CONSERVATOIRE NATIONAL DES ARTS ET MÉTIERS

Centre Régional Languedoc-Roussillon

Spécialité : INFORMATIQUE

MÉMOIRE

Création d'une application intégrée pour la gestion et l'analyse de données protéomiques

---

Soutenu le 5 juillet 2005

par

Cédric BOUTTES

JURY

Président : Jean-Yves RANCHIN, Professeur, CNAM
Membres : Isabelle MOUGENOT, Maître de conférence, Université Montpellier II
Marc NANARD, Professeur, CNAM
Michel ROSSIGNOL, Directeur de Recherche, INRA
Thierry HOTELIER, Ingénieur, ENSA Montpellier

REMERCIEMENTS

Ce travail a été effectué au sein de l'unité de recherche 1199 en protéomique de l'INRA de Montpellier. Je remercie tout d'abord Michel ROSSIGNOL, directeur de cette unité, pour m'avoir aidé à réaliser ce mémoire dans les meilleures conditions.

J'exprime spécialement ma reconnaissance à Isabelle MOUGENOT pour sa gentillesse et son aide précieuse pendant la rédaction de ce mémoire. J'ai particulièrement apprécié ses conseils et les efforts fournis pour la correction de ce document.

Je remercie aussi Marc NANARD pour m'avoir permis de réaliser ce mémoire sur ce sujet.

Je tiens également à remercier Olivier MARTIN pour ses conseils et son aide lors de la relecture.

Mes remerciements vont également à ma famille qui m'a soutenu et encouragé tout au long de la rédaction de ce manuscrit.

Enfin, je remercie William MOTHES ainsi que toutes les personnes qui m'ont apporté leur soutient.

CONVENTIONS TYPOGRAPHIQUES

Une abréviation de type [G] est une référence dans le glossaire.

Ensuite, les abréviations suivantes introduisent des références dans la partie bibliographique :
[i] : est une référence sur un lien internet
[a] : est une référence sur un lien article
[L] : est une référence sur un livre
[r] : est une référence sur un rapport d'étudiant

SOMMAIRE

1 Introduction

Les espèces végétales sont à la base de l'alimentation de l'homme depuis ses origines et font, de ce fait, l'objet d'une attention toute particulière avec un souci constant d'amélioration de leurs qualités (nutritives, de rendement, de résistance aux maladies …). Derrière toute variété améliorée se trouve un ensemble de milliers de gènes dont la combinaison explique les caractéristiques de la plante. Alors que dans les années 1980, on étudiait ces gènes un par un, la génomique *(science de l'étude des gènes)* permet maintenant d'espérer dresser le catalogue de tous les gènes d'un organisme grâce au décryptage du produit du séquencage de l'ADN[1]. Ce travail est rendu possible grâce à des moyens performants dans la gestion et l'analyse des données de séquences apportés par l'utilisation de la bioinformatique, discipline visant à mettre les sciences formelles dont l'informatique à la disposition de la biologie.

L'objectif est désormais de comprendre la fonction des gènes présents au niveau des chromosomes ainsi que leurs différents niveaux de régulation et d'interactions. Dans ce nouveau champ d'investigation appelé génomique fonctionnelle (ou post-génomique), une des activités principales est d'étudier et d'analyser le niveau d'expression des gènes (à partir des ARNs[2] transcrits) ainsi que leur produit d'expression (les protéines). L'étude du niveau d'expression des gènes est au centre d'une discipline appelée transcriptomique (ou étude des ARNs transcrits) tandis que la protéomique s'attache à décrire l'ensemble des protéines produites par les gènes d'un organisme à un instant donné.

Les techniques permettant l'acquisition des données transcriptomiques et protéomiques génèrent des quantités importantes de données qu'il est souvent nécessaire de corréler entre elles pour avancer dans la compréhension des mécanismes de régulation de l'activité des gènes. Ces données sont elles même à relier en amont avec les caractéristiques de ces gènes : séquences des gènes, positions des gènes sur la carte des chromosomes, niveaux de variabilité et mutations de ces gènes …

La conception d'un système d'information capable de gérer, intégrer et mettre en relation toutes ces données se révèle alors un atout majeur pour le biologiste qui veut à la fois connaître tout ce qui est disponible sur la donnée sur laquelle il travaille, mais également découvrir par la même occasion, de nouveaux liens sur des données connexes et donc s'orienter vers de nouvelles pistes dans ses projets de recherches. Différents outils de bioinformatique peuvent ensuite être utilisés pour vérifier la qualité des données, les analyser et en faciliter l'extraction de connaissances par le biologiste.

Tout ceci définit les axes d'un programme de bioinformatique ambitieux mis en place par le consortium Génoplante, qui est un programme marquant de génomique végétale mettant en synergie à la fois recherche publique et recherche privée. Le système d'information GpiIS *(Genoplante-info information system),* sous la responsabilité de l'équipe du centre de ressource bioinformatique de Génoplante (Génoplante Info), a pour objectif de réunir, au sein d'un même entrepôt, l'ensemble des données végétales de toutes natures, générées en masse, par les différents laboratoires partenaires du consortium.

Mon travail s'inscrit dans le cadre du développement d'un des modules du système d'information GpiIS. Ce module appelé GnpProt *(GeNoPlante Proteomic module)* a pour rôle de prendre en charge la gestion et l'analyse de données protéomiques ainsi que leur intégration avec les autres données biologiques contenues dans GpiIS. Un autre besoin s'est fait jour et concerne les laboratoires de protéomique impliqués dans Génoplante. Une version appropriée de GnpProt leur était nécessaire pour la gestion et l'analyse de leurs données en local. Il s'agissait donc dans ce contexte, de m'adapter à leurs besoins et de leur proposer une « version locale » de GnpProt, nommée alors de manière à la différencier, ProteomIs (Proteome Information system).

---

[1] *Acide DésoxyriboNucléique, molécule support de l'hérédité*

[2] *Acide RiboNucléique, molécule servant d'intermédiaire entre l'ADN et les protéines*

Je présente dans ce mémoire, mes différentes contributions, que cela soit des contributions uniquement liées à GnpProt, des contributions uniquement liées à ProteomIs ou bien encore des contributions communes aux deux (ProteomIs/GnpProt).

Les réalisations présentées, sont à mettre à l'actif de l'unité de recherche en protéomique de l'INRA de Montpellier dirigé par Michel Rossignol, dans son ensemble. Dans ce contexte, j'ai bénéficié de l'aide de Thierry Hotelier, ingénieur d'étude en bioinformatique, et j'ai encadré le travail de trois stagiaires sur diverses fonctionnalités du système. J'ai aussi conduit une collaboration active avec les biologistes pour l'évaluation des besoins ainsi qu'avec les informaticiens de Génoplante Info.

Ce mémoire décrit l'ensemble de mon travail et la démarche qui m'a permis d'aboutir à une version fonctionnelle de ProteomIs/GnpProt. Le plan de ce document est organisé de la manière suivante :

A la suite de cette première partie introductive, le **chapitre 2** décrit les principaux partenaires du projet ProteomIs/GnpProt et apporte un niveau de précision supplémentaire sur les objectifs de ma mission.

Le **chapitre 3** apporte des notions du vocabulaire biologique indispensables pour la compréhension du travail présenté dans ce mémoire.

Le **chapitre 4** est une étude prospective sur l'existant dans la littérature en matière de méthodologies, concepts, technologies et applications afférents à la bioinformatique. Nous nous focaliserons plus volontiers sur les sources de données du domaine ainsi que sur les outils de traitement qui se révèleront d'intérêt pour le projet.

