

THESIS / THÈSE

MASTER EN SCIENCES INFORMATIQUES

Analyse quantitative de gels d'électrophorèse bidimensionnelle par traitement d'image

Henin, Vincent

Award date:
1989

Awarding institution:
Université de Namur

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

ANALYSE QUANTITATIVE DE GELS
D'ELECTROPHORESE BIDIMENSIONNELLE
PAR TRAITEMENT D'IMAGE.

Vincent HENIN

Promoteur :
Mme N. Noirhomme-Fraiture.

Année académique 88 - 89

Mémoire présenté pour
l'obtention du diplôme
de Licencié et Maître
en Informatique.

Je tiens à exprimer mes remerciements à Madame Noïrhomme qui a bien voulu diriger ce mémoire, et dont les conseils furent précieux.

Je tiens également à remercier Monsieur Remacle qui m'a permis de réaliser ce travail.

Mes remerciements vont aussi à tous ceux qui, de près ou de loin, m'ont aidé et encouragé. Je pense particulièrement à Dominique, Elise et Simon à qui je dédie ce mémoire.

table des matières.

	Page
1. Introduction : la technique d'électrophorèse 2D.	2
2. Le matériel disponible.	5
3. Les fonctionnalités attendues du système.	6
4. Architecture logicielle du système.	8
5. Extraction de l'information présente dans l'image: module extraction.	10
5.1. Détermination du background.	10
5.1.1. Inversion de l'image.	11
5.1.2. Lissage de l'image.	11
5.1.3. Calcul du background ligne par ligne.	11
5.1.4. Lissage du background.	12
5.2. Quantification des spots.	14
5.2.1. Modélisation.	14
5.2.2. Constitution d'une tache.	15
5.2.3. Soustraction du background d'une tache.	15
5.2.4. Résolution des spots d'une tache.	16
5.2.4.1. Recherche des centres de spots au sein d'une tache.	16
5.2.4.2. Approximation des spots d'une tache par le modèle de Gauss, selon l'algorithme de Schroeder.	19
5.2.4.3. Intégration des spots d'une tache.	25
5.3. Représentation de l'information extraite.	25
5.4. Sauvegarde de l'information extraite.	26
6. Représentation des données sur l'ecran video: module video.	26
7. L'accès aux données: module accès-BD.	28
7.1. Organisation des données.	28
7.2. Les primitives de base.	29
7.3. Les procédures offertes par le module accès-BD.	30

8. L'implémentation des fonctionnalités du système: module fonction.	31
8.1. Principe.	31
8.2. Gestion des artefacts et intégrité des données.	32
8.3. Numérotation automatique des spots.	32
9. Ce que le système propose: module menu.	33
10. Discussion.	38
10.1. Performances de l'algorithme de quantification des spots.	38
10.1.1. Complexité des calculs.	38
10.1.2. Solutions proposées.	40
10.1.2.1. Seuillage des taches.	40
10.1.2.2. Coupure des taches.	40
10.1.2.3. Installation d'un coprocesseur arithmétique.	42
10.1.2.4. Simplification des calculs.	42
10.2. L'organisation des données.	42
10.3. Perspectives de développement.	43
11. Bibliographie.	47
12. Annexes.	49

1. INTRODUCTION : LA TECHNIQUE D'ELECTROPHORESE 2D.

L'électrophorèse à deux dimensions est une technique qui permet de séparer les protéines présentes dans un échantillon biologique, sur base de deux de leurs propriétés: le point isoélectrique et le poids moléculaire (O' Farrell, 1975).

Le point isoélectrique (pI) d'une protéine représente le niveau d'acidité, mesuré en terme de pH, auquel la solution qui la contient doit être pour que cette protéine présente une charge électrique globale nulle.

Le poids moléculaire (pM) est quant à lui l'expression en "kilo-daltons" de la taille d'une protéine.

Concrètement, l'échantillon à analyser est déposé à l'extrémité d'un gel de polyacrylamide dans lequel existe un gradient de pH. En soumettant ce gel à un champ électrique, chaque protéine va migrer jusqu'à ce que sa charge électrique devienne nulle, c-à-d jusqu'à l'endroit où le pH du gel est égal à son pI (cette technique de séparation porte le nom d'électrofocusing). Une seconde migration est ensuite réalisée, perpendiculairement à la première, dans un gel à pH constant mais de densité croissante, c-à-d dont les "mailles" sont de plus en plus fines à mesure que l'on s'éloigne du gel d'électrofocusing. De ce fait, les protéines migrent d'autant plus vite et plus loin dans ce gel que leur taille est petite.

Ainsi, la position finale d'une protéine dans un gel d'électrophorèse 2D (bidimensionnelle) lui est propre puisque directement liée à son pI et son pM.

Cette position est révélée soit par une coloration adéquate des protéines, soit par exposition d'un film photographique sur le gel si celui-ci contient des protéines préalablement marquées à l'aide d'un isotope radioactif. A chaque protéine présente dans l'échantillon analysé correspond donc dans le gel un spot dont l'intensité de coloration et les dimensions sont proportionnelles à la quantité de protéine présente à cet endroit.

Lorsque la technique d'électrophorèse 2D est appliquée à un mélange de protéines, comme par exemple un surnageant de cellules en culture in vitro, le gel présente une multitude (environ mille) spots qui identifient chacun une protéine.

Un exemple de gel d'électrophorèse 2D est repris à la figure 1.

La quantité d'informations présentes dans un gel d'électrophorèse 2D le rend donc inexploitable manuellement même si souvent l'expérimentateur ne s'intéresse qu'à un nombre réduit de protéines, en évaluant leur quantité à travers différentes conditions expérimentales par exemple.

C'est la raison pour laquelle l'Unité de Biochimie Cellulaire souhaitait voir développer un système qui génère automatiquement l'information présente dans un gel d'électrophorèse 2D sous une forme utilisable par l'expérimentateur.



Fig. 1

Un exemple de gel d'électrophorèse 2D

2. LE MATERIEL DISPONIBLE.

L'unité de Biochimie Cellulaire dispose du matériel suivant:

hardware:

- Une camera fournissant des images digitalisées (binarisées) d'une taille maximale de 512 x 512 points (pixels). L'intensité lumineuse de chaque point est codée sur 1 byte, ce qui représente 256 niveaux de gris allant du noir (0) au blanc (255). La mémorisation d'une image complète nécessite donc un minimum de 262 Kbytes de mémoire.
- Un Olivetti PC M24 avec un disque dur de 20 Mbytes et une carte vision supplémentaire qui constitue la "mémoire image", pouvant contenir 4 images de 512 x 512 points.
- Un moniteur pour la visualisation des images.

software:

- Parallèlement au matériel de prise de vue, le laboratoire a acquis un logiciel de traitement d'image, PCSCOPE, qui fournit un certain nombre de fonctions de base:
 - enregistrement d'une image,
 - détermination / modification de l'intensité lumineuse d'un point,
 - soustraction point par point de deux images,
 - etc.

Ce logiciel ne permet pas, tel quel, l'analyse de gels d'électrophorèse 2D. Par contre, la bibliothèque de fonctions (écrites en langage C) qu'il offre sera utilisée pour développer l'application qui nous intéresse.

3. FONCTIONNALITES ATTENDUES DU SYSTEME.

La technique d'électrophorèse 2D n'a été que récemment développée à l'Unité de Biochimie Cellulaire. L'expérience en interprétation de gels 2D est donc très limitée. Toutefois certaines fonctionnalités ont pu être précisées, que le système devrait offrir à partir de l'image d'un gel:

- obtenir les renseignements suivants pour un spot déterminé:
 - l'intensité par rapport à l'ensemble des spots du gel,
 - l'intensité par rapport à un spot de référence,
 - les pI et pM estimés,
 - le numéro d'identification, si un tel numéro a été attribué,
 - le nom (s'il est connu);
- attribuer un numéro à un spot déterminé;
- attribuer automatiquement un numéro à chaque spot du gel, ce numéro devant donner une idée de son pI et son pM.
- attribuer un nom à un spot déterminé;
- désigner un spot comme étant le spot de référence pour le calcul des intensités;
- supprimer un spot considéré comme un artéfact (cette opération doit être réversible);
- attribuer à un endroit ou un spot un pI et un pM connus par l'utilisateur.

Pour ces fonctions, le spot concerné doit pouvoir être désigné soit

- par la souris,
 - par son numéro manuel,
 - par son numéro automatique,
 - par son nom.
- obtenir le pI et le pM d'un endroit désigné dans le gel.

L'information de départ étant l'image binarisée d'un gel, le traitement à accomplir sur cette information pour réaliser les fonctionnalités attendues présente deux aspects fondamentaux:

- 1) L'extraction de l'information relative aux différents spots présents dans le gel (processus non interactif).
- 2) La gestion de cette information, via un interface adéquat, de façon à réaliser les fonctionnalités proprement dites (processus interactif).

4. ARCHITECTURE LOGICIELLE DU SYSTEME.

Les différents modules du système ainsi que les relations logicielles qui existent entre eux sont représentés à la figure 2

Le module **fonction** offre les services qui correspondent aux différentes fonctions révélées par l'analyse fonctionnelle.

Le module **extraction** offre les services permettant l'extraction de l'information présente dans l'image d'un gel 2D source.

Le module **accès_BD** offre tous les services d'accès aux données demandés par les deux modules précédents. Ce module rend le système indépendant d'un SGBD particulier.

Le module **vidéo** offre les services qui permettent la visualisation de différents types d'informations à l'écran vidéo.

Le module **PCSCOPE** offre les services de base relatifs au traitement d'image, à usage général, c-à-d indépendants d'une application particulière.

La liste des différentes primitives, ou procédures, offertes par ces modules est reprise en annexe, de même que les spécifications de ces procédures.

Le module **menu** joue quant à lui le rôle de coordinateur en appelant les fonctions sélectionnées par l'utilisateur. Il réalise également l'initialisation de la carte-vision lors de la mise en marche du système.

Chacun de ces modules est implémenté sous forme d'une unité de compilation séparée.

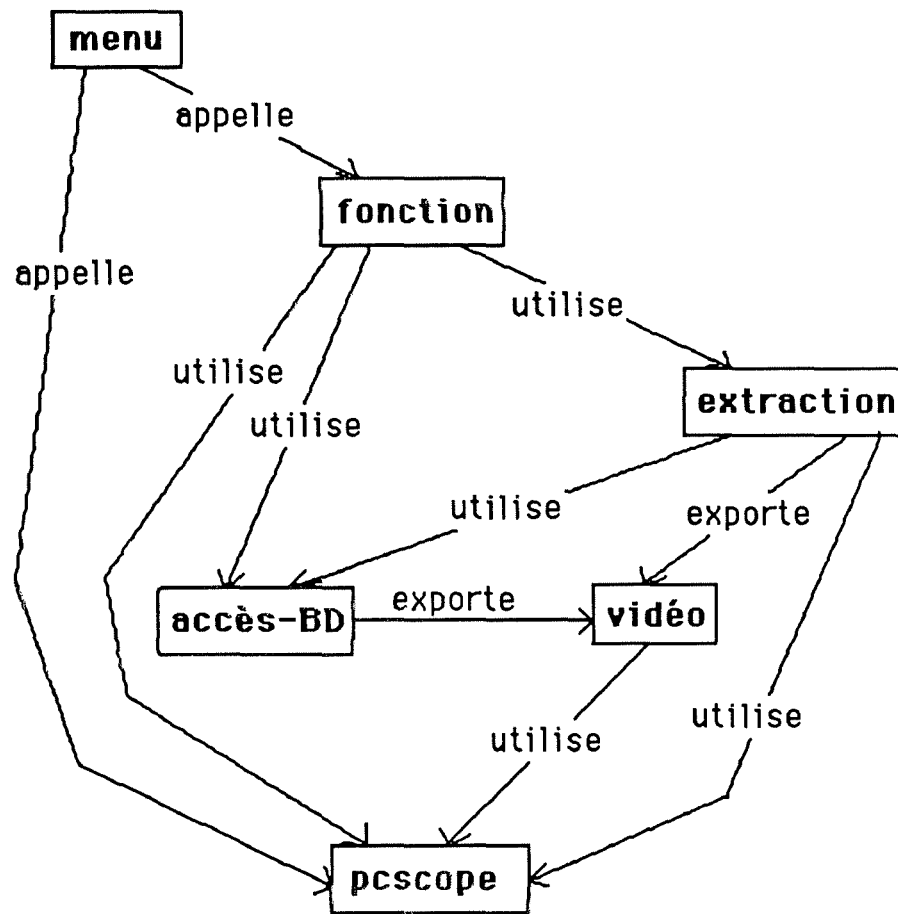


Fig. 2

Architecture logicielle du système.

5. EXTRACTION DE L'INFORMATION PRESENTE DANS L'IMAGE : module extraction.

La figure 1 montre clairement que le gel présente une certaine intensité de coloration en dehors des spots de protéines. Cette coloration, qui constitue le background, devra donc être soustraite de la coloration des spots avant la quantification de ces derniers au moyen d'un modèle adéquat.

5.1. Détermination du background.

Le background n'étant pas homogène sur l'ensemble du gel, il doit être recalculé en chaque point de l'image. Différentes stratégies ont été développées dans le cadre de systèmes plus ou moins sophistiqués. Les plus complexes (Skolnick 1982, Vincens 1986) font appel à la morphologie mathématique, et le background y est déterminé au moyen d'éléments dits structurants. Une telle approche nécessite l'utilisation d'un minicomputer doté d'un processeur parallèle. Dans des systèmes un peu moins évolués (Anderson 1981) le background au niveau d'un point est calculé à partir d'une zone plus ou moins étendue, englobant ce point, ou à partir de plusieurs zones distribuées autour du point (Lemkin 1981). Ces systèmes utilisent des minicomputers équipés d'array processors. Dans un des premiers systèmes développés (Garrels 1979 (1984)), l'intensité lumineuse du gel est fournie point par point à l'aide d'un scanner qui parcourt toute la surface du gel, ligne après ligne. Le background est calculé simultanément, sur base d'une (dé)croissance plus ou moins forte entre deux points successifs, et directement soustrait de l'intensité du dernier point lu.

Compte tenu du matériel disponible, la solution imaginée pour notre système se situe entre ces deux derniers types d'approches.

La détermination du background est réalisée en quatre étapes.

5.1.1. Inversion de l'image.

Après que l'image ait été chargée dans le quadrant n°0 de la mémoire image, les 256 niveaux de gris sont inversés de manière à obtenir un background absolu de zéro et non plus 255 (la couleur de fond du gel tendant dès lors vers le noir, les spots de faible intensité apparaissent beaucoup plus clairement à l'utilisateur). Cette opération correspond à une fonction offerte par PCSCOPE. L'image résultante est mémorisée dans le quadrant n°1 de la mémoire image.

5.1.2. Lissage de l'image.

Le calcul du background étant basé sur l'évolution de l'intensité lumineuse entre deux points adjacents, les petites variations de gris dues au bruit présent lors de la prise de vue doivent être "gommées". L'image du gel est donc lissée en recalculant l'intensité de chaque point en fonction de celle de ses voisins selon le "masque" suivant:

1	2	1
2	4	2
1	2	1

L'intensité du point situé au centre du masque est égale à la somme des intensités des points compris dans ce masque, chacune multipliée par le coefficient qui lui correspond, le tout divisé par la somme des coefficients (= 16). Cette opération correspond à la fonction "filtrage passe-bas" offerte par PCSCOPE.

5.1.3. Calcul du background ligne par ligne.

Tant que l'augmentation d'intensité entre deux points successifs (c-à-d la pente) reste en dessous d'une certaine valeur (*seuilpente*), le background prend la valeur de l'intensité

du dernier point lu dans l'image inversée. Cette valeur de background est mémorisée dans le quadrant n°2 de la mémoire image au niveau du point qui y occupe la même position relative que celle du point lu dans le quadrant n°1 (image inversée). Lorsque la pente devient supérieure à *seuilpente*, ce qui correspond à l'entrée dans un pic (un spot au niveau de l'image), le background conserve la dernière valeur précédemment lue, et ce tant que l'intensité lue lui demeure supérieure ou égale. Lorsque l'intensité lue devient inférieure à la valeur du background, celui-ci est mis à jour en prenant cette dernière valeur.

Après avoir effectué une première passe de gauche à droite et de haut en bas, la lecture point à point réalise une seconde passe de droite à gauche (toujours de haut en bas) de manière à déterminer correctement le background lorsque celui-ci présente globalement un gradient dans un sens ou dans l'autre. Lors de cette seconde passe, la valeur de background mémorisée dans le quadrant n°2 est la plus grande des deux valeurs calculées en ce point.

Lorsque ce scanning est terminé, le quadrant n°2 de la mémoire image contient le background calculé en chaque point de l'image.

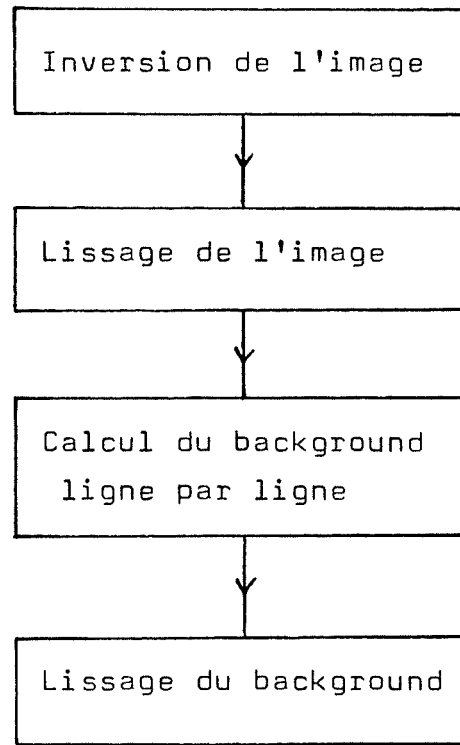
5.1.4. Lissage du background.

Le background ayant été calculé indépendamment sur chaque ligne, il peut en résulter une certaine irrégularité d'une ligne à l'autre. Cet artéfact est éliminé en réalisant un lissage du quadrant n°2 par la même fonction que celle utilisée ci-dessus.

Ces quatre opérations sont réalisées par la procédure **background**.

La figure 3 résume le processus de détermination du background ainsi que le contenu final de la mémoire image.

A.



B.

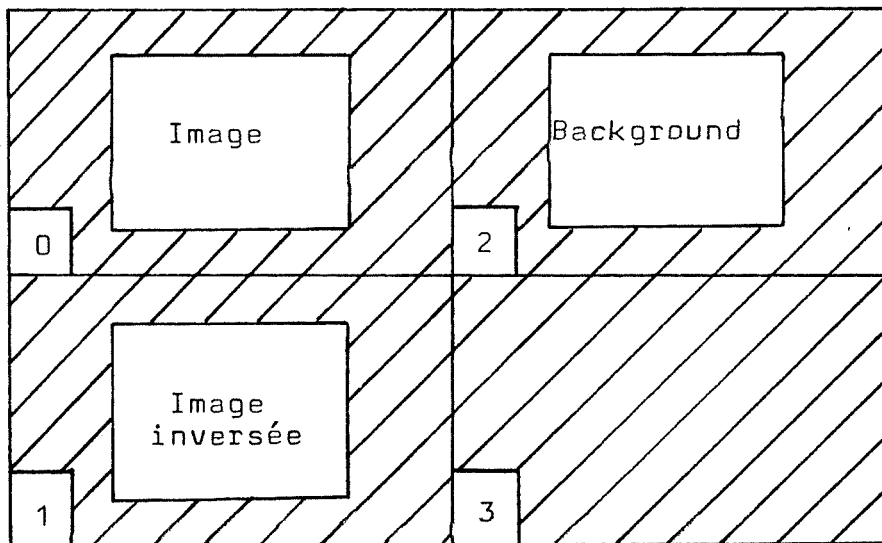


FIG 3.

A: le processus de détermination du background.

B: l'état de la mémoire image lors de la terminaison de la procédure Background.

5.2. Quantification des spots. (procédure quantifspot)

5.2.1. Modélisation.

L'examen de la figure 1 montre que les spots de protéines sont rarement isolés. Souvent ils se chevauchent plus ou moins fortement pour constituer des agglomérats ou taches formés d'un nombre variable de spots.

Bien que certains systèmes, parmi les plus sophistiqués (Lemkin 1981, Skolnick 1982), n'imposent pas un modèle particulier pour la résolution des spots, la grande majorité font appel tôt (Anderson 1981, Taylor 1979) ou tard (Vincens 1986) au modèle de courbe de Gauss à deux dimensions pour la quantification des spots, isolés ou non.

Il est en effet généralement admis que la distribution d'une protéine dans un gel d'électrophorèse peut être assez fidèlement représentée par une distribution normale à deux dimensions.

Le modèle adopté dans notre système étant la courbe de Gauss, la quantification des spots contenus dans une tache revient dès lors à analyser un mélange de distributions normales de probabilité.

Etant donné le nombre de taches élevé que l'on peut rencontrer dans un gel, l'algorithme effectuant la résolution des spots présents dans chaque tache doit avoir un temps d'exécution limité si l'on veut obtenir un temps raisonnable pour le traitement complet d'un gel.

Dans l'algorithme proposé par Young (Young 1970) le nombre de distributions supposées présentes dans le mélange peut être fixé arbitrairement au départ. En cours d'itération, un facteur de pondération associé à chaque distribution tend vers zéro si la distribution qui lui correspond n'est pas nécessaire pour rendre compte de la distribution du mélange. Un tel algorithme nécessite cependant un nombre élevé d'itérations pour estimer correctement ces facteurs de pondération et les paramètres des courbes de Gauss. Il ne peut donc être utilisé dans le cadre qui nous intéresse.

L'algorithme retenu pour notre système (Schroeder 1975) analyse un mélange de distributions sur base du nombre de distributions qui lui est soumis au départ. Il a cependant l'avantage de converger rapidement si les paramètres attribués initialement aux courbes de Gauss du modèle sont relativement proches des valeurs optimales.

Le processus de quantification des spots d'une image consiste, dans notre système, à y délimiter les taches, et pour chacune d'elles, en soustraire le background et résoudre les spots qui la constituent via la modélisation par un mélange de distributions normales 2D.

5.2.2. Constitution d'une tache.

L'image du gel, présente dans le quadrant n°1, est parcourue point par point jusqu'à obtention d'un point dont l'intensité est supérieure ou égale à celle du background en cet endroit, plus un certain seuil (*seuilbkgrd*); la valeur du background en un point de l'image est obtenue par lecture du point correspondant dans le quadrant n°2. Lorsqu'un tel point est obtenu, les points qui lui sont adjacents et qui vérifient la même propriété sont également collectés, et ainsi de suite avec les points adjacents à ces derniers (si dans un graphe deux sommets reliés par une arête représentaient deux points adjacents et d'intensité suffisante, une telle façon de procéder correspondrait à la recherche des composantes simplement connexes du graphe). L'ensemble des points ainsi collectés à partir d'un premier point constitue une tache.

5.2.3. Soustraction du background d'une tache. (procédure **nivtache**)

Une fois la tache constituée, son background moyen est calculé sur base des valeurs mémorisées dans les points correspondants du quadrant n°2. Cette valeur est alors déduite de l'intensité de chaque point qui constitue la tache. En procédant de la sorte, le background soustrait à chaque tache est calculé indépendamment d'une tache à l'autre; de plus les taches ne subissent aucune déformation puisque pour chacune d'entre elles c'est une valeur unique et constante qui est soustraite en chaque point.

5.2.4. Résolution des spots d'une tache. (procédure **normalisation2D**)

La première phase du processus de normalisation d'une tache consiste à rechercher le plus précisément possible les positions des centres des spots qui la constituent, de façon à ce que l'algorithme qui approxime ensuite ces spots par des courbes de Gauss converge rapidement.

5.2.4.1. Recherche des centres de spots au sein d'une tache.

L'algorithme qui réalise cette opération s'inspire de celui développé dans Garrels 1979. Brièvement, celui-ci réalise la détection et l'intégration des pics présents sur chaque ligne de l'image. Pour chaque ligne analysée une liste est créée reprenant les positions des centres des pics qui y ont été détectés, et leur intensité intégrée. Ces listes sont ensuite réorganisées de manière à regrouper dans de nouvelles listes les centres de pics adjacents et situés pratiquement sur une même colonne dans l'image. Chaque liste contient dès lors les centres de pics (et leur intensité intégrée) appartenant à un même spot, ou appartenant à plusieurs spots situés sensiblement sur la même colonne dans l'image. La détection des centres de pics et leur intégration sont réalisées sur le contenu de chacune de ces nouvelles listes, générant ainsi les positions des centres de spots (cf. ci-dessous) et leur intensité intégrée.

Dans notre système, seule la recherche des positions des centres de spots nous intéresse, d'autant plus que la quantification (intégration) des spots selon Garrels n'essaie pas de résoudre des pics qui se chevauchent en les approximant par des courbes de Gauss.

Une procédure **detectcentre** a donc été développée, qui réalise la détection des centres de pics sur toute suite de points qui lui est soumise.

La figure 4 illustre son principe de fonctionnement.

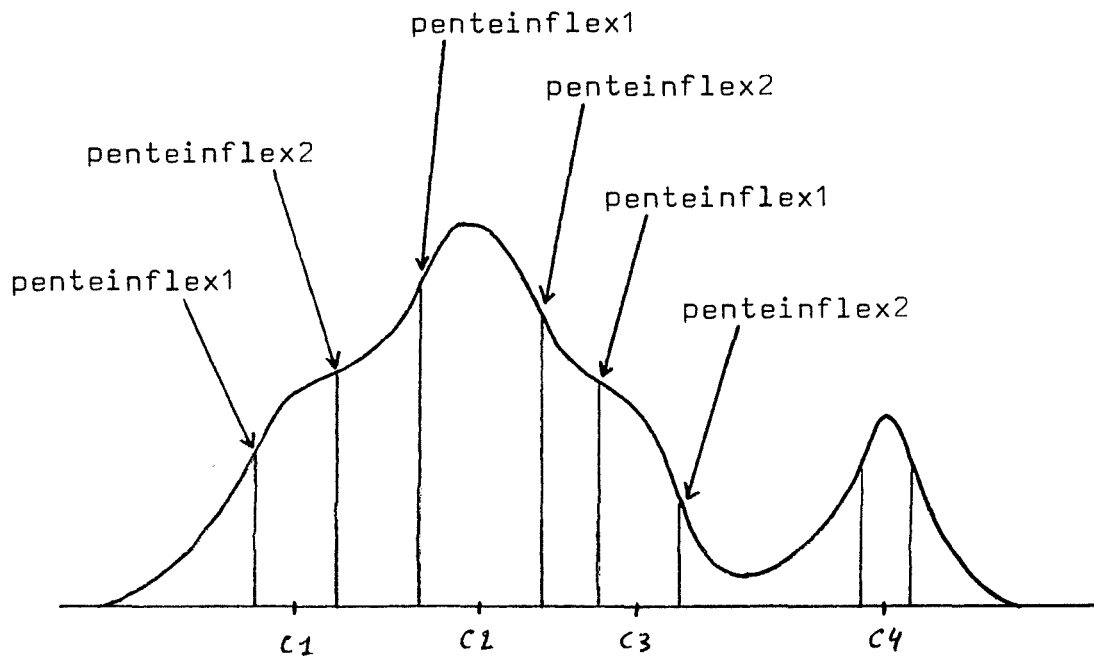


FIG 4 .

