

"Typol'98" : un programme de classification rapide en Pascal objet pour Windows 98, NT 4 ou XP

○=○=○

Table des matières

"Typol'98" : un programme de classification rapide en Pascal objet pour Windows 98, NT 4 ou XP.....	1
INTRODUCTION.....	2
1. PRINCIPE DE L'ALGORITHME.....	3
1.1. Quantification des différences entre objets.....	3
i) Transformation de la matrice en une matrice "normalisée".....	3
ii) Prise en compte d'une pondération des caractères.....	3
iii) Calcul des distances.....	4
1.2. Classement des objets.....	4
2. TESTS DU PROGRAMME.....	4
2.1. Un cas qui "ne marche pas".....	4
2.2. Un cas qui "marche".....	5
3. INFLUENCE DE LA QUESTION POSEE.....	6
3.1. Premier traitement.....	7
3.2. Deuxième traitement.....	8
4. EXEMPLE DETAILLE D'APPLICATION.....	9
4.1. Classement de plans d'églises romanes.....	9
4.2. Analyse des résultats obtenus.....	12
CONCLUSION GENERALE.....	14
REFERENCES.....	15
ANNEXE : Planche « Regroupement des plans par classes ».....	16

○=○=○

Avertissement :

Ce texte est une version adaptée et mise à jour d'un article publié dans la littérature en 1985 [0].

Les nombres entre crochets renvoient aux références placées en fin d'article.

INTRODUCTION

Les méthodes de classification automatique, qui permettent de mettre de l'ordre au sein d'un matériel décrit sous un certain nombre d'aspects, constituent un vaste champ d'étude. La littérature spécialisée, abondante et foisonnante, le montre bien. Le livre de J.L. CHANDON et S. PINSON [1] constitue une bonne introduction au sujet. Les spécialistes des sciences humaines ont donc à leur disposition un arsenal de méthodes de traitement de données susceptible de s'appliquer à leur matériel.

Il semble cependant encore exister un large fossé entre le numéricien qui développe une méthode automatique de classification et l'historien ou l'archéologue, par exemple, confrontés à leur matériel.

En effet, les spécialistes en analyse numérique doivent se consacrer aux difficultés théoriques et pratiques que soulève la mise en œuvre des procédures numériques de classification. Leurs passionnantes discussions ou publications le reflètent bien, qui souvent sont inaccessibles à l'humaniste "moyen" sans formation mathématique particulière (cf. [2], par exemple). Il faut aussi savoir qu'entre le moment où un programme "tourne" effectivement (c'est-à-dire que son auteur peut l'utiliser pour en tirer des résultats) et le moment où il est réellement opérationnel (c'est-à-dire qu'il peut être utilisé par une personne ignorant la programmation), il faut en général faire un important travail de mise en forme (du côté de l'auteur du programme), de formation et de familiarisation (du côté de l'utilisateur).

Ainsi, le domaine de la classification automatique paraît souvent à l'utilisateur un monde distant et ésotérique. Son attitude peut alors appartenir à un large spectre, depuis la réaction de rejet pure et simple jusqu'à l'attirance motivée par le désir d'être "à la page", en passant par l'attitude circonspecte du chercheur qui a une grande masse de données à traiter et songe à utiliser un ordinateur, mais demeure perplexe devant la difficulté à choisir une méthode, à la mettre en œuvre et à évaluer la qualité des résultats numériques qu'il peut en attendre (s'il n'est pas perplexe, il peut parfois le devenir à la lecture des préoccupations - justifiées - de certains mathématiciens [5]).

Il est donc nécessaire de jeter des ponts entre les deux spécialités. Dans cette optique, il faut saluer la contribution de S.CLEUZIOU et J.P. DEMOULE [4] qui abordent les méthodes de classification automatique en les situant à leur juste place par rapport à l'ensemble de la démarche archéologique. Le présent article est une tentative pour apporter une pierre à l'édifice ; son but est donc double :

- présenter un algorithme de classification automatique,
- en montrer le fonctionnement sur quelques exemples en essayant d'en tirer quelques enseignements pour l'utilisateur potentiel.

A cet effet, une première partie donne les grandes lignes du principe de l'algorithme. Ensuite, la deuxième partie montre deux applications tests du programme à des données artificielles, pour cerner les performances et limites de validité de la procédure, puis la troisième partie est consacrée au traitement d'une petite matrice simple permettant de mettre en évidence l'influence de la façon de poser le problème sur les résultats. Enfin, la quatrième partie traite une application réaliste.

1. PRINCIPE DE L'ALGORITHME

L'utilisateur est supposé avoir constitué un échantillon composé d'un certain nombre d'objets, décrits par une suite d'attributs, et il souhaite les ranger pour rapprocher ceux qui se ressemblent. Il est bien évident que la qualité du résultat obtenu dépend des attributs choisis et, par hypothèse, les attributs retenus sont suffisants pour caractériser le matériel ; pour cette raison, ils sont aussi appelés "traits caractéristiques" ou, en abrégé, caractères. Il est en général également intéressant d'ordonner les caractères de façon à rapprocher ceux qui varient de façon similaire sur l'ensemble des objets.

Par construction, pour chaque objet, chaque caractère prend une valeur. Celle-ci peut être quantitative (dimensions, année,...) ou qualitative (du type absence - présence ; ce cas peut être ramené au précédent en codant "absence" par "0" et "présence" par "1"). Cependant, en raison du caractère souvent lacunaire du matériel archéologique ou historique, quelques données manquantes sont tolérées.

Les données constituent alors une matrice descriptive ; à partir de ces données, il est possible de construire, pour chaque paire d'objets, un nombre unique qui traduit globalement la dissemblance entre ces deux objets en prenant en compte tous les caractères ; ce nombre s'appelle un indice de dissimilarité (définition d'après [1], p.48) ou, plus simplement, dans le cas du programme présenté ici, une distance. De façon analogue, il est possible de définir une distance entre caractères. La manière de calculer cette distance serait trop longue à détailler ici mais les opérations qu'elle implique sont décrites succinctement ci-après.