Dans le **chapitre 5**, nous entamons la discussion concernant le choix des solutions permettant de répondre à chacun des objectifs et sous-objectifs de notre projet. Cette réflexion s'appuie sur l'existant disponible au sein des laboratoires mais aussi sur les solutions présentées dans le chapitre 4 Etat de l'art.

Dans le **chapitre 6**, nous abordons le choix d'une solution proprement dite au travers d'une démarche d'analyse et de conception. La **partie analyse** est conduite à l'aide d'UML selon une approche itérative guidée par les cas d'utilisation et inspirée de la méthode UP. L'ensemble des modèles obtenus sert ainsi de support à la conception détaillée du projet. La **partie conception** est organisée autour de la présentation des solutions architecturales et programmatiques qui ont conduit à l'implémentation des grandes fonctionnalités de l'application. Cependant, avant de présenter chacune de ces solutions techniques, le choix de celles-ci sera argumenté et la solution choisie sera éventuellement comparée à d'autres solutions imaginables.

Le **chapitre 7** présentera l'aboutissement du travail d'implémentation montrant l'application dans son état d'avancement par rapport aux objectifs du projet.

Enfin dans le **chapitre 8** je conclurai sur l'ensemble du projet et de ma démarche accomplie et donnerai des perspectives au projet.

2 Le contexte

Il s'agit ici de donner un aperçu de la problématique et des enjeux du projet au travers notamment des organismes et des laboratoires impliqués. Les lignes directrices qui ont guidé notre travail sont présentées dans la partie « lettre de mission ».

2.1 Les partenaires du projet

Afin de faire mieux comprendre le contexte collaboratif de GnpProt/ProteomIs, nous allons présenter le consortium Génoplante ainsi que les différents laboratoires associés au projet. Le centre de recherche de l'INRA de Montpellier et notamment son unité protéomique qui avait en charge le développement du module protéomique font l'objet d'une section spécifique.

2.1.1 Présentation du consortium Génoplante et de Génoplante-Info

➢ Le consortium Génoplante

L'importance de fédérer, au niveau national, les efforts consentis autour de la génomique végétale n'est plus à démontrer. Il en va de notre capacité à maîtriser les enjeux économiques et environnementaux qui en sont à la clé. Génoplante a été créé à ce titre et associe des établissements scientifiques publics (CNRS, INRA, IRD) ou semi-publics (CIRAD) ainsi que les principales sociétés privées impliquées dans l'amélioration et la protection des cultures (Biogemma, Bayer Cropscience, Bioplante).

Des programmes analogues à Génoplante existent de par le monde. Nous pouvons ainsi citer le « Plant Genome Initiative » aux Etats-Unis, les programmes Zigia et Gabi en Allemagne, ou encore le « Rice Genome Research Program » au Japon. Pour illustration, il s'agit pour ces programmes de concentrer leurs efforts sur les ensembles de gènes d'intérêt afin de mieux cultiver et de mieux produire. Les impacts sont à tout niveau : au niveau du semencier, de la filière agricole dans son ensemble ou encore du consommateur.

Génoplante peut ainsi être vu comme un dispositif structuré de recherche et de valorisation qui va profiter à tous (partenaires publics et privés) et qui va favoriser la pluridisciplinarité. Par pluridisciplinarité, nous entendons que des travaux sont menés au niveau de la génomique mais aussi au niveau de la transcriptomique ou encore de la protéomique. De même, pour mener à bien les différentes tâches, Génoplante s'est doté en interne d'un département de bioinformatique appelé Genoplante-Info. Génoplante-Info est à l'origine de ProteomIs / GnpProt et est présenté plus en avant ci-dessous.

➢ Génoplante-Info

Génoplante-Info **[i1]** a été créé en 2000 par Génoplante et est abrité par Infobiogen à Evry. Genoplante-Info correspond, de manière partielle, à une unité de recherche informatique en génomique de l'INRA et peut, en outre, être vu comme un centre de ressource de bioinformatique.

Les objectifs de Génoplante-Info sont nombreux puisqu'il s'agit d'une unité de recherche, cependant d'un point de vue « services offerts », il lui faut mettre à disposition de la communauté des outils de traitement et d'analyse des données biologiques (Predotar **[i80]**, ...). Il lui faut également collecter, structurer et stocker les multiples données générées par une centaine de projets scientifiques de Génoplante **[a1]**. Nous pouvons citer, à ce titre, les systèmes d'information FlagDB++ **[i42]** ou GeneFarm **[i79]**.

Depuis récemment, Génoplante-info a entrepris une démarche d'intégration en proposant la construction de GpiIS, un système d'information unique pour l'ensemble des données biologiques collectées dans les différents projets.

GpiIS contient plusieurs volets dont GnpProt et nous y reviendrons plus en détail dans la lettre de mission.

Plusieurs remarques méritent d'être relevées, à ce niveau. Tout d'abord, la construction de GpiIS doit faire l'objet d'une interaction étroite entre les informaticiens de Génoplante-Info et les informaticiens appartenant à d'autres unités de l'INRA. Il en va ainsi pour le module GnpProt dont nous sommes responsables sur Montpellier. En second lieu, les projets coordonnés par Génoplante ne sont pas tous publics et par voie de conséquence, les données produites sont pour certaines à accès restreint. Cet état de fait oblige à une politique de restriction et de contrôle des accès et oblige à une certaine rigueur dans la transmission des données et briques logicielles développées.

2.1.2 Présentation du centre INRA de Montpellier

Créé en 1946, l'Institut national de la recherche agronomique **[i2]** est un établissement public à caractère scientifique et technologique, placé sous la double tutelle des ministères chargés de la Recherche et de l'Agriculture.

Le centre INRA de Montpellier **[i3]** (682 agents, dont 276 scientifiques et ingénieurs) possède 13 implantations en Languedoc-Roussillon (dont 500 ha de terrains expérimentaux). La plupart des laboratoires se trouvent rassemblés sur le campus de l'Ecole Nationale supérieure Agronomique de Montpellier (ENSA-M). L'ensemble est lui-même intégré au complexe international Agropolis. Le caractère pluridisciplinaire des activités développées sur le centre favorise son insertion dans des problématiques régionales et permet d'instaurer le développement de nombreuses collaborations avec d'autres organismes de recherche et avec l'ensemble des partenaires du monde agricole et agro-alimentaire.

Les activités du centre INRA de Montpellier, sont réunies autour de quatre axes fédérateurs combinant à la fois des activités de recherche de base, des programmes de développement et de formation.

Un premier axe concerne les agros-industries en tentant de répondre aux exigences de qualité et de diversification des industries agroalimentaire. Les programmes menés sont axés principalement sur les produits de la vigne et des céréales.

Un deuxième axe d'activité concerne la diversité génétique avec l'études des ressources phytogénétiques, l'utilisation de la diversité génétique et la variabilité et génie génétique chez les pathogènes d'invertébrés.

Un troisième axe concerne le génie agroécologique dans lequel l'importance est donnée aux culture/environnement pour la gestion du milieu.

Enfin le quatrième axe d'activité est la biologie du développement. A ce niveau les recherches s'orientent dans trois domaines qui sont la physiologie de la fructification, l'adaptation aux facteurs du milieu et la modification de l'expression des gènes au cours du développement.

En collaboration avec les acteurs du monde agricole et agroalimentaire régional, le centre INRA de Montpellier met en oeuvre un programme de recherche-développement (recherches pour et sur le développement régional), en partenariat avec les partenaires économiques et les institutions publiques en Languedoc - Roussillon. Le centre développe de nombreuses relations avec des instituts de recherche implantés partout dans le monde, notamment par l'intermédiaire d'Agropolis. Il est également un centre d'accueil pour les scientifiques étrangers et participe directement au transfert des connaissances et à l'établissement de réseaux communautaires et internationaux, principalement en Méditerranée.