Illustration du principe de fonctionnement de la
procédure Déetectcentre.

En se basant sur les changements de signe de la dérivée seconde la procédure détecte les points d'inflexion. Si un premier point d'inflexion est suivi par un second alors qu'aucun sommet n'a encore été atteint entre les deux, ce deuxième point d'inflexion traduit le chevauchement du premier pic par un second à condition que la différence entre les pentes mesurées en chacun des deux points d'inflexion soit suffisamment grande ($\text{penteinflex2} \leq \text{factm}(d) \times \text{penteinflex1}$ avec $\text{factm}(d) = 0,7$), ceci afin d'éviter de scinder la courbe en un trop grand nombre de pics. La position du centre d'un pic ainsi que son intensité sont calculés à mi-distance entre deux points d'inflexion.

Si l'intensité calculée au centre du pic est supérieure à un certain seuil (*seuilpic*), les résultats sont placés dans une liste qui va recueillir ces informations pour chaque centre de pic détecté dans la courbe.

Si deux points d'inflexion sont séparés par un sommet, le centre du pic est estimé d'office à mi distance entre ces deux points. Le facteur $\text{factm}(d)$ et le seuil d'intensité (*seuilpic*) à dépasser pour qu'un pic soit considéré en tant que tel déterminent le niveau de sensibilité de la méthode.

Une tache étant représentée dans notre système par une liste de points qui la constituent, une phase préliminaire consiste à reproduire cette tache dans le quadrant n°3 de la mémoire image.

. Détection des centres de pics dans chaque ligne de la tache.
(procédure **déTECTcentre**)

Chaque ligne du quadrant n°3 qui contient au moins un point de la tache est soumise à la procédure **déTECTcentre** qui retourne pour chacune d'elles une liste des centres de pics détectés, avec leur coordonnée et leur intensité. Ces listes sont donc triées par numéro de ligne.

. Réorganisation des centres de pics.

En comparant les positions des centres de pics détectés sur deux lignes successives, la procédure **réorgcentre** regroupe dans de nouvelles listes les pics dont les positions sont suffisamment proches l'une de l'autre, c-à-d ne s'écartant pas de plus de *seuilreorg* colonnes (typiquement, *seuilreorg* = 1). En réalisant cette opération sur l'ensemble des listes correspondant aux lignes de l'image, les nouvelles listes ainsi générées contiennent les données relatives aux centres de pics appartenant à un même spot ou à plusieurs spots se chevauchant et situés sensiblement sur une même verticale.

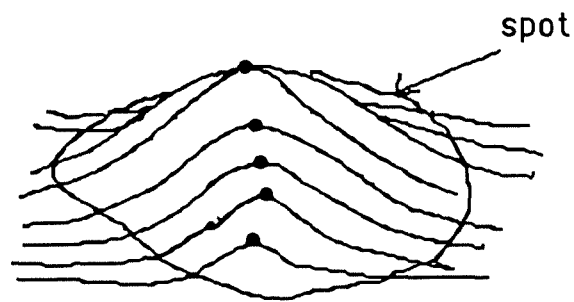
. Détection des centres de spots.

Le contenu de chacune de ces nouvelles listes est soumis à la procédure **détecentre** pour y déterminer les centres de pics. Les points sur lesquels s'effectue cette détection étant en fait les sommets de "tranches" réalisées ligne par ligne au sein d'un même spot, le sommet qui y est détecté est donc le sommet, ou centre, de ce spot (voir figure 5). Si plusieurs centres de pics sont détectés dans la liste, ceux-ci correspondent aux centres (sommets) de spots qui se chevauchent et sont quasi-alignés verticalement.

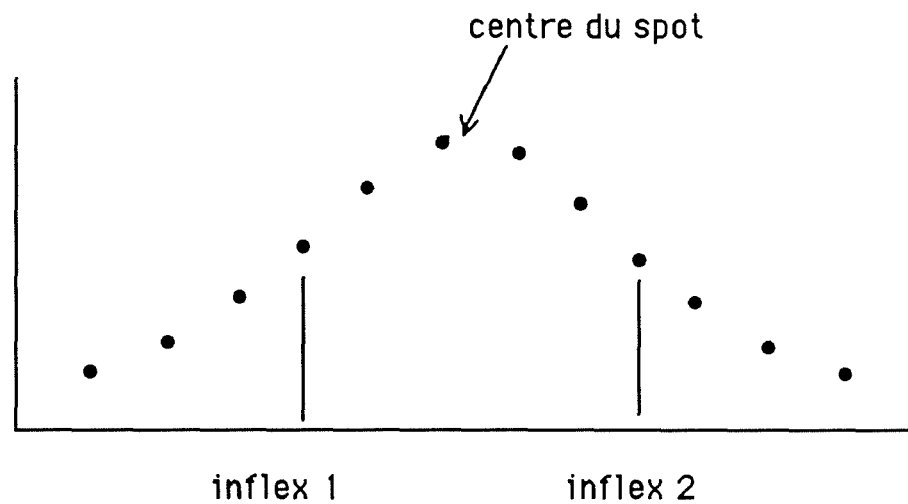
En réalisant la détection des centres de pics sur chacune de ces listes, la procédure **normalisation2D** détermine donc la position des centres de tous les spots présents dans une tache.

5.2.4.2. Approximation des spots d'une tache par le modèle de Gauss selon l'algorithme de Schroeder. (procédure fitting)

Si l'on modélise les spots présents dans une tache par des courbes de Gauss, résoudre ces spots qui se chevauchent équivaut à déterminer les distributions normales présentes dans un mélange de distributions. L'algorithme proposé par Schroeder permet d'estimer à partir d'un échantillon fini les paramètres des k distributions intervenant dans le mélange (k étant supposé connu).



Détection des centres de pics ligne par ligne : une liste par ligne; 1 centre de pic par liste.
Tous les • sont regroupés dans une nouvelle liste par la procédure réorgcentre.



Détection du centre du spot parmi les centres de pics

Fig. 5

Illustration de la détection du centre d'un spot.

Il s'agit d'un algorithme itératif détectant parallèlement une partition en classes de l'échantillon observé, et les distributions associées à ces classes. Chaque itération consiste à passer d'une partition à un ensemble de distributions en améliorant une fonction critère choisie qui est fonction du couple (partition, ensemble de distributions).

. Schéma général de l'algorithme.

Les paramètres d'entrée:

- l'échantillon à analyser (taille et dimension quelconques),
- la forme de la famille des lois de probabilité cherchées,
- le nombre (k) de composants recherchés dans le mélange.

Les paramètres de sortie:

- l'échantillon partitionné en k classes,
- les k lois de probabilité attribuées à ces classes.

Soient:

E = sous ensemble fini de $\mathbf{R}^q$, l'échantillon dont on dispose.

$(f_\lambda / \lambda \in L)$ la famille des densités de probabilité à laquelle on suppose que les distributions des différents composants appartiennent:

λ = paramètre réel ou vectoriel.

L = espace de définition de λ .

S'il s'agit de distributions gaussiennes, $\lambda = (\mu, V)$ où $\mu \in \mathbf{R}^q$ et V est la matrice de variance-covariance, $L = \mathbf{R}^q \times \mathbf{S}$ où $\mathbf{S}$ est l'espace des matrices réelles symétriques définies positives, de rang q .

L'algorithme se déroule de la manière suivante:

Partant d'une partition initiale quelconque $(P_1^0, P_2^0, \dots, P_k^0)$ $\mathbf{P}$ de E , ou bien d'un quelconque k -uplet de valeurs du paramètre inconnu $(\lambda_1^0, \lambda_2^0, \dots, \lambda_k^0) \in L^k$, on applique successivement les deux fonctions suivantes jusqu'à obtention d'éléments stables de L^k et $\mathbf{P}^k$;

$$\begin{aligned} \text{A) } f: L^k &\longrightarrow P^k, \\ L &\longrightarrow P, \end{aligned}$$

avec $L = (\lambda_1, \lambda_2, \dots, \lambda_k)$ et $P = (P_1, P_2, \dots, P_k)$ tels que
 $\forall i, 1 \leq i \leq k : P_i$ est l'ensemble des observations qui sont plus "proches" de la loi f_{λ_i} que de toutes les autres (avec choix du plus petit indice en cas d'égalité).

La distance d'une observation $x \in E$ à une distribution f_{λ} est mesurée par une fonction

$$\begin{aligned} D: E \times L &\longrightarrow \mathbb{R}^+, \\ (x, \lambda) &\longrightarrow D(x, \lambda) \end{aligned}$$

Une observation x peut être considérée comme "proche" de la valeur du paramètre inconnu, si la valeur $f_{\lambda}(x)$ est grande, ou bien si x est proche de l'espérance mathématique de f_{λ} .

$$\text{B) } g: P^k \longrightarrow L^k,$$

$$P \longrightarrow L, \text{ tel que}$$

$$\forall i, 1 \leq i \leq k : \lambda_i \text{ est tel que } \sum_{x \in P_i} D(x, \lambda_i) = \inf_{\lambda \in L} \sum_{x \in P_i} D(x, \lambda)$$

La fonction critère à améliorer (minimiser) à chaque j ème itération sera

$$W(L^j, P^j) = \sum_{1 \leq i \leq k} \sum_{x \in P_i^j} D(x, \lambda_i^j)$$

Le schéma de l'algorithme prend donc la forme suivante:

$$\begin{array}{ccc} & \xrightarrow{f} & \\ L^k & & P^k \\ & \xleftarrow{g} & \end{array}$$

Sa convergence est démontrée dans Schroeder 1975.

. Application de l'algorithme à notre problème.

$q=2$ puisque nous travaillons sur des images à deux dimensions.

Paramètres d'entrée:

- l'échantillon est l'ensemble des points constituant une tache:
un point de coordonnées x, y et d'intensité z équivaut à z observations du couple (x, y) ,
- la forme de la famille des lois de probabilité cherchées est la distribution gaussienne,

- le nombre de distributions recherchées dans le mélange est égal au nombre (k) de centres de spots détectés dans la tache.

Paramètres de sortie:

- les points de la tache partitionnés en k classes,
- les k distributions normales attribuées à ces classes.

L'implémentation de l'algorithme dans notre système est schématisée à la figure 6.

. Prépartition. (procédure **prépartition**)

L'ensemble des points constituant une tache est partitionné sur base de la distance euclidienne entre les points et chaque centre de spot détecté au préalable. Chaque point est affecté à la classe qui correspond au spot le plus proche (au spot de plus petit indice en cas d'égalité de distance).

. Paramétrisation. (procédure **paramètre**)

Pour chaque classe de points P_i , $1 \leq i \leq k$, la fonction de paramétrisation estime par le maximum de vraisemblance la moyenne μ_i et la matrice de variance-covariance V_i de la distribution normale qui lui est associée.

Ces estimations sont:

$$\mu_i = 1/n_i \sum_{x \in P_i} x \quad V_i = 1/n_i \sum_{x \in P_i} (x - \mu_i)(x - \mu_i)^T$$

où $n_i = \text{card}(\text{classe } P_i)$

Rem: si la tache est constituée d'un seul spot, seule la procédure **paramètre** est appelée.

. Partition. (procédure **partition**)

La partition des points se fait cette fois sur base de la distance associée à la matrice variance-covariance de chaque distribution. Pour chaque point de la tache, cette distance est calculée, entre le point et chaque moyenne de distribution.

$\forall i, 1 \leq i \leq k :$

$$P_i = \{x \text{ tache} \mid \log \det V_i + d_{V_i}^2(x, \mu_i) \leq \log \det V_j + d_{V_j}^2(x, \mu_j), \forall j \neq i\}$$

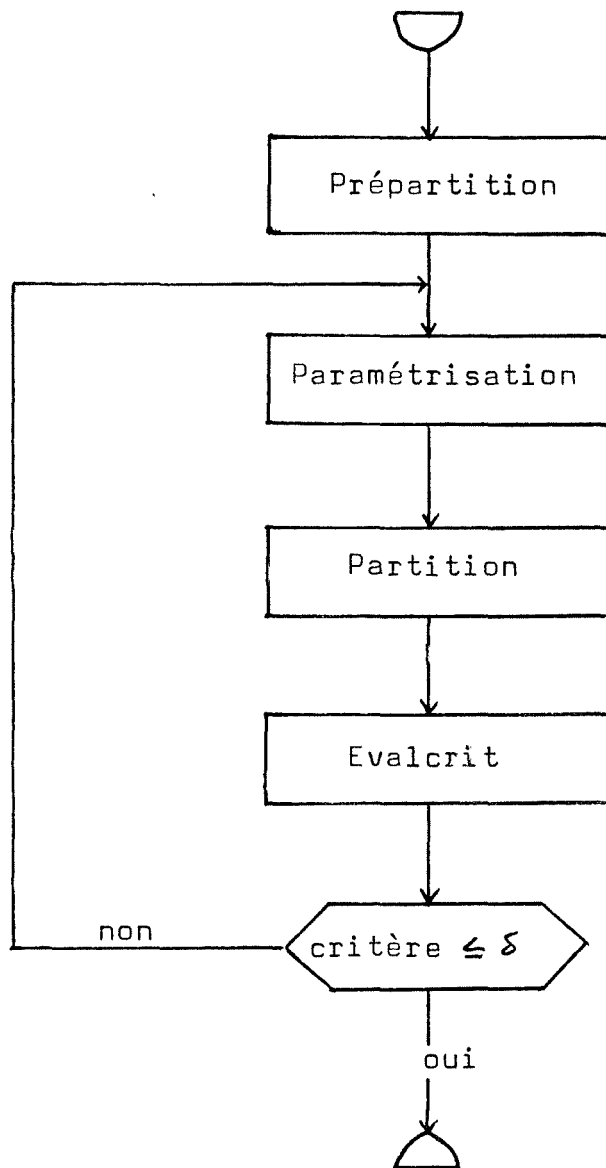


FIG 6 .

L'algorithme de Schroeder.

. Evaluation du critère. (procédure **évalcrit**)

L'algorithme converge vers un minimum local de la fonction critère

$$\text{crit} = \text{cste} + 1/2 \sum_{1 \leq i \leq k} (n_i \log \det V_i + \sum_{x \in P_i} d_{V_i}^2(x, \mu_i))$$

5.2.4.3. Intégration des spots d'une tache.
(procédure **intégration**)

Les spots d'une tache ayant été modélisés par autant de distributions normales 2D, la dernière étape du processus de leur quantification consiste à calculer, pour chaque point de la tache, la densité de probabilité au niveau de chacune de ces distributions de probabilité.

$$f_i(x) = 1/2 \pi (\det V_i)^{-1/2} \exp(- (x - \mu_i)^T V_i^{-1} (x - \mu_i)/2)$$

L'intensité lumineuse du point est alors répartie entre les différents spots proportionnellement aux densités de probabilité calculées.

Lorsque ce processus a été répété pour tous les points de la tache, l'intensité lumineuse calculée d'un spot correspond à l'intégration d'une fraction de l'intensité lumineuse de chaque point de la tache, cette fraction étant calculée sur base des distributions normales qui modélisent chacune un des spots de la tache. L'intensité intégrée d'un spot tient ainsi compte des chevauchements entre ce spot et ses voisins.

Rem: telle quelle, cette forme de quantification n'a guère de signification. Aussi, l'intensité d'un spot sera exprimée dans notre système par rapport à l'intensité globale, c-à-d la somme des intensités de tous les spots de l'image (gel), ou par rapport à l'intensité d'un spot de référence si celui-ci existe.

5.3. Représentation de l'information extraite.

Lorsqu'un spot a été intégré, celui-ci est remplacé sur l'écran du moniteur vidéo par une croix centrée sur la position de la moyenne de la distribution gaussienne 2D qui le modélise. Cette opération est réalisée par appel à la procédure **video_spot** du module vidéo.

5.4. Sauvegarde de l'information extraite.

L'extraction de l'information présente dans un gel donne lieu à la création de deux fichiers: un premier fichier, d'extension ".spt", recueille pour chaque spot résolu les paramètres suivants:

- la position du centre de la distribution 2D qui le modélise, par rapport au coin supérieur gauche de l'image,
- la variance en x (direction horizontale),
- la variance en y (direction verticale),
- l'intensité intégrée.

Cette opération est réalisée par appel à la procédure **addspot**. Une procédure **init_fichref** crée le second fichier, d'extension ".ref", et y mémorise les dimensions de l'image du gel, en nombre de lignes et de colonnes.

Ces deux procédures sont offertes par le module fichier.

6. Représentation des données sur l'écran vidéo: module vidéo.

La procédure **video_spot** permet de symboliser un spot sur l'écran vidéo par une croix centrée sur la position de la moyenne de la distribution gaussienne 2D qui le modélise. Les longueurs des branches horizontale et verticale de cette croix sont proportionnelles à la variance de la distribution dans chacune de ces deux directions. La taille de la croix, autrement dit la somme des longueurs de ces deux branches, est proportionnelle au logarithme de l'intensité intégrée du spot. Ce type de représentation graphique permet de distinguer facilement les spots même lorsqu'ils sont fortement imbriqués les uns dans les autres, tant en donnant une idée de leur forme originale (voir figure 7).

La procédure **clignoter_point** fait apparaître pendant un court laps de temps un carré gris centré sur la position de tout point qui lui est donné en entrée. Lors d'un appel à cette procédure, l'utilisateur peut voir à l'écran quel spot ou point est effectivement concerné par l'opération en cours (cf. primitives de base du module fichier).

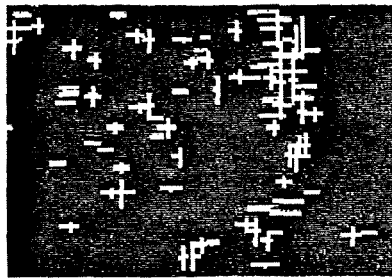


Fig. 7

Une partie d'un gel 2D source, et le gel synthétique qui lui correspond.

7. L'ACCES AUX DONNEES : module accès BD.

7.1. Organisation des données.

Les informations manipulées par le système sont de trois types:

- l'information "brute" de départ qui est l'image digitalisée d'un gel 2D.
- les paramètres relatifs aux spots qui ont été extraits de cette image.
- les points de référence (pl, pM, intensité) introduits par l'expérimentateur.

A chacun de ces trois types d'information correspond un type de fichier qui lui est propre:

- Fichier *.fim*.

Ce type de fichier contient entre autres l'information relative à la taille de l'image ainsi que l'intensité codée sur un byte de chaque point de l'image. La gestion d'un tel fichier se fait entièrement à l'aide de fonctions offertes par PCSCOPE. (Pour de plus amples informations, consulter le manuel d'utilisation PCSCOPE).

- Fichier *.spt*.

Un tel fichier contient les informations relatives aux différents spots qui ont été extraits d'une même image. A chaque spot correspond dans ce fichier un enregistrement qui reprend les valeurs de ses différents paramètres. La structure d'un tel enregistrement est reprise en annexe. Les enregistrements sont rangés dans ce fichier par ordre d'extraction des spots auxquels ils correspondent.

- Fichier *.ref*.

Ce fichier, qui est également créé lors du traitement de l'image d'un gel peut contenir l'information relative aux dimensions de cette image, aux étalons de pl, pM et intensité qui seraient désignés par l'utilisateur ainsi qu'un commentaire éventuel. La structure de ce fichier est reprise en annexe.

Ces trois types de fichiers constituent des fichiers à accès direct et/ou séquentiel.

Dans notre système, l'information relative à un gel 2D se retrouve donc dans trois fichiers distincts. Les noms de ces trois fichiers sont constitués du même préfixe, en l'occurrence le nom donné par l'utilisateur lors de l'enregistrement de l'image, tandis que leur extension caractérise leur type. Le respect de cette contrainte est garanti par les primitives offertes par le module fichier.

7.2. Les primitives de base.

L'analyse des différentes fonctions offertes par le système met en évidence quatre modes d'accès fondamentaux. En effet, que ce soit pour lire l'information relative à un spot où pour la modifier, il faut pouvoir retrouver cette information dans le fichier d'extension. *fim* sur base de la position du spot dans le gel, son nom, son numéro manuel ou son numéro automatique. Quatre primitives, ou procédures, ont donc été développées, qui réalisent chacune une de ces recherches.

Dans la procédure **chercher_spot_coordee**, l'information en entrée est la position du spot désignée à l'écran, au moyen de la souris, par l'utilisateur. Comme il est peu probable que celui-ci parvienne à positionner le curseur vidéo exactement au centre de la croix qui symbolise le spot (ce qui serait par ailleurs fastidieux), cette position n'est qu'approximative. Aussi, cette procédure recherche dans le fichier le spot dont le centre est le plus proche (distance euclidienne) de cette position. Cet algorithme ne considère toutefois que les spots qui sont effectivement visualisés puisque seuls ceux-ci sont susceptibles d'être désignés à l'écran par l'utilisateur. Dans cette recherche, l'ensemble du fichier *fim* est parcouru.

La procédure **chercher_spot_nom** reçoit en entrée le nom d'un spot et parcourt le fichier jusqu'à obtention d'un enregistrement dont le nom est identique.

La procédure **chercher_spot_numman** reçoit en entrée le numéro attribué par l'utilisateur à un spot donné, et parcourt le fichier jusqu'à ce qu'elle trouve un enregistrement dont le numéro manuel correspond au paramètre entré.

La procédure **chercher_spot_numaut** travaille comme la précédente mais sur base d'un numéro attribué au préalable par un algorithme de numérotation automatique.

Lorsque le spot recherché a été trouvé, un signal vidéo apparaît sur l'écran à son emplacement, de manière à ce que l'utilisateur puisse se rendre compte, par exemple, si le spot sur lequel il travaille est bien celui auquel il pense. Ceci est réalisé par appel à la procédure **clignoter_point** du module vidéo.

Ces quatre primitives retournent la position du début de l'enregistrement recherché dans le fichier *.fim*, via un pointeur vers le bloc de contrôle d'accès au fichier, qui contient entre autres l'indicateur de position.

Lorsque le fichier a été correctement positionné, des primitives de base d'un second type peuvent être utilisées pour consulter ou modifier l'information ainsi accédée. Ces primitives ont toutes un paramètre d'entrée commun : un pointeur vers le bloc descripteur de fichier.

Citons entre autres les possibilités de :

- lire le contenu de l'enregistrement (**lire_spot**)
- écrire le nom d'un spot
- écrire le numéro manuel d'un spot
- modifier le byte de visualisation
- copier le spot étalon d'intensité dans le fichier *.ref*.

7.3. Les procédures offertes par le module accès_BD.

En combinant les procédures de base deux à deux on obtient autant de primitives directement utilisables par les procédures du module fonction, ces dernières réalisant les diverses fonctionnalités du système. Par exemple, la procédure **consult_spot_nom** est obtenue par combinaison de **chercher_spot_nom** et **lire_spot**.

D'autres primitives sont également offertes, qui ne résultent pas de la combinaison de ces sous-procédures. Citons par exemple :

- le calcul de l'intensité globale d'un gel.
- la consultation des étalons d'intensité, pl ou pM.
- écrire l'information relative à un nouveau spot.
- etc,.....

Toutes ces primitives d'accès aux données, qui sont offertes par le module fichier aux modules de niveau supérieur, ont comme caractéristique commune de recevoir comme paramètre d'entrée (entre autres) non pas les noms des trois fichiers sur lesquels elles travaillent, mais simplement le nom donné par l'utilisateur à l'image (au gel) lors de son enregistrement.

8. L'IMPLEMENTATION DES FONCTIONNALITES OFFERTES : **module fonction.**

8.1. Principe.

Chaque procédure offerte par le module fonction réalise une des fonctionnalités proposées par le système. Ces procédures sont directement appelées lors de leur sélection, par l'utilisateur, dans les différents menus. Les algorithmes qui correspondent à ces procédures sont tous basés sur le même principe :

- saisie des informations que l'utilisateur doit fournir (si nécessaire).
- appel à une (aux) primitive(s) d'accès aux données offerte(s) par le module fichier.
- impression des résultats (si nécessaire) ou d'un message d'erreur si un problème s'est manifesté lors de l'accès aux données.

Les procédures qui sont susceptibles d'être utilisées de manière répétée par l'utilisateur se traduisent par des algorithmes itératifs résultant ainsi en une plus grande commodité d'emploi.. C'est le cas de toutes celles où l'information en entrée (ou une partie de cette information) est la coordonnée d'un point désigné par l'utilisateur au niveau de l'écran vidéo, à l'aide de la souris.

Exemples: **info_spot_souris**
attrib_nom_spot_souris
masq_artef_souris

L'exécution d'une telle procédure se termine lorsque l'utilisateur appuie sur le bouton droit de la souris.

8.2. Gestion des artéfacts et intégrité des données.

L'intensité relative d'un spot s'exprime soit par rapport à celle d'un spot de référence, soit par rapport à l'ensemble des spots de protéines présents dans le gel. Dans ce dernier cas, l'intensité globale du gel ne doit donc pas tenir compte des spots qui traduisent en réalité des artéfacts. Dès lors, les procédures qui permettent de masquer un spot, c-à-d le désigner comme artéfact, recalculent également l'intensité globale du gel par appel à la procédure **calcul_int_gel** du module fichier. C'est le cas des procédures **masq_artef_souris**, **masq_artef_nom**, **masq_artef_numman**, **masq_artef_numaut**.