1.1. Quantification des différences entre objets

i) Transformation de la matrice en une matrice "normalisée"

L'intérêt de cette transformation est montré par l'étude du cas de trois personnes à comparer d'après leur taille et le temps qu'elles mettent à courir cent mètres :

		Objets		
		Personne X	Personne Y	Personne Z
Caractères	taille en cm	180	180	160
	temps en sec	10	11	12

A l'évidence X et Y sont presque semblables (du point de vue des caractères choisis !) mais Y et Z sont différents. Au vu de ce tableau, un ordinateur "pensera" que ces personnes sont surtout différentes à cause de la taille (différence de 20) plutôt qu'à cause du temps (différence de 1 seulement). Toutefois, si la taille est exprimée en mètres (1,8 pour Y et 1,6 pour Z) et le temps en millièmes de seconde (11000 pour Y et 12000 pour Z), c'est cette fois la différence sur la taille (0,2) qui paraît négligeable devant celle sur le temps (1000). Les chiffres traduisent pourtant la même réalité physique. Pour éviter de semblables distorsions, tous les intervalles de variation des caractères sont ramenés automatiquement à [0,1] par une transformation linéaire des valeurs.

ii) Prise en compte d'une pondération des caractères.

La transformation précédente supprime une pondération inconsciente des caractères mais le programme offre à l'utilisateur la possibilité de donner volontairement plus ou moins d'importance à certains caractères en multipliant par une quantité donnée (ponds) tous les écarts entre objets apparaissant sur ce caractère. Cette possibilité n'est pas offerte dans le cas des calculs de distances entre caractères, où tous les objets sont réputés équivalents.

iii) Calcul des distances

Les distances entre objets (ou entre caractères) sont calculées par la formule dite de la distance rectangulaire pondérée [1].

1.2. Classement des objets

En appliquant ces trois opérations, l'algorithme commence par calculer toutes les distances entre les objets (ou les caractères, suivant la phase de traitement en cours) pris deux à deux. Dans la suite, on raisonne sur le cas des objets mais tout se transpose au cas du rangement des caractères.

Une fois ces distances disponibles, il s'agit de trouver un rangement plaçant à côté de chaque objet celui qui lui ressemble le plus. La procédure présentée ici consiste à partir d'un objet, à mettre en deuxième position le plus proche parmi les autres, puis en troisième position le plus proche du deuxième parmi ceux qui restent etc., et on construit ainsi progressivement un "chemin" jusqu'à épuisement de tous les objets.

En fait, chaque objet est essayé comme point de départ et on retient le "meilleur" rangement de tous ceux ainsi obtenus, c'est-à-dire celui qui correspond au chemin le plus court.

Remarque : ce processus n'essaye pas tous les rangements possibles et ne fournit donc pas la garantie d'obtenir le chemin le plus court dans l'absolu. Cette méthode permet néanmoins, dans la plupart des cas, d'arriver à un résultat satisfaisant en peu de temps.

En fin d'exécution, le programme donne l'ordre obtenu pour les objets (et la représentation graphique de la matrice rangée) ainsi que des propositions de coupures pour définir des classes, numérotées dans l'ordre d'intérêt décroissant.

2. TESTS DU PROGRAMME

L'application du programme à deux petites matrices artificielles est présentée ci-après, en commençant par un cas permettant de fixer les limites de ce type d'approche.

2.1. Un cas qui "ne marche pas"

Soit cinq objets O1, O2, O3, O4, O5 décrits par cinq caractères C1, C2, C3, C4, C5 du type "absence -présence" comme l'indique la matrice M1 du tableau où 1 signifie "présence" et 0 signifie "absence".

		Objets				
		O1	O2	O3	O4	O5
Caractères	C1	1	0	0	0	0
	C2	0	1	0	0	0
	C3	0	0	1	0	0
	C4	0	0	0	1	0
	C5	0	0	0	0	1

Tableau 1 : matrice M1

Le test consiste à donner comme entrée au programme une forme désordonnée M2 de la même matrice, celle décrite par le tableau 2 par exemple, et à lui demander de ranger à la fois les objets et les caractères.

		Objets				
		O4	O2	O3	O5	O1
Caractères	C1	0	0	0	0	1
	C4	1	0	0	0	0
	C2	0	1	0	0	0
	C5	0	0	0	1	0
	C3	0	0	1	0	0

Tableau 2 : matrice M2

En résultat, le programme redonne la matrice M2 et n'améliore donc pas du tout le rangement.

L'explication réside dans la nature très "creuse" de la matrice (beaucoup plus de 0 que de 1). L'interprétation du cas extrême présenté ici est la mise en évidence d'un choix de caractères mal adapté : chaque caractère ne convient vraiment qu'à un seul objet. En conséquence, la distance entre deux objets, quels qu'ils soient, est égale à 1 et tous les rangements sont équivalents !

Des algorithmes capables de ranger des matrices très creuses existent (voir par exemple [5]), mais ils ne peuvent pas faire de miracle : une matrice descriptive creuse risque d'être l'indication d'un mauvais choix de la grille descriptive et, dans ce cas, effectuer le rangement n'apporte aucune information utile.

2.2. Un cas qui "marche"

Cette fois, sont considérés cinq objets décrits par la matrice M3 du tableau 3.

		Objets				
		O1	O2	O3	O4	O5
Caractères	C1	1	0,5	0	0	0
	C2	1	1	0,5	0	0
	C3	0,5	1	1	0,5	0
	C4	0	0,5	1	1	0,5
	C5	0	0	0,5	1	1

Tableau 3 : matrice M3 (données quantitatives imaginaires)

Ensuite, la forme M4 de la même information, brouillée par une mise en désordre manuelle montrée dans le tableau 4, est soumise au programme.

		Objets				
		O4	O3	O5	O1	O2
Caractères	C5	1	0,5	1	0	0
	C2	0	0,5	0	1	1
	C4	1	1	0,5	0	0,5
	C1	0	0	0	1	0,5
	C3	0,5	1	0	0,5	1

Tableau 4 : matrice M4

En résultat, le programme retrouve l'ordre de la matrice initiale comme le visualisent les représentations graphiques de la figure 1.