2.1.3 Présentation de l'unité de recherche en protéomique de l'INRA de Montpellier et des laboratoires partenaires du projet

Les unités de recherche en protéomique, partenaires du projet, qui ont activement collaboré pour la réalisation du cahier des charges de ProteomIs/GnpProt sont listées ci-dessous.

- **UR 1199 de l'INRA de Montpellier**
- **UMR 5546 du pôle de Biotechnologie Végétale de Toulouse**
- **UMR 5168 du CEA de Grenoble**
- **l'URPVI de l'INRA de Nantes**

Ces unités vont être vues, non seulement, comme acteurs impliqués dans les développements mais aussi comme acteurs producteurs des données. Elles ont pour point commun de disposer de plateformes de protéomique particulièrement performantes et de travailler majoritairement sur le protéome de la plante modèle Arabidopsis thaliana (l'arabette des dames).

Je présente en **annexe 5** l'ensemble de ces unités, à l'exception de l'unité de recherche en protéomique de l'INRA de Montpellier à laquelle je suis rattaché et qui, de ce fait, sera présentée ici.

➢ **Unité de Recherche en Protéomique de l'INRA de Montpellier (UR 1199) [i70] :**

Le personnel permanent de l'unité se compose de 2 chercheurs, 2 ingénieurs et 4 techniciens et administratifs, renforcé par 3 ingénieurs et post-doctorants. Le directeur de cette unité (Michel Rossignol) est mon interlocuteur le plus proche dans la définition des besoins et la mise à jour du cahier des charges dans le projet ProteomIs/GnpProt.

L'unité protéomique de l'INRA de Montpellier (équipée depuis 1995 pour le séquençage chimique de protéines et de peptides purifiés) a bénéficié en 1999 et 2000, du soutien de l'INRA, de la région Languedoc-Roussillon et de la Génopole de Montpellier pour évoluer vers une plate-forme protéomique à haut débit.

Ainsi, l'installation récente d'équipements performants pour la séparation des protéines (électrophorèse bidimensionnelle), la préparation des échantillons (robotique) et leur identification (spectrométrie de masse MALDI TOF et ESI MS) a permis l'évolution vers une approche permettant un débit "élevé" compatible avec des objectifs de génomique fonctionnelle.

Les activités de recherche sont ciblées sur l'analyse du protéome de Arabidopsis thaliana (ou Arabette des dames). Il s'agit d'une petite plante à fleurs, de la famille des crucifères, dont le génome de petite taille facilite son étude. De ce fait, Arabidopsis thaliana, à l'instar des autres organismes modèles, sert de tremplin pour amorcer l'étude de génomes/protéomes plus complexes.

Arabidopsis thaliana

Deux axes de recherche complémentaires sur le protéome d'Arabidopis sont principalement conduits au laboratoire :

- Un premier programme concerne la construction des répertoires de protéines exprimées dans les principaux organes de la plante. L'approche comprend l'établissement de cartes de référence par électrophorèse bidimensionnelle et spectrométrie de masse. L'ensemble des informations est ensuite structuré sous forme de carte cliquable avec une interface web.
- La deuxième action concerne l'étude de la réponse à la carence en phosphate. L'analyse de séries temporelles de gels d'électrophorèse bidimensionnelle permet d'identifier les classes de protéines dont l'expression est affectée de façon similaire (protéomique quantitative différentielle). L'identification des fonctions cellulaires correspondantes est ensuite réalisée par confrontation avec les ressources établies précédemment. Ces approches conduisent à la fois à une analyse physiologique intégrée et à l'identification de classes de protéines nouvelles associées au statut nutritionnel des plantes.

Des problématiques connexes concernent aussi :

- la caractérisation à large échelle de modifications post-traductionnelles en réponse aux contraintes abiotiques : signalisation du stress ferrique chez *Arabidopsis thaliana*.
- l'élaboration de modèles statistiques pour l'analyse des données protéomiques d'expression.

Par ailleurs, la plate-forme a pour mission, en fonction de ses capacités, de répondre aux besoins d'identification de protéines de différents laboratoires en assurant une activité de prestation de service.

Elle assure ce rôle en association avec le site du CCIPE de Montpellier, instrumentalisé de manière complémentaire et orientée vers la protéomique structurale. Le site de l'unité de recherche en protéomique de Montpellier (UR199) et le site CCIPE constituent à eux deux la plate-forme protéomique de la Génopole de Montpellier **[i13]**.

2.2 La lettre de mission

Les données biologiques sont, par essence même, complexes, évolutives et incomplètes. Ces données peuvent par être exemple des séquences de marqueurs génétiques (EST, FST, SNP, ...) ou encore des cartes génétiques pour la génomique, des profils d'expression de gènes pour la transcriptomique ou encore des patterns de domaines fonctionnels pour la protéomique. Les laboratoires, qui sont à l'origine de ces données, les produisent en masse à l'aide de biotechnologies de plus en plus performantes et sont, en outre, distribuées géographiquement. Par exemple et si l'on se réfère aux laboratoires précédemment cités, les analyses menées sur les plateformes de protéomique comprennent de multiples étapes et génèrent des données sur une large échelle.

Un des premiers problèmes qui se posent alors, se révèle être le stockage des données au sein même du laboratoire. Ainsi ce stockage peut avoir une visée uniquement interne avec par exemple le stockage des données sous forme de fichiers au format Excel ou dans un format texte. Ce stockage peut également avoir une visée externe avec la nécessiter de publier et de partager ces données avec le reste de la communauté.

Enfin, la comparaison des données de même nature comme la confrontation de données de différentes natures pour faire émerger ou valider de nouveaux modèles biologiques sont des activités essentielles en biologie. Les systèmes d'information vont venir en support de ces activités.

Ces considérations ont conduit Génoplante-Info à décider de la construction du système GpiIS **[i6]** (**annexe 6**) qui a pour ambition la gestion et la restitution de toute donnée biologique provenant des projets pluridisciplinaires Génoplante. A cet effet, et pour respecter les différentes problématiques biologiques sous-jacentes, GpiIS possède un socle commun et est organisé en modules. Ainsi GnpProt en constitue le volet protéome de GpiIS. Un tel système à visée intégrative se veut une réponse aux questions complexes que sont amenés à se poser les biologistes et qui font appel à toutes sortes de données.

De manière exhaustive, GpiIS fait l'objet de cinq modules qui chacun s'articule autour d'une catégorie de données biologiques particulière :

- GnpSeq (Genoplante Sequence) pour la gestion des données de séquences (EST, mRNA)
- GnpMap (Genoplante Map) pour la gestion des données de cartographie génétiques (cartes physiques, cartes génétiques, QTL)
- GnpArray (Genoplante Array) pour la gestion des expériences sur le transcriptome (puces à ADN)
- GnpSNP (Genoplante SNP) pour la gestion des données sur le polymorphisme génétique
- GnpProt (Genoplante Protéome) pour la gestion des données protéomiques

Les différents modules sont construits de manière indépendante et pour l'instant, se trouvent à différents stades d'élaboration. Leur connexion se fera ensuite de manière progressive. Les contraintes imposées par l'assemblage portent à la fois sur la politique de collaboration entre les équipes, le modèle de données unificateur et les technologies retenues.

Le travail présenté dans ce mémoire porte sur l'analyse, la conception et l'implémentation du module GnpProt avec l'idée sous-jacente de l'intégration des données protéomiques avec les données provenant des autres modules. Les données protéomiques qui alimenteront la base intégrative seront des données qui auront été au préalable validées par les biologistes et disponibles pour être consultés par tous les membres des laboratoires partenaires qui auront un accès au système.