Remarquons que la décision de considérer un spot comme étant un artéfact et non pas une protéine dépend uniquement de l'utilisateur. Il se peut que celui-ci soit amené, suite à diverses expériences, à conclure qu'un spot qu'il avait considéré dans un premier temps comme un artéfact corresponde en fait à une protéine. Notre système offre donc la possibilité de démasquer un spot. Comme celui-ci n'apparaît plus à l'écran, l'utilisateur ne peut avoir qu'une idée approximative de son emplacement. Une procédure **demasquer** lui permet dès lors de déterminer une zone dans laquelle les spots masqués doivent réapparaître à l'écran. Cette procédure recalcule également l'intensité globale du gel.

Il est également possible de démasquer un spot identifié par son nom, son numéro manuel ou automatique pour autant que de tels attributs aient été affectés à ce spot.

8.3. Numérotation automatique des spots.

En plus de leur position dans le gel, les spots peuvent être identifiés par un nom ou un numéro dit "manuel" attribués par l'utilisateur. Ce sera généralement le cas des spots auxquels il s'intéresse particulièrement. Il est cependant possible d'identifier l'ensemble des spots d'un gel à l'aide d'un algorithme de numérotation automatique. Celui-ci, implémenté par la procédure **numeroter_2d**, attribue à chaque spot, et donc à chaque protéine, un numéro entier basé sur son pI et son pM estimés.

Le principe est le suivant:

numéro automatique: **XXYYZZZ**

où **ZZZ** = le pM en Kdaltons,
XXYY = le pI multiplié par 100.

Le pI et le pM sont estimés par les sous-procédures **eval_pi** et **eval_pm** du module accès-BD, sur base de la position du spot concerné, d'une part, et des positions et valeurs des points étalons introduits au préalable par l'utilisateur, d'autre part (les données relatives aux étalons de pI et de pM sont contenues dans le fichier .ref).

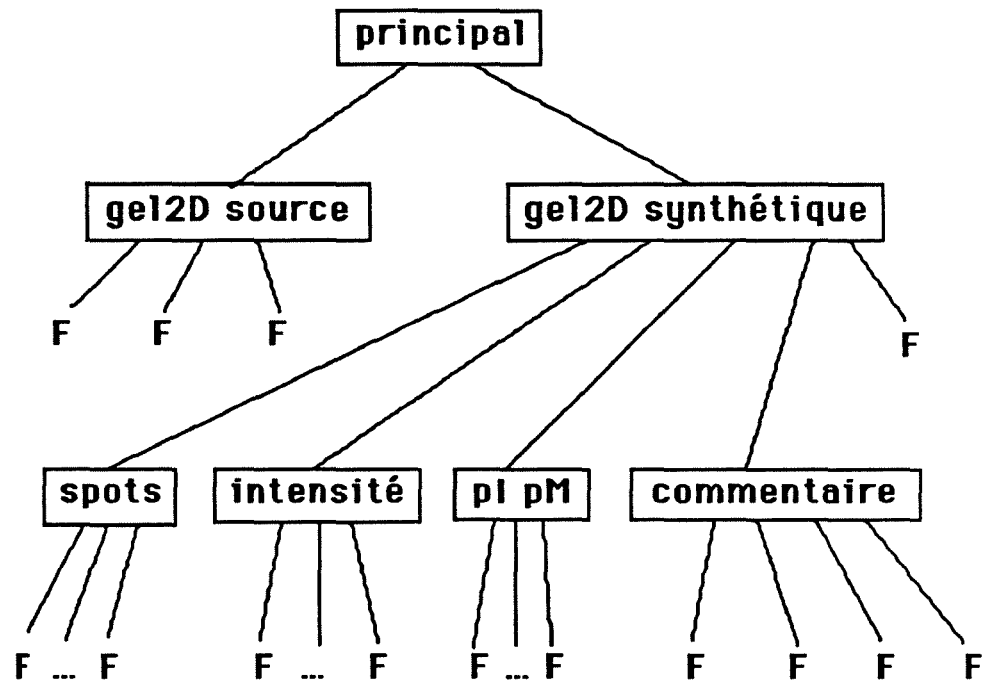
Le numéro automatique attribué à chaque spot est donc directement lié à sa position dans le gel, laquelle l'identifie au sein du gel.

9. **CE QUE LE SYSTEME PROPOSE : module menu.**

Les fonctionnalités offertes par le système sont proposées à travers des menus hiérarchisés dont la structure est reprise à la figure 8.

Les contenus de ces menus sont repris à la figure 9.

A chaque menu ou opération proposé correspond un numéro. L'utilisateur opère sa sélection en introduisant au clavier le numéro qui correspond à son choix.



F = fonction

Fig. 8

Hiérarchie des menus.

Menu principal:

- gel2D source (1)
- gel2D synthétique (2)
- quitter (3)

Menu gel2D source:

- visionner gel2D (2)
- traiter gel2D (3)
- quitter (4)

Menu gel2D synth:

- visionner gel2D (1)
- spots (2)
- intensité (3)
- pl pM (4)
- commentaire (5)
- quitter (6)

Menu spots:

- consulter spot par souris (1)
- nom (2)
- n° manuel (3)
- n° automatique (4)
- attribuer nom par souris (5)
- nom (6)
- n° manuel (7)
- n° automatique (8)
- attribuer n° par souris (9)
- nom (10)
- n° manuel (11)
- n° automatique (12)
- quitter (13)

Fig. 9

Les fonctionnalités offertes par les différents menus.

Menu gel2d intensité:

- consulter intensité globale du gel (1)
- consulter étalon intensité (2)
- désigner étalon par souris (3)
 - nom (4)
 - n° manuel (5)
 - n° automatique (6)
- masquer artéfact par souris (7)
 - nom (8)
 - n° manuel (9)
 - n° automatique (10)
- démasquer zone par souris (11)
 - spot par nom (12)
 - n° manuel (13)
 - n° automatique (14)
- quitter (15)

Menu gel2D plpM:

- consulter étalons (1)
- estimer point (2)
- désigner spot étalon pl par souris (3)
 - nom (4)
 - n° manuel (5)
 - n° automatique (6)
- désigner spot étalon pM par souris (7)
 - nom (8)
 - n° manuel (9)
 - n° automatique (10)
- désigner point étalon pl par souris (11)
- désigner point étalon pM par souris (12)
- numérotation automatique (13)
- quitter (14)

Fig. 9 (suite)

- Menu com2D synth:

- lire commentaire gel courant (1)
- écrire commenaire gel courant (2)
- lire commentaire autre gel (3)
- écrire commentaire autre gel (4)

Fig. 9 (suite et fin)

10. DISCUSSION.

10.1. Performances de l'algorithme de quantification des spots.

10.1.1. Complexité des calculs.

L'algorithme de quantification des spots (voir figure 10) est un algorithme itératif dont le nombre d'itérations est fonction du nombre de taches présentes dans un gel. Chaque boucle de cet algorithme contient elle-même l'algorithme itératif de Schroeder dont les calculs sont relativement lourds. Les complexités théoriques des procédures de ce sous algorithme sont les suivantes:

Prépartition : $O(n \times s)$

Paramètre : $O(n)$

Partition : $O(n \times s)$

Evalcrit : $O(n)$

où n = le nombre de points constituant la tache,
 s = le nombre de spots dans la tache.

Si T = le nombre de taches dans l'image,
 N = le nombre de points constituant l'image,
 S = le nombre de spots dans l'image,

la complexité de l'algorithme de quantification des spots est de l'ordre de

$$T \times (N/T \times S/T)$$

où N/T représente le nombre moyen de points par tache et S/T représente le nombre moyen de spots par tache; ou encore

$$(N \times S)/T .$$

N et S étant fixés pour une image, il faut augmenter T pour réduire les temps de calcul.

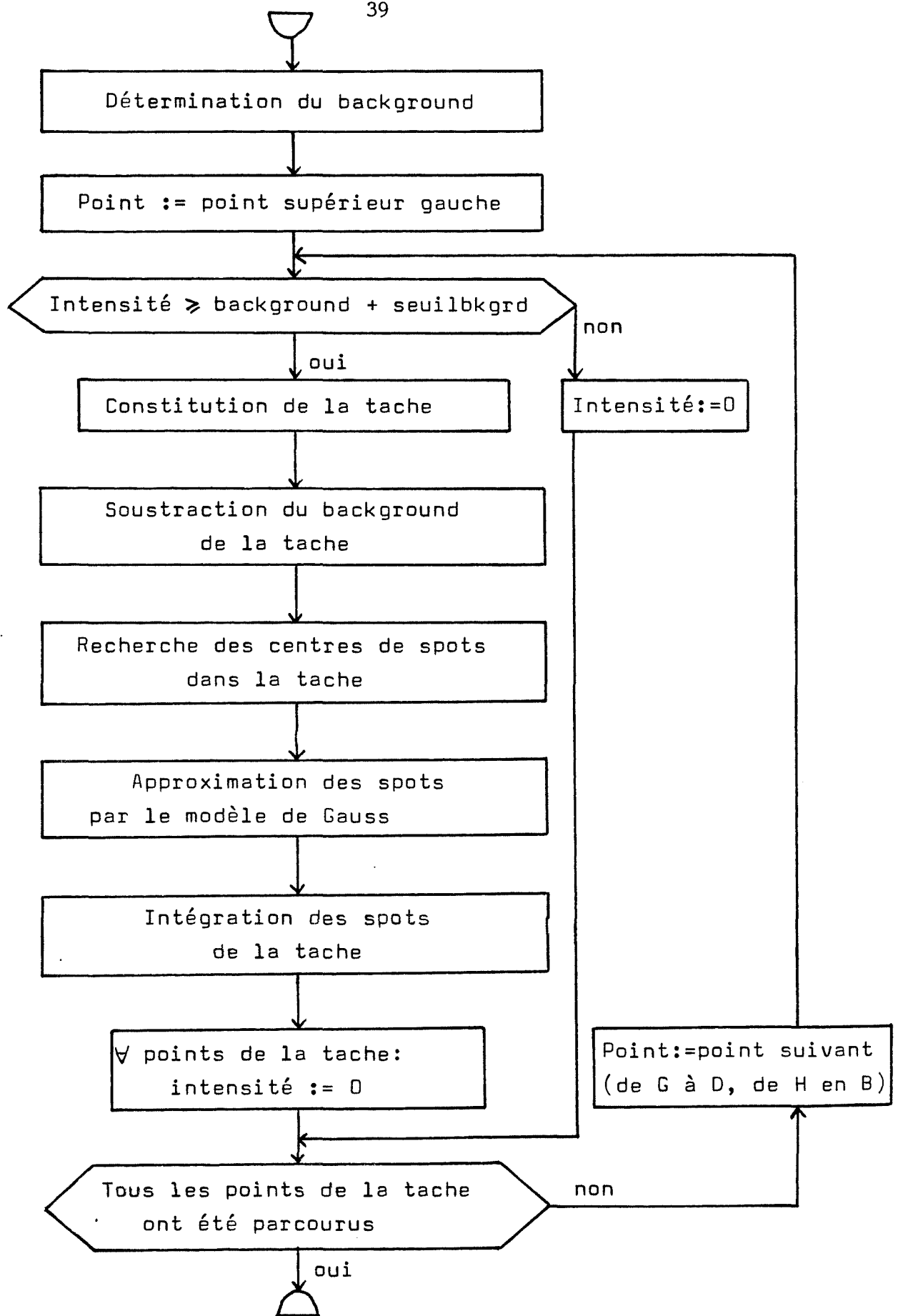


FIG 10

Algorithme d'extraction de l'information présente dans l'image

Il y a donc intérêt, si l'on veut obtenir un temps de traitement raisonnable pour la totalité de l'image, à travailler sur un nombre élevé de petites taches (n et s petits) plutôt que sur un nombre restreint de taches étendues.

En effet, pour peu qu'une tache contienne quelques milliers de points et une vingtaine de spots, le traitement de cette seule tâche se chiffre en heures.

10.1.2. Solutions proposées.

10.1.2.1. Seuillage des taches.

Lors de la constitution d'une tache (cf 5.2.2) un point est sélectionné si son intensité dépasse celle du background correspondant d'une valeur au moins égale à *seuilbkgrd*. Il est donc possible en donnant à ce paramètre une valeur suffisante, de ne pas sélectionner les points de faible intensité se trouvant par exemple dans les zones où des spots ne se chevauchent que très légèrement. De ce fait, la tache n'est constituée que d'un nombre réduit de spots qui se chevauchent suffisamment.

Ce principe est illustré à la figure 11.

Le choix d'une valeur faible (5-10 niveaux de gris) pour le paramètre *seuilbkgrd* n'a pratiquement aucune répercussion sur la quantification des spots car seuls les points situés à leur extrême périphérie (points de faible intensité) ne sont pas repris. En outre, seul le background est soustrait de l'intensité des points collectés (et non pas le background augmenté de la valeur de *seuilbkgrd*).

10.1.2.2. Coupure de taches. (procédure **coupure**)

Bien que notre système suppose des gels d'électrophorèse de bonne qualité (Garrels 1979), il n'est pas impossible que des stries y apparaissent de façon assez marquée. De tels artefacts font alors la jonction entre des taches qui seraient normalement distinctes. Il n'est pas possible de jouer sur le valeur de *seuilbkgrd* pour "gommer" de telles stries puisque cette valeur doit rester relativement faible.

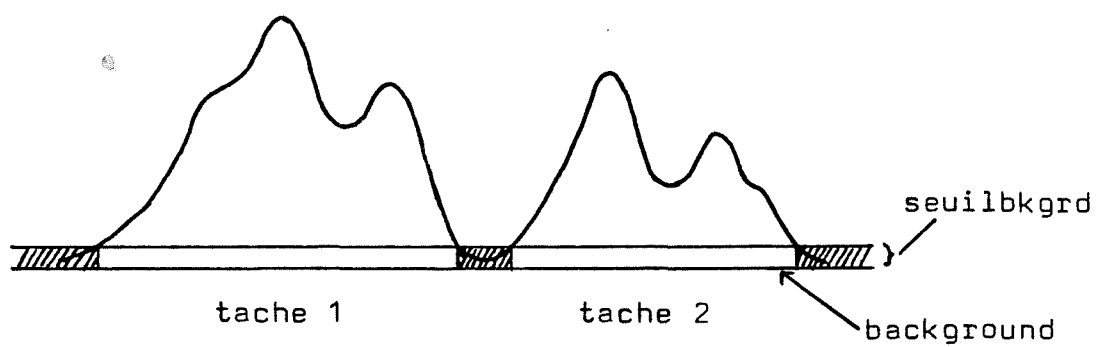


FIG 11

Illustration du principe de seuillage des taches.

Il est donc donné à l'utilisateur la possibilité de couper une strie (une tache en toute généralité) en désignant à l'écran et à l'aide de la souris deux points entre lesquels la coupure doit être réalisée. Cette opération est offerte par la procédure **coupure** du module extraction.

10.1.2.3. Installation d'un coprocesseur arithmétique.

Cette solution devrait se traduire par une réduction notable de la durée du traitement global d'un gel.

10.1.2.4. Simplification des calculs.

Si malgré l'adjonction d'un coprocesseur, le traitement d'un gel 2D s'avérait encore trop long, la complexité des calculs liés à l'algorithme de Schroeder (et donc la durée du traitement) pourrait être réduite, par exemple, en limitant les classes auxquelles un point peut être affecté à celles qui correspondent aux 3 (4,5,6 ?) distributions les plus proches de ce point et non plus à l'ensemble des distributions présentes dans la tache qui contient ce point.

10.2. L'organisation des données.

L'exploitation des données dans notre système fait appel aux possibilités offertes par le langage C. Les trois types de fichiers utilisés, *.fim*, *.spt* et *.ref*, sont donc des fichiers séquentiels et à accès direct. L'extraction de l'information présente dans l'image d'un gel se traduit par une réduction considérable de l'espace mémoire nécessaire pour stocker l'information utile. En effet, si l'image de départ est une image de 500 X 500 pixels contenant 1000 spots, le fichier *.fim* contenant l'image occupera 250 kbytes ce qui est énorme en égard aux limites imposées par le hardware. L'information extraite d'une telle image sera mémorisée dans un fichier *.spt* de 50 kbytes seulement. Ce dernier fichier contenant toute l'information nécessaire, avec le fichier *.ref*, il n'est dès lors plus indispensable de conserver dans le système le fichier *.fim*.

Un fichier *.spt* contient l'information relative aux différents spots présents dans une même image, à l'exception de leurs intensités relatives, et de leurs *pl* et *pM* estimés (cf structure fichier *.spt*). Lors d'une consultation, ces paramètres sont calculés sur base de l'intensité absolue et la position du spot (fichier *.spt*) et de l'information relative aux différents étalons (fichier *.ref*).

Ce procédé permet d'économiser $3 \times 4 = 12$ bytes par article dans le fichier *.spt* soit 12 kbytes si ce fichier contient l'information relative à un millier de spots. De plus, aucune opération de mise à jour n'est nécessaire dans le fichier *.spt*, lors de toute modification apportée par l'utilisateur dans l'information relative aux divers étalons.

Il est actuellement trop tôt pour dire si cette organisation des données est la plus adéquate en fonction du taux d'utilisation qui sera fait des différentes fonctionnalités offertes par le système. Il se peut également que des développements qui seraient réalisés à partir de notre système nécessitent une organisation (exploitation) différente des données. De telles adaptations pourraient être réalisées sans aucun problème grâce à l'architecture du système. En effet, seules les procédures du module fichier devraient être réécrites, l'interface avec les modules de plus haut niveau restant inchangé puisque celui-ci est réalisé au moyen de primitives dont les paramètres sont indépendants de l'organisation des données (le paramètre principal est le nom de l'image (du gel) à laquelle l'utilisateur s'intéresse).

10.3. Perspectives de développement.

La possibilité pour l'utilisateur de sélectionner les menus et fonctionnalités à l'aide de la souris pourrait s'avérer intéressante. Si tel était le cas, seul le module "menu" devrait subir quelques modifications.

L'enregistrement des images se fait actuellement par l'utilisation du logiciel de traitement d'image PCSCOPE. Cette opération pourrait être intégrée dans notre système et proposée dans le menu "gel 2D source". Si cela s'avérait nécessaire, ce menu ainsi que le menu "gel 2D synthétique" pourraient offrir la possibilité d'effectuer un zoom sur une partie d'un gel.

A plus long terme, le système pourrait offrir la possibilité de comparer automatiquement plusieurs gels entre eux ainsi que d'établir des statistiques sur l'évolution de certaines protéines à travers les gels correspondant à différentes conditions expérimentales.

Comparer deux gels entre eux implique de pouvoir identifier chaque protéine sur chacun de ces deux gels. Or, malgré une technique de préparation de gel très rigoureuse des distorsions apparaissent si bien qu'une même protéine n'occupe jamais exactement la même position dans deux gels différents. L'algorithme chargé de l'appariement des spots situés sur deux gels devrait donc tenir compte de ces distorsions. Celles-ci peuvent être estimées si l'expérimentateur reconnaît et désigne lui-même des spots de référence sur chacun des gels à apparier (KUICK 1986, LIPKIN 1980). Chaque spot à apparier est alors identifié par sa position par rapport aux spots de référence qui l'entourent.

Le volume de calculs nécessaires à cette opération serait non négligeable.

L'expérimentateur peut être aidé dans sa désignation des spots de référence par la présence et la mise à jour régulière d'une liste de tels spots, pour un échantillon biologique donné (LEMKIN 1981 (II)). En répétant cette opération pour chaque appariement de gels, il serait ainsi possible de constituer un gel de référence, ou dictionnaire des protéines, à utiliser pour la comparaison de gels (APPEL 1986).

Le volume d'informations manipulé par ce système constituerait une base de données importante (CELIS 1988, GARRELS 1986, LEMKIN 1981 (III)) qui nécessiterait vraisemblablement l'utilisation d'un SGBD.

Le passage du système actuel à un système aussi évolué impliquerait donc probablement l'abandon du microcomputer actuel pour un minicomputer.

Cependant, à la demande des utilisateurs, un des premiers développements à réaliser devrait être la quantification des gels d'électrophorèse à une dimension. Dans cette technique, les protéines sont présentes dans le gel sous forme de bandes (et non plus de spots) parallèles et alignées entre elles dans un même canal (voir figure 12).

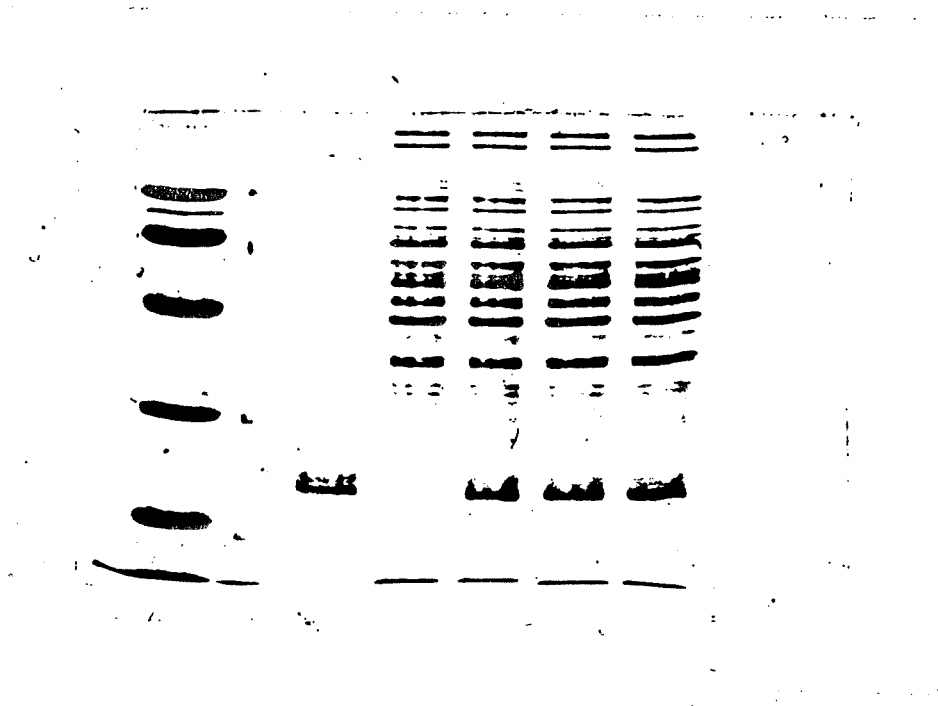


Fig. 12

Un exemple de gel d'électrophorèse à une dimension.

Le profil de chacune de ces bandes (qui correspondrait à une coupe perpendiculaire à la bande) correspond toutefois à une distribution gaussienne à une dimension. La quantification de telles bandes se ferait selon le principe suivant :

- 1- Détermination du background à l'aide des procédures **backligned** et **backligneg**.
- 2- Pour chaque ligne de points constituant un canal :
 - détection des centres de pics par la procédure **detectcentre**.
 - résolution des pics par l'algorithme de schroeder adapté à une seule dimension.
 - intégration des pics par répartition de l'intensité de chaque point de la ligne en fonction des densités de probabilité calculées pour chaque distribution.
 - constitution d'une liste reprenant pour chaque pic la position du centre ou sommets et l'intensité intégrée.
- 3- Réorganisation des listes de centres par la procédure **reorgcentre**.
- 4- A ce stade, chaque liste contient les intensités intégrées des "coupes" réalisées à travers une même bande. Il ne reste plus qu'à totaliser ces intensités pour obtenir l'intensité globale d'une bande.

Comme on le voit, le nombre de procédures à écrire pour réaliser cette nouvelle fonctionnalité est limité puisque les procédures **backligned**, **backligneg**, **detectcentre** et **reorgcentre** sont réutilisables telles qu'elles.

11. BIBLIOGRAPHIE.

Anderson, N.L., Taylor, J., Scandora, A.E., Coulter, B.P., Anderson, N.G. The TYCHO system for computer analysis of two-dimensional gel electrophoresis patterns. *Clin. Chem.* **27** (11), 1807-1820 (1981).

Appel, R., Funk, M., Hochstrasser, D., Müller, A.F., Pellegrini, C. MELANIE. Unsysteme expert d'aide à l'analyse et à l'interprétation d'électrophorèses bidimensionnelles. *Progrès Récents en Electrophorèse Bidimensionnelle* . Press. Univ. Nancy . 1986, 107-116 (1986).

Celis, J.E., Ratz, G.P., Celis, A., Madsen, P., Gesser, B., Kwee, S., Madsen, P.S., Nielsen, H.V., Yde, H., Lauridsen, J.B., Basse, B. Towards establishing comprehensive databases of cellular proteins from transformed human epithelial amnion cells (AMA) and normal peripheral blood mononuclear cells. *Leukemia*. **2** (9), 561-602 (1988).

Garrels, J.I. Two-dimensional gel electrophoresis and computer analysis of proteins synthesized by clonal cell lines. *J. Biol. Chem.* **254** (16), 7961-7977 (1979).

Garrels, J.I., Franza Jr, B.R. Building protein databases with the Quest system of computer-analysed two-dimensional gel electrophoresis. *Progrès Récents en Electrophorèse Bidimensionnelle*. Press . Univ. Nancy. 1986, 85-90 (1986).

Kuick, R., Sing, C., Hanash, S. An interactive multigel matching strategy. *Progrès Récents en Electrophorèse Bidimensionnelle*. Press .Univ. Nancy. 1986, 91-96 (1986).

Lemkin, P.F., Lipkin, L.E. GELLAB: a computer system for 2D gel electrophoresis analysis .1. Segmentation of spots and system preliminaries. *Computers and Biomedical Research*. **14**, 272-297 (1981).

Lemkin, P.F., Lipkin, L.E. GELLAB: a computer system for 2D gel electrophoresis analysis .2. Pairing spots. *Computers and Biomedical Research*. **14**, 355-380. (1981).

Lemkin, P.F., Lipkin, L.E. GELLAB: a computer system for 2D gel electrophoresis analysis .3. Multiple two-dimensional gel analysis. *Computers and Biomedical Research*. **14**, 407-446 (1981).

Lipkin, L.E., Lemkin, P.F. Data-base techniques for multiple two-dimensional polyacrylamide gel electrophoresis analyses. *Clin. Chem.* **26** (10), 1403-1412 (1980).

O' Farrell, P.H. High resolution two-dimensional electrophoresis of proteins. *J. Biol. Chem.* **250**, 4007 (1975).

Schroeder, A. Analyse d'un mélange de distributions de probabilités de même type. IRIA, rapport de recherche n°104 (1975).

Skolnick, M.M., Sternberg, S.R., Neel, J.V. Computer programs for adapting two-dimensional gels to the study of mutation. *Clin. Chem.* **28** (4), 969-978 (1982).