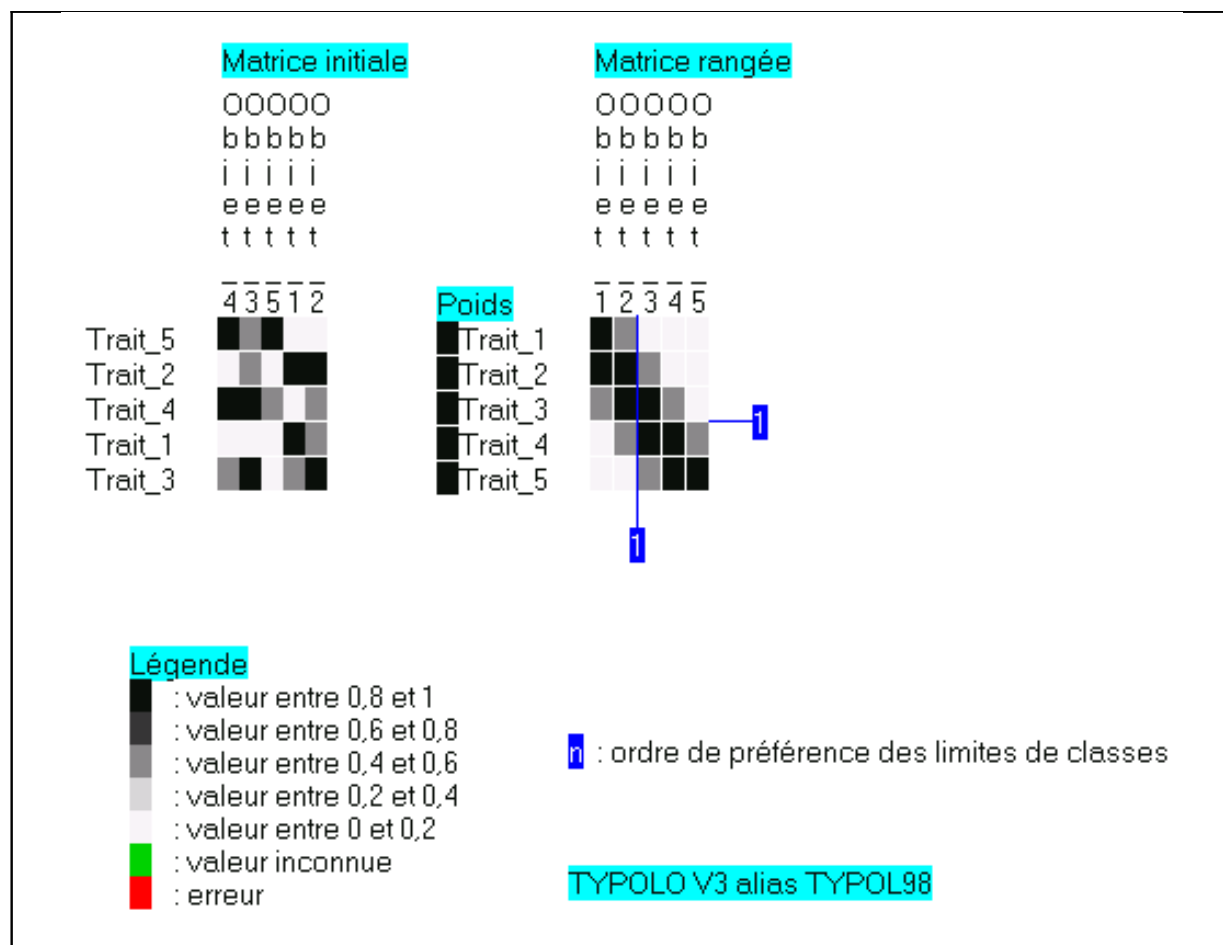


Figure 1 : représentation graphique des données et résultats

3. INFLUENCE DE LA QUESTION POSEE

Une même information peut conduire à deux interprétations différentes suivant la question posée à son propos, c'est-à-dire suivant la façon d'appliquer le programme. L'exemple très simple suivant illustre ce fait.

L'information de base traitée ici est tirée d'un exemple donné par J. BERTIN [6], sous l'intitulé "le problème de la viande dans le marché commun", et résumée par le tableau 5 qui donne la répartition de différentes catégories de cheptel entre différents pays du Marché Commun, en pourcentage, pour une année donnée.

	Italie	RFA	Benelux	France	Pays-Bas	Total
Veaux	12	15	3	60	10	100
Ovins	20	5	1	70	4	100
Porcs	9	45	6	29	11	100
Équidés	24	6	6	60	4	100
Gros bovins	17	32	6	38	7	100

Tableau 5 : répartition du cheptel

Face à une telle information, faut-il considérer les pays comme les objets et les catégories de cheptel comme les caractères ou le contraire ? En fait, le premier choix correspond plutôt à la question "Comment se classent les producteurs de viande ?" tandis que le second est plutôt associé à la question "Par qui la production de viande est-elle assurée ?".

Comme la normalisation de la matrice mentionnée au paragraphe 1 n'a pas le même effet sur les objets et sur les caractères, les deux choix conduisent à des résultats différents.

3.1. Premier traitement

Un premier traitement effectué avec les pays comme objets et les catégories de cheptel comme caractères conduit aux résultats représentés graphiquement par la figure 2.