Une seconde idée forte, en plus de la stratégie intégrative, est de proposer GnpProt sous une déclinaison locale. Dans ce contexte, la base de données sera dénommée ProteomIs et permettra d'apporter une gestion des données interne a chacun des laboratoires partenaires. Nous désignerons dans le mémoire le système par ProteomIs/GnpProt lorsqu'il s'agira de sa vision d'ensemble, par GnpProt lorsqu'il s'agira de la brique logicielle intégrée et par ProteomIs lorsqu'il s'agira du système local (**document 1**).

GnpProt/ProteomIs est à concevoir comme un système d'information et nous considérons non seulement les aspects statiques des données (au travers de la base de données) mais aussi les aspects dynamiques (c'est à dire les traitements sur ces données). Les données biologiques nécessitent de nombreux traitements en vue de les analyser ou bien de contrôler leur qualité. Dans ce sens, plusieurs développements sont à mettre en oeuvre :

- contrôle d'intégrité : une des priorités sera d'éliminer la redondance des séquences protéiques inhérente aux approches du domaine au travers d'une technique de « clustering »
- caractérisation des séquences : une batterie d'outils est à envisager pour effectuer une recherche de similarité ou encore une recherche de motifs fonctionnels **[G14]**.

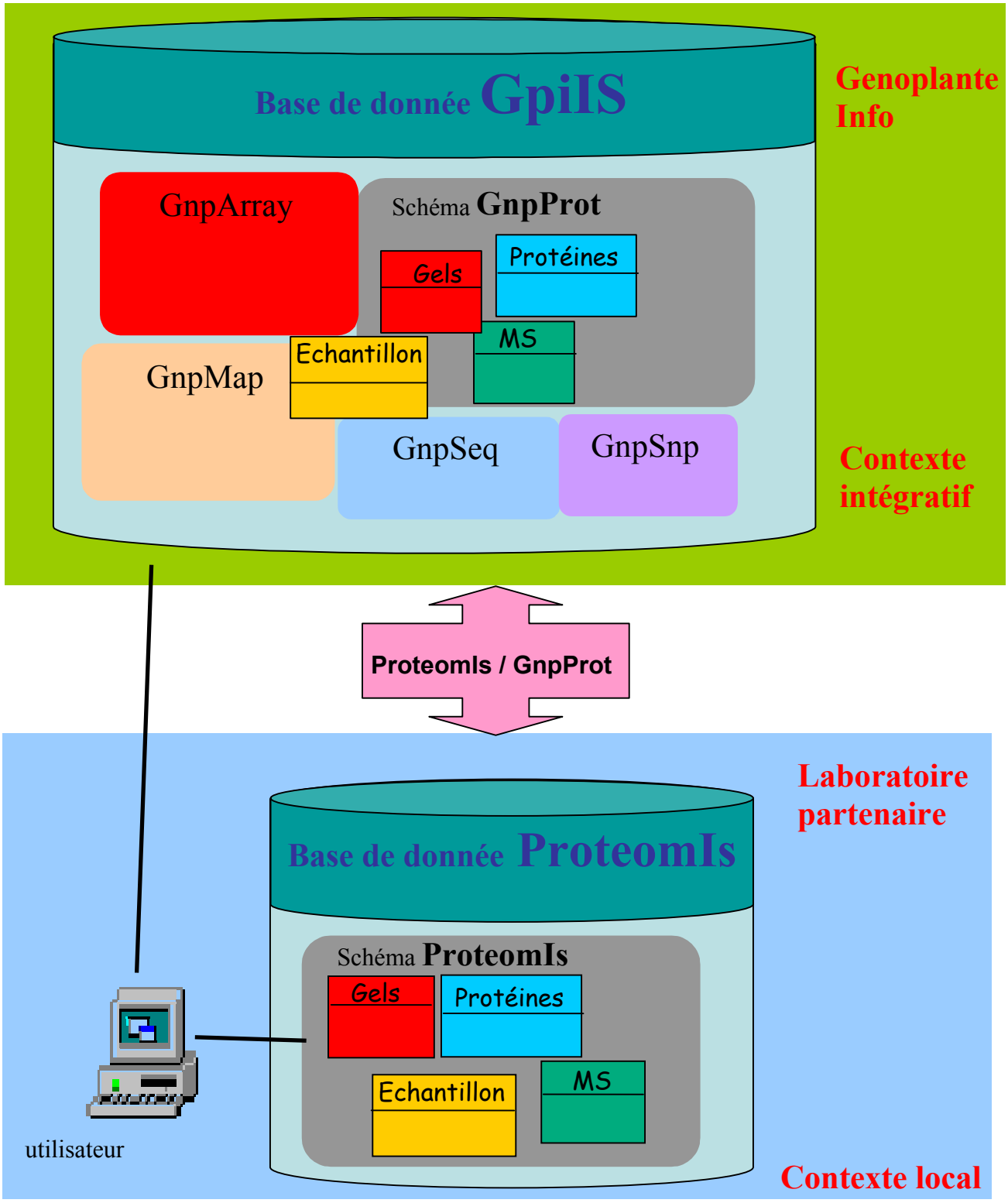
De manière synthétique, le travail présenté tout au long de ce mémoire s'articule autour d'une double approche :

- une approche collective autour du module GnpProt avec la réalisation d'un cahier des charges puis le développement du module proprement dit avec le respect des contraintes inhérentes à GpiIS.
- une approche plus personnelle avec une proposition de mise en oeuvre autonome du module (ProteomIs) et des extensions orientées à la fois vers une prise en charge des traitements (similarité, recherche de motifs, ...) et une connexion avec les bases de données factuelles du domaine.

Avant d'entreprendre une réflexion sur une solution informatique pouvant répondre aux différents objectifs du projet nous devrons avant présenter un état de l'art des solutions existantes.

Nous ferons alors un inventaire des différentes applications existantes dans la gestion des données biologiques puis plus spécifiquement protéomiques.

Ensuite, nous verrons les méthodes disponibles pour analyser les données protéomiques d'un point de vue bioinformatique. Enfin, nous analyserons les approches courantes permettant d'assurer l'interopérabilité au niveau des bases de données en bioinformatique. Dans l'immédiat afin de mieux comprendre la suite du mémoire, il est nécessaire de posséder quelques prérequis sur la biologie moléculaire, la bioinformatique en général et l'approche protéomique. C'est l'objectif de la partie suivante.



Document 1 : Les deux contextes d'utilisation de la base de donnée protéomique ProteomIs /GnpProt

3 Prérequis

3.1 Notions de biologie moléculaire

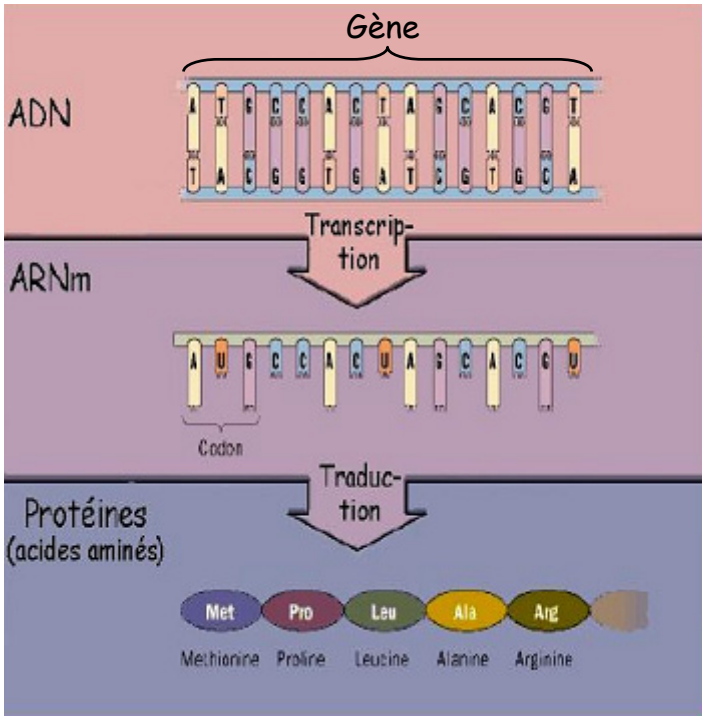
➢La synthèse des protéines, l'expression des gènes

Tout d'abord quelques éléments de biologie pour comprendre ultérieurement la nécessité de l'outil informatique pour gérer et analyser les données en biologie moléculaire. Tout être vivant est constitué de cellules qui contiennent toutes dans leurs noyau une molécule nommée **ADN** (Acide Désoxyribo Nucléique) constituant essentiel de nos **chromosomes**.

