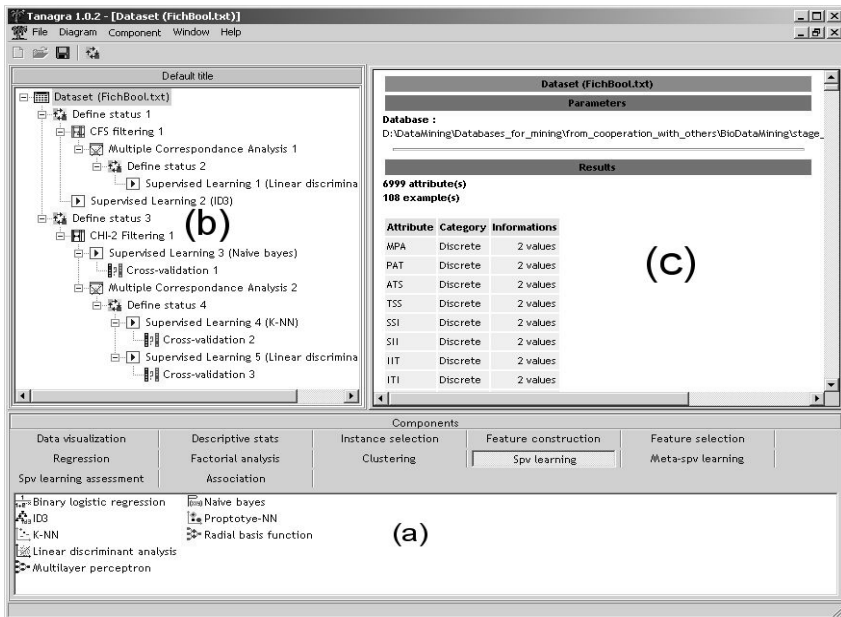


FIG. 1 – La fenêtre principale de TANAGRA.

### 2.3 Algorithmes de traitement

A l’instar de tous les logiciels de recherche, toutes les méthodes de traitement de données sont dûment référencées. Le code source étant accessible, il est de plus possible pour tout un chacun de vérifier l’implémentation réalisée.

Les algorithmes sont regroupés en grandes familles, certains peuvent être discutables mais il ne nous semblait pas approprié de trop multiplier les catégories. Grosso modo, nous distinguerons deux grandes super-familles, à savoir les algorithmes d’obédience statistique : statistique descriptive, statistique inférentielle, analyse de données et économétrie ; et les algorithmes issus des publications en apprentissage automatique et bases de données : filtrage d’individus et de variables, apprentissage supervisé, règles d’association. Nous ne revendiquons nullement la pertinence du découpage choisi, il fallait à la fois composer avec la pratique des utilisateurs et une ergonomie plus ou moins heureuse.

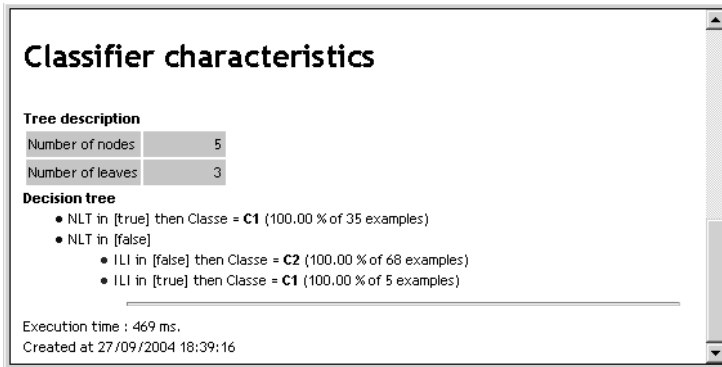


FIG. 2 – Un exemple de résultats d'induction d'arbres de décision au format HTML.

Un composant représente un algorithme de traitement de données. Les composants ont pour point commun de prendre en entrée des données en provenance du composant qui le précède ; de procéder à des calculs donnant lieu à un affichage des résultats sous forme de page HTML (Figure 2); ils sont le plus souvent paramétrables ; et enfin, ils transmettent aux composants en aval les données en y ajoutant parfois des données produites localement, les prédictions par exemple pour les méthodes supervisées.

La possibilité d'enchaîner des méthodes d'apprentissage à travers le diagramme de traitements est un atout indéniable, en effet, il rend aisé la combinaison des méthodes sans avoir pour autant à se lancer dans l'apprentissage d'un langage de script (Figure 3). La plupart des logiciels commerciaux du marché, même ceux qui disposent à l'origine d'un langage de programmation, proposent aujourd'hui ce mode de représentation qui fait référence.

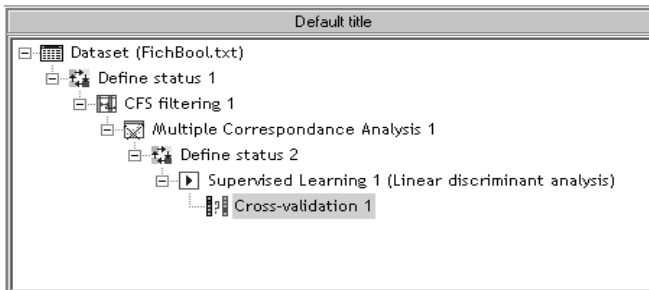


FIG. 3 – Un diagramme de traitements implémentant : une sélection de variables, une analyse des correspondances sur les variables sélectionnées, une analyse discriminante sur les axes factoriels, une évaluation par validation croisée.

## 2.4 Performances

TANAGRA est développé avec le langage de programmation DELPHI. Une version gratuite du compilateur est disponible sur le site de BORLAND. Le programme est donc compilé, il est distribué tel quel, son exécution ne nécessite aucune bibliothèque supplémentaire. En revanche, il ne fonctionne que sous Windows.