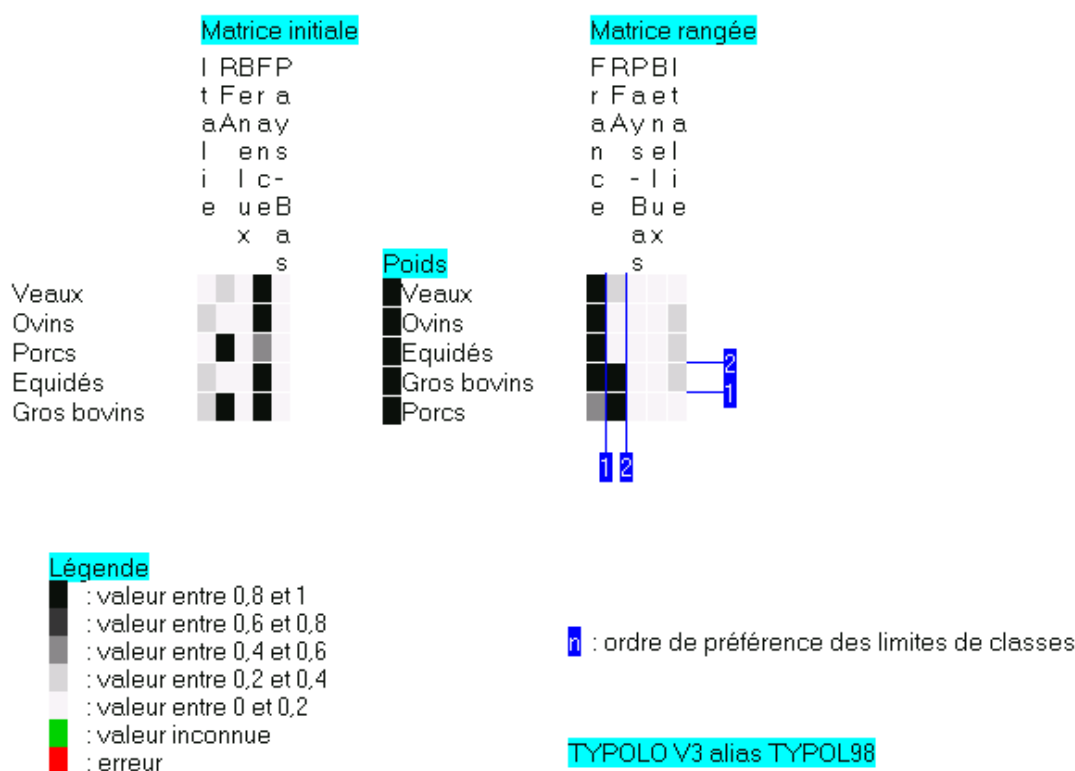


Figure 2 : représentation graphique des résultats du premier traitement.

Le programme propose, pour les objets (pays), l'ordre et le découpage en trois classes suivants (l'épaisseur du trait de séparation entre classes augmente avec la priorité de la coupure).

Classe 1	Classe 2	Classe 3
France	RFA	Pays-Bas, Benelux, Italie

et, pour les caractères (cheptel) :

Classe 1	Classe 2	Classe 3
Veaux, Ovins, Équidés	Gros bovins	Porcs

Ces résultats s'interprètent aisément : les pays sont classés en fonction de l'importance de la production de viande, avec la première coupure proposée entre la RFA et la France, ce qui suggère de reconnaître deux classes, l'une composée de la France seule et l'autre des quatre

autres pays (avec cinq pays, il ne semble pas judicieux de prendre en compte plus d'une coupure). De même, pour les caractères, on peut introduire deux classes : la première (veaux, ovins, équidés, gros bovins) est celle qui donne à la France sa position privilégiée alors qu'elle n'a pas l'avantage pour la seconde classe (porcs).

3.2. Deuxième traitement

Le deuxième traitement effectué avec le cheptel en objet et les pays comme caractères conduit aux résultats représentés graphiquement par la figure 3.

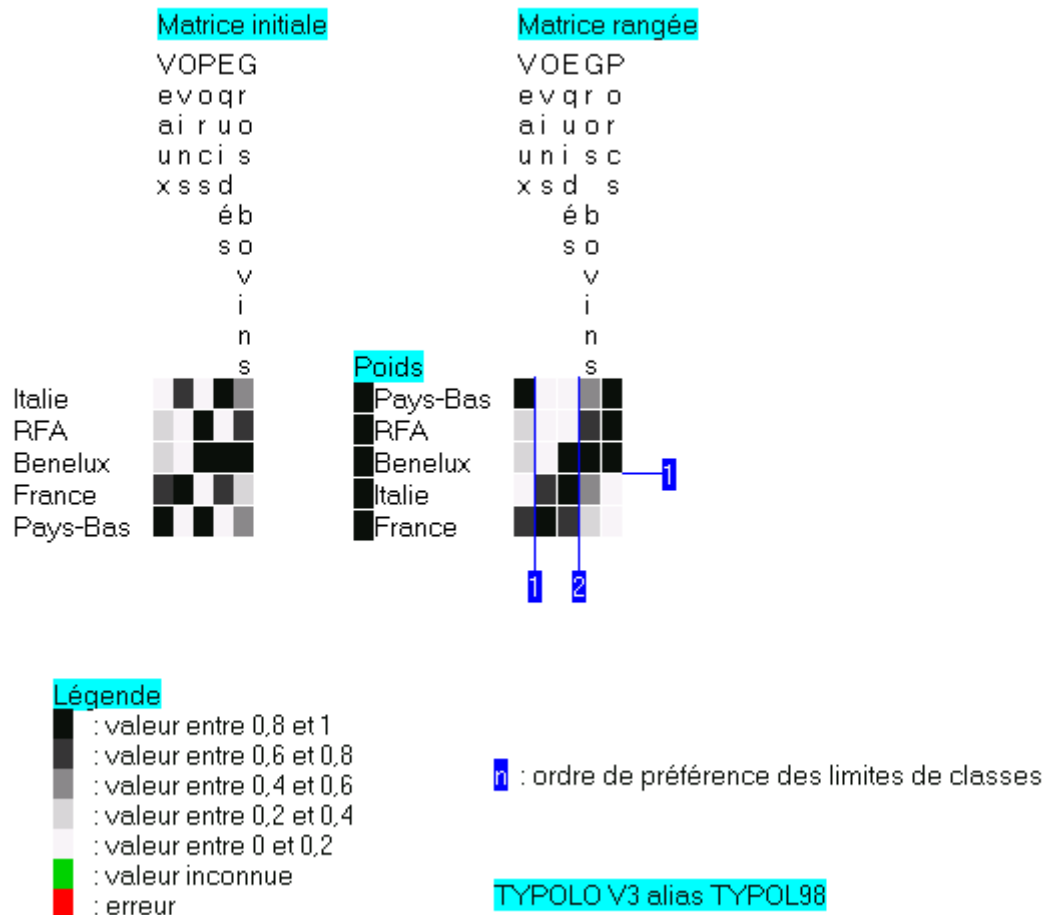


Figure 3 : représentation graphique des résultats du deuxième traitement

Le programme propose, pour les objets (viandes), l'ordre et le découpage en trois classes suivants :