La molécule d'ADN peut être comparée à un livre dans lequel serait écrit la totalité de notre information génétique qui est le **génome**. L'originalité de ce livre est qu'il est écrit dans un alphabet à 4 lettres A, T, C et G (pour Adénine, Thymine, Cytosine et Guanine) qui sont les 4 bases de l'ADN. Cette information est transformée en partie en **protéines** qui sont les éléments constructifs de nos cellules. La portion d'ADN codant pour une protéine correspond à un **gène**.

Les explications et schémas ci-dessous (**document 2**) résument succinctement le passage du gène à la molécule fonctionnelle, la protéine. Ce passage correspondant à l'**expression des gènes** se fait en deux grandes étapes que sont la transcription et la traduction, décrites ci dessous :

1. La transcription : Dans cette première étape, l'ADN est transcrit en **ARNm** (Acide Ribo Nucléique messager) par une enzyme appelé RNA-polymérase. Dans cette étape, la Thymine est transformée en Uracile. L'ensemble des transcrits (les ARN messagers) d'un organisme à un instant donné est nommé **transcriptome**.
2. La traduction : Dans cette deuxième étape l'ARNm va être décodé par des organites cellulaires appelés ribosomes qui réalisent la synthèse de la protéine. Ces protéines sont constituées d'**acides aminés** dont l'ordre et la nature mis en place découlent de la séquence de l'ARNm. En effet trois nucléotides de l'ARNm forment un codon et à un codon correspond un acide aminé.



Document 2 : L'expression d'un gène : de l'ADN à la protéine

➢ Les modifications post-traductionnelles

Une fois la protéine synthétisée, on assiste à de nombreuses modifications chimiques appelées **modifications post-traductionnelles**. On les distingue des modifications co-traductionnelles qui se produisent alors que la traduction se poursuit encore. Les modifications post-traductionnelles peuvent correspondre à l'addition de groupements chimiques qui permettent de réguler le comportement de la protéines (**phosphorylation**, glycosilation et acétylation sont des exemples courants). D'autre changements incluant des clivages concernent les mécanismes d'adressage de la cellule, qui dictent la destination des protéines qu'elle produit et donc leur localisation. Toutes ces modifications peuvent avoir lieu dans la cellule, dans les organites ou hors de la cellule.

➢ Le séquençage et le décryptage des génomes

Ainsi chaque organisme vivant peut être décrit et suivi au cours de son évolution par ses gènes et les protéines qu'il synthétise. Il est intéressant pour divers domaines (p. ex en médecine ou en agronomie) de traiter et de stocker ces séquences de gènes ou ces séquences de protéines. Dans ce contexte un des objectifs actuels majeur de la biologie moléculaire est par exemple de construire la carte génétique de l'homme ou celle de plantes cultivées. Sur cette carte doivent être localisées de nombreux gènes dont les plus recherchés sont, par exemple dans le cas de l'homme ceux impliqués dans des maladies génétiques comme la myopathie (un des objectifs du Généthon) et, dans le cas des plantes cultivées, ceux d'intérêt agronomique. Les gènes une fois localisés sont séquencés. C'est à dire que l'on doit déchiffrer la portion d'ADN correspondant en suite de lettres ATCG. Enfin, on détermine la séquence de la protéine correspondante pour caractériser la fonction du gène. Le séquençage peut aussi être fait sur la totalité du génome sans se soucier de la position des gènes. Ce n'est que par la suite les séquences sont analysées et leurs fonctions déterminées (p. ex : en les comparant avec d'autres séquences connues). On parle alors de décryptage des génomes.

3.2 Les bases de données de séquences

Sachant que plusieurs milliers de gènes peuvent piloter le fonctionnement d'un organisme, la gestion des séquences n'est possible qu'en collaboration avec des outils informatiques adaptés. Avant d'effectuer une présentation des systèmes les plus connus, nous commencerons par présenter les principaux formats de stockage utilisés et l'origine de leur utilisation. Cette partie sera notamment pour nous l'occasion de comprendre la notion de numéro d'accession qui sera utilisé tout au long de ce mémoire.

3.2.1 Les formats de stockage

1. Les banques de données :

Les banques de données ont été les premiers systèmes utilisés par les biologistes pour stocker les données sur les séquences. Ces banques de données correspondent à des collections de fichiers obéissant à un format précis et organisé dans différents répertoires. Pour accéder à ces données les biologistes utilisent des systèmes comme le logiciel SRS (Sequence Retrieval System **[i52]**) conçu spécialement pour l'indexation et l'interrogation de ces banques. La popularité de ces systèmes de stockage remonte à la période de « l'après séquençage » dans les années 80 où les biologistes commençaient à fournir en masse leurs données de séquences sous la forme de fichiers. Aujourd'hui, la procédure de soumission des séquences est toujours la même. Les biologistes fournissent des fiches d'annotation aux centres responsable de la gestion des séquences.

Celles-ci contiennent les séquences du gène ou de la protéine découverte, les noms donnés à ceux-ci par le laboratoire, leur localisation dans la cellule, mais aussi et surtout, des commentaires sur ces données, comme la fonction biologique putative d'une protéine ou encore les maladies pouvant être impliquées par un gène. Chacune de ces soumissions est ensuite convertie dans le format spécifique de la banque à laquelle la séquence est soumise. Ce format a été conçu pour être facilement lu par des humains mais aussi par des programmes appelés aussi « parser » **[G5]** conçus par les utilisateurs des banques pour récupérer les informations de certains champs en particulier. Pour cela, par exemple dans une entrée de la base de donnée protéiques SWISSPROT, chaque fichier de séquence obéit à un format propre à base d'étiquettes (2 lettres) qui renseignent sur la nature du champ d'information.