Taylor, J., Anderson, N.L., Coulter, B.P., Scandora Jr, A.E., Anderson, N.G. Estimation of two-dimensional electrophoretic spot intensities and positions by modeling. *Electrophoresis* **79**. Walter de Gruyter co., Berlin, New York. 329-339 (1980).

Vincens, P., Rabilloud, C., Gaboriaud, C., Therre, H., Pennetier, J-L., Hubert, M., Tarroux, P. A complete system for the automatic processing of two-dimensional electrophoresis images. *Progrès Récents en Electrophorèse Bidimensionnelle*. Press. Univ. Nancy. 1986, 121-130 (1986).

Young, T.Y., Coraluppi, G. Stochastic estimation of a mixture of normal density functions using an information criterion. *IEEE Trans. Information Theory*, IT-16 (3), 258-263 (1970).

12. ANNEXES.

<u>Table des matières.</u>	Page
1. Constantes propres au système (macros).	A.1
2. Structures manipulées par le système.	A.2
3. Chaînage des listes de structures <i>pixel</i> .	A.4
4. Structure des fichiers.	A.5
5. Liste des procédures offertes par les différents modules.	A.6
6. Spécifications des procédures.	A.9
7. Glossaire.	A.10

1. Constantes propres au système. (macros)

longcommentaire	61
longnomfich	31
longnomspot	21
maxcentre	256
maxchaîne	50
maxclasse	3
maxcol	256
maxdistrib	200
maxligne	256
maxpoint	30000
maxtab	100
seuilbkgrd	5
seuilfit	3
seuilpente	1
seuilreorg	1
xfactd	0.7
xfactm	0.7
xseuilpic	0
yfactd	0.7
yfactm	0.7
yseuilpic	0.7

2. Structures manipulées par le système.

```
struct artspot { float col;
                 float ligne;
                 float varx;
                 float vary;
                 float intensité;
                 long numman;
                 long numaut;
                 char vision;
                 char nom[longnomspot];
                 }
```

```
struct dim { short dx;
             short dy;
             }
```

```
struct distrib { float col;
                 float ligne;
                 float varx;
                 float vary;
                 float covarxy;
                 float intensite;
                 float taille;
                 float distance;
                 }
```

```
struct pixel { short col;
               short ligne;
               unsigned char intensite;
               struct pixel *next;
               }
```

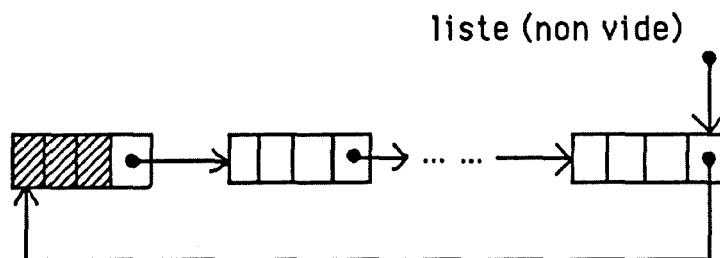
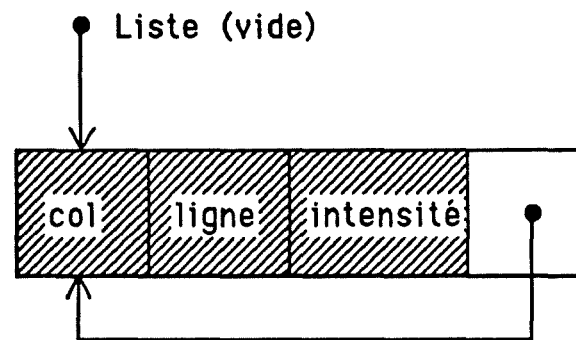
```
struct point { short col;
               short ligne;
               short intensite;
               short classe;
               }
```

```
struct pointref { float var;
                  float col;
                  float ligne;
                  }
```

A.3

```
struct proteine { char nom[longnomspot];  
                  long num1;  
                  long num2;  
                  float int1;  
                  float int2;  
                  float col;  
                  float ligne;  
                  float pi;  
                  float pm;  
                  char vision;  
                }
```

3. Chainage des listes de structures pixel



4. Structure des fichiers.

Fichier.ref :

Bytes	Type	Contenu
1 à 4	struct dim	dimensions du gel
5 à 16	struct pointref	étalon de pl n°1
17 à 28	struct pointref	étalon de pM n°1
29 à 40	struct pointref	étalon de pl n°2
41 à 52	struct pointref	étalon de pM n°2
53 à 56	float	intensité globale de gel
57 à 106	struct artspot	spot étalon d'intensité
107 à 167	string	commentaire

Fichier.spt :

artspot, artspot, artspot, ... , ... , ... , artspot.

artspot:

Bytes	Type	Contenu
1 à 4	float	colonne (position en x)
5 à 8	float	ligne (position en y)
9 à 12	float	variance en x
13 à 16	float	variance en y
17 à 20	float	intensité absolue du spot
21 à 24	long int	numéro manuel
25 à 28	long int	numéro automatique
29	char	(non-)visualisation à l'écran
30 à 50	string	nom du spot (protéine)

5. Liste des procédures offertes par les modules.

<u>Module accès-BD.</u>	Page.
addspot	S.2
calcul_int_gel	S.18
charger_gel2d_source	S.19
charger-gel2d_synth	S.20
chercher_spot_cooronnee	S.21
chercher_spot_nom	S.22
chercher_spot_numaut	S.23
chercher_spot_numman	S.24
consult_etalon_int	S.27
consult_int_gel	S.28
consult_point	S.29
consult_spot_cooronnee	S.30
consult_spot_nom	S.31
consult_spot_numaut	S.32
consult_spot_numman	S.34
demasquer_spot	S.40
demasquer_spot_nom	S.41
demasquer_spot_numaut	S.42
demasquer_spot_numman	S.43
demasquer_zone	S.44
ecrire_com2d_synth	S.48
ecrire_etat_int	S.49
ecrire_etat_int_cooronnee	S.51
ecrire_etat_int_nom	S.52
ecrire_etat_int_numaut	S.53
ecrire_etat_int_numman	S.54
ecrire_etalon_pi	S.55
ecrire_etalon_pm	S.56
ecrire_nomspot_cooronnee	S.57
ecrire_nomspot_nom	S.58
ecrire_nomspot_numaut	S.59
ecrire_nomspot_numman	S.60
ecrire_numspot_cooronnee	S.61
ecrire_numspot_nom	S.62
ecrire_numspot_numaut	S.63
ecrire_numspot_numman	S.64
ecrire_spot_nom	S.65
ecrire_spot_numman	S.66

eval_pi	S.67
eval_pm	S.68
init_fichref	S.93
lire_com2d_synth	S.95
lire_etalon_pi	S.96
lire_etalon_pm	S.97
lire_spot	S.99
masquer_spot	S.105
masquer_spot_coordonnee	S.106
masquer_spot_nom	S.107
masquer_spot_numaut	S.108
masquer_spot_numman	S.109
numaut_2d	S.112

Module extraction.

addpixel	S.1
backcolb	S.12
backcolh	S.13
backligned	S.16
backligneg	S.17
background	S.14
coupure	S.35
detectcentre	S.45
evalcrit	S.69
fitting	S.70
integration	S.94
nivtache	S.110
normalisation2d	S.111
parametre	S.114
partition	S.115
prepartition	S.116
quantif	S.118
reorgcentre	S.119
videpixel	S.122

<u>Module fonction.</u>	<u>Page.</u>
attrib_nom_spot_nom	S.4
attrib_nom_spot_numaut	S.5
attrib_nom_spot_numman	S.6
attrib_nom_spot_souris	S.7
attrib_num_spot_nom	S.8
attrib_num_spot_numaut	S.9
attrib_num_spot_numman	S.10
attrib_num_spot_souris	S.11
commentaire_synth	S.26
demasquer	S.36
demasq_artef_nom	S.37
demasq_artef_numaut	S.38
demasq_artef_numman	S.39
ecran_prot	S.47
fix_etal_int_nom	S.71
fix_etal_int_numaut	S.72
fix_etal_int_numman	S.73
fix_etal_int_souris	S.74
fix_etal_pi_point	S.75
fix_etal_pi_spot_nom	S.76
fix_etal_pi_spot_numaut	S.77
fix_etal_pi_spot_numman	S.78
fix_etal_pi_spot_souris	S.79
fix_etal_pm_point	S.80
fix_etal_pm_spot_nom	S.81
fix_etal_pm_spot_numaut	S.82
fix_etal_pm_spot_numman	S.83
fix_etal_pm_spot_souris	S.84
info_etalon_int	S.85
info_etalons_pipm	S.86
info_intensite_gel	S.87
info_point	S.88
info_spot_nom	S.89
info_spot_numaut	S.90
info_spot_numman	S.91
info_spot_souris	S.92
masq_artef_nom	S.101
masq_artef_numaut	S.102
masq_artef_numman	S.103

masq_artef_souris	S.104
numeroter_2d	S.113
traiter_gel2d_source	S.120
vision_gel2d_source	S.123
vision_gel2d_synth	S.124

Module vidéo.**Page.**

clignoter_point	S.25
video_spot	S.121

6. Spécifications des procédures.

PROCEDURE : **addpixel (&liste, col, ligne, intensité)**

MODULE : **Extraction**

TYPE : **Gestion de listes de pixels**

PARAMETRES D'ENTREE : **&liste = pointeur sur un pointeur de liste de
pixels
Type : pointeur**

**col, ligne = coordonnées d'un pixel
Type : short int**

**intensité = niveau de gris d'un pixel
Type : unsigned char**

PARAMETRES DE SORTIE : **la liste pointée par *&liste* contient une
structure *pixel* de plus où se trouvent les
valeurs de col, ligne et intensité**

DESCRIPTION : **Ajout d'un élément à la liste de structures *pixel*
pointée par *liste***

APPEL PRELIMINAIRE :


```
spot.intensite = intensité;  
spot.numman = 0;  
spot.numaut = 0;  
spot.vision = 'y';  
spot.nom = '-';
```

APPEL PRELIMINAIRE :

PROCEDURE : attrib_nom_spot_nom (nomgel, quad, x0, y0)

MODULE : Fonction

TYPE : Gestion de spots

PARAMETRES D'ENTREE :

- nomgel = pointeur sur la chaîne de caractères contenant le nom d'un gel 2D
Type : pointeur
- quad = numéro du quadrant où est visualisé l'image du gel
Type : short int
- x0, y0 = coordonnées du pont en haut et à gauche où est positionnée l'image du gel. Ce point doit être situé dans le quadrant *quad*
Type : short int
- nomspot = pointeur sur la chaîne de caractères contenant le nom d'un spot du gel
Type : pointeur
entrée au clavier
- nvnomspot :pointeur sur la chaîne de caractères contenant le nom de spot à attribuer
Type : pointeur
entrée au clavier

PARAMETRES DE SORTIE :

- Le message "erreur lecture/écriture" est généré si un problème est apparu lors d'un accès au fichier
Type : string
Sortie à l'écran PC

DESCRIPTION :

- attribution du nom *nvnomspot* au spot désigné au sein du gel synthétique *nomgel* par *nomspot*
L'emplacement correspondant au spot clignote à l'écran vidéo

APPEL PRELIMINAIRE :

PROCEDURE : attrib_nom_spot_numaut (nomgel, quad, x0, y0)

MODULE : Fonction

TYPE : Gestion de spots

PARAMETRES D'ENTREE :

nomgel = pointeur sur la chaîne de caractères contenant le nom d'un gel 2D
Type : pointeur

quad = numéro du quadrant où est visualisé l'image du gel
Type : short int

x0, y0 = coordonnées du point en haut et à gauche où est positionnée l'image du gel. Ce point doit être situé dans le quadrant *quad*
Type : short int

numspotaut = numéro automatique d'un spot du gel
Type : long int
entrée au clavier

nvnomspot :pointeur sur la chaîne de caractères contenant le nom de spot à attribuer
Type : pointeur
entrée au clavier

PARAMETRES DE SORTIE : Le message "erreur lecture/écriture" est généré si un problème est apparu lors d'un accès au fichier
Type : string
Sortie à l'écran PC

DESCRIPTION : attribution du nom *nvnomspot* au spot désigné au sein du gel synthétique *nomgel* par *numspotaut*
L'emplacement correspondant au spot clignote à l'écran vidéo

APPEL PRELIMINAIRE :

PROCEDURE : attrib_nom_spot_numman (nomgel, quad, x0, y0)

MODULE : Fonction

TYPE : Gestion de spots

PARAMETRES D'ENTREE :

nomgel = pointeur sur la chaîne de caractères
contenant le nom d'un gel 2D
Type : pointeur

quad = numéro du quadrant où est visualisé
l'image du gel
Type : short int

x0, y0 = coordonnées du point en haut et à
gauche où est positionnée l'image du gel. Ce
point doit être situé dans le quadrant *quad*
Type : short int

numspotman = numéro manuel d'un spot du gel
Type : long int
entrée au clavier

nvnomspot :pointeur sur la chaîne de caractères
contenant le nom de spot à attribuer
Type : pointeur
entrée au clavier

PARAMETRES DE SORTIE :

Le message "erreur lecture/écriture" est généré
si un problème est apparu lors d'un accès au
fichier
Type : string
Sortie à l'écran PC

DESCRIPTION :

attribution du nom *nvnomspot* au spot désigné
au sein du gel synthétique *nomgel* par
numspotman
L'emplacement correspondant au spot clignote à
l'écran vidéo

APPEL PRELIMINAIRE :

PROCEDURE : attrib_nom_spot_souris (nomgel, quad, x0, y0)

MODULE : Fonction

TYPE : Gestion de spots

PARAMETRES D'ENTREE :

nomgel = pointeur sur la chaîne de caractères contenant le nom d'un gel 2D
Type : pointeur

quad = numéro du quadrant où est visualisé l'image du gel
Type : short int

x0, y0 = coordonnées du point en haut et à gauche où est positionnée l'image du gel. Ce point doit être situé dans le quadrant *quad*
Type : short int

x,y = coordonnées d'un pont du gel
Type : short int
entrée à la souris

nvnomspot :pointeur sur la chaîne de caractères contenant le nom de spot à attribuer
Type : pointeur
entrée au clavier

PARAMETRES DE SORTIE :

Le message "erreur lecture/écriture" est généré si un problème est apparu lors d'un accès au fichier
Type : string
Sortie à l'écran PC

DESCRIPTION :

attribution du nom *nvnomspot* au spot le plus proche du point désigné par la souris au sein du gel synthétique *nomgel*
L'utilisateur sélectionne le point en appuyant sur le bouton gauche de la souris
La procédure se répète jusqu'à ce qu'il appuie sur le bouton droit de la souris
L'emplacement correspondant au spot clignote à l'écran vidéo

APPEL PRELIMINAIRE : **vision_gel2d_synth**

PROCEDURE : attrib_num_spot_nom (nomgel, quad, x0, y0)

MODULE : Fonction

TYPE : Gestion de spots

PARAMETRES D'ENTREE :

nomgel = pointeur sur la chaîne de caractères
contenant le nom d'un gel 2D
Type : pointeur

quad = numéro du quadrant où est visualisée
l'image du gel
Type : short int

x0, y0 = coordonnées du point en haut et à
gauche où est positionnée l'image du gel. Ce
point doit être situé dans le quadrant *quad*
Type : short int

nomspot = pointeur sur la chaîne de caractères
contenant le nom du gel
Type : pointeur
entrée au clavier

numspot = numéro à attribuer au spot
Type : long int
entrée au clavier

PARAMETRES DE SORTIE : Le message "erreur lecture/écriture" est généré
si un problème est apparu lors d'un accès au
fichier
Type : string
sortie à l'écran PC

DESCRIPTION :

attribution du numéro manuel *numspot* au spot
désigné au sein du gel synthétique *nomgel* par
nomspot
L'emplacement correspondant au spot clignote à
l'écran vidéo

APPEL PRELIMINAIRE :

PROCEDURE : attrib_num_spot_nummat (nomgel, quad, x0, y0)

MODULE : Fonction

TYPE : Gestion de spots

PARAMETRES D'ENTREE :

nomgel = pointeur sur la chaîne de caractères contenant le nom d'un gel 2D
Type : pointeur

quad = numéro du quadrant où est visualisée l'image du gel
Type : short int

x0, y0 = coordonnées du point en haut et à gauche où est positionnée l'image du gel. Ce point doit être situé dans le quadrant *quad*
Type : short int

numspotaut = numéro automatique d'un spot du gel
Type : long int
entrée au clavier

numspot = numéro à attribuer au spot
Type : long int
entrée au clavier

PARAMETRES DE SORTIE :

Le message "erreur lecture/écriture" est généré si un problème est apparu lors d'un accès au fichier
Type : string
sortie à l'écran PC

DESCRIPTION :

attribution du numéro manuel *numspot* au spot désigné au sein du gel synthétique *nomgel* par *numspotaut*
L'emplacement correspondant au spot clignote à l'écran vidéo

APPEL PRELIMINAIRE :

PROCEDURE : attrib_num_spot_numman (nomgel, quad, x0, y0)

MODULE : Fonction

TYPE : Gestion de spots

PARAMETRES D'ENTREE :

nomgel = pointeur sur la chaîne de caractères contenant le nom d'un gel 2D
Type : pointeur

quad = numéro du quadrant où est visualisée l'image du gel
Type : short int

x0, y0 = coordonnées du point en haut et à gauche où est positionnée l'image du gel. Ce point doit être situé dans le quadrant *quad*
Type : short int

numspotman = numéro manuel d'un spot du gel
Type : long int
entrée au clavier

numspot = numéro à attribuer au spot
Type : long int
entrée au clavier

PARAMETRES DE SORTIE : Le message "erreur lecture/écriture" est généré si un problème est apparu lors d'un accès au fichier
Type : string
sortie à l'écran PC

DESCRIPTION : attribution du numéro manuel *numspot* au spot désigné au sein du gel synthétique *nomgel* par *numspotman*
L'emplacement correspondant au spot clignote à l'écran vidéo

APPEL PRELIMINAIRE :

PROCEDURE : attrib_num_spot_souris (nomgel, quad, x0, y0)

MODULE : Fonction

TYPE : Gestion de spots

PARAMETRES D'ENTREE :

nomgel = pointeur sur la chaîne de caractères contenant le nom d'un gel 2D
Type : pointeur

quad = numéro du quadrant où est visualisée l'image du gel
Type : short int

x0, y0 = coordonnées du point en haut et à gauche où est positionnée l'image du gel. Ce point doit être situé dans le quadrant *quad*
Type : short int

x,y =coordonnées d'un point du gel
Type : short int
entrée à la souris

numspot = numéro à attribuer au spot
Type : long int
entrée au clavier

PARAMETRES DE SORTIE : Le message "erreur lecture/écriture" est généré si un problème est apparu lors d'un accès au fichier
Type : string
sortie à l'écran PC

DESCRIPTION :

attribution du numéro manuel *numspot* au spot le plus proche désigné par la souris au sein du gel synthétique *nomgel* . L'utilisateur sélectionne le point en appuyant sur le bouton gauche de la souris. La procédure se répète jusqu'à ce qu'il appuie sur le bouton droit de la souris.
L'emplacement correspondant au spot clignote à l'écran vidéo

APPEL PRELIMINAIRE : **vision_gel2d_synth**

PROCEDURE : backcolb (x0, y0, dx, dy, xf, yf, pente)

MODULE : Extraction

TYPE : Détermination du background

PARAMETRES D'ENTREE :

x0, y0 = coordonnées du point en haut et à gauche où est positionnée l'image inversée d'un gel 2D source
Type : short int

dx, dy = nombre de colonnes et de lignes de l'image du gel
Type : short int

xf, yf = coordonnées du point où est mémorisée la valeur du background déterminée au point x0, y0
Type : short int

pente = différence entre les niveaux de gris de deux points lus successivement, au delà de laquelle le background est maintenu constant
Type : short int

PARAMETRES DE SORTIE :

DESCRIPTION :

Cette procédure détermine le background en tout point de l'image inversée d'un gel 2D source, en lisant chaque point du gel de haut en bas et de gauche à droite.
La valeur du background en un point x,y est le maximum entre le background déterminé par la procédure en ce point, et la valeur mémorisée au point $x+(xf-x0)$, $y+(yf-y0)$. Cette valeur maximale est mémorisée au point $x+(xf-x0)$, $y+(yf-y0)$.

APPEL PRELIMINAIRE :

PROCEDURE : backcolh (x0, y0, dx, dy, xf, yf, pente)

MODULE : Extraction

TYPE : Détermination du background

PARAMETRES D'ENTREE :

x0, y0 = coordonnées du point en haut et à gauche où est positionnée l'image inversée d'un gel 2D source
Type : short int

dx, dy = nombre de colonnes et de lignes de l'image du gel
Type : short int

xf, yf = coordonnées du point où est mémorisée la valeur du background déterminée au point x0, y0
Type : short int

pente = différence entre les niveaux de gris de deux points lus successivement, au delà de laquelle le background est maintenu constant
Type : short int

PARAMETRES DE SORTIE :

DESCRIPTION : Cette procédure détermine le background en tout point de l'image inversée d'un gel 2D source, en lisant chaque point du gel de bas en haut et de gauche à droite.
La valeur du background en un point x,y est le maximum entre le background déterminé par la procédure en ce point, et la valeur mémorisée au point $x+(xf-x0)$, $y+(yf-y0)$. Cette valeur maximale est mémorisée au point $x+(xf-x0)$, $y+(yf-y0)$.

APPEL PRELIMINAIRE :

PROCEDURE : background (nomgel, xinit, yinit, &x0, &y0, &dx, &dy, &xbkgrd, &ybkgrd, &fstat)

MODULE : Extraction

TYPE : Détermination du background

PARAMETRES D'ENTREE : nomgel = pointeur sur la chaîne de caractère contenant le nom d'un gel 2D source
Type : pointeur

xinit, yinit = coordonnées du point en haut et à gauche où doit être positionnée l'image du gel source
Type : short int

&x0, &y0 = pointeurs sur les variables recevant les coordonnées du coin supérieur gauche de l'image inversée du gel
Type : pointeur

&dx, &dy = pointeurs sur les variables recevant les dimensions (nombre de colonnes et de lignes) de l'image du gel
Type : pointeur

&xbkgrd, &ybkgrd = pointeurs sur les variables recevant les coordonnées du coin supérieur gauche de la zone où sont mémorisées les valeurs de background
Type : pointeur

&fstat = pointeur sur l'entier qui contiendra l'état des accès au fichier
Type : pointeur

PARAMETRES DE SORTIE : x0, y0 = coordonnées du coin supérieur gauche de l'image inversée du gel
Type : short int

dx, dy = nombre de colonnes et de lignes de l'image du gel
Type : short int

xbkgrd, ybkgrd = coordonnées du coin supérieur gauche de la zone où sont mémorisées les valeurs de background

Type : short int

fstat = 0 si un problème est apparu lors des accès au fichier

= 1 si aucun problème n'est apparu

Type : short int

DESCRIPTION :

Détermination du background d'un gel 2D

L'image du gel 2D source est positionnée au point xinit, yinit.

L'image inversée du gel est positionnée au point x0, y0.

La zone contenant les valeurs de background est positionnée au point xbkgrd, ybkgrd.

APPEL PRELIMINAIRE :

PROCEDURE : backligneg (x0, y0, dx, dy, xf, yf, pente)

MODULE : Extraction

TYPE : Détermination du background

PARAMETRES D'ENTREE : x0, y0 = coordonnées du point en haut et à gauche où est positionnée l'image inversée d'un gel 2D source
Type : short int

dx, dy = nombre de colonnes et de lignes de l'image du gel
Type : short int

xf, yf = coordonnées du point où est mémorisée la valeur du background déterminée au point x0, y0
Type : short int

pente = différence entre les niveaux de gris de deux points lus successivement, au delà de laquelle le background est maintenu constant
Type : short int

PARAMETRES DE SORTIE :

DESCRIPTION : Cette procédure détermine le background en tout point de l'image inversée d'un gel 2D source, en lisant chaque point du gel de droite à gauche et de haut en bas.
La valeur du background en un point x,y est le maximum entre le background déterminé par la procédure en ce point, et la valeur mémorisée au point $x+(xf-x0)$, $y+(yf-y0)$. Cette valeur maximale est mémorisée au point $x+(xf-x0)$, $y+(yf-y0)$.

APPEL PRELIMINAIRE :

PROCEDURE : `calcul_int_gel (nomgel, &fstat)`

MODULE : Accès-BD

TYPE: Gestion d'un gel synthétique

PARAMETRES D'ENTREE : `nomgel` = pointeur sur la chaîne de caractères
 contenant le nom d'un gel 2D
 Type : pointeur

`&fstat` = pointeur sur l'entier qui contiendra
 l'état des accès aux fichiers

PARAMETRES DE SORTIE : `fstat` = 0 si un problème est apparu lors des
 accès aux fichiers
 = 1 si aucun problème n'est apparu
 Type : short int

DESCRIPTION : Cette procédure calcule l'intensité globale du
 gel synthétique contenu dans le fichier
 nomgel.spt . Le résultat est sauvé dans le
 fichier *nomgel.ref*

APPEL PRELIMINAIRE :

PROCEDURE : charger_gel2d_source (nomgel, xo,yo, comment, &fstat)

MODULE : Accès-BD

TYPE: Gestion de gels 2D sources

PARAMETRES D'ENTREE :

nomgel = pointeur sur la chaîne de caractères contenant le nom d'un gel 2D
Type : pointeur

xo, yo = coordonnées du point en haut et à gauche où doit se positionner l'image du gel
Type : short int

comment = pointeur sur la chaîne de caractères recevant le commentaire associé au gel
Type : pointeur

&fstat = pointeur sur l'entier qui contiendra l'état des accès au fichier

PARAMETRES DE SORTIE :

fstat = 0 si un problème est apparu lors des accès au fichier
= 1 si aucun problème n'est apparu
Type : short int

DESCRIPTION :

Cette procédure réalise le chargement en mémoire image du gel 2D source contenu dans le fichier *nomgel.fim* , ainsi que la lecture du commentaire inclus dans ce fichier.