Classe 1	Classe 2	Classe 3
Porcs, gros bovins	Equidés, ovins	Veaux

et, pour les caractères (pays) :

Classe 1	Classe 2
Pays-Bas, RFA, Benelux	Italie, France

Ce résultat est différent du précédent et il est intéressant de noter que l'ordre des pays obtenu est pratiquement celui présenté par J.BERTIN (à l'intervention près de la RFA et des Pays-Bas) et, surtout, que la distinction en deux classes est celle qu'il propose en l'interprétant comme l'indice de politiques opposées ou complémentaires ; ceci se vérifie à l'aide de l'ordre obtenu pour les viandes : les deux premières sont plutôt privilégiées par la première classe de pays tandis que les deux suivantes sont plutôt privilégiées par la deuxième classe, le veau étant un peu à part car privilégié pratiquement par les Pays-Bas.

Naturellement, l'exemple qui vient d'être présenté est un peu simpliste mais il a néanmoins le mérite d'attirer l'attention sur un point crucial. D'abord, de façon ponctuelle, il indique que, lorsque le contexte n'impose pas à l'évidence le choix des objets et des caractères, celui-ci doit faire l'objet d'une réflexion approfondie. Ensuite, de façon plus générale, cet exemple montre que le fait d'utiliser un ordinateur n'ajoute pas de science à la démarche effectuée. Comme le burin dans la main du sculpteur, l'ordinateur est un instrument aveugle. C'est à son utilisateur qu'incombe la responsabilité de la qualité de l'étude, à tous les niveaux :

- choix de l'échantillon d'objets,
- définition de la grille descriptive,
- utilisation judicieuse des résultats pour l'interprétation.

4. EXEMPLE DETAILLE D'APPLICATION

4.1. Classement de plans d'églises romanes

L'application détaillée dans ce paragraphe a pour but le classement de 17 églises romanes d'après la forme de leurs plans. Ce sont les suivantes, prises dans différentes régions de France, dans un ordre aléatoire :

- Notre-Dame du Port à Clermont-Ferrand (Auvergne),
- La Trinité de Caen (Basse Normandie),
- Saint -Trophime d'Arles (Provence),
- Eglise d'Eschau (Alsace),
- Église d'Issoire (Auvergne),
- Saint-Paul-Trois-Châteaux (Provence),
- Eglise de Bernay (Haute Normandie),
- Notre-Dame des Doms à Avignon (Provence),
- Saint -Saturnin (Auvergne),
- Eglise de Rosheim (Alsace),
- Eglise d'Orcival (Auvergne),
- Saint-Nicolas de Caen (Basse Normandie),
- Saint-Restitut (Provence),
- Eglise de Saint-Nectaire (Auvergne),
- Saint-Gabriel (Provence),
- Abbatale de Lessay (Basse Normandie),
- Eglise de Boscherville (Haute Normandie).

Les 7 caractères sélectionnés pour traduire la forme des plans sont les proportions des principaux éléments du plan :

- la longueur de la nef sur la longueur totale de l'église (caractère noté : "Longueur nef/totale")
- la longueur du chœur sur la longueur totale ("Longueur choeur/totale")

- la largeur du transept sur la largeur totale de la nef et de ses collatéraux ("Largeur transept/totale")
- la largeur de la nef seule sur la largeur totale de la nef et de ses collatéraux ("Largeur nef/totale")
- la largeur totale de la nef et de ses collatéraux sur la longueur de la nef ("Largeur totale/longueur nef")
- la longueur du narthex sur la longueur de la nef ("Longueur narthex/longueur nef")
- la largeur du chœur sur la longueur du chœur ("Largeur chœur/longueur chœur")

A ces caractères, traduisant purement la forme, sont également ajoutés certains caractères traduisant plutôt la structure du bâtiment. Ce sont :

- le nombre de collatéraux,
- le nombre de travées de la nef,
- le nombre d'absidioles du chœur,
- le nombre d'absidioles du transept.

Tous ces caractères reçoivent un poids égal à 1.