Voici un extrait de quelques un des champs de ce format pour un fichier d'une séquence :

```
AC   P02144;          <-- L'identifiant unique de la séquence (appelé aussi numéro d'accession)
DT   21-JUL-1986 (Rel. 01, Created)
DE   Myoglobin.
GN   MB.
OS   Homo sapiens (Human).
RL   Biochim. Biophys. Acta 251:482-488(1971).
DR   EMBL; M14603; AAA59595.1; -.
DR   PIR; A02464; MYHU.
DR   InterPro; IPR000971; Globin.
KW   Heme; Oxygen transport; Transport; Muscle; Polymorphism;
FT   INIT_MET     0     0
FT   METAL       64    64       IRON (HEME DISTAL LIGAND).
SQ   SEQUENCE   153 AA;  17053 MW;  5F84A2C481B8F0D5 CRC64;
     GLSDGEWQLV LNVWGKVEAD IPGHGQEVLI RLFKGHPETL EKFDKFKHLK SEDEMKASED
//
```

Le champ **AC** est l'identifiant d'accès à la séquence. En effet chacune des séquences soumises (appelé aussi entrée) dans la base doit pouvoir être identifiée de manière unique et permanente (comme pour nous notre numéro de sécurité sociale). Pour cela on attribut à chacune des fiches résultantes un identifiant unique couramment appelé **numéro d'accession** dans la communauté des biologistes. Chaque banque de données utilise sont propre format d'accession pour identifier ses fiches sur les séquences (voir **annexe 7** les différents formats). Le champ **DT** (**D**a**T**e) renseigne sur les différentes dates concernant l'entrée (création, modification de séquence ou d'annotation). Le champ **DE** (**DE**scripteur) renseigne sur la nature de la protéine et le champ **GN** est le **N**om de **G**ene. Le champ **OS** contient le nom (latin et anglais) de l'espèce et de l'organisme (**O**rganism **S**pecie). Les différents champs **RN** (RP, RX, RT, RA, RL) concernent les références bibliographiques de séquences. La ligne **DR** fournit des **liens croisés** sur les autres banques de données. Ceci signifie que l'on obtient le numéro d'accession de la séquence correspondante dans d'autres bases de données. Le champ **KW** (**K**ey**W**ord ou mot clé) et le champ **FT** (**F**eature **T**able) est pour les informations et les annotations concernant la séquence. Si les informations sont non vérifiées expérimentalement, le mot "potential" ou "conflict" est ajouté. Enfin, le dernier champ est **SQ** pour SéQuence. Le terminateur d'entrée est //

2. Les bases de données :

La tendance actuelle tend à généraliser de plus en plus l'usage des bases de données pour la gestion des données biologiques afin de corriger les faiblesses des banques de données et profiter de la puissance des SGBD en matière de gestion des données. La culture des fichiers à plat étant encore cependant fortement ancré dans la communauté des biologistes, les propriétaires de bases de données biologiques fournissent toujours une distribution de leur données sous la forme d'une banque de données sur leur site ftp. Dans un soucis de simplification nous utiliserons dans la suite de ce mémoire le terme de base de données pour désigner les systèmes pouvant se présenter sous les deux formes.

3.2.2 Classification des bases de données de séquences

Nous effectuerons ici une classification rapide des systèmes les plus connues. Nous insisterons sur ceux utilisés pour alimenter la base ProteomIs/GnpProt en séquences protéiques et vers lesquels celle-ci devra offrir un certain nombre de liens hypertextes. En effet, l'ensemble des bases de données présentées est disponible sur le réseau Internet. Ceci présente le double avantage de permettre un accès rapide et ouvert aux données par les biologistes et une intégration forte des différentes banques de données entre elles grâce à l'utilisation de liens hypertextes.

➢ Les bases de données de séquences généralistes

Elles accueillent des données d'une nature particulière : par exemple des séquences nucléotidiques, protéiques, les références bibliographiques … Elles sont généralistes dans le sens où elles accueillent des données associées à tout organisme vivant ou toutes familles de molécules. En ce qui concerne les séquences nucléotidiques, trois systèmes se partagent toute l'information mondiale. Il s'agit de la banque de séquence EMBL (European Moleculary Biology Laboratory) **[i81]** pour l'Europe, du système GenBank **[i82]** pour les USA et du système DDBJ (DNA Database of Japan) **[i83]** pour le Japon. Ces trois systèmes sont partenaires et ont mis en place une démarche d'échange de l'information depuis 1990. Concernant les séquences protéiques, nous retiendrons également trois systèmes avec lesquelles il devra être possible de confronter les données de ProteomIs/GnpProt :

- GenPept est la traduction automatique en acides aminés de GenBank/EMBL/DDBJ **[i192]**.
- SwissProt **[a14] [i85]** est actuellement considérée comme la banque de référence. Elle est particulièrement riche en références croisées avec d'autres banques et en annotations (annotations, expertise, bibliographie).
- Uniprot **[i86]** a été crée pour regrouper les données de SwissProt et PIR (Protein Information Resource)

L'objectif aussi bien des bases de données nucléotidiques que des bases de données protéiques est de rendre publiques les séquences qui ont été archivées, ainsi un des premiers intérêts de ces banques est la masse de séquences qu'elles contiennent : en 2000, plus d'un milliard de bases étaient recensées dans la banque EMBL. et plus de 31 millions d'acides aminés dans celle de Swissprot.

➢ Les bases de données de séquences spécialisées

Nous trouvons ici les bases de données ayant pour objectif de réunir les séquences d'une même espèce et d'en enrichir les annotations pour diminuer ou lever les ambiguïtés laissées par les grandes banques publiques. Nous présentons ici les bases de données de séquences sur Arabidopsis dont la majeure partie des séquences de ProteomIs/GnpProt est issue. Parmi ces bases de données ont peu citer :

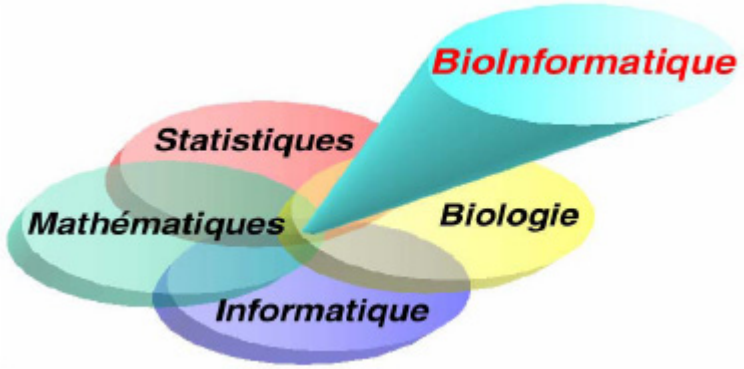
- TAIR **[i5] [a4]** : TAIR *(The Arabidopsis Information Resource)* donne accès à de nombreuses données destinées à la communauté des scientifiques travaillant sur Arabidopsis thaliana. Il s'agit d'une base de données relationnelle contenant une carte des 5 chromosomes d'Arabidopsis avec possibilité de faire des recherches sur les gènes ou les séquences de cet organisme. Les séquences annotées des gènes contenus dans TAIR proviennent des bases de donnée des centre TIGR et MIPS. Ces deux centres se sont mis d'accord avec les responsables de TAIR pour mettre en place la nomenclature AGI *(Arabidopsis Genome Initiative)* **[G7]** chargée d'identifier ces séquences.
- MatDB **[a5] [i11]**: La base de donnée MatDB *(MIPS Arabidopsis thaliana Database)* est un projet du centre d'analyse du projet européen sur A. thaliana ESSA *(European Scientists Sequencing Arabidopsis)* et AGI. Toutes les séquences d'Arabidopsis du projet AGI ont été intégrées dans MatDB et annotées notamment à l'aide d'outils d'analyse de séquences automatique comme PEDANT **[i15] [a7]**.

- TIGR **[i56]** : TIGR *(THE INSTITUTE FOR GENOMIC RESEARCH)* est l'homologue américain qui maintient de manière alternative une base de donnée génomique sur Arabidopsis thaliana. Ce centre est responsable d'une grande partie du séquençage et de l'annotation du génome d'Arabidopsis.
- ARAMEMNON **[a6] [i12]** : ARAMEMNON *(A specialized database (DB) for Arabidopsis membrane proteins, ARAMEMNON)* est une base de donnée spécialisée sur l'étude des segments transmembranaires dans les séquences protéiques d'Arabidopsis.