APPEL PRELIMINAIRE :

PROCEDURE : charger_gel2d_synth (nomgel, xo, yo, &fstat)

MODULE : Accès-BD

TYPE: Gestion de gels 2D synthétiques

PARAMETRES D'ENTREE : nomgel = pointeur sur la chaîne de caractères
contenant le nom d'un gel 2D
Type : pointeur

xo, yo = coordonnées du point en haut et à gauche
où doit se positionner l'image du gel
Type : short int

&fstat = pointeur sur l'entier qui contiendra
l'état des accès aux fichiers
Type : pointeur

PARAMETRES DE SORTIE : fstat = 0 si un problème est apparu lors des
accès aux fichiers
= 1 si aucun problème n'est apparu
Type : short int

DESCRIPTION : Cette procédure réalise le chargement en
mémoire image du gel 2D synthétique contenu
dans le fichier *nomgel.spt*

APPEL PRELIMINAIRE :

PROCEDURE : chercher_spot_coordonnee (&streamspt, xo ,yo,
 x, y, &fstat)

MODULE : Accès-BD

TYPE : Gestion de spots

PARAMETRES D'ENTREE : &stremspt = pointeur sur un fichier d'extension
 .spt
 Type : pointeur

 xo ,yo = coordonnées du point en haut et à
 gauche où est positionnée l'image du gel
 correspondant au fichier *.spt*
 Type : short int

 x, y = coordonnées d'un point, par rapport au
 coin supérieur gauche du gel
 Type : short int

 &fstat = pointeur sur l'entier qui recevra l'état
 des accès au fichier

PARAMETRES DE SORTIE : streamspt = fichier positionné au début de
 l'enregistrement contenant les données
 relatives au spot le plus proche du point de
 coordonnées x,y
 Type : FILE

 fstat = 0 si un problème est apparu lors des
 accès au fichier
 = 1 si aucun problème n'est apparu
 Type : short int

DESCRIPTION : Recherche, dans un fichier d'extension *.spt* , du
 spot le plus proche d'un point donné
 Le fichier est positionné au début de
 l'enregistrement contenant ce spot
 L'emplacement correspondant au spot clignote
 à l'écran vidéo

APPEL PRELIMINAIRE :

PROCEDURE : chercher_spot_nom (&streamspt, xo ,yo,
 nomspot, &fstat)

MODULE : Accès-BD

TYPE : Gestion de spots

PARAMETRES D'ENTREE :

 &streamspt = pointeur sur un fichier
 d'extension *.spt*
 Type : pointeur

 xo ,yo = coordonnées du point en haut et à
 gauche où est positionnée l'image du gel
 correspondant au fichier *.spt*
 Type : short int

 nomspot = pointeur sur la chaîne de caractères
 contenant le nom d'un spot au sein du gel
 Type : pointeur

 &fstat = pointeur sur l'entier qui recevra l'état
 des accès au fichier

PARAMETRES DE SORTIE :

 streamspt = fichier positionné au début de
 l'enregistrement contenant les données
 relatives au spot dont le nom a la valeur
 pointée par *nomspot*
 Type : FILE

 fstat = 0 si un problème est apparu lors des
 accès au fichier
 = 1 si aucun problème n'est apparu
 = 2 si le fichier ne contient aucun spot
 dont le nom a la valeur pointée par *nomspot*
 Type : short int

DESCRIPTION : Recherche, dans un fichier d'extension *.spt* , du
 spot désigné par un nom donné
 Le fichier est positionné au début de
 l'enregistrement contenant ce spot
 L'emplacement correspondant au spot clignote
 à l'écran vidéo

APPEL PRELIMINAIRE :

PROCEDURE : chercher_spot_numaut (&streamspt, x0,y0,
 numspotaut, &fstat)

MODULE : Accès-BD

TYPE : Gestion de spots

PARAMETRES D'ENTREE : &streamspt = pointeur sur un fichier d'extension
 .spt
 Type : pointeur

 x0,y0 = coordonnées du point en haut et à
 gauche où est positionnée l'image du gel
 correspondant au fichier *.spt*
 Type : short int

 numspotaut = numéro automatique d'un spot au
 sein du gel
 Type : long int

 &fstat = pointeur sur l'entier qui recevra l'état
 des accès au fichier

PARAMETRES DE SORTIE : streamspt = fichier positionné au début de
 l'enregistrement contenant les données
 relatives au spot dont le numéro automatique
 est *numspotaut*
 Type : FILE

 fstat = 0 si un problème est apparu lors des
 accès au fichier
 = 1 si aucun problème n'est apparu
 = 2 si le fichier ne contient aucun spot
 dont le numéro automatique est *numspotaut*
 Type : short int

DESCRIPTION : Recherche, dans un fichier d'extension *.spt*, du
 spot désigné par un numéro automatique donné
 Le fichier est positionné au début de
 l'enregistrement contenant ce spot
 L'emplacement correspondant au spot clignote
 à l'écran vidéo

APPEL PRELIMINAIRE :

PROCEDURE : chercher_spot_numman (&streamspt, xo ,yo,
numspotman, &fstat)

MODULE : Accès-BD

TYPE : Gestion de spots

PARAMETRES D'ENTREE : &streamspt = pointeur sur un fichier
d'extension *.spt*
Type : pointeur

 xo ,yo = coordonnées du point en haut et à
gauche où est positionnée l'image du gel
correspondant au fichier *.spt*
Type : short int

 numspotman = numéro manuel d'un spot au sein
du gel
Type : long int

 &fstat = pointeur sur l'entier qui recevra l'état
des accès au fichier

PARAMETRES DE SORTIE : streamspt = fichier positionné au début de
l'enregistrement contenant les données
relatives au spot dont le numéro manuel est
numspotman
Type : FILE

 fstat = 0 si un problème est apparu lors des
accès au fichier
 = 1 si aucun problème n'est apparu
 = 2 si le fichier ne contient aucun spot
dont le numéro manuel est *numspotman*
Type : short int

DESCRIPTION : Recherche, dans un fichier d'extension *.spt* , du
spot désigné par un numéro manuel donné
Le fichier est positionné au début de
l'enregistrement contenant ce spot
L'emplacement correspondant au spot clignote
à l'écran vidéo

APPEL PRELIMINAIRE :

PROCEDURE : clignoter_point (x0, y0,x,y)

MODULE : Vidéo

TYPE : Visualisation d'un point à l'écran vidéo

PARAMETRES D'ENTREE : x0, y0 = coordonnées du point en haut et à
gauche où est positionnée l'image d'un gel en 2D
Type : short int

 x,y = coordonnées d'un point, par rapport au coin
supérieur gauche de l'image du gel
Type : float

PARAMETRES DE SORTIE :

DESCRIPTION : Un carré gris, centré sur le point de coordonnées
x,y apparaît à l'écran vidéo pendant 2 secondes

APPEL PRELIMINAIRE :

PROCEDURE : commentaire__synth (nomgel, choix, mode,)

MODULE : Fonction

TYPE : Gestion de gels synthétiques

PARAMETRES D'ENTREE : nomgel = pointeur sur la chaîne de caractères
 contenant le nom d'un gel 2D
 Type : pointeur

 choix = 0 si le commentaire est celui du gel
 visualisé à l'écran vidéo : *nomgel*
 = 1 le commentaire est celui d'un autre gel
 que celui qui est visualisé à l'écran
 Type : short int

 mode = 'l' : lecture du commentaire
 'e' : écriture du commentaire
 Type : char

PARAMETRES DE SORTIE : Le message "erreur lecture ou "erreur écriture"
 est généré si un problème apparaît lors d'un
 accès (mode 'l' ou mode 'e') au fichier
 Type : string
 sortie à l'écran PC

DESCRIPTION : Cette procédure permet, selon le choix de
 l'utilisateur, de lire ou écrire le commentaire du
 gel visualisé à l'écran vidéo, ou d'un autre gel

APPEL PRELIMINAIRE : **vision_gel2d_synth** si choix = 0

```
PROCEDURE :      consult_etalon_int (nomgel, xo, yo, &prot,
                                &fstat)

MODULE :         Accès-BD

TYPE:            Gestion de l'étalon d'intensité

PARAMETRES D'ENTREE : nomgel = pointeur sur la chaîne de caractères
                        contenant le nom d'un gel 2D
                        Type : pointeur

                        xo, yo = coordonnées du point en haut et à gauche
                        où doit se positionner l'image du gel
                        Type : short int

                        &prot = pointeur sur la variable recevant les
                        informations relatives au spot étalon du gel
                        Type : pointeur

                        &fstat = pointeur sur l'entier qui contiendra
                        l'état de l'accès au fichier

PARAMETRES DE SORTIE : prot = les informations relatives au spot étalon
                          d'intensité pour le gel
                          Type : struct protéine

                          fstat = 0 si un problème est apparu lors de
                          l'accès au fichier
                              = 1 si aucun problème n'est apparu
                          Type : short int

DESCRIPTION :     Lecture des informations relatives au spot
                  étalon d'intensité d'un gel 2D.
                  L'emplacement correspondant au spot clignote à
                  l'écran vidéo.

APPEL PRELIMINAIRE :
```

PROCEDURE : consult_int_gel (nomgel, &intensité, &fstat)

MODULE : Accès-BD

TYPE: Gestion de gels synthétiques

PARAMETRES D'ENTREE : nomgel = pointeur sur la chaîne de caractères
contenant le nom d'un gel 2D
Type : pointeur

&intensité = pointeur sur la variable recevant la
valeur de l'intensité globale du gel
Type : pointeur

&fstat = pointeur sur l'entier qui contiendra
l'état de l'accès au fichier
Type : pointeur

PARAMETRES DE SORTIE : intensité = intensité globale du gel synthétique
Type : float

fstat = 0 si un problème est apparu lors de
l'accès au fichier
= 1 si aucun problème n'est apparu
Type : short int

DESCRIPTION : Lecture de l'intensité globale d'un gel
synthétique.

APPEL PRELIMINAIRE :

PROCEDURE : consult_point (nomgel, x, y, &pi, &pm, &fstat)

MODULE : Accès-BD

TYPE: Gestion de gels 2D synthétiques

PARAMETRES D'ENTREE : nomgel = pointeur sur la chaîne de caractères
contenant le nom d'un gel 2D
Type : pointeur

x,y = coordonnées d'un point par rapport au coin
supérieur gauche de l'image du gel
Type : short int

&pi = pointeur sur la variable recevant la valeur
de pi estimée en ce point
Type : pointeur

&pm = pointeur sur la variable recevant la
valeur de pM estimée en ce point
Type : pointeur

&fstat = pointeur sur l'entier recevant l'état des
accès au fichier
Type : pointeur

PARAMETRES DE SORTIE : pi = valeur de pi estimée au point de
coordonnées x,y
Type : float

pm = valeur de pM estimée au point de
coordonnées x,y
Type : float

fstat = 0 si un problème est apparu lors des
accès au fichier
= 1 si aucun problème n'est apparu
Type : short int

DESCRIPTION : Evaluation du pi et du pM en un point donné d'un
gel 2D

APPEL PRELIMINAIRE :

```
PROCEDURE :      consult_spot_cooronnee (nomgel, xo, yo,x, y,  
                                &prot, &fstat)  
  
MODULE :         Accès-BD  
  
TYPE:            Gestion de spots  
  
PARAMETRES D'ENTREE : nomgel = pointeur sur la chaîne de caractères  
                        contenant le nom d'un gel 2D  
                        Type : pointeur  
  
                  xo, yo = coordonnées du point en haut et à gauche  
                          où est positionnée l'image du gel  
                          Type : short int  
  
                  x,y = coordonnées d'un point, par rapport au coin  
                        supérieur gauche de l'image du gel  
                        Type : short int  
  
                  &prot = pointeur sur la variable recevant les  
                           informations relatives au spot  
                           Type : pointeur  
  
                  &fstat = pointeur sur l'entier qui contiendra  
                            l'état des accès aux fichiers  
  
PARAMETRES DE SORTIE : prot = les informations relatives au spot du gel  
                          nomgel, le plus proche du point de coordonnées  
                          x,y  
                          Type : struct proteine  
  
                  fstat = 0 si un problème est apparu lors des  
                          accès aux fichiers  
                         =1 si aucun problème n'est apparu  
                          Type : short int  
  
DESCRIPTION :     Lecture des informations relatives au spot le  
                  plus proche d'un point donné au sein d'un gel 2D.  
                  L'emplacement correspondant au spot clignote à  
                  l'écran vidéo.
```

APPEL PRELIMINAIRE :

```

PROCEDURE :      consult_spot_nom (nomgel, xo, yo, nomspot,
                                &prot, &fstat)

MODULE :         Accès-BD

TYPE:            Gestion de spots

PARAMETRES D'ENTREE :  nomgel = pointeur sur la chaîne de caractères
                                contenant le nom d'un gel 2D
                                Type : pointeur

                                xo, yo = coordonnées du point en haut et à gauche
                                où doit se positionner l'image du gel
                                Type : short int

                                nomspot = pointeur sur la chaîne de caractères
                                contenant le nom du spot
                                Type : pointeur

                                &prot = pointeur sur la variable recevant les
                                informations relatives au spot
                                Type : pointeur

                                &fstat = pointeur sur l'entier qui contiendra
                                l'état des accès aux fichiers

PARAMETRES DE SORTIE :  prot = les informations relatives au spot du gel
                                nomgel , désigné par nomspot
                                Type : struct proteine

                                fstat = 0 si un problème est apparu lors des
                                accès aux fichiers
                                        = 1 si aucun problème n'est apparu
                                Type : short int

DESCRIPTION :      Lecture des informations relatives à un spot
                                désigné par son nom au sein d'un gel 2D
                                L'emplacement correspondant au spot clignote à
                                l'écran vidéo.

```

PROCEDURE : consult_spot_numaut (nomgel, xo, yo,
 numspotaut, &prot, &fstat)

MODULE : Accès-BD

TYPE: Gestion de spots

PARAMETRES D'ENTREE : nomgel = pointeur sur la chaîne de caractères
 contenant le nom d'un gel 2D
 Type : pointeur

 xo, yo = coordonnées du point en haut et à gauche
 où est positionnée l'image du gel
 Type : short int

 numspotaut = numéro automatique du spot
 Type : long int

 &prot = pointeur sur la variable recevant les
 informations relatives au spot
 Type : pointeur

 &fstat = pointeur sur l'entier qui contiendra
 l'état des accès aux fichiers

PARAMETRES DE SORTIE : prot = les informations relatives au spot du gel
 nomgel , désigné par numspotaut
 Type : struct proteine

 fstat = 0 si un problème est apparu lors des
 accès aux fichiers
 = 1 si aucun problème n'est apparu
 Type : short int

DESCRIPTION : Lecture des informations relatives à un spot
 désigné par son numéro automatique au sein d'un
 gel 2D
 L'emplacement correspondant au spot clignote à
 l'écran vidéo.

APPEL PRELIMINAIRE :

PROCEDURE : `consult_spot_numman (nomgel, xo, yo, numspotman, &prot, &fstat)`

MODULE : Accès-BD

TYPE: Gestion de spots

PARAMETRES D'ENTREE :

`nomgel` = pointeur sur la chaîne de caractères contenant le nom d'un gel 2D
Type : pointeur

`xo, yo` = coordonnées du point en haut et à gauche où doit se positionner l'image du gel
Type : short int

`numspotman` = numéro manuel du spot
Type : long int

`&prot` = pointeur sur la variable recevant les informations relatives au spot
Type : pointeur

`&fstat` = pointeur sur l'entier qui contiendra l'état des accès aux fichiers

PARAMETRES DE SORTIE :

`prot` = les informations relatives au spot du gel *nomgel* , désigné par `numspotman`
Type : struct protéine

`fstat` = 0 si un problème est apparu lors des accès aux fichiers
 = 1 si aucun problème n'est apparu
Type : short int

DESCRIPTION :

Lecture des informations relatives à un spot désigné par son numéro manuel au sein d'un gel 2D
L'emplacement correspondant au spot clignote à l'écran vidéo.

APPEL PRELIMINAIRE :

PROCEDURE : coupure ()

MODULE : Extraction

TYPE : Definition de taches

PARAMETRES D'ENTREE : les coordonnées de points qui délimitent le
 pourtour d'une tache au sein d'un gel visualisé à
 l'écran vidéo
 Type : short int
 entrée au clavier

PARAMETRES DE SORTIE : les points (pixels) qui joignent sur l'écran vidéo
 les points en entrée ont une intensité nulle

DESCRIPTION : Cette procédure permet à l'utilisateur de
 scinder une ou plusieurs taches en différentes
 parties en y traçant des segments constitués de
 points d'intensité nulle

APPEL PRELIMINAIRE : **background**

PROCEDURE :	demasquer (nomgel, quad, x0, y0)
MODULE :	Fonction
TYPE :	Gestion d'artéfacts
PARAMETRES D'ENTREE :	nomgel = pointeur sur la chaîne de caractères contenant le nom d'un gel 2D Type : pointeur quad = numéro du quadrant où est visualisé l'image du gel Type : short int x0, y0 = coordonnées du point en haut et à gauche où est positionnée l'image du gel. Ce point doit être situé dans le quadrant <i>quad</i> Type : short int x1 , y1 = coordonnées du point supérieur gauche de la zone à démasquer Type : short int entrée à la souris x2,y2 = coordonnées du point inférieur droit de la zone à démasquer Type : short int entrée à la souris
PARAMETRES DE SORTIE :	Le message "erreur lecture/écriture" est généré si un problème apparaît lors d'un accès au fichier Type : string sortie à l'écran PC
DESCRIPTION :	Cette procédure permet à l'utilisateur de définir une zone au sein du gel synthétique <i>nomgel</i> dans laquelle les spots éventuellement masqués sont démasqués. L'intensité globale du gel est mise à jour
APPEL PRELIMINAIRE :	

PROCEDURE : demasq_artef_nom (nomgel, quad, x0, y0)

MODULE : Fonction

TYPE : Gestion d'artéfacts

PARAMETRES D'ENTREE : nomgel = pointeur sur la chaîne de caractères
contenant le nom d'un gel 2D
Type : pointeur

quad = numéro du quadrant où est visualisé
l'image du gel
Type : short int

x0, y0 = coordonnées du point en haut et à gauche
où est positionnée l'image du gel. Ce point doit
être situé dans le quadrant *quad*
Type : short int

nomspot = pointeur sur la chaîne de caractères
contenant le nom d'un spot du gel
Type : pointeur
entrée au clavier

PARAMETRES DE SORTIE : le message "erreur lecture/écriture" est généré
si un problème est apparu lors des accès au
fichier
sortie à l'écran PC

DESCRIPTION : Cette procédure permet à l'utilisateur de
démasquer au sein d'un gel 2D un spot identifié
par son nom
L'intensité globale du gel est mise à jour

APPEL PRELIMINAIRE :

PROCEDURE : demasq_artef_numaut (nomgel, quad, x0, y0)

MODULE : Fonction

TYPE : Gestion d'artéfacts

PARAMETRES D'ENTREE :

- nomgel = pointeur sur la chaîne de caractères contenant le nom d'un gel 2D
Type : pointeur
- quad = numéro du quadrant où est visualisé l'image du gel
Type : short int
- x0, y0 = coordonnées du point en haut et à gauche où est positionnée l'image du gel. Ce point doit être situé dans le quadrant *quad*
Type : short int
- numspotaut = numéro automatique d'un spot du gel
Type : long int
entrée au clavier

PARAMETRES DE SORTIE : le message "erreur lecture/écriture" est généré si un problème est apparu lors des accès au fichier
sortie à l'écran PC

DESCRIPTION : Cette procédure permet à l'utilisateur de démasquer au sein d'un gel 2D un spot identifié par son numéro automatique
L'intensité globale du gel est mise à jour

APPEL PRELIMINAIRE :

PROCEDURE : demasq_artef_numman (nomgel, quad, x0, y0)

MODULE : Fonction

TYPE : Gestion d'artéfacts

PARAMETRES D'ENTREE : nomgel = pointeur sur la chaîne de caractères
contenant le nom d'un gel 2D
Type : pointeur

quad = numéro du quadrant où est visualisé
l'image du gel
Type : short int

x0, y0 = coordonnées du point en haut et à gauche
où est positionnée l'image du gel. Ce point doit
être situé dans le quadrant *quad*
Type : short int

numspotman = numéro manuel d'un spot du gel
Type : long int
entrée au clavier

PARAMETRES DE SORTIE : le message "erreur lecture/écriture" est généré
si un problème est apparu lors des accès au
fichier
sortie à l'écran PC

DESCRIPTION : Cette procédure permet à l'utilisateur de
démasquer au sein d'un gel 2D un spot identifié
par son numéro manuel
L'intensité globale du gel est mise à jour

APPEL PRELIMINAIRE :

PROCEDURE : demasquer_spot_numaut (nomgel, xo, yo, numspotaut, &fstat)

MODULE : Accès-BD

TYPE: Gestion d'artéfacts

PARAMETRES D'ENTREE : nomgel = pointeur sur la chaîne de caractères contenant le nom d'un gel 2D
Type : pointeur

xo, yo = coordonnées du point en haut et à gauche où est positionnée l'image du gel
Type : short int

numspotaut : numéro automatique du spot
Type : long int

&fstat = pointeur sur l'entier qui contiendra l'état des accès au fichier
Type : pointeur

PARAMETRES DE SORTIE : fstat = 0 si un problème est apparu lors des accès au fichier
= 1 si aucun problème n'est apparu

DESCRIPTION : Démasquage d'un spot désigné par son numéro automatique au sein d'un gel 2D.

APPEL PRELIMINAIRE :

PROCEDURE : demasquer_spot_numman (nomgel, xo, yo, numspotman, &fstat)

MODULE : Accès-BD

TYPE: Gestion d'artéfacts

PARAMETRES D'ENTREE : nomgel = pointeur sur la chaîne de caractères contenant le nom d'un gel 2D
Type : pointeur

xo, yo = coordonnées du point en haut et à gauche où est positionnée l'image du gel
Type : short int

numspotman = numéro manuel du spot
Type : long int

&fstat = pointeur sur l'entier qui contiendra l'état des accès au fichier
Type : pointeur

PARAMETRES DE SORTIE : fstat = 0 si un problème est apparu lors des accès au fichier
= 1 si aucun problème n'est apparu

DESCRIPTION : Démasquage d'un spot désigné par son numéro manuel au sein d'un gel 2D.

APPEL PRELIMINAIRE :

PROCEDURE : demasquer_zone (nomgel, x0, y0, x1, y1, x2, y2, &fstat)

MODULE : Accès-BD

TYPE: Gestion d'artéfacts

PARAMETRES D'ENTREE : nomgel = pointeur sur la chaîne de caractères contenant le nom d'un gel 2D
Type : pointeur

x0, y0 = coordonnées du point en haut et à gauche où est positionnée l'image du gel
Type : short int

x1, y1 = coordonnées du coin supérieur gauche de la zone du gel à démasquer, par rapport au coin supérieur gauche de l'image du gel
Type : short int

x2, y2 = coordonnées du coin inférieur droit de la zone du gel à démasquer, par rapport au coin supérieur gauche de l'image du gel
Type : short int

&fstat = pointeur sur l'entier qui contiendra l'état des accès au fichier
Type : pointeur

PARAMETRES DE SORTIE : fstat = 0 si un problème est apparu lors des accès au fichier
= 1 si aucun problème n'est apparu
Type : short int

DESCRIPTION : Démasquage des spots masqués au sein d'une zone définie dans un gel 2D.

APPEL PRELIMINAIRE :

PROCEDURE : detectcentre (seuil, factm, factd, tab, long, ¢re, &ncentre)

MODULE : Extraction

TYPE : Gestion de listes de pixels

PARAMETRES D'ENTREE :

seuil = niveau de gris que doit atteindre un centre de pic pour être considéré en tant que tel
Type : short

factm = facteur de sensibilité pour la détection des points d'inflexion sur la face ascendante d'un pic
Type : float

factd = facteur de sensibilité pour la détection des points d'inflexion sur la face descendante d'un pic
Type : float

tab = pointeur sur un tableau de 3 X *maxcol* entiers courts. Chaque colonne de ce tableau contient les valeurs col, ligne et intensité d'un pixel
Type : pointeur

long = nombre de pixels introduits dans le tableau *tab*
Type : short int

¢re = pointeur sur un pointeur de liste de pixels
Type : pointeur

&ncentre = pointeur sur la variable recevant le nombre de centres de pics détectés dans le tableau *tab*

PARAMETRES DE SORTIE : le contenu de la liste pointée par ¢re est augmenté des structures *pixel* qui contiennent les informations relatives aux centres de pics/spots détectés

ncentre = nombre de centres de pics détectés
par la procédure
Type : short int

DESCRIPTION :

Détection des centres de pics parmi une suite de
pixels mémorisés dans le tableau *tab*
Chaque centre de pic est mémorisé dans une
structure *pixel* qui est ajoutée au contenu de la
liste pointée par *¢re*

APPEL PRELIMINAIRE :

PROCEDURE : ecran_prot (prot)

MODULE : Fonction

TYPE : Affichage de données

PARAMETRES D'ENTREE : nomgel = pointeur sur la chaîne de caractères
 contenant le nom d'un gel 2D
 Type : pointeur

 prot = l'ensemble des renseignements relatifs à
 un spot de protéines
 Type : structure protéine

PARAMETRES DE SORTIE : Les informations contenues dans la variable
 spot
 Type : struct protéine
 sortie à l'écran PC

DESCRIPTION : Affichage à l'écran PC des informations
 contenues dans une variable de type *protéine*

APPEL PRELIMINAIRE : L'une des procédures suivantes :
 info_spot_souris
 info_spot_nom
 info_spot_numman
 info_spot_numaut
 info_etalon_int

```
PROCEDURE :      ecrire_com2d_synth_ (nomgel, commentaire,  
                                     &fstat)  
  
MODULE :         accès-BD  
  
TYPE :          Gestion de gels 2D synthétiques  
  
PARAMETRES D'ENTREE : nomgel = pointeur sur la chaîne de caractères  
                       contenant le nom du gel 2D  
                       Type : pointeur  
  
                       commentaire = pointeur sur la chaîne de  
                       caractères contenant le commentaire à écrire  
                       dans le fichier nomgel.ref  
                       Type : pointeur  
  
                       &fstat = pointeur sur l'entier qui contiendra  
                       l'état de l'accès au fichier  
                       Type : pointeur  
  
PARAMETRES DE SORTIE : fstat = 0 si un problème est apparu lors des  
                       accès au fichier  
                       = 1 si aucun problème n'est apparu  
                       Type : short int  
  
DESCRIPTION :    Sauvegarde d'un commentaire relatif à un gel 2D  
                 synthétique.  
  
APPEL PRELIMINAIRE :
```

PROCEDURE : `ecrire_etat_int (&streamspt, &streamref, &fstat)`

MODULE : Accès-BD

TYPE : Gestion de l'étalon d'intensité

PARAMETRES D'ENTREE :

- `&streamspt` = pointeur sur un fichier d'extension `.spt` positionné au début d'un enregistrement
Type : pointeur
- `&streamref` = pointeur sur le fichier d'extension `.ref` portant le même nom que le fichier `streamspt`
Type : pointeur
- `&fstat` = pointeur sur l'entier qui recevra l'état des accès aux fichiers
Type : pointeur

PARAMETRES DE SORTIE :

- `streamspt` = fichier `.spt` positionné en fin de l'enregistrement
Type : FILE
- `streamref` = fichier `.ref` dans lequel le contenu de l'enregistrement lu dans le fichier `.spt` a été recopié à l'endroit réservé au spot étalon d'intensité (cf structure du fichier `.ref`)
Le fichier est positionné à la fin de l'enregistrement écrit.
- `fstat` = 0 si un problème est apparu lors des accès aux fichiers
 = 1 si aucun problème n'est apparu
Type : short int

DESCRIPTION :

L'information relative à un spot contenu dans le fichier `.spt` est recopiée dans l'enregistrement du fichier `.ref` destiné à contenir l'information relative au spot étalon d'intensité du gel 2D qui correspond à ces fichiers

APPEL PRELIMINAIRE :

Une des procédures suivantes :

chercher_spot_cooronnee

chercher_spot_nom

chercher_spot_numaut

chercher_spot_numman

PROCEDURE : ^{S51} `ecrire_etat_int_cooronnee (nomgel, xo, yo, x, y, &fstat)`

MODULE : Accès-BD

TYPE : Gestion de l'étalon d'intensité.