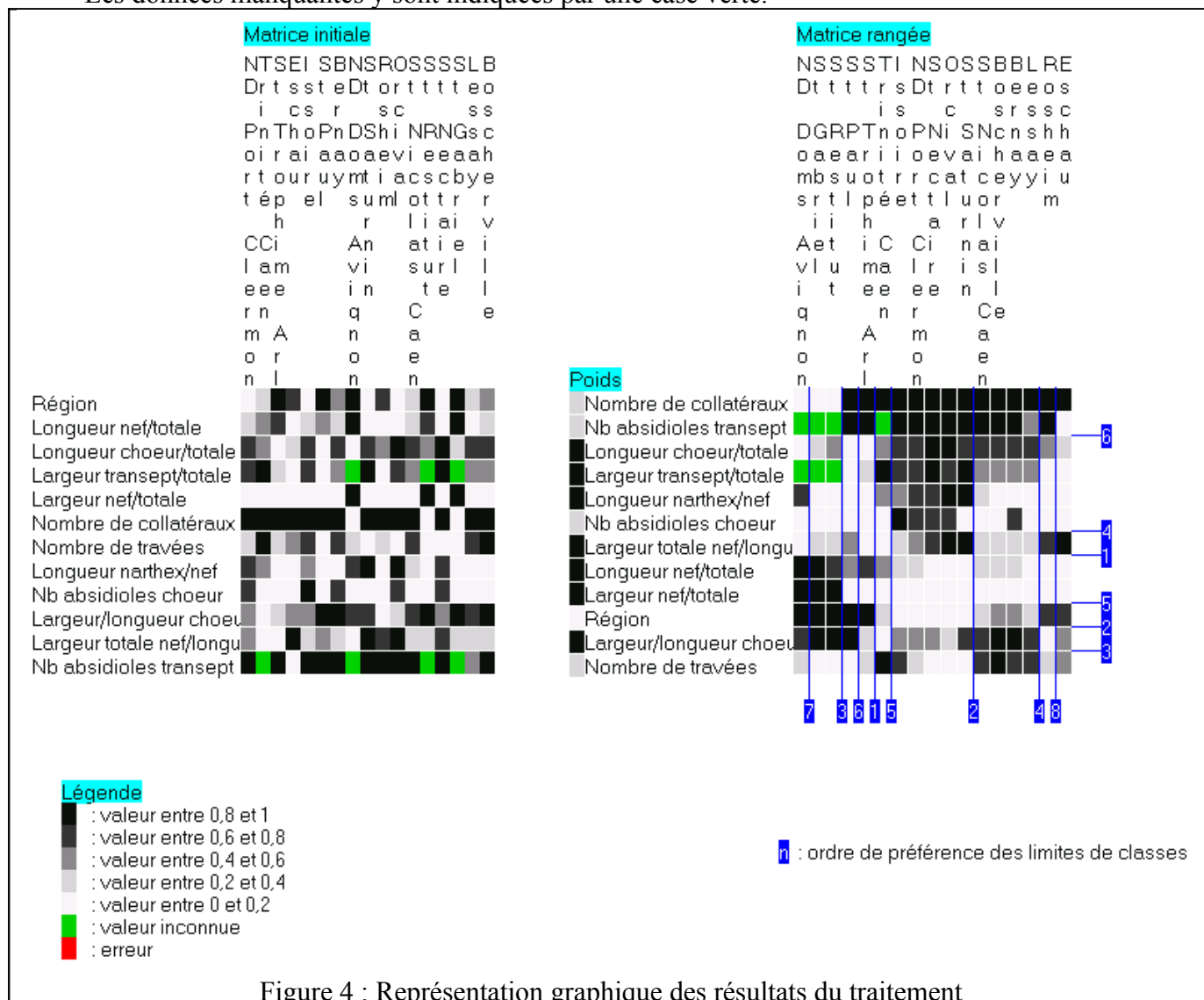
Enfin, on introduit, pour mémoire, un caractère extrinsèque traduisant la région (1 : Auvergne, 2 : Basse Normandie, 3 : Haute Normandie, 4 : Alsace, 5 : Provence) ; ce caractère reçoit un poids nul pour éviter qu'il soit pris en compte par le programme lors des opérations de classement.

Le tableau 6 donne la matrice descriptive. Dans cette matrice, à sept endroits, l'indication "?" signale, de façon conventionnelle, une donnée "manquante", afin d'illustrer la capacité du programme à traiter une information comportant quelques lacunes.

Caractères	Poids		Cl er mont	Tri nit é de Ca en	Arl es	Es ch au	Iss oir e	St- Pa ul- 3- ch âte au x	Be rn ay	Avi gn on	St- Sa tur nin	Ro sh ei m	Or civ al	St- Ni col as de Ca en	St- Re stit ut	St- Ne cta ire	St- Ga briel	Le ss ay	Bo sc her vill e
Région	0		1	2	5	4	1	5	3	5	1	4	1	2	5	1	5	2	3
Longueur nef/totale	1		0,59	0,66	0,74	0,57	0,6	0,68	0,6	0,87	0,54	0,57	0,5	0,6	0,78	0,52	0,82	0,57	0,58
Longueur chœur / totale	1		0,28	0,22	0,13	0,18	0,28	0,12	0,27	0,12	0,27	0,24	0,33	0,25	0,22	0,33	0,18	0,28	0,26
Largeur transept / totale	1		1,72	1,83	1,56	1,34	1,71	1,39	1,67	?	1,8	1,37	1,71	1,63	?	1,9	?	1,64	1,62
Largeur nef / totale	1		0,46	0,52	0,54	0,52	0,47	0,44	0,45	1	0,51	0,44	0,48	0,52	1	0,5	1	0,47	0,51
Largeur totale / longueur nef	1		0,63	0,41	0,37	0,99	0,54	0,64	0,5	0,39	0,92	0,76	0,88	0,54	0,51	0,75	0,5	0,53	0,51
Longueur narthex / longueur nef	1		0,24	0,15	0	0	0,19	0	0	0,26	0,33	0	0,31	0,1	0	0,25	0	0	0
Largeur chœur / Longueur chœur	1		1,17	0,64	1,09	1,22	1,14	1,68	1,61	1,44	1,45	0,81	1,12	1,58	1,68	1,2	1,67	1,51	1,84
Nombre de collatéraux	0,25		2	2	2	2	2	2	2	0	2	2	2	2	0	2	0	2	2
Nombre de travées de la nef	0,25		5	9	5	6	7	3	7	5	3	5	4	7	3	4	3	7	8
Nombre d'absidioles du chœur	0,25		4	0	0	0	5	0	3	0	0	0	4	1	0	3	0	1	1
Nombre d'absidioles du transept	0,25		2	?	2	0	2	2	2	?	2	2	2	2	?	2	?	1	2

Tableau 6 : matrice descriptive des plans

La figure 4 montre la représentation graphique de cette matrice descriptive après rangement. Les données manquantes y sont indiquées par une case verte.



4.2. Analyse des résultats obtenus

Le premier résultat obtenu est l'ordre du rangement effectué par le programme. Il y a lieu de rappeler que le caractère "région", affecté d'un poids nul, n'est pas pris en compte par le programme pour ranger les plans. Néanmoins, l'ordre obtenu rapproche les églises par régions, à deux exceptions près :

- la Trinité de Caen est intercalée entre des églises provençales et des églises auvergnates,
- les églises de Haute-Normandie et de Basse Normandie sont mélangées.

Le deuxième apport du programme est une proposition de formation de classes d'objets. Les coupures entre classes sont proposées dans un ordre de priorité qui traduit l'importance de la distance entre les deux objets situés de part et d'autre de la coupure : la première coupure est placée entre les deux objets voisins dans le classement les plus distants, la seconde entre les deux objets, voisins dans le classement, séparés par la distance immédiatement inférieure à la plus grande, etc. ; toutefois, il n'est pas nécessairement opportun de retenir l'intégralité des coupures en descendant jusqu'au bas de la liste proposée.