3.3 Définition de la bioinformatique

➢ Présentation générale

La bioinformatique a fait son apparition un peu avant les années 1980 avec les premières banques de biomolécules (EMBL et GenBank). Fondée sur les acquis de la biologie et de sciences formelles telles que les mathématiques, les statistiques et l'informatique, son objectif est de contribuer à une meilleure compréhension de l'information biologique.



Le terme bioinformatique français regroupe en fait deux disciplines anglo-saxones. La première se nomme "bioinformatics" et se consacre spécifiquement à l'étude des séquences et des structures. La deuxième est utilisée sous le terme générique de "biocomputing" et concerne le traitement sur ordinateur des données biologiques. Plusieurs revues internationales traitent de ces différentes disciplines et donc de la bioinformatique en général : Bioinformatics (Cabios ou Computer Applications in the Biosciences jusqu'en 1997) et Journal of Computational Biology. Enfin, la bioinformatique est un carrefour pluridisciplinaire qui a également stimulé la création de conférences comme JOBIM (Journées Ouvertes Biologie Informatique et Mathématiques), de formations (en particulier, des filières universitaires nouvelles comme les DESS en bioinformatique) et d'unités de bioinformatique dans le secteur académique et privé. Dans ces unités des chercheurs et ingénieurs bioinformaticiens ont développé des compétences pour accélérer l'approche "in silico" de la recherche en biologie : leur formation première peut être la biologie, les mathématiques ou l'informatique et ils ont suivi une formation complémentaire dans "l'autre domaine".

➢ Domaines d'intervention de la bioinformatique

L'un des volets de la bioinformatique consiste en la représentation, le stockage, et la distribution des données biologiques. Une conception adaptée des formats de données et des schémas de bases de données, le développement d'outils d'interrogation de ces bases de données, et d'interfaces permettant à l'utilisateur d'interroger les données via des requêtes complexes, comptent parmi les réalisations en bioinformatique.

Un deuxième volet est constituée par l'ensemble des concepts et des techniques nécessaires à l'interprétation de l'information génétique (séquences) et structurale (p.ex repliement 3-D des séquences). C'est le décryptage de la "bio-information" ("Computational Biology" en anglais).

Son but, comme tout volet théorique d'une discipline, est d'effectuer la synthèse des données disponibles (à l'aide de modèles et de théories), d'énoncer des hypothèses généralisatrices (ex. : comment les protéines se replient ou comment les espèces évoluent), et de formuler des prédictions (ex. : localiser un gène sur la séquence et en déduire sa fonction). Cet aspect prédictif de la bioinformatique contribue en partie à épargner le coût que nécessiterait une expérience « aveugle » en laboratoire pour analyser une séquence et déduire la fonction des gènes. La démarche expérimentale intervient alors uniquement pour tester et valider les hypothèses obtenues grâce aux prédictions bioinformatiques.

3.4 L'analyse protéomique

Le protéome désigne l'ensemble des protéines exprimées par le génome d'une cellule, d'un tissu ou d'un organe à un moment donné (voir partie **3.1**). L'analyse protéomique consiste à identifier à un instant t les protéines qui sont produites par un organisme. Ceci à partir de tel ou tel type cellulaire et telle ou telle condition qui va faire varier la nature des protéines produites. Un individu ne possède donc pas un seul protéome mais plusieurs. En résumé l'objectif de la protéomique est à partir d'un extrait de tissu organique, d'aboutir à l'identification des protéines qu'il contient.

L'approche protéomique est basée sur le couplage de plusieurs technologies de pointe :

1) L'**électrophorèse bidimensionnelle** (Gel 2D) (décrite **annexe 3** partie **I)** ou mono dimensionnelle (Gel 1D) **[G15]** permet de **séparer** plusieurs milliers de protéines d'un même échantillon. La technique de chromatographie en phase liquide (LC) **[G12]** peut également être utilisée comme technique de séparation des protéines. L'extrait protéique analysé par la LC peut dans ce cas provenir d'un échantillon mais aussi d'un spot prélevé à partir d'un Gel 1D.

2) La **spectrométrie de masse** permet ensuite d'identifier les protéines séparées dans les techniques précédentes ; ceci à partir du produit de leur fragmentation correspondant à des quantités infimes de l'ordre du picomolaire. La spectrométrie de masse permet en fait de disposer d'un spectre avec les masses expérimentales des peptides spécifique d'une protéine. Ces masses expérimentales vont être ensuite comparées à l'aide d'un logiciel comme Mascot **[i4]** avec les masses théoriques calculées des protéines contenues dans des bases de données spécifiques. C'est tout ce processus qui permettra l'identification des protéines. Le laboratoire de Montpellier utilise la spectrométrie de masse Maldi tof (**annexe 3** partie **IV**). La spectrométrie de masse MS/MS (**annexe 4**) peut également être utilisée et est généralement couplée en amont avec une chromatographie en phase liquide. La MS/MS est utilisée par exemple par la plate-forme protéomique du CEA de Grenoble. L'unité protéomique de Montpellier a elle aussi récemment investi dans cette technologie. La spectrométrie de masse MS/MS permet de faire une analyse plus fine qu'en Maldi tof des protéines étudiées en apportant en plus des informations concernant leur séquence. Par contre en comparaison la spectrométrie de masse Maldi tof est plus orientée vers une approche haut débit en terme d'analyse en permettant d'identifier plus rapidement un plus grand nombre de protéines.

Sur le **document 3** est présentée la démarche expérimentale couplant technique d'électrophorèse bidimensionnelle et spectrométrie de masse Maldi tof. L'utilisation de l'analyse protéomique est appelée à se généraliser, dans le cadre d'une complémentarité avec la génomique, tant en recherche fondamentale qu'en bio médecine ou en pharmacologie, où elle constitue désormais un outil puissant pour l'identification de marqueurs associés à une pathologie et de cibles thérapeutiques.

Document 3 : Démarche expérimentale en protéomique utilisant le couplage des techniques d'électrophorèse bidimensionnelle et de spectrométrie de Masse Maldi tof

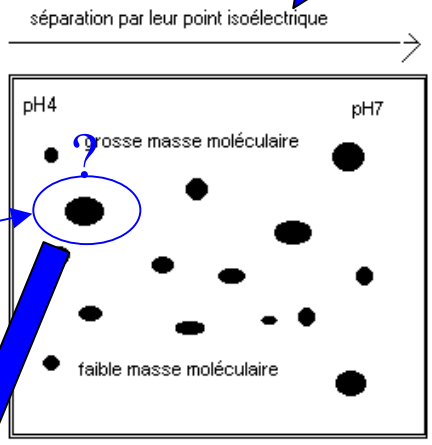
Objectif : Caractériser les protéines inconnues contenues dans un extrait organique

Echantillon végétal

Extraction des protéines

Extrait protéique

La première étape de la démarche correspond à la séparation des protéines contenues dans un extrait protéique à l'aide d'un gel d'électrophorèse bidimensionnel (gel 2D) (**annexe 5 étape I**).