PARAMETRES D'ENTREE :

`nomgel` = pointeur sur la chaîne de caractères contenant le nom du gel 2D
 Type : pointeur

`xo, yo` = coordonnées du point en haut à gauche où est positionnée l'image du gel
 Type : short int

`x, y` = coordonnées d'un point, par rapport au coin supérieur gauche de l'image du gel
 Type : short int

`&fstat` = pointeur sur l'entier qui contiendra l'état des accès aux fichiers
 Type : pointeur

PARAMETRES DE SORTIE :

`fstat` = 0 si un problème est apparu lors de l'accès aux fichiers
 = 1 si aucun problème n'est apparu
 Type = short int

DESCRIPTION :

Le spot le plus proche d'un point donné est enregistré en tant qu'étalon d'intensité pour un gel 2D. L'emplacement correspondant au spot clignote à l'écran vidéo.

APPEL PRELIMINAIRE :

PROCEDURE : écrire_etal_int_nom (nomgel, xo, yo, nomspot,
&fstat)

MODULE : Accès-BD

TYPE : Gestion de l'étalon d'intensité

PARAMETRES D'ENTREE :

- nomgel = pointeur sur la chaîne de caractères contenant le nom du gel 2D
Type : pointeur
- xo, yo = coordonnées du point en haut à gauche où est positionnée l'image du gel
Type : short int
- nomspot = pointeur sur la chaîne de caractères contenant le nom du spot
Type : pointeur
- &fstat = pointeur sur l'entier qui contiendra l'état des accès aux fichiers
Type : pointeur

PARAMETRES DE SORTIE :

- fstat = 0 si un problème est apparu lors de l'accès aux fichiers
= 1 si aucun problème n'est apparu
Type = short int

DESCRIPTION : Un spot désigné par son nom est enregistré en tant qu'étalon d'intensité pour un gel 2D. L'emplacement correspondant au spot clignote à l'écran vidéo.

APPEL PRELIMINAIRE :

PROCEDURE : `ecrire_etat_int_numaut (nomgel, x0, y0, numspotaut, &fstat)`

MODULE : Accès-BD

TYPE : Gestion de l'étalon d'intensité

PARAMETRES D'ENTREE :

`nomgel` = pointeur sur la chaîne de caractères contenant le nom du gel 2D
 Type : pointeur

`x0, y0` = coordonnées du point en haut à gauche où est positionnée l'image du gel
 Type : short int

`numspotaut` = numéro automatique du spot
 Type : long int

`&fstat` = pointeur sur l'entier qui contiendra l'état des accès aux fichiers
 Type : pointeur

PARAMETRES DE SORTIE :

`fstat` = 0 si un problème est apparu lors de l'accès aux fichiers
 = 1 si aucun problème n'est apparu
 Type = short int

DESCRIPTION : Un spot désigné par son numéro automatique est enregistré en tant qu'étalon d'intensité pour un gel 2D. L'emplacement correspondant au spot clignote à l'écran vidéo.

APPEL PRELIMINAIRE :

PROCEDURE : `ecrire_etat_int_numman (nomgel, xo, yo, numspotman, &fstat)`

MODULE : Accès-BD

TYPE : Gestion de l'étalon d'intensité

PARAMETRES D'ENTREE :

`nomgel` = pointeur sur la chaîne de caractères contenant le nom du gel 2D
Type : pointeur

`xo, yo` = coordonnées du point en haut à gauche où est positionnée l'image du gel
Type : short int

`numspotman` = numéro manuel du spot
Type : long int

`&fstat` = pointeur sur l'entier qui contiendra l'état des accès aux fichiers
Type : pointeur

PARAMETRES DE SORTIE :

`fstat` = 0 si un problème est apparu lors de l'accès aux fichiers
 = 1 si aucun problème n'est apparu
Type = short int

DESCRIPTION : Un spot désigné par son numéro manuel est enregistré en tant qu'étalon d'intensité pour un gel 2D. L'emplacement correspondant au spot clignote à l'écran vidéo.

APPEL PRELIMINAIRE :

PROCEDURE : `ecrire_etalon_pi_(nomgel, pi, x, y, numetalon, &fstat)`

MODULE : Accès-BD

TYPE : Gestion des étalons de point isoélectrique

PARAMETRES D'ENTREE :

`nomgel` = pointeur sur la chaîne de caractères contenant le nom du gel 2D
Type : pointeur

`pi` = valeur de pI de l'étalon
Type : float

`x, y` = coordonnées de l'étalon, par rapport au coin supérieur gauche de l'image du gel
Type : float

`numetalon` = numéro de l'étalon du spot : 1 ou 2
Type : short int

`&fstat` = pointeur sur l'entier qui contiendra l'état de l'accès au fichier

PARAMETRES DE SORTIE :

`fstat` = 0 si un problème est apparu lors de l'accès au fichier
 = 1 si aucun problème n'est apparu
Type : short int

DESCRIPTION : Enregistrement dans le fichier *nomgel.ref* des informations relatives à un des deux points étalons de pI dans le gel 2D.

APPEL PRELIMINAIRE :

PROCEDURE : `ecrire_etalon_pm_(nomgel, pm, x, y, numetalon, &fstat)`

MODULE : Accès-BD

TYPE : Gestion des étalons de point isoélectrique

PARAMETRES D'ENTREE :

`nomgel` = pointeur sur la chaîne de caractères contenant le nom du gel 2D
Type : pointeur

`pm` = valeur de pM de l'étalon
Type : float

`x, y` = coordonnées de l'étalon, par rapport au coin supérieur gauche de l'image du gel
Type : float

`numetalon` = numéro de l'étalon du spot : 1 ou 2
Type : short int

`&fstat` = pointeur sur l'entier qui contiendra l'état de l'accès au fichier

PARAMETRES DE SORTIE :

`fstat` = 0 si un problème est apparu lors de l'accès au fichier
 = 1 si aucun problème n'est apparu
Type : short int

DESCRIPTION : Enregistrement dans le fichier *nomgel.ref* des informations relatives à un des deux points étalons de pM dans le gel 2D.

APPEL PRELIMINAIRE :

PROCEDURE : `ecrire_nomspot_coordonnée (nomgel, xo,yo,x, y, nomspot, &fstat)`

MODULE : Accès-BD

TYPE : Gestion de spots

PARAMETRES D'ENTREE : `nomgel` = pointeur sur la chaîne de caractères contenant le nom du gel 2D
Type : pointeur

`xo, yo` = coordonnées d'un point en haut à gauche où est positionnée l'image du gel
Type : short int

`x, y` = coordonnées d'un point, par rapport au coin supérieur gauche de l'image du gel
Type : short int

`nomspot` = pointeur sur la chaîne de caractères contenant le nom à attribuer au spot
Type : pointeur

`&fstat` = pointeur sur l'entier qui contiendra l'état des accès au fichier
Type : pointeur

PARAMETRES DE SORTIE : `fstat` = 0 si un problème est apparu lors des accès au fichier
 = 1 si aucun problème n'est apparu
Type : short int

DESCRIPTION : Affectation d'un nom au spot le plus proche d'un point donné au sein d'un gel 2D. L'emplacement correspondant au spot clignote à l'écran vidéo.

APPEL PRELIMINAIRE :

PROCEDURE : `ecrire_nomspot_nom (nomgel, xo,yo, nomspot, nvnomspot, &fstat)`

MODULE : Accès-BD

TYPE : Gestion de spots

PARAMETRES D'ENTREE : `nomgel` = pointeur sur la chaîne de caractères contenant le nom du gel 2D
Type : pointeur

`xo, yo` = coordonnées du point en haut à gauche où est positionnée l'image du gel
Type : short int

`nomspot` = pointeur sur la chaîne de caractères contenant le nom du spot
Type : pointeur

`nvnomspot` = pointeur sur la chaîne de caractères contenant le nouveau nom à attribuer au spot
Type : pointeur

`&fstat` = pointeur sur l'entier qui contiendra l'état des accès au fichier
Type : pointeur

PARAMETRES DE SORTIE : `fstat` = 0 si un problème est apparu lors des accès au fichier
 = 1 si aucun problème n'est apparu
Type : short int

DESCRIPTION : Affectation d'un nom à un spot désigné par son nom au sein d'un gel 2D. L'emplacement correspondant au spot clignote à l'écran vidéo.

APPEL PRELIMINAIRE :

PROCEDURE : écrire_nomspot_numaut (nomgel, xo,yo, numspotaut,nomspot, &fstat)

MODULE : Accès-BD

TYPE : Gestion de spots

PARAMETRES D'ENTREE : nomgel = pointeur sur la chaîne de caractères contenant le nom du gel 2D
Type : pointeur

xo, yo = coordonnées d'un point en haut à gauche où est positionné l'image du gel
Type : short int

numspotaut = numéro automatique du spot
Type : long int

nomspot = pointeur sur la chaîne de caractères contenant le nom à attribuer spot
Type : pointeur

&fstat = pointeur sur l'entier qui contiendra l'état des accès au fichier
Type : pointeur

PARAMETRES DE SORTIE : fstat = 0 si un problème est apparu lors des accès au fichier
= 1 si aucun problème n'est apparu
Type : short int

DESCRIPTION : Affectation d'un nom à un spot désigné par son numéro automatique au sein d'un gel 2D. L'emplacement correspondant au spot clignote à l'écran vidéo.

APPEL PRELIMINAIRE :

PROCEDURE : `ecrire_nomspot_numman (nomgel, x0,y0, numspotman, nomspot, &fstat)`

MODULE : Accès-BD

TYPE : Gestion de spots

PARAMETRES D'ENTREE :

`nomgel` = pointeur sur la chaîne de caractères contenant le nom du gel 2D
 Type : pointeur

`x0, y0` = coordonnées d'un point en haut à gauche où est positionné l'image du gel
 Type : short int

`numspotman` = numéro manuel du spot
 Type = long int

`nomspot` = pointeur sur la chaîne de caractères contenant le nouveau nom à attribuer spot
 Type : pointeur

`&fstat` = pointeur sur l'entier qui contiendra l'état des accès au fichier
 Type : pointeur

PARAMETRES DE SORTIE :

`fstat` = 0 si un problème est apparu lors des accès au fichier
 = 1 si aucun problème n'est apparu
 Type : short int

DESCRIPTION :

 Affectation d'un nom à un spot désigné par son numéro manuel au sein d'un gel 2D.
 L'emplacement correspondant au spot clignote à l'écran vidéo.

APPEL PRELIMINAIRE :

PROCEDURE : écrire_numspot_cooronnée (nomgel, xo,yo,x, y,
 numspot, &fstat)

MODULE : Accès-BD

TYPE : Gestion de spots

PARAMETRES D'ENTREE : nomgel = pointeur sur la chaîne de caractères
 contenant le nom du gel 2D
 Type : pointeur

 xo, yo = coordonnées du point en haut à gauche où
 est positionnée l'image du gel
 Type : short int

 x, y = coordonnées d'un point, par rapport au coin
 supérieur gauche de l'image du gel
 Type : short int

 numspot =numéro manuel à attribuer au spot
 Type : long int

 &fstat = pointeur sur l'entier qui contiendra
 l'état des accès au fichier
 Type : pointeur

PARAMETRES DE SORTIE : fstat = 0 si un problème est apparu lors des
 accès au fichier
 = 1 si aucun problème n'est apparu
 Type : short int

DESCRIPTION : Affectation d'un numéro manuel au spot le plus
 proche d'un point donné au sein d'un gel 2D.
 L'emplacement correspondant au spot clignote à
 l'écran vidéo.

APPEL PRELIMINAIRE :

PROCEDURE : `ecrire_numspot_nom (nomgel, xo,yo, nomspot, numspot, &fstat)`

MODULE : Accès-BD

TYPE : Gestion de spots

PARAMETRES D'ENTREE :

`nomgel` = pointeur sur la chaîne de caractères contenant le nom du gel 2D
 Type : pointeur

`xo, yo` = coordonnées d'un point en haut à gauche où est positionné l'image du gel
 Type : short int

`nomspot` = pointeur sur la chaîne de caractères contenant le nom du spot
 Type : pointeur

`numspot` = numéro manuel à attribuer au spot
 Type: long int

`&fstat` = pointeur sur l'entier qui contiendra l'état des accès au fichier
 Type : pointeur

PARAMETRES DE SORTIE :

`fstat` = 0 si un problème est apparu lors des accès au fichier
 = 1 si aucun problème n'est apparu
 Type : short int

DESCRIPTION : Affectation d'un numéro manuel à un spot désigné par son nom au sein d'un gel 2D.
 L'emplacement correspondant au spot clignote à l'écran vidéo.

APPEL PRELIMINAIRE :

PROCEDURE : `ecrire_numspot_numaut (nomgel, xo,yo, numspotaut, numspot, &fstat)`

MODULE : Accès-BD

TYPE : Gestion de spots

PARAMETRES D'ENTREE : `nomgel` = pointeur sur la chaîne de caractères contenant le nom du gel 2D
Type : pointeur

`xo, yo` = coordonnées d'un point en haut à gauche où est positionné l'image du gel
Type : short int

`numspotaut` = numéro automatique du spot
Type : long int

`numspot` =numéro manuel à attribuer au spot
Type : long int

`&fstat` = pointeur sur l'entier qui contiendra l'état des accès au fichier
Type : pointeur

PARAMETRES DE SORTIE : `fstat` = 0 si un problème est apparu lors des accès au fichier
 = 1 si aucun problème n'est apparu
Type : short int

DESCRIPTION : Affectation d'un numéro manuel à un spot désigné par son numéro automatique au sein d'un gel 2D. L'emplacement correspondant au spot clignote à l'écran vidéo.

APPEL PRELIMINAIRE :

```

PROCEDURE :      ecrire_numspot_numman (nomgel, xo,yo,
                    numspotman, numspot, &fstat)

MODULE :         Accès-BD

TYPE :           Gestion de spots

PARAMETRES D'ENTREE :  nomgel = pointeur sur la chaîne de caractères
                        contenant le nom du gel 2D
                        Type : pointeur

                        xo, yo = coordonnées d'un point en haut à gauche
                        où est positionné l'image du gel
                        Type : short int

                        numspotman = numéro manuel du spot
                        Type = long int

                        numspot = numéro manuel à attribuer au spot
                        Type : long int

                        &fstat = pointeur sur l'entier qui contiendra
                        l'état des accès au fichier
                        Type : pointeur

PARAMETRES DE SORTIE : fstat = 0 si un problème est apparu lors des
                        accès au fichier
                        = 1 si aucun problème n'est apparu
                        Type : short int

DESCRIPTION :      Affectation d'un numéro manuel à un spot
                    désigné par son numéro manuel au sein d'un gel
                    2D. L'emplacement correspondant au spot
                    clignote à l'écran vidéo.

```

PROCEDURE : `ecrire_spot_nom (&streamspt, nomspot, &fstat)`

MODULE : Accès-BD

TYPE : Gestion de spots

PARAMETRES D'ENTREE :

`&streamspt` = pointeur sur un fichier
 d'extension *.spt* positionné au début d'un
 enregistrement
 Type : pointeur

`nomspot` = pointeur sur la chaîne de caractères
 contenant le nom à attribuer au spot
 Type : pointeur

`&fstat` = pointeur sur l'entier qui recevra l'état
 des accès aux fichiers
 Type : pointeur

PARAMETRES DE SORTIE :

`streamspt` = fichier *.spt* dans lequel le nom du
 spot contenu dans l'enregistrement sur lequel
 il était positionné a pris la valeur pointée par
 nomspot
 Le fichier est positionné à la fin de
 l'enregistrement

`fstat` = 0 si un problème est apparu lors des
 accès au fichier
 = 1 si aucun problème n'est apparu
 Type : short int

DESCRIPTION :

APPEL PRELIMINAIRE : Une des procédures suivantes :

chercher_spot_cooronnee
 chercher_spot_nom
 chercher_spot_numaut
 chercher_spot_numman

PROCEDURE : `ecrire_spot_numman (&streamspt, numspotman, &fstat)`

MODULE : Accès-BD

TYPE : Gestion de spots

PARAMETRES D'ENTREE :

`&streamspt` = pointeur sur un fichier d'extension *.spt* positionné au début d'un enregistrement
 Type : pointeur

`numspotman` = numéro manuel à attribuer au spot
 Type : long int

`&fstat` = pointeur sur l'entier qui recevra l'état des accès aux fichiers
 Type : pointeur

PARAMETRES DE SORTIE :

`streamspt` = fichier *.spt* dans lequel le numéro manuel du spot contenu dans l'enregistrement sur lequel il était positionné a pris la valeur *numspotman*
 Le fichier est positionné à la fin de l'enregistrement

`fstat` = 0 si un problème est apparu lors des accès au fichier
 = 1 si aucun problème n'est apparu
 Type : short int

DESCRIPTION :

APPEL PRELIMINAIRE : Une des procédures suivantes :
 chercher_spot_cooronnee
 chercher_spot_nom
 chercher_spot_numaut
 chercher_spot_numman

PROCEDURE :	e_val_pi (x, y, pi1, x1, y1, pi2, x2, y2, &pi)
MODULE :	Accès-BD
TYPE :	Gestion des points isoélectriques
PARAMETRES D'ENTREE :	x,y = coordonnées d'un point Type : float pi1 = valeur de pi du 1er étalon de pi Type : float x1, y1 = coordonnées du 1er étalon de pi Type : float pi2 = valeur de pi du 2ème étalon de pi Type : float x2, y2 = coordonnées du 2ème étalon de pi Type : float &pi = pointeur sur la variable recevant la valeur de pi estimée au point de coordonnée x,y Type : pointeur
PARAMETRES DE SORTIE :	pi = valeur de pi estimée au point de coordonnée x,y Type : float
DESCRIPTION :	Détermination du pi en un point d'un gel sur base de deux points étalons de pi Toutes les coordonnées sont relatives au coin supérieur gauche de l'image du gel.
APPEL PRELIMINAIRE :	

PROCEDURE : eval_pm (x, y, pm1, x1, y1, pm2, x2, y2, &pm)

MODULE : Accès-BD

TYPE : Gestion des poids moléculaires

PARAMETRES D'ENTREE : x,y = coordonnées d'un point
Type : float

pm1 = valeur de pM du 1er étalon de pM
Type : float

x1, y1 = coordonnées du 1er étalon de pM
Type : float

pm2 = valeur de pM du 2ème étalon de pM
Type : float

x2, y2 = coordonnées du 2ème étalon de pM
Type : float

&pm = pointeur sur la variable recevant la
valeur de pM estimée au point de coordonnée
x,y
Type : pointeur

PARAMETRES DE SORTIE : pm = valeur de pM estimée au point de
coordonnée x,y
Type : float

DESCRIPTION : Détermination du pM en un point d'un gel sur
base de deux points étalons de pM
Toutes les coordonnées sont relatives au coin
supérieur gauche de l'image du gel.

APPEL PRELIMINAIRE :

PROCEDURE : evalcrit (tache, taille, nspot, ldistrib, &total)

MODULE : Extraction

TYPE : Modélisation gaussienne

PARAMETRES D'ENTREE :

- tache = pointeur sur le tableau de structures *point* contenant les données relatives aux pixels d'une tache, la valeur du background moyen de la tache ayant été soustraite de l'intensité de chaque point
Type : pointeur
- taille = nombre de points constituant la tache
Type : unsigned short
- nspot = nombre de centres de spots détectés dans la tache
Type : short int
- ldistrib = pointeur sur le tableau de structures *distrib* contenant les paramètres des distributions gaussiennes 2D qui modélisent les spots de la tache
Type : pointeur
- &total = pointeur sur la variable recevant la valeur prise par la fonction critère
Type : pointeur

PARAMETRES DE SORTIE : total = valeur prise par la fonction critère
Type : float

DESCRIPTION : Calcul de la valeur prise par la fonction critère sur base de la partition des points constituant une tache et des paramètres des distributions gaussiennes associées aux classes de cette partition

APPEL PRELIMINAIRE : **partition**

PROCEDURE : fitting (tache, taille, nspot, ¢reclasse, ldistrib)

MODULE : Extraction

TYPE : Modélisation gaussienne

PARAMETRES D'ENTREE :

tache = pointeur sur le tableau de structures *point* contenant les données relatives aux pixels d'une tache, la valeur du background moyen de la tache ayant été soustraite de l'intensité de chaque point
Type : pointeur

taille = nombre de points constituant la tache
Type : unsigned short

nspot = nombre de centres de spots détectés dans la tache
Type : short int

¢reclasse = pointeur sur la liste de structures pixels qui contiennent les données relatives aux centres de spots détectés dans la tache
Type : pointeur

ldistrib = pointeur sur un tableau de structures *distrib* contenant au moins *nspot* éléments
Type : pointeur

PARAMETRES DE SORTIE : chacun des *nspot* premiers éléments du tableau pointé par *ldistrib* contient la position du centre d'un spot, sa variance en x et en y, sa covariance et son intensité intégrée.

DESCRIPTION : La position, la forme et l'intensité intégrée des spots présents dans une tache sont estimés sur base des positions de leur centre et des points qui constituent la tache en modélisant ces spots pour des distributions normales à deux dimensions.