En effet, l'analyse typologique vise à satisfaire deux objectifs antinomiques : pour avoir valeur de "type", les classes distinguées doivent avoir à la fois un effectif suffisant et une certaine homogénéité. Il est évident que la meilleure homogénéité des classes est obtenue lorsqu'elles comportent chacune un seul objet (si les objets sont tous différents) mais il n'est alors plus guère possible de parler de "classes". Il faut donc augmenter l'effectif mais sans atteindre l'autre extrême qui consisterait à regrouper toute la population dans une seule classe fourre-tout, sans grande valeur ajoutée. Trouver le bon compromis entre ces deux extrêmes est l'une des difficultés de l'analyse.

Le programme TYPOL'98 effectue déjà un premier tri en ne proposant de coupures qu'aux intervalles plus grands que la moyenne. Cependant le nombre de coupures proposées est encore souvent trop grand et conduit à des classes de trop petits effectifs. Ceci conduit à éliminer certaines coupures en queue de liste de priorité.

En pratique, le plus simple est d'examiner la dernière coupure proposée par le programme : si son introduction n'engendre que des petites¹ classes, elle est éliminée ; dans ce cas, on procède ensuite de même avec la nouvelle coupure qui est devenue la dernière ; et ainsi de suite jusqu'à trouver une coupure dont l'introduction crée au moins une classe de taille significative, qui est retenue comme dernière coupure ; toutes les coupures de rang supérieur sont alors conservées, quelles que soient les tailles de classes engendrées par leur introduction.

Dans le cas présent, le déroulement de ce processus est résumé par le tableau suivant :

N° de coupure	Tailles des classes engendrées	Décision
8	1 et 1	éliminée
7	1 et 2	éliminée
6	1 et 1	éliminée
5	1 et 5	conservée
4	4 et 2	conservée

Les coupures 1 à 5 sont donc conservées.

En définitive, en première analyse, les classes données par le tableau 7 paraissent devoir être distinguées.

Classe 1	Classe 2	Classe 3	Classe 4	Classe 5	Classe 6
Avignon, St-Gabriel, St-Restitut	St-Paul-3-ch., Ste-Trophime d'Arles	Trinité	Issoire, Clermont, St-Nectaire, Orcival, St-Saturnin	St-Nicolas, Boscherville, Bernay, Lessay	Rosheim, Eschau

Tableau 7 : classes en première analyse

La troisième étape consiste à exploiter le classement obtenu pour les caractères. Le programme rapproche les caractères qui se ressemblent, c'est-à-dire les caractères qui varient de façon corrélée sur l'ensemble des objets. Les caractères fortement corrélés forment une classe. Ceci est intéressant pour caractériser les types d'objets car un seul caractère peut être retenu pour "représenter" une classe de caractères corrélés et la définition des types est ainsi facilitée.

Dans le cas présent, en appliquant la même tactique que pour les objets, on doit, en première analyse, retenir l'intégralité des coupures proposées car, bien que les coupures 3, 4 et 5 créent des classes à un seul caractère, la coupure 6 crée deux classes de 4 et de 2 objets. Les classes de caractères distinguées sont alors données par le tableau 8.

Classe 1	Classe 2	Classe 3	Classe 4	Classe 5	Classe 6	Classe 7
Nb de collatéraux, Nb. Absid. Transept	Long. Chœur/tot., Larg. trans./tot., Long. Narthex/nef, Nb absid. Chœur	Larg. Tot/long. Nef	Long. Nef/tot., Larg. nef/tot.	(Région)	Larg./long. Chœur	Nb travées nef

Tableau 8 : classes de caractères en première analyse

¹ L'appréciation du sens à donner à « petites » doit prendre en compte le nombre total d'articles considérés. Une classe à deux objets est petite dans une matrice décrivant 50 objets mais ne l'est pas dans une matrice à 5 objets.

L'examen visuel des plans (cf. annexe) et de la matrice aide à compléter cette analyse par les commentaires suivants.

- la première classe peut être représentée par le caractère “ Nb de collatéraux ” qui traduit, en fait, la présence ou l'absence de collatéraux ;
- la deuxième classe peut être représentée par le caractère “ larg. Trans/tot. ”,
- dans la classe 4, seul le caractère “Long. Nef/tot.” a un sens car “larg. Nef/tot.” est surtout lié à l'absence ou à la présence de transept,
- le caractère “région”, qui n'a pas servi à classer les plans, est seul dans sa classe ; ceci montre un résultat intéressant : la région n'est pas étroitement corrélée à un ensemble de caractères en petit nombre ni même à un caractère unique ; le classement obtenu traduit donc une corrélation entre la région et un ensemble varié de caractères.

Finalement, le classement effectué conduit à la caractérisation typologique synthétique présentée par le tableau 9.

Plans→ ↓ Traits discriminants	Classe 1 (Provence 1)	Classe 2 (Provence 2)	Classe 3 (Trinité)	Classe 4 (Auvergne)	Classe 5 (Normandie)	Classe 6 (Alsace)
Collatéraux	Non	Oui	Oui	Oui	Oui	Oui
Larg. Trans./tot.	Pas de trans.	Faible	Fort	Moyen- Fort	Moyen- Fort	Faible
Larg. Tot./long. nef	Faible	Faible-moyen	Faible	Moyen-fort	Faible	Fort
Long nef/tot.	Fort	Moyen	Moyen	Faible	Faible	Faible
Larg./long. Chœur	Fort	Varié	Faible	Moyen-faible	Moyen-fort	Faible-moyen
Nb de travées nef	Faible	Faible	Fort	Moyen-faible	Moyen-fort	Faible-Moyen

Tableau 9 : synthèse typologique

En résumé, l'analyse fait apparaître un plan inclassable, celui de la Trinité de Caen qui se rapprocherait plutôt des plans auvergnats, et 5 types régionaux de plans caractérisés chacun par une signature composée de l'ensemble des valeurs des 6 traits caractéristiques pertinents indiqués dans le tableau 9.

En conclusion, à l'exception de la Trinité et à condition de considérer que Haute et Basse Normandie forment une seule région, cette étude met en évidence l'existence d'un style régional de plan, en ouvrant la voie à l'interprétation.