La principale faiblesse du logiciel réside dans l'obligation de charger, sous forme recodée, la totalité des données en mémoire. Un fichier de 1 000 000 d'observations avec 1000 variables exclusivement continues occupe approximativement 382 Mo en mémoire centrale. On peut relativiser ce goulot d'étranglement en ce qui concerne les fichiers usuellement rencontrés. Un PC de bureau doté de 512 Mo de mémoire vive par exemple peut traiter directement l'ensemble des clients d'une grande banque régionale pour un ciblage marketing. En revanche, traiter l'ensemble des transactions journalières d'une enseigne de grande distribution en chargeant les données en mémoire paraît inconcevable.

En ce qui concerne le temps de traitement, pour donner un ordre d'idées sur l'implémentation, la création d'un arbre de décision avec la méthode ID3 de Quinlan sur le fichier « Forest CoverType » du serveur UCI (Hettich et Bay 1999) comportant 580 000 individus et 56 variables (les 10 variables continues ont été discrétisées), est réalisée en 9 secondes sur un Pentium 4 à 3 Ghz fonctionnant sous Windows 2000. L'arbre final comporte 927 feuilles.

## 3 Tanagra et les logiciels de fouille de données

L'offre de logiciels de fouille de données est pléthorique. Malheureusement pour les chercheurs, ces offres sont le plus souvent le fait d'entreprises commerciales et, très souvent, les algorithmes implémentés ne sont ni documentés, ni référencés, rendant très difficile la publication d'articles.

TANAGRA s'inscrit dans la lignée des plates-formes d'expérimentations ouvertes qui se sont rapidement répandues depuis le début des années 90. On peut citer rapidement les bibliothèques IND (Buntine 1991), ou encore MLC++ (Kohavi et Sommerfield 2002). A l'heure actuelle, le projet WEKA (Witten et Frank 2000) est certainement celui qui nous a le plus inspiré. Si ces références, pour la plupart en provenance de la communauté de l'apprentissage automatique, semblent assez récentes, il ne faut pas perdre de vue que mutualiser des algorithmes de traitement de données a été depuis très longtemps mis en place dans la communauté des statisticiens, sous forme de code FORTRAN ou de scripts de haut niveau.

TANAGRA fait suite à plusieurs projets développés au sein de notre laboratoire depuis plusieurs années. Le plus connu d'entre eux a été le projet SIPINA (Zighed et al. 1992) piloté par D. Zighed depuis une vingtaine d'années. Nous avons intégré en cours de route le développement de la version 2.5 en 1994, puis nous avons été le principal maître d'œuvre de la version recherche dont l'implémentation a réellement commencé en 1998. Au-delà de la disponibilité de l'outil sur le web (<http://eric.univ-lyon2.fr/~ricco/sipina.html>) et des contacts que nous avons pu nouer avec de nombreux chercheurs dans le monde, cette plate-forme nous a beaucoup servi pour développer nos propres expérimentations qui ont donné lieu à des publications. SIPINA était avant tout dédié à l'apprentissage supervisé, il nous est apparu au fil du temps que son architecture n'était plus adaptée, notamment parce qu'il n'était pas

possible d'enchaîner automatiquement des méthodes de construction et de sélection automatique de variables. De plus, il était nécessaire pour chaque méthode ajoutée de définir une interface de visualisation spécifique. TANAGRA a donc intégré dès le départ les spécifications adéquates pour dépasser ces limitations qui étaient devenues contraignantes.

## 4 Conclusion

TANAGRA est avant tout destiné à la recherche, en ce sens nous nous engageons à ce que le logiciel soit toujours gratuit et le code source accessible. Le choix de la licence ne fut pas aisé, le concept de logiciel libre, aussi séduisant soit-il, laissait la porte ouverte à l'appropriation commerciale de l'outil par de tierces personnes, avec des contraintes de publications de codes certes, mais difficile à faire respecter.

Notre premier enjeu aujourd'hui est d'assurer la diffusion du logiciel afin qu'il soit utilisé dans différents domaines, les retours de ces utilisateurs nous permettent d'affiner les fonctionnalités du logiciel, améliorant ainsi son efficacité. Depuis le début de l'année 2004, nous comptons une vingtaine de visiteurs par jour sur notre site web. Notre second objectif est de fédérer les bonnes volontés pour élargir la bibliothèque des méthodes de fouille de données. Ce deuxième objectif est un peu plus délicat, seuls quelques chercheurs dans l'entourage proche de notre laboratoire l'ont réellement réalisé à ce jour.

## Références

- Buntine W. (1991), About the IND tree package, Technical Report, NASA Ames Research Center, Moffet Field, California, September 1991.
- Hettich, S., Bay S. (1999). The UCI KDD Archive [<http://kdd.ics.uci.edu>]. Irvine, CA: University of California, Department of Information and Computer Science.
- Kohavi R., Sommerfield D. (2002), MLC++. In Will Klossgen and Jan M. Zytkow, editors, Handbook of Data Mining and Knowledge Discovery, chapter 24.1.2, pages 548-553. Oxford University Press, 2002.
- Witten I., Frank E. (2000), Data Mining: Practical machine learning tools with Java implementations, Morgan Kaufmann, San Francisco, 2000.
- Zighed D., Auray J.P., Duru G. (1992), SIPINA : Méthode et logiciel, Lacassagne, 1992.

## Summary

TANAGRA is an open source software available on the web. It tries to reconcile two kinds of users. On the one hand, non-specialists can use the soft, which proposes a user-friendly GUI. On the other hand, a simplified architecture makes it possible to the researchers to concentrate their efforts on the development and the evaluation of new data mining algorithms. In this paper, we present the main functionalities of this new data mining software.