APPEL PRELIMINAIRE :

PROCEDURE : `fix_etal_int_nom (nomgel, quad, x0, y0 )`

MODULE : Fonction

TYPE : Gestion de l'étalon d'intensité

PARAMETRES D'ENTREE : `nomgel` = pointeur sur la chaîne de caractères
contenant le nom d'un gel 2D
Type : pointeur

`quad` = numéro du quadrant où est visualisé
l'image du gel
Type : short int

`x0, y0` = coordonnées du point en haut et à
gauche où est positionnée l'image du gel. Ce
point doit être situé dans le quadrant *quad*

`nomspot` = pointeur sur la chaîne de caractères
contenant le nom d'un spot du gel
Type : ponteur
entrée au clavier

PARAMETRES DE SORTIE : Le message "erreur lecture/écriture" est généré
si un problème est apparu lors d'un accès au
fichier
Type : short int
Sortie à l'écran PC

DESCRIPTION : Cette procédure permet à l'utilisateur de
désigner un spot au sein du gel synthétique
nomgel par *nomspot*
Celui-ci est alors enregistré comme étant
l'étalon d'intensité.
L'emplacement correspondant au spot clignote à
l'écran vidéo

APPEL PRELIMINAIRE :

PROCEDURE : `fix_etal_int_numaut (nomgel, quad, x0, y0 )`

MODULE : Fonction

TYPE : Gestion de l'étalon d'intensité

PARAMETRES D'ENTREE :

`nomgel` = pointeur sur la chaîne de caractères contenant le nom d'un gel 2D
Type : pointeur

`quad` = numéro du quadrant où est visualisé l'image du gel
Type : short int

`x0, y0` = coordonnées du point en haut et à gauche où est positionnée l'image du gel. Ce point doit être situé dans le quadrant *quad*

`numspotaut` = numéro automatique d'un spot du gel
Type : long int
entrée au clavier

PARAMETRES DE SORTIE : Le message "erreur lecture/écriture" est généré si un problème est apparu lors d'un accès au fichier
Type : short int
Sortie à l'écran PC

DESCRIPTION : Cette procédure permet à l'utilisateur de désigner un spot au sein du gel synthétique *nomgel* par *numspotaut*. Celui-ci est alors enregistré comme étant l'étalon d'intensité. L'emplacement correspondant au spot clignote à l'écran vidéo

APPEL PRELIMINAIRE :

PROCEDURE : `fix_etal_int_numman (nomgel, quad, x0, y0 )`

MODULE : Fonction

TYPE : Gestion de l'étalon d'intensité

PARAMETRES D'ENTREE :

`nomgel` = pointeur sur la chaîne de caractères contenant le nom d'un gel 2D
Type : pointeur

`quad` = numéro du quadrant où est visualisé l'image du gel
Type : short int

`x0 , y0` = coordonnées du point en haut et à gauche où est positionnée l'image du gel. Ce point doit être situé dans le quadrant *quad*

`numspotman` = numéro manuel d'un spot du gel
Type : long int
entrée au clavier

PARAMETRES DE SORTIE :

Le message "erreur lecture/écriture" est généré si un problème est apparu lors d'un accès au fichier
Type : short int
Sortie à l'écran PC

DESCRIPTION :

Cette procédure permet à l'utilisateur de désigner un spot, au sein du gel synthétique *nomgel*, par *numspotman*
Celui-ci est alors enregistré comme étant l'étalon d'intensité.
L'emplacement correspondant au spot clignote à l'écran vidéo

APPEL PRELIMINAIRE :

PROCEDURE :	<code>fix_etal_int_souris (nomgel, quad, x0, y0)</code>
MODULE :	Fonction
TYPE :	Gestion de l'étalon d'intensité
PARAMETRES D'ENTREE :	<code>nomgel</code> = pointeur sur la chaîne de caractères contenant le nom d'un gel 2D Type : pointeur <code>quad</code> = numéro du quadrant où est visualisé l'image du gel Type : short int <code>x0, y0</code> = coordonnées du point en haut et à gauche où est positionnée l'image du gel. Ce point doit être situé dans le quadrant <i>quad</i> <code>x,y</code> = coordonnées d'un point du gel Type : short int entrée à la souris
PARAMETRES DE SORTIE :	Le message "erreur lecture/écriture" est généré si un problème est apparu lors d'un accès au fichier Type : short int Sortie à l'écran PC
DESCRIPTION :	Cette procédure permet à l'utilisateur de désigner un spot au sein du gel synthétique <i>nomgel</i> par la souris Le spot le plus proche de ce point est alors enregistré comme étant l'étalon d'intensité. L'emplacement correspondant au spot clignote à l'écran vidéo
APPEL PRELIMINAIRE :	<code>vision_gel2d_synth</code>

PROCEDURE :	<code>fix_etal_pi_point (nomgel, quad, x0, y0)</code>
MODULE :	Fonction
TYPE :	Gestion des étalons de point isoélectrique
PARAMETRES D'ENTREE :	<code>nomgel</code> = pointeur sur la chaîne de caractères contenant le nom d'un gel 2D Type : pointeur <code>quad</code> = numéro du quadrant où est visualisée l'image du gel Type : short int <code>x0, y0</code> = coordonnées du point en haut et à gauche où est positionnée l'image du gel. Ce point doit être situé dans le quadrant <i>quad</i> <code>x,y</code> = coordonnées du point désigné par la souris Type : short int entrée à la souris <code>pi</code> = valeur du point isoélectrique en ce point Type : float entrée au clavier <code>numéro</code> = numéro d'étalon à attribuer à ce point :1 ou 2 Type : short entrée au clavier
PARAMETRES DE SORTIE :	le message "erreur écriture" est généré si un problème apparaît lors d'un accès au fichier Type : string sortie à l'écran PC
DESCRIPTION :	Désignation d'un point au sein du gel synthétique <i>nomgel</i> à l'aide de la souris. Ce point est alors enregistré comme étant un des deux points étalons pour le <code>pi</code>
APPEL PRELIMINAIRE :	<code>vision_gel2d_synth</code>

PROCEDURE : `fix_etat_pi_spot_nom (nomgel, quad, x0,y0 )`

MODULE : Fonction

TYPE : Gestion des étalons de point isoélectrique

PARAMETRES D'ENTREE :

- `nomgel` = pointeur sur la chaîne de caractères contenant le nom d'un gel 2D
Type : pointeur
- `quad` = numéro du quadrant où est visualisée l'image du gel
Type : short int
- `x0, y0` = coordonnées du point en haut et à gauche où est positionnée l'image du gel. Ce point doit être situé dans le quadrant *quad*
- `nomspot` = pointeur sur la chaîne de caractères contenant le nom d'un spot du gel
Type : pointeur
entrée au clavier
- `pi` = valeur du point isoélectrique correspondant au spot
Type : float
entrée au clavier
- `numetalon` = numéro d'étalon à attribuer au spot : 1 ou 2
Type : short int
entrée au clavier

PARAMETRES DE SORTIE :

- le message "erreur lecture/écriture" est généré si un problème est apparu lors d'un accès au fichier
Type : string
sortie à l'écran PC

DESCRIPTION :

Le spot désigné au sein du gel synthétique *nomgel*/par *nomspot* est enregistré comme étant l'un des deux points étalons pour le pi
L'emplacement correspondant au spot clignote à l'écran vidéo

APPEL PRELIMINAIRE :

PROCEDURE : `fix_etal_pi_spot_numaut (nomgel, quad, x0,y0 )`

MODULE : Fonction

TYPE : Gestion des étalons de point isoélectrique

PARAMETRES D'ENTREE :

- `nomgel` = pointeur sur la chaîne de caractères contenant le nom d'un gel 2D
Type : pointeur
- `quad` = numéro du quadrant où est visualisée l'image du gel
Type : short int
- `x0, y0` = coordonnées du point en haut et à gauche où est positionnée l'image du gel. Ce point doit être situé dans le quadrant *quad*
- `numspotaut` = numéro automatique d'un spot du gel
Type : long int
entrée au clavier
- `pi` = valeur du point isoélectrique correspondant au spot
Type : float
entrée au clavier
- `numetalon` = numéro d'étalon à attribuer au spot : 1 ou 2
Type : short int
entrée au clavier

PARAMETRES DE SORTIE : le message "erreur lecture/écriture" est généré si un problème est apparu lors d'un accès au fichier
Type : string
sortie à l'écran PC

DESCRIPTION : Le spot désigné au sein du gel synthétique *nomgel* par *numspotaut* est enregistré comme étant l'un des deux points étalons pour le pi
L'emplacement correspondant au spot clignote à l'écran vidéo

APPEL PRELIMINAIRE :

PROCEDURE :	<code>fix_etal_pi_spot_numman (nomgel, quad, x0 ,y0)</code>
MODULE :	Fonction
TYPE :	Gestion des étalons de point isoélectrique
PARAMETRES D'ENTREE :	<code>nomgel</code> = pointeur sur la chaîne de caractères contenant le nom d'un gel 2D Type : pointeur <code>quad</code> = numéro du quadrant où est visualisée l'image du gel Type : short int <code>x0, y0</code> = coordonnées du point en haut et à gauche où est positionnée l'image du gel. Ce point doit être situé dans le quadrant <i>quad</i> <code>numspotman</code> = numéro manuel d'un spot du gel Type : long int entrée au clavier <code>pi</code> = valeur du point isoélectrique correspondant au spot Type : float entrée au clavier <code>numetalon</code> = numéro d'étalon à attribuer au spot : 1 ou 2 Type : short int entrée au clavier
PARAMETRES DE SORTIE :	le message "erreur lecture/écriture" est généré si un problème est apparu lors d'un accès au fichier Type : string sortie à l'écran PC
DESCRIPTION :	Le spot désigné au sein du gel synthétique <i>nomgel</i> par <i>numspotman</i> est enregistré comme étant l'un des deux points étalons pour le <i>pi</i> L'emplacement correspondant au spot clignote à l'écran vidéo
APPEL PRELIMINAIRE :	

PROCEDURE : `fix_etal_pi_spot_souris (nomgel, quad, x0,y0 )`

MODULE : Fonction

TYPE : Gestion des étalons de point isoélectrique

PARAMETRES D'ENTREE :

`nomgel` = pointeur sur la chaîne de caractères contenant le nom d'un gel 2D
Type : pointeur

`quad` = numéro du quadrant où est visualisée l'image du gel
Type : short int

`x0, y0` = coordonnées du point en haut et à gauche où est positionnée l'image du gel. Ce point doit être situé dans le quadrant *quad*

`x,y` = coordonnées d'un point du gel
Type : short int
entrée à la souris

`pi` = valeur du point isoélectrique correspondant au spot
Type : float
entrée au clavier

`numetalon` = numéro d'étalon à attribuer au spot : 1 ou 2
Type : short int
entrée au clavier

PARAMETRES DE SORTIE : le message "erreur lecture/écriture" est généré si un problème est apparu lors d'un accès au fichier
Type : string
sortie à l'écran PC

DESCRIPTION : Le spot le plus proche du point désigné au sein du gel synthétique *nomgel* est enregistré comme étant l'un des deux points étalons pour le *pi*
L'emplacement correspondant au spot clignote à l'écran vidéo

APPEL PRELIMINAIRE : **vision_gel2d_synth**

PROCEDURE :	<code>fix_etal_pm_point (nomgel, quad, x0,y0)</code>
MODULE :	Fonction
TYPE :	Gestion des étalons de poids moléculaire
PARAMETRES D'ENTREE :	<code>nomgel</code> = pointeur sur la chaîne de caractères contenant le nom d'un gel 2D Type : pointeur <code>quad</code> = numéro du quadrant où est visualisée l'image du gel Type : short int <code>x0, y0</code> = coordonnées du point en haut et à gauche où est positionnée l'image du gel. Ce point doit être situé dans le quadrant <i>quad</i> <code>x,y</code> = coordonnées du point désigné par la souris Type : short int entrée à la souris <code>pm</code> = valeur du poids moléculaire en ce point Type : float entrée au clavier <code>numetalon</code> = numéro d'étalon à attribuer à ce point : 1 ou 2 Type : short entrée au clavier
PARAMETRES DE SORTIE :	le message "erreur écriture" est généré si un problème apparaît lors d'un accès au fichier Type : string sortie à l'écran PC
DESCRIPTION :	Désignation d'un point au sein du gel synthétique <i>nomgel</i> à l'aide de la souris. Ce point est alors enregistré comme étant un des deux points étalons pour le pM
APPEL PRELIMINAIRE :	<code>vision_gel2d_synth</code>

PROCEDURE : `fix_etal_pm_spot_nom` (`nomgel`, `quad`, `x0`, `y0`)

MODULE : Fonction

TYPE : Gestion des étalons de poids moléculaire

PARAMETRES D'ENTREE :

`nomgel` = pointeur sur la chaîne de caractères contenant le nom d'un gel 2D
Type : pointeur

`quad` = numéro du quadrant où est visualisée l'image du gel
Type : short int

`x0`, `y0` = coordonnées du point en haut et à gauche où est positionnée l'image du gel. Ce point doit être situé dans le quadrant *quad*

`nomspot` = pointeur sur la chaîne de caractères contenant le nom d'un spot du gel
Type : pointeur
entrée au clavier

`pm` = valeur du poids moléculaire correspondant au spot
Type : float
entrée au clavier

`numetalon` = numéro d'étalon à attribuer au spot : 1 ou 2
Type : short int
entrée au clavier

PARAMETRES DE SORTIE : le message "erreur lecture/écriture" est généré si un problème est apparu lors d'un accès au fichier
Type : string
sortie à l'écran PC

DESCRIPTION : Le spot désigné au sein du gel synthétique *nomgel* par *nomspot* est enregistré comme étant l'un des deux points étalons pour le pM. L'emplacement correspondant au spot clignote à l'écran vidéo

APPEL PRELIMINAIRE :

PROCEDURE :	<code>fix_etal_pm_spot_numman</code> (<code>nomgel</code> , <code>quad</code> , <code>x0</code> , <code>y0</code>)
MODULE :	Fonction
TYPE :	Gestion des étalons de poids moléculaire
PARAMETRES D'ENTREE :	<code>nomgel</code> = pointeur sur la chaîne de caractères contenant le nom d'un gel 2D Type : pointeur <code>quad</code> = numéro du quadrant où est visualisée l'image du gel Type : short int <code>x0</code>, <code>y0</code> = coordonnées du point en haut et à gauche où est positionnée l'image du gel. Ce point doit être situé dans le quadrant <i>quad</i> <code>numspotman</code> = numéro manuel d'un spot du gel Type : long int entrée au clavier <code>pm</code> = valeur du poids moléculaire correspondant au spot Type : float entrée au clavier <code>numetalon</code> = numéro d'étalon à attribuer au spot : 1 ou 2 Type : short int entrée au clavier
PARAMETRES DE SORTIE :	le message "erreur lecture/écriture" est généré si un problème est apparu lors d'un accès au fichier Type : string sortie à l'écran PC
DESCRIPTION :	Le spot désigné au sein du gel synthétique <i>nomgel</i> par <i>numspotman</i> est enregistré comme étant l'un des deux points étalons pour le pM L'emplacement correspondant au spot clignote à l'écran vidéo
APPEL PRELIMINAIRE :	

PROCEDURE : `fix_etal_pm_spot_souris (nomgel, quad, x0,y0 )`

MODULE : Fonction

TYPE : Gestion des étalons de poids moléculaire

PARAMETRES D'ENTREE :

`nomgel` = pointeur sur la chaîne de caractères contenant le nom d'un gel 2D
Type : pointeur

`quad` = numéro du quadrant où est visualisée l'image du gel
Type : short int

`x0, y0` = coordonnées du point en haut et à gauche où est positionnée l'image du gel. Ce point doit être situé dans le quadrant *quad*

`x,y` = coordonnées d'un point du gel
Type : short int
entrée à la souris

`pm` = valeur du poids moléculaire correspondant au spot
Type : float
entrée au clavier

`numetalon` = numéro d'étalon à attribuer au spot : 1 ou 2
Type : short int
entrée au clavier

PARAMETRES DE SORTIE : le message "erreur lecture/écriture" est généré si un problème est apparu lors d'un accès au fichier
Type : string
sortie à l'écran PC

DESCRIPTION : Le spot le plus proche du point désigné au sein du gel synthétique *nomgel* est enregistré comme étant l'un des deux points étalons pour le pM
L'emplacement correspondant au spot clignote à l'écran vidéo

APPEL PRELIMINAIRE :

PROCEDURE :	info_etalon_int (nomgel,quad, x0 ,y0)
MODULE :	Fonction
TYPE :	Gestion de l'étalon d'intensité
PARAMETRES D'ENTREE :	nomgel = pointeur sur la chaîne de caractères contenant le nom d'un gel 2D Type : pointeur quad = numéro du quadrant où est visualisée l'image du gel Type : short int x0, y0 = coordonnées du point en haut et à gauche où est positionnée l'image du gel. Ce point doit être situé dans le quadrant <i>quad</i>
PARAMETRES DE SORTIE :	les informations relatives au spot étalon d'intensité Type : struct protéine sortie à l'écran PC le message "erreur lecture" est généré si un problème apparaît lors de l'accès au fichier Type : string sortie à l'écran
DESCRIPTION :	Affichage des renseignements relatifs au spot qui constitue l'étalon pour le calcul des intensités relatives dans le gel synthétique <i>nomgel</i> L'emplacement correspondant au spot étalon clignote à l'écran vidéo
APPEL PRELIMINAIRE :	

PROCEDURE : info_etalons_pipm (nomgel, quad, x0,y0)

MODULE : Fonction

TYPE : Gestion des étalons de pl et de pM

PARAMETRES D'ENTREE : nomgel = pointeur sur la chaîne de caractères
contenant le nom d'un gel 2D
Type : pointeur

quad = numéro du quadrant où est visualisée
l'image du gel
Type : short int

x0 , y0 = coordonnées du point en haut et à
gauche où est positionnée l'image du gel. Ce
point doit être situé dans le quadrant *quad*

PARAMETRES DE SORTIE : les informations relatives aux quatre points
étalons
Type : struct pointref
sortie à l'écran PC

le message "erreur lecture" est généré si un
problème apparaît lors de l'accès au fichier
Type : string
sortie à l'écran PC

DESCRIPTION : Cette procédure affiche à l'écran PC les
positions et valeurs de pl des deux points
étalons du gel synthétique nomgel, et de même
pour les deux points étalons de poids
moléculaire
Les emplacements correspondant à ces points
clignotent à l'écran vidéo

APPEL PRELIMINAIRE :

PROCEDURE : info_intensite_gel (nomgel)

MODULE : Fonction

TYPE : Gestion de l'intensité globale d'un gel

PARAMETRES D'ENTREE : nomgel = pointeur sur la chaîne de caractères
contenant le nom d'un gel 2D
Type : pointeur

PARAMETRES DE SORTIE : intglobale = intensité globale du gel
Type : float
sortie à l'écran PC

le message "erreur lecture" est généré si un
problème apparaît lors de l'accès au fichier
Type : string
sortie à l'écran PC

DESCRIPTION : Cette procédure affiche à l'écran PC l'intensité
globale du gel synthétique *nomgel*

APPEL PRELIMINAIRE :

PROCEDURE : info_point (nomgel, quad, x0, y0)

MODULE : Fonction

TYPE : Gestion de gel 2D synthétiques

PARAMETRES D'ENTREE : nomgel = pointeur sur la chaîne de caractères
contenant le nom d'un gel 2D
Type : pointeur

quad = numéro du quadrant où est visualisée
l'image du gel
Type : short int

x0, y0 = coordonnées du point en haut et à
gauche où est positionnée l'image du gel. Ce
point doit être situé dans le quadrant *quad*

x,y = coordonnées du point désigné par la souris
sur l'écran vidéo
Type : short int
entrée à la souris

PARAMETRES DE SORTIE : pi,pm = valeurs de pl et de pM estimées au point
de coordonnées x,y
Type : float
sortie à l'écran PC

x,y = coordonnées du point désigné par la souris
Type : short int
sortie à l'écran PC

le message "erreur lecture" est généré si un
problème apparaît lors de l'accès au fichier
Type : string
sortie à l'écran PC

DESCRIPTION : Cette procédure affiche à l'écran PC les valeurs
de pl et de pM et position d'un point désigné à
l'aide de la souris au sein du gel synthétique
nomgel visualisé sur l'écran vidéo

APPEL PRELIMINAIRE : **vision_gel2d_synth**

PROCEDURE : info_spot_nom (nomgel, quad, x0, y0)

MODULE : Fonction

TYPE : Gestion de spots

PARAMETRES D'ENTREE :

nomgel = pointeur sur la chaîne de caractères contenant le nom d'un gel 2D
Type : pointeur

quad = numéro du quadrant où est visualisée l'image du gel
Type : short int

x0, y0 = coordonnées du point en haut et à gauche où est positionnée l'image du gel. Ce point doit être situé dans le quadrant *quad*

nomspot = pointeur sur la chaîne de caractères contenant le nom d'un spot du gel
Type : pointeur
entrée au clavier

PARAMETRES DE SORTIE :

les informations relatives au spot désigné
Type : struct protéine
sortie à l'écran PC

le message "erreur lecture" est généré si un problème apparaît lors de l'accès au fichier
Type : string
sortie à l'écran PC

DESCRIPTION :

Affichage à l'écran PC des renseignements relatifs à un spot désigné par son nom au sein du gel synthétique *nomgel*
L'emplacement correspondant au spot clignote à l'écran vidéo

APPEL PRELIMINAIRE :

PROCEDURE : info_spot_numaut (nomgel, quad, x0,y0)

MODULE : Fonction

TYPE : Gestion de spots

PARAMETRES D'ENTREE :

nomgel = pointeur sur la chaîne de caractères contenant le nom d'un gel 2D
Type : pointeur

quad = numéro du quadrant où est visualisée l'image du gel
Type : short int

x0, y0 = coordonnées du point en haut et à gauche où est positionnée l'image du gel. Ce point doit être situé dans le quadrant *quad*

numspotaut = numéro automatique d'un spot du gel
Type : long int
entrée au clavier

PARAMETRES DE SORTIE :

les informations relatives au spot désigné
Type : struct protéine
sortie à l'écran PC

le message "erreur lecture" est généré si un problème apparaît lors de l'accès au fichier
Type : string
sortie à l'écran PC

DESCRIPTION :

Affichage à l'écran PC des renseignements relatifs à un spot désigné par son numéro automatique au sein du gel synthétique *nomgel*
L'emplacement correspondant au spot clignote à l'écran vidéo

APPEL PRELIMINAIRE :

PROCEDURE : info_spot_numman (nomgel, quad, x0,y0)

MODULE : Fonction

TYPE : Gestion de spots

PARAMETRES D'ENTREE : nomgel = pointeur sur la chaîne de caractères contenant le nom d'un gel 2D
Type : pointeur

quad = numéro du quadrant où est visualisé l'image du gel
Type : short int

x0 , y0 = coordonnées du point en haut et à gauche où est positionnée l'image du gel. Ce point doit être situé dans le quadrant *quad*

numspotman = numéro manuel d'un spot du gel
Type : long int
entrée au clavier

PARAMETRES DE SORTIE : les informations relatives au spot désigné
Type : struct protéine
sortie à l'écran PC

le message "erreur lecture" est généré si un problème apparaît lors de l'accès au fichier
Type : string
sortie à l'écran PC

DESCRIPTION : Affichage à l'écran PC des renseignements relatifs à un spot désigné par son numéro manuel au sein du gel synthétique *nomgel*
L'emplacement correspondant au spot clignote à l'écran vidéo

APPEL PRELIMINAIRE :

PROCEDURE :	info_spot_souris (nomgel, quad, x0,y0)
MODULE :	Fonction
TYPE :	Gestion de spots
PARAMETRES D'ENTREE :	nomgel = pointeur sur la chaîne de caractères contenant le nom d'un gel 2D Type : pointeur quad = numéro du quadrant où est visualisée l'image du gel Type : short int x0, y0 = coordonnées du point en haut et à gauche où est positionnée l'image du gel. Ce point doit être situé dans le quadrant <i>quad</i> x,y = coordonnées d'un point du gel Type : short int entrée à la souris
PARAMETRES DE SORTIE :	les informations relatives au spot désigné Type : struct proteine sortie à l'écran PC le message "erreur lecture" est généré si un problème apparaît lors de l'accès au fichier Type : string sortie à l'écran PC
DESCRIPTION :	Affichage à l'écran PC des renseignements relatifs au spot le plus proche du point désigné par la souris au sein du gel synthétique <i>nomgel</i> L'utilisateur sélectionne le point en appuyant sur le bouton gauche de la souris. La procédure se répète jusqu'à ce qu'il appuie sur le bouton droit de la souris. L'emplacement correspondant au spot clignote à l'écran vidéo.
APPEL PRELIMINAIRE :	vision_gel2d_synth

PROCEDURE : init_fichref (nomgel, dx, dy, &fstat)

MODULE : Accès-BD

TYPE : Gestion de gels 2D synthétiques

PARAMETRES D'ENTREE : nomgel = pointeur sur la chaîne de caractères
 contenant le nom du gel 2D
 Type : pointeur

 dx = nombre de colonnes de l'image du gel
 Type : short int

 dy = nombre de lignes de l'image du gel
 Type : short int

 &fstat = pointeur sur l'entier qui contiendra
 l'état d'accès au fichier
 Type : pointeur

PARAMETRES DE SORTIE : fstat = 0 si un problème est apparu lors des
 accès au fichier
 = 1 si aucun problème n'est apparu
 Type : short int

DESCRIPTION : Création et initialisation du fichier *nomgel.ref*.
 Les valeurs affectées aux différents champs
 sont les suivantes :
 { dx, dy } { 0,0,0 } { 0,0,0 } { 0,0,0 } { 0,0,0 } { 1 }
 { 0,0,0,0,1,0,0,'y','-' } { "aucun commentaire" }

APPEL PRELIMINAIRE :

PROCEDURE : integration (tache, taille, nspot, ldistrib)

MODULE : Extraction

TYPE : Modélisation gaussienne

PARAMETRES D'ENTREE :

 tache = pointeur sur le tableau de structures
 point contenant les données relatives aux
 pixels d'une tache, la valeur du background
 moyen de la tache ayant été soustraite de
 l'intensité de chaque point
 Type : pointeur

 taille = nombre de points (pixels) constituant la
 tache
 Type : unsigned short

 nspot = nombre de centres de spots détectés
 dans la tache
 Type : short int

 ldistrib = pointeur sur le tableau de structures
 distrib contenant les paramètres des
 distributions gaussiennes 2D qui modélisent les
 spots de la tache
 Type : pointeur

PARAMETRES DE SORTIE :

 l'intensité contenue dans chacune des *nspot*
 premières structures du tableau pointé par
 ldistrib est l'intensité intégrée du spot
 modélisé par la distribution correspondante

DESCRIPTION :

 L'intensité de chaque point de la tache est
 répartie entre les différentes distributions
 proportionnellement à la densité de probabilité
 calculée en ce point pour chaque distribution

APPEL PRELIMINAIRE :

PROCEDURE : lire_com2d_synth (nomgel, commentaire,
&fstat)

MODULE : accès-BD

TYPE : Gestion de gels 2D synthétiques

PARAMETRES D'ENTREE : nomgel = pointeur sur la chaîne de caractères
contenant le nom du gel 2D
Type : pointeur

commentaire = pointeur sur la chaîne de
caractères recevant le commentaire à lire dans
le fichier *nomgel.ref*
Type : pointeur

&fstat = pointeur sur l'entier qui contiendra
l'état de l'accès au fichier
Type : pointeur

PARAMETRES DE SORTIE : &fstat = 0 si un problème est apparu lors des
accès au fichier
= 1 si aucun problème n'est apparu
Type : short int

DESCRIPTION : Lecture d'un commentaire relatif à un gel 2D
synthétique

APPEL PRELIMINAIRE :

PROCEDURE : lire_etalon_pi (nomgel, xo, yo, &pi, &x, &y, numetalon, &fstat)

MODULE : Accès-BD

TYPE : Gestion des étalons de point isoélectrique.

PARAMETRES D'ENTREE : nomgel = pointeur sur la chaîne de caractères contenant le nom d'un gel 2D
Type : pointeur

xo, yo = coordonnées du point en haut et à gauche où est positionnée l'image du gel
Type : short int

&pi = pointeur sur la variable recevant la valeur de pi de l'étalon
Type : pointeur

&x, &y = pointeurs sur les variables recevant les coordonnées de l'étalon
Type : pointeur

numetalon = numéro de l'étalon pi à lire : 1 ou 2
Type : short int

&fstat = pointeur sur l'entier qui contiendra l'état de l'accès au fichier
Type : pointeur

PARAMETRES DE SORTIE : pi = valeur de pi de l'étalon
Type : float

x, y = coordonnées de l'étalon, par rapport au coin supérieur gauche de l'image du gel
Type : float

fstat = 0 si un problème est apparu lors de l'accès au fichier
= 1 si aucun problème n'est apparu
Type : short int

DESCRIPTION : Lecture des informations relatives à un des deux points étalons de pi au sein d'un gel 2D. L'emplacement correspondant à l'étalon lu clignote à l'écran vidéo.

APPEL PRELIMINAIRE :

PROCEDURE : lire_etalon_pm (nomgel, xo, yo, &pm, &x, &y, numetalon, &fstat)

MODULE : Accès-BD

TYPE : Gestion des étalons de poids moléculaire

PARAMETRES D'ENTREE : nomgel = pointeur sur la chaîne de caractères contenant le nom d'un gel 2D
Type : pointeur

xo, yo = coordonnées du point en haut et à gauche où est positionnée l'image du gel
Type : short int

&pm = pointeur sur la variable recevant la valeur de pM de l'étalon
Type : pointeur

&x, &y = pointeurs sur les variables recevant les coordonnées de l'étalon
Type : pointeur

numetalon = numéro de l'étalon pM à lire : 1 ou 2
Type : short int

&fstat = pointeur sur l'entier qui contiendra l'état de l'accès au fichier
Type : pointeur

PARAMETRES DE SORTIE : pm = valeur de pM de l'étalon
Type : float

x, y = coordonnées de l'étalon, par rapport au coin supérieur gauche de l'image du gel
Type : float

fstat = 0 si un problème est apparu lors de l'accès au fichier
= 1 si aucun problème n'est apparu
Type : short int

DESCRIPTION : Lecture des informations relatives à un des deux points étalons de pM au sein d'un gel 2D. L'emplacement correspondant à l'étalon lu clignote à l'écran vidéo.