CONCLUSION GENERALE

L'utilisation d'un programme tel que celui présenté ici suppose un important travail avant et après le traitement purement informatique.

Avant, il faut analyser le matériel pour trouver une bonne grille de description convenant à la fois au matériel et aux questions posées. Il faut aussi se persuader qu'un mauvais conditionnement de la grille de description ne peut pas être compensé par l'emploi d'un algorithme, quelles que puissent être ses performances.

Après, le travail d'interprétation est essentiel et l'exemple détaillé montre que les conclusions de cette interprétation doivent se développer à partir des résultats bruts fournis par le programme.


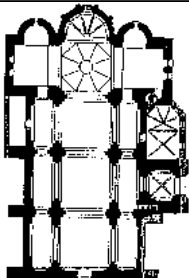
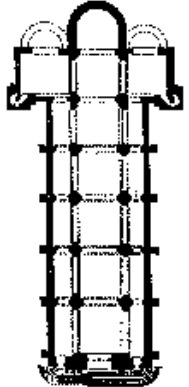
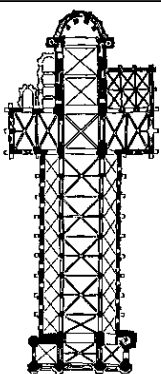
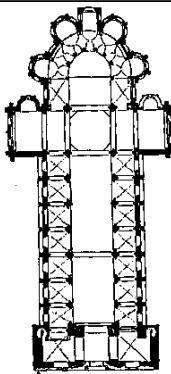
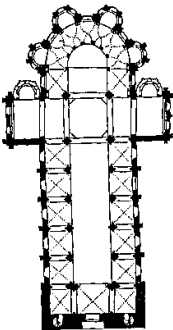
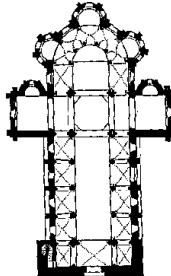
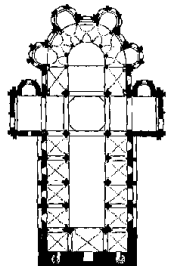
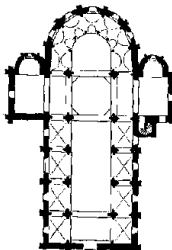
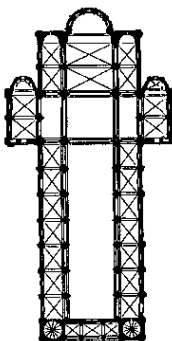
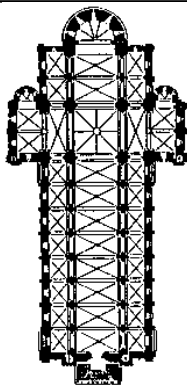
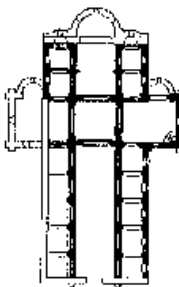
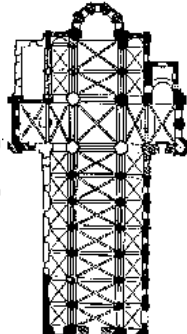
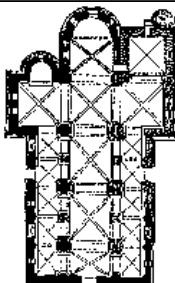
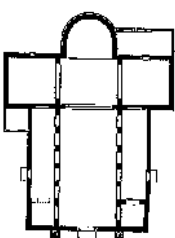
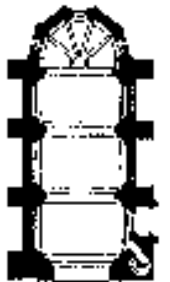
Ainsi, le passage sur ordinateur ne peut ni dispenser d'une bonne connaissance du matériel, ni fournir directement un résultat définitif.

Néanmoins, il constitue à la fois un intermédiaire et un guide, et il peut s'avérer irremplaçable dans les cas où l'on doit exploiter un grand nombre de données.

REFERENCES

- [0] N. FERNANDEZ, G. FERNANDEZ, "MATRIX, un programme de classification rapide en BASIC", Archéologues et ordinateurs, 1985.
- [1] S. PINSON et J.L. CHANDON, Analyse typologique, théories et applications, Paris, Masson, 1981.
- [2] B. JAULIN, "Mesure de la ressemblance en archéologie", Colloque International du C.N.R.S. sur l'emploi des calculateurs en archéologie, Marseille, 7-12 avril 1969, pp.343-356.
- [3] K. SPARCK-JONES, "*The evaluation of archaeological classifications*", Colloque C.N.R.S., Marseille 7-12 avril 1969, pp 245-274.
- [4] S. CLEUZIQU et J.P. DEMOULE, "Enregistrer, gérer, traiter les données archéologiques", l'Archéologie aujourd'hui, Hachette, Paris, 1980, pp.87-150.
- [5] M.S. LAGRANGE, Analyse sémiologique et Histoire de l'Art, Paris, Éditions Klincksieck, 1973 (Voir chap. VI : traitement sur ordinateur", pp.60 et svtes.).
- [6] J. BERTIN, Sémiologie graphique, Gauthier-Villars, Paris, 1973, p.260.

ANNEXE : Planche « Regroupement des plans par classes »

Classe 1 Provence	Classe 2 Provence	Classe 3 Trinité	Classe 4 Auvergne	Classe 5 Normandie	Classe 6 Alsace
 <p>Avignon↑</p>	 <p>St-Paul-3-ch. ↑</p>  <p>Ste-Trophime↑</p>	 <p>Trinité Caen↑</p>	 <p>↑Issoire</p>  <p>Clermont↑</p>  <p>↑St-Nectaire</p>  <p>Orcival↑</p>  <p>St-Saturnin↑</p>	 <p>↑St-Nicolas</p>  <p>Boscherville↑</p>  <p>↑Bernay</p>  <p>Lessay↑</p>	 <p>Rosheim↑</p>  <p>Eschau↑</p>
 <p>St-Restitut↑</p>					