APPEL PRELIMINAIRE :

PROCEDURE : lire_spot (&streamspt, &streamref, &prot, &fstat)

MODULE : Accès-BD

TYPE : Gestion de spots

PARAMETRES D'ENTREE :

- &streamspt = pointeur sur un fichier d'extension *.spt* positionné au début d'un enregistrement
Type : pointeur
- &streamref = pointeur sur le fichier d'extension *.ref* portant le même nom que le fichier *.spt*
Type : pointeur
- &prot = pointeur la variable recevant les informations lues dans les fichiers
Type : pointeur
- &fstat = pointeur sur l'entier qui contiendra l'état des accès aux fichiers
Type : pointeur

PARAMETRES DE SORTIE :

- streamspt = fichier *.spt* positionné en fin de l'enregistrement lu
Type : FILE
- streamref = fichier *.ref* positionné à la fin de l'enregistrement correspondant au 4ème point étalon
Type : FILE
- prot = variable contenant les informations relatives au spot lu dans le fichier *.spt*
Type : structure *proteine*
- fstat = 0 si un problème est apparu lors des accès aux fichiers
 = 1 si aucun problème n'est apparu
Type : short int

DESCRIPTION : Lecture des paramètres d'un spot contenus dans le fichier *.spt* et calcul des pI et pM de ce spot sur base des points étalons contenus dans le fichier *.ref*. L'ensemble de ces informations est repris dans une structure *protéine*

APPEL PRELIMINAIRE : Une des procédures suivantes :
chercher_spot_cooronnee
chercher_spot_nom
chercher_spot_numout
chercher_spot_nummon

PROCEDURE : masq_artef_nom (nomgel, quad, x0,y0)

MODULE : Fonction

TYPE : Gestion d'artéfacts

PARAMETRES D'ENTREE : nomgel = pointeur sur la chaîne de caractères
 contenant le nom d'un gel 2D
 Type : pointeur

 quad = numéro du quadrant où est visualisée
 l'image du gel
 Type : short int

 x0, y0 = coordonnées du point en haut et à
 gauche où est positionnée l'image du gel. Ce
 point doit être situé dans le quadrant *quad*

 nomspot = pointeur sur la chaîne de caractères
 contenant le nom d'un spot du gel
 Type : pointeur
 entrée au clavier

PARAMETRES DE SORTIE : le(s) message(s) "erreur lecture/écriture" et/ou
 "erreur lors calcul intensité globale du gel"
 est(sont) généré(s) si un problème apparaît lors
 d'un accès au fichier
 Type : string
 sortie à l'écran PC

DESCRIPTION : Cette procédure masque, au sein du gel
 synthétique *nomgel* le spot désigné par *nomspot*
 L'emplacement correspondant au spot /artéfact
 clignote à l'écran vidéo.
 L'intensité globale du gel est mise à jour

APPEL PRELIMINAIRE :

PROCEDURE : masq_artef_numaut (nomgel, quad, x0,y0)

MODULE : Fonction

TYPE : Gestion d'artéfacts

PARAMETRES D'ENTREE : nomgel = pointeur sur la chaîne de caractères
 contenant le nom d'un gel 2D
 Type : pointeur

 quad = numéro du quadrant où est visualisée
 l'image du gel
 Type : short int

 x0, y0 = coordonnées du point en haut et à
 gauche où est positionnée l'image du gel. Ce
 point doit être situé dans le quadrant *quad*

 numspotaut = numéro automatique d'un spot du
 gel
 Type : long int
 entrée au clavier

PARAMETRES DE SORTIE : 1e(s) message(s) "erreur lecture/écriture" et/ou
 "erreur lors calcul intensité globale du gel"
 est(sont) généré(s) si un problème apparaît lors
 d'un accès au fichier
 Type : string
 sortie à l'écran PC

DESCRIPTION : Cette procédure masque, au sein du gel
 synthétique *nomgel* le spot désigné par
 numspotaut
 L'emplacement correspondant au spot /artéfact
 clignote à l'écran vidéo.
 L'intensité globale du gel est mise à jour

APPEL PRELIMINAIRE :

PROCEDURE : `masq_artef_numman (nomgel, quad, x0,y0 )`

MODULE : Fonction

TYPE : Gestion d'artéfacts

PARAMETRES D'ENTREE : `nomgel` = pointeur sur la chaîne de caractères
contenant le nom d'un gel 2D
Type : pointeur

`quad` = numéro du quadrant où est visualisée
l'image du gel
Type : short int

`x0, y0` = coordonnées du point en haut et à
gauche où est positionnée l'image du gel. Ce
point doit être situé dans le quadrant *quad*

`numspotman` = numéro manuel d'un spot du gel
Type : long int
entrée au clavier

PARAMETRES DE SORTIE : le(s) message(s) "erreur lecture/écriture" et/ou
"erreur lors calcul intensité globale du gel"
est(sont) générée(s) si un problème apparaît lors
d'un accès au fichier
Type : string
sortie à l'écran PC

DESCRIPTION : Cette procédure masque, au sein du gel
synthétique *nomgel* le spot désigné par
numspotman
L'emplacement correspondant au spot /artéfact
clignote à l'écran vidéo.
L'intensité globale du gel est mise à jour

APPEL PRELIMINAIRE :

PROCEDURE : **masq_artef_souris (nomgel, quad, x0,y0)**

MODULE : **Fonction**

TYPE : **Gestion d'artéfacts**

PARAMETRES D'ENTREE : **nomgel = pointeur sur la chaîne de caractères
contenant le nom d'un gel 2D
Type : pointeur**

**quad = numéro du quadrant où est visualisée
l'image du gel
Type : short int**

**x0, y0 = coordonnées du point en haut et à
gauche où est positionnée l'image du gel. Ce
point doit être situé dans le quadrant *quad***

**x,y = coordonnées d'un point du gel
Type : short int
entrée à la souris**

PARAMETRES DE SORTIE : **le(s) message(s) "erreur lecture/écriture" et/ou
"erreur lors calcul intensité globale du gel"
est(sont) généré(s) si un problème apparaît lors
d'un accès au fichier
Type : string
sortie à l'écran PC**

DESCRIPTION : **Cette procédure masque, au sein du gel
synthétique *nomgel* le spot le plus proche du
point désigné à l'aide de la souris. L'utilisateur
sélectionne le point en appuyant sur le bouton
gauche de la souris. La procédure se répète
jusqu'à ce qu'il appuie sur le bouton droit de la
souris.
L'emplacement correspondant au spot /artéfact
clignote à l'écran vidéo.
L'intensité globale du gel est mise à jour
lorsque la procédure se termine.**

APPEL PRELIMINAIRE : **vision_gel2d_synth**

PROCEDURE : masquer_spot_nom (nomgel, xo, yo, nomspot,
 &fstat)

MODULE : Accès-BD

TYPE : Gestion de spots

PARAMETRES D'ENTREE : nomgel = pointeur sur la chaîne de caractères
 contenant le nom d'un gel 2D
 Type : pointeur

 xo, yo = coordonnées du point en haut et à gauche
 où est positionnée l'image du gel
 Type : short int

 nomspot = pointeur sur la chaîne de caractères
 contenant le nom du spot
 Type : pointeur

 &fstat = pointeur sur l'entier qui contiendra
 l'état des accès au fichier
 Type : pointeur

PARAMETRES DE SORTIE : fstat = 0 si un problème est apparu lors de
 l'accès au fichier
 = 1 si aucun problème n'est apparu
 Type : short int

DESCRIPTION : Masquage d'un spot désigné par son nom au sein
 d'un gel 2D.
 L'emplacement correspondant au spot clignote à
 l'écran vidéo.

APPEL PRELIMINAIRE :

PROCEDURE : masquer_spot_numaut (nomgel, xo, yo,
 numspotaut, &fstat)

MODULE : Accès-BD

TYPE : Gestion de spots

PARAMETRES D'ENTREE : nomgel = pointeur sur la chaîne de caractères
 contenant le nom d'un gel 2D
 Type : pointeur

 xo, yo = coordonnées du point en haut et à gauche
 où est positionnée l'image du gel
 Type : short int

 numspotaut = numéro automatique du spot
 Type : long int

 &fstat = pointeur sur l'entier qui contiendra
 l'état des accès au fichier
 Type : pointeur

PARAMETRES DE SORTIE : fstat = 0 si un problème est apparu lors de
 l'accès au fichier
 = 1 si aucun problème n'est apparu
 Type : short int

DESCRIPTION : Masquage d'un spot désigné par son numéro
 automatique au sein d'un gel 2D.
 L'emplacement correspondant au spot clignote à
 l'écran vidéo.

APPEL PRELIMINAIRE :

PROCEDURE : masquer_spot_numman (nomgel, xo, yo, numspotman, &fstat)

MODULE : Accès-BD

TYPE : Gestion de spots

PARAMETRES D'ENTREE : nomgel = pointeur sur la chaîne de caractères contenant le nom d'un gel 2D
Type : pointeur

xo, yo = coordonnées du point en haut et à gauche où est positionnée l'image du gel
Type : short int

numspotman = numéro manuel du spot
Type : long int

&fstat = pointeur sur l'entier qui contiendra l'état des accès au fichier
Type : pointeur

PARAMETRES DE SORTIE : fstat = 0 si un problème est apparu lors de l'accès au fichier
= 1 si aucun problème n'est apparu
Type : short int

DESCRIPTION : Masquage d'un spot désigné par son numéro manuel au sein d'un gel 2D.
L'emplacement correspondant au spot clignote à l'écran vidéo.

APPEL PRELIMINAIRE :

PROCEDURE : nivtache (x0, y0, xbkgrd, ybkgrd, tache, taille)

MODULE : Extraction

TYPE : Détermination du background

PARAMETRES D'ENTREE :

x0, y0 = coordonnées du point en haut et à gauche où est positionnée l'image inversée du gel 2D source
Type : short int

xbkgrd, ybkgrd = coordonnées du coin supérieur gauche de la zone où sont mémorisées les valeurs de background de chaque point de l'image inversée du gel
Type : short int

tache = pointeur sur le tableau de structures *point* contenant les données relatives aux points (pixels) d'une tache déterminée dans l'image
Type : pointeur

&taille = pointeur sur la variable contenant le nombre de points qui constituent la tache
Type : pointeur

PARAMETRES DE SORTIE :

taille = nombre de points ayant une intensité supérieure au background moyen de la tache
Type : unsigned short

les *taille* premiers éléments du tableau pointé par *tache* contiennent les données relatives aux points dont l'intensité est supérieure au background moyen de la tache. L'intensité de chacun de ces points a été diminuée de la valeur de ce background moyen.

DESCRIPTION :

APPEL PRELIMINAIRE :

PROCEDURE : normalisation2d (tache, taille, ldistrib, &nspot)

MODULE : Extraction

TYPE : Modélisation gaussienne

PARAMETRES D'ENTREE : *tache* = pointeur sur le tableau de structures
 point contenant les données relatives aux
 points qui constituent une tache
 Type : pointeur

taille = nombre de points constituant la tache
 Type : unsigned short

ldistrib = pointeur sur un tableau de structures
 distrib contenant *maxdistrib* éléments
 Type : pointeur

 &nspot = pointeur sur la variable recevant le
 nombre de spots détectés et quantifiés dans la
 tache
 Type : pointeur

PARAMETRES DE SORTIE : les *nspot* premiers éléments du tableau pointé
 par *ldistrib* contiennent les données relatives
 aux distributions normales 2D qui modélisent
 les spots détectés dans la tache, ainsi que
 l'intensité intégrée de ceux-ci

nspot = nombre de spots détectés et quantifiés
 dans la tache
 Type : short int

DESCRIPTION : Détection et quantification des spots présents
 dans une tache

APPEL PRELIMINAIRE : **nivtache**

PROCEDURE : numaut_2d (nomgel, &fstat)

MODULE : Accès-BD

TYPE : Gestion de spots

PARAMETRES D'ENTREE : nomgel = pointeur sur la chaîne de caractères
contenant le nom d'un gel 2D

Type : pointeur

&fstat = pointeur sur l'entier qui contiendra
l'état des accès aux fichiers

Type = short int

PARAMETRES DE SORTIE : fstat = 0 si un problème est apparu lors des accès
aux fichiers

= 1 si aucun problème n'est apparu

Type : short int

DESCRIPTION : Numérotation automatique de tous les spots d'un
gel 2D synthétique.

Le numéro d'un spot est construit sur base de ses
pI et pM de la manière suivante :

xxyyzzz où zzz = pM exprimé en Kdaltons

xxyy = pI multiplié par cent

APPEL PRELIMINAIRE :

PROCEDURE : numeroter_2d (nomgel)

MODULE : Fonction

TYPE : Gestion des spots

PARAMETRES D'ENTREE : nomgel = pointeur sur la chaîne de caractères
 contenant le nom d'un gel 2D
 Type : pointeur

PARAMETRES DE SORTIE : le message "erreur lecture/écriture" est généré
 si un problème est apparu lors d'un accès aux
 fichiers
 Type : string
 Sortie à l'écran PC

DESCRIPTION : Numérotation automatique des spots du gel 2D
 synthétique contenu dans le fichier *nomgel.spt*

APPEL PRELIMINAIRE :

PROCEDURE : parametre (tache, taille, nspot, ldistrib)

MODULE : Extraction

TYPE : Modélisation gaussienne

PARAMETRES D'ENTREE :

tache = pointeur sur le tableau de structures *point* contenant les données relatives aux pixels d'une tache, la valeur du background moyen de la tache ayant été soustraite de l'intensité de chaque point
Type : pointeur

taille = nombre de points (pixels) constituant la tache
Type : unsigned short

nspot = nombre de centres de spots détectés dans la tache
Type : short int

ldistrib = pointeur sur le tableau de structures *distrib* contenant les paramètres des distributions gaussiennes 2D qui modélisent les spots de la tache
Type : pointeur

PARAMETRES DE SORTIE :

chacun des *nspot* premiers éléments du tableau pointé par *ldistrib* contient les paramètres d'une distribution gaussienne estimés sur base des points affectés à la classe qui correspond à cette distribution

DESCRIPTION :

Calcul des paramètres de chaque distribution qui modélise un des spots d'une tache, sur base de la partition des points qui constituent cette tache.

APPEL PRELIMINAIRE : **prepartition** ou **evalcrit**

PROCEDURE : partition (tache, taille, nspot, ldistrib)

MODULE : Extraction

TYPE : Modélisation gaussienne

PARAMETRES D'ENTREE :

tache = pointeur sur le tableau de structures *point* contenant les données relatives aux pixels d'une tache, la valeur du background moyen de la tache ayant été soustraite de l'intensité de chaque point
Type : pointeur

taille = nombre de points (pixels) constituant la tache
Type : unsigned short

nspot = nombre de centres de spots détectés dans la tache
Type : short int

ldistrib = pointeur sur le tableau de structures *distrib* contenant les paramètres des distributions gaussiennes 2D qui modélisent les spots de la tache
Type : pointeur

PARAMETRES DE SORTIE :

le numéro de classe contenu dans chacun des *taille* premiers éléments du tableau pointé par *tache* est le numéro de la distribution la plus proche du point contenu dans cet élément (= numéro de la classe à laquelle le point a été affecté)

DESCRIPTION :

Pour chaque point de la tache, calcul de la distance associée à la matrice au carré de chaque distribution. Chaque point est affecté à la classe correspondant à la distribution la plus proche

APPEL PRELIMINAIRE : **parametre**

PROCEDURE : prepartition (tache, taille, nspot,
 ¢reclasse, ldistrib)

MODULE : Extraction

TYPE : Modélisation gaussienne

PARAMETRES D'ENTREE : tache = pointeur sur le tableau de structures
 point contenant les données relatives aux
 pixels d'une tache, la valeur du background
 moyen de la tache ayant été soustraite de
 l'intensité de chaque point
 Type : pointeur

 taille = nombre de points (pixels) constituant la
 tache
 Type : unsigned short

 nspot = nombre de centres de spots détectés
 dans la tache
 Type : short int

 ¢reclasse = pointeur sur la liste de
 structures *pixels* qui contiennent les données
 relatives aux centres de spots détectés dans la
 tache
 Type : pointeur

 ldistrib = pointeur sur un tableau de structures
 distrib contenant au moins *nspot* éléments
 Type : pointeur

PARAMETRES DE SORTIE : le numéro de classe contenu dans chacun des
 taille premiers éléments du tableau pointé par
 tache est le numéro de la distribution la plus
 proche (distance euclidienne) du point contenu
 dans cet élément

 les *nspot* premiers éléments du tableau pointé
 par *ldistrib* contiennent les positions des
 distributions qui modélisent les spots de la
 tache

DESCRIPTION : Les positions des centres de spots détectés dans une tache sont affectées aux positions des distributions qui modélisent ces spots. Chaque point de la tache est affecté à la classe qui correspond à la distribution la plus proche de ce point, au sens de la distance euclidienne.

APPEL PRELIMINAIRE :

PROCEDURE : quantif (nomgel, xo, yo, dx, dy, xbkgrd, ybkgrd, &fstat)

MODULE : Extraction

TYPE : Quantification d'un gel 2D source.

PARAMETRES D'ENTREE : nomgel = pointeur sur la chaîne de caractères contenant le nom du gel 2D visualisé à l'écran vidéo.
 Type : pointeur

 xo, yo = coordonnées du point en haut et à gauche où est positionné l'image inversée du gel.
 Type : short int

 dx, dy = dimensions de l'image du gel, en nombre de colonnes et de lignes.
 Type : short int

 xbkgrd, ybkgrd = coordonnées du coin supérieur gauche de la zone où sont mémorisées les valeurs de background de chaque point de l'image inversée du gel.

 &fstat = pointeur sur l'entier qui contiendra l'état des accès aux fichiers.
 Type : pointeur

PARAMETRES DE SORTIE : fstat = 0 si un problème est apparu lors des accès aux fichiers
 = 1 si aucun problème n'est apparu
 Type : short int

DESCRIPTION : Quantification d'un gel 2D source à partir de son image inversée. Les résultats sont sauvés dans les fichiers *nomgel.spt* et *nomgel.ref*, créés par cette procédure.

APPEL PRELIMINAIRE : **background**

PROCEDURE : reorgcentre (liste1, liste2, nligne, &nligne)

MODULE : Extraction

TYPE : Gestion de listes de pixels

PARAMETRES D'ENTREE : liste1 = pointeur sur le tableau de pointeurs de listes de pixel. Chacune des *nligne* premières listes contient les centres de pics détectés sur des lignes de pixels d'une tâche.
Type : pointeur

liste2 = pointeur sur un tableau de pointeurs de listes vides de structures *pixel*.
Type : pointeur

nligne = nombre de pointeurs du tableau pointé par *liste1* qui pointent sur des listes non vides.
Type : pointeur

&nchaîne = pointeur sur l'entier qui reçoit le nombre de pointeurs du tableau pointé par *liste2* qui pointent sur des listes non vides.
Type : pointeur

PARAMETRES DE SORTIE : Dans le tableau pointé par *liste1*, tous les pointeurs pointent sur des listes de pixels vides. Dans le tableau pointé par *liste2*, les *nchaîne* premiers pointeurs pointent sur les listes de pixels générées par la procédure.

nchaîne = nombre de pointeurs du tableau *liste2* qui pointent sur des listes non vides.
Type : short int

DESCRIPTION : En comparant les positions des centres de pics d'une liste à l'autre dans le tableau *liste1*, cette procédure regroupe les centres de pics qui ne s'écartent pas de plus de *seuilreorg* colonnes l'un de l'autre. Les centres de pics sont regroupés dans des listes pointées par les pointeurs du tableau *liste2*.

APPEL PRELIMINAIRE :

PROCEDURE : traîter_gel2d_source (nomgel, quad, x0,y0)

MODULE : Fonction

TYPE : Gestion d'un gel 2D source

PARAMETRES D'ENTREE : nomgel = pointeur sur la chaîne de caractères
recevant le nom d'un gel 2D à traiter
Type : pointeur
entrée au clavier

 quad = numéro du quadrant actif (visualisé) en
fin de traitement
Type : short int

 x0, y0 = coordonnées du point en haut et à
gauche où est positionnée l'image du gel 2D
source. La totalité de cette image doit être
contenue dans le quadrant 0 (zéro) de la
mémoire image.

 nom = nom d'un gel 2D source à traiter
Type : string
entrée au clavier

PARAMETRES DE SORTIE : nomgel = pointeur sur la chaîne de caractères
contenant le nom du gel traité

 Le message "erreur lecture" est généré si un
problème apparaît lors de la lecture du fichier
contenant l'image linéarisée du gel source
Type : string
sortie à l'écran PC

 le message "erreur écriture" est généré si un
problème apparaît lors de l'écriture sur disque
des informations extraites de l'image
Type : string
sortie à l'écran PC

DESCRIPTION : Constitution d'un gel synthétique à partir d'un
gel 2D source. Les fichiers relatifs à ces deux
gels sont de même nom mais d'extension
différente : *.spt* et *.fim* respectivement.

APPEL PRELIMINAIRE :

PROCEDURE : video_spot (x0 ,y0 , x, y, varx, vary, intensité, couleur)

MODULE : Vidéo

TYPE : Visualisation d'un spot à l'écran vidéo

PARAMETRES D'ENTREE :

x0 ,y0 = coordonnées du point en haut et à gauche où est positionnée l'image d'un gel 2D
Type : short int

x,y = position du centre d'un spot, par rapport au coin supérieur gauche de l'image du gel
Type : float

varx, vary = variances en x (colonne) et en y (ligne) du spot
Type : float

intensité = intensité intégrée du spot
Type : float

couleur = niveau de gris de la croix qui synthétise le spot
Type : unsigned char

PARAMETRES DE SORTIE :

DESCRIPTION :

Une croix apparaît à l'écran vidéo, centrée sur le point de coordonnées x,y
Le niveau de gris de la croix est égal à *couleur*
La somme des longueurs des deux branches de la croix est proportionnelle au log 10 de l'intensité intégrée du spot
Les longueurs relatives des branches horizontale et verticale sont proportionnelles à la variance du spot en x et en y respectivement

APPEL PRELIMINAIRE :

PROCEDURE : videpixel (&liste)

MODULE : Extraction

TYPE : Gestion de listes de pixels

PARAMETRES D'ENTREE : & liste = pointeur sur un pointeur de liste de structures *pixel*.
Type : pointeur

PARAMETRES DE SORTIE : la liste pointée par &liste est une liste vide de structures *pixel*.

DESCRIPTION : Vidange d'une liste de structures *pixel* par libération de la mémoire qu'elle occupait. *liste* pointe sur une liste de pixels vide.

APPEL PRELIMINAIRE :

PROCEDURE : vision _gel2d_source (nomgel, quad, xo, yo)

MODULE : fonction

TYPE: gestion d'un gel 2D source

PARAMETRES D'ENTREE :

nom = nom d'un gel 2D source
 Type : string
 entrée au clavier

nomgel = pointeur sur la chaîne de caractères
 contenant le nom d'un gel 2D
 Type : pointeur

quad = numéro du quadrant où est visualisée
 l'image du gel
 Type : short int

xo, yo = coordonnées du point en haut et à gauche
 où est positionnée l'image du gel. Ce point doit
 être situé dans le quadrant *quad*
 Type : short int

PARAMETRES DE SORTIE :

nomgel = pointeur sur la chaîne de caractères
 contenant le nom du gel visionné (*nom*)

commentaire = commentaire associé au gel 2D
 source *nom*
 Type : string
 sortie à l'écran PC

le message "erreur lecture" est généré si un
 problème apparaît lors d'un accès au fichier
 contenant l'image linéarisée du gel source.
 Type : string
 sortie à l'écran PC

DESCRIPTION : Visualisation d'un gel 2D source par chargement
 dans le quadrant zéro de la mémoire image, et
 affichage du commentaire qui lui est associé.

APPEL PRELIMINAIRE :

PROCEDURE :	vision _gel2d_synth (nomgel, quad, xo, yo)
MODULE :	fonction
TYPE:	gestion d'un gel 2D synthétique
PARAMETRES D'ENTREE :	nom = nom d'un gel 2D synthétique Type : string entrée au clavier nomgel = pointeur sur la chaîne de caractères contenant le nom d'un gel 2D Type : pointeur quad = numéro du quadrant où est visualisée l'image du gel Type : short int xo, yo = coordonnées du point en haut et à gauche où est positionnée l'image du gel. Ce point doit être situé dans le quadrant <i>quad</i> Type : short int
PARAMETRES DE SORTIE :	nomgel = pointeur sur la chaîne de caractères contenant le nom du gel visionné (<i>nom</i>) le message "erreur lecture" est généré si un problème apparaît lors d'un accès au fichier. Type : string sortie à l'écran PC
DESCRIPTION :	Visualisation d'un gel 2D synthétique par chargement dans le quadrant zéro de la mémoire image.
APPEL PRELIMINAIRE :	

7. Glossaire.

Background : intensité lumineuse due à la coloration du gel en dehors des taches.

Enregistrement : dans un fichier.spt, il s'agit de l'information équivalant au contenu d'une structure artspot.

Mémoire image : partie de la carte vision destinée à mémoriser l'information visualisée à l'écran vidéo.

Numéro automatique : numéro attribué par l'algorithme de numérotation automatique à un spot.

Numéro manuel : numéro attribué par l'utilisateur à un spot.

Point : le plus petit élément constitutif d'une image. L'intensité lumineuse d'un point est codée sur 1 byte dans la mémoire image.

Pointeur : référence à une variable sur base d'un adressage à 32 bits (large or huge memory model).

Problème : lors d'un accès à un fichier (cf. spécifications de procédures), il peut s'agir d'un fichier non présent sur disque, d'un manque de place sur disque, d'une erreur de lecture ou d'écriture, d'un spot non trouvé dans un fichier.spt ...

Spot : ensemble des points qui correspondent à la localisation d'une protéine dans l'image d'un gel.

Tache : ensemble de points dont l'intensité lumineuse est supérieure à celle de leur background, et adjacents entre eux.