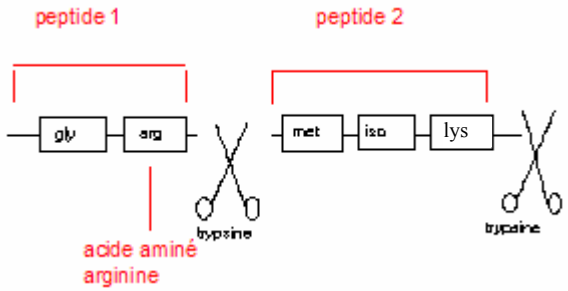


Gel 2D

1 tâche = 1 spot = 1 ou plusieurs protéines inconnues

Prélèvement

Digestion trypsique

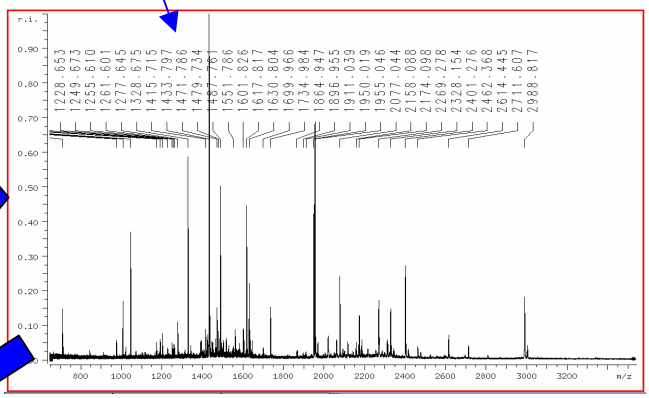


acide aminé arginine

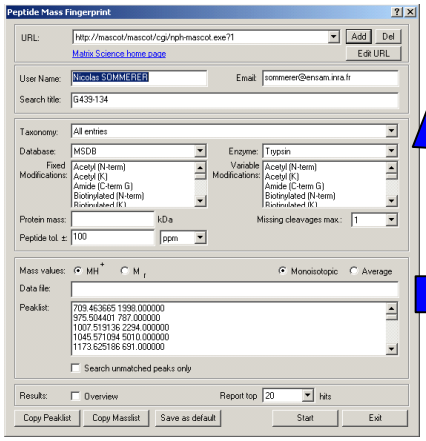
Les différentes protéines sont ensuite extraites du gel après prélèvement des spots (**annexe 5 étape II**) et digérées par une enzyme nommée trypsine (**annexe 5 étape III**). Cela va nous permettre d'obtenir une combinaison (empreinte) de peptides caractéristiques de la protéine.

Le but est ensuite de caractériser la protéine inconnue. Pour cela la solution est d'obtenir la masse des différents peptides à l'aide la technique de spectrométrie de masse (**annexe 5 étape IV**). Cet ensemble de masses peptidiques va alors constituer une **empreinte peptidique massique** caractéristique de la protéine analysée.

Liste de masses des peptides = empreinte peptidique massique = spectre de masse



Cette liste de masses expérimentales va alors être comparée à l'aide d'un logiciel à une banque de données de masses théoriques calculées pour de protéines déjà connues car référencées dans les banques (**annexe 5 étape V**). Le résultat de cette comparaison aboutit à l'identification des protéines analysées



Mascot Search Results

User : SOMMERER
Email : sommerer@ensam.inra.fr
Search title : 665
Database : MSDB 20020122)
Timestamp : 20 Jun 2002 at 12:42:23 GMT
Top Score : 142 for T04820, *aconitate hydratase*

Protéine caractérisée

Résultat d'identification MASCOT

4 Etat de l'art

Dans ce chapitre nous procédons à une présentation des solutions informatiques décrites dans la littérature et en rapport avec les objectifs du projet ProteomIs/GnpProt. Cet état de l'art sera par la suite un prérequis pour la discussion du choix de solutions dans le chapitre 5. La première partie sera consacrée à une présentation des outils existants en matière de gestion de données protéomiques. En deuxième partie nous aborderons les problématiques bioinformatiques. Enfin dans la dernière partie nous présenterons les solutions courantes utilisables pour garantir l'intégration de diverses sources de données biologiques.

4.1 Les outils de gestion des données protéomiques

Après avoir effectué une présentation générale des bases de données biologiques généralistes et spécialisées nous allons nous intéresser plus spécifiquement aux outils de gestion de données protéomique. La première partie concerne les cahiers de laboratoires informatisés (ou LIMS) nécessaires dans le cadre de plate-forme protéomique et la seconde partie décrit les bases de données spécialisées en protéomiques

4.2.1 Les LIMS

Pour faire face aux problématiques spécifiques des laboratoires et pour prendre en compte les information qui peuvent y être produites ou stockées, de nouveaux logiciels sont apparus, les LIMS (*Laboratory Information Management System*). Il s'agit de produits proposant un mini système d'information pour la gestion du laboratoire. Un certain nombre de ces outils existent pour la gestion des données protéomiques : LWS **[i18]** développé par *Amersham Biosciences* et *Cimarron Software*, SQL*LIMS appliquée à la plate-forme protéomique du Génopôle Toulouse Midi-Pyrénées **[i7]** …. Ils disposent de fonctions telles que la gestion des échantillons, du matériel, la tenue d'un journal des expériences… Ces logiciels sont aujourd'hui couramment proposés par les éditeurs ou constructeurs de matériel expérimental, mais la plupart du temps pour un coût important. En comparaison, ProteomIs/GnpProt n'a pas pour objectif de fournir toutes les fonctionnalités de gestion de données de laboratoire que fournit un logiciel de type LIMS, mais plutôt de faciliter l'exploration et l'analyse des données validées par les biologistes. Il peut donc être par-là considéré en ce sens comme un outil complémentaire aux LIMS et l'on peut envisager la possibilité qu'aurait ProteomIs/GnpProt à venir se « brancher » directement sur un LIMS pour récupérer l'ensemble des données valides d'un laboratoire.

4.2.2 Les bases de données orientées protéomiques

C'est dans cette catégorie d'outils que se place la base de donnée de ProteomIs/GnpProt. Il s'agit de décrire ici les principales bases de données permettant de faciliter la gestion et la diffusion des connaissances en protéomique au sein de la communauté scientifique. Certains de ces outils comme ProteomIs vont plus loin en permettant d'analyser ces données. De manière générale, il existe une multitude de base de données présentant les protéines identifiées sur un gel d'électrophorèse (technique présentée en **3.3**). Par contre, très peu de ces outils présentent les données de spectrométrie de masse.

➢ **PPMDB**

Avant le projet Génoplante, un projet nommé protéome vert **[i16]** fut une des premières initiatives pour rassembler les données protéomiques végétales de divers partenaires (dont l'UR1199 de Montpellier) dans un réceptacle commun. L'outil, conçu à cet effet, se nomme PPMDB **[a11]**.

PPMDB *(Plant Plasma Membrane Database)* est une base de donnée protéomique consacrée à l'étude des protéines membranaires de l'espèce végétale Arabidopsis et conçue pour être consultable via le web. Elle s'intéresse particulièrement aux modifications post-traductionnelles des protéines. L'organisation des données dans PPMDB est structurée en deux niveaux. Le premier niveau concerne la gestion des images des gels 2D et des liens sur les protocoles utilisés. Le deuxième niveau traite les données qualitatives et quantitatives sur les spots identifiées sur ces gels : point isoelectrique (pI), poids moléculaire (mw), nom de la protéine correspondante, séquence de cette protéine, localisation subcellulaire. PPMDB intègre l'utilisation du logiciel ANREP (Mehldau et Myers, 1993) qui permet de retrouver les spots en faisant une recherche à partir de la séquence des protéines correspondantes. L'application fournit également des liens hypertextes sur les autres bases de données (EMBL, GenBank, GenPep, and SWISS-PROT et TrEMBL). Ces liens sont disponibles à partir des fiches HTML descriptives sur les spots et référencent les protéines du spot dans les autres bases. Les fiches HTML descriptives des spots sont générées dynamiquement à partir de script Perl CGI qui vont interroger une base de donnée relationnelle Sybase pour récupérer les données et les mettre en forme. PPMDB fut une des premières bases de données à fournir une véritable interface interactive permettant de visualiser les informations sur les spots dans un gel 2D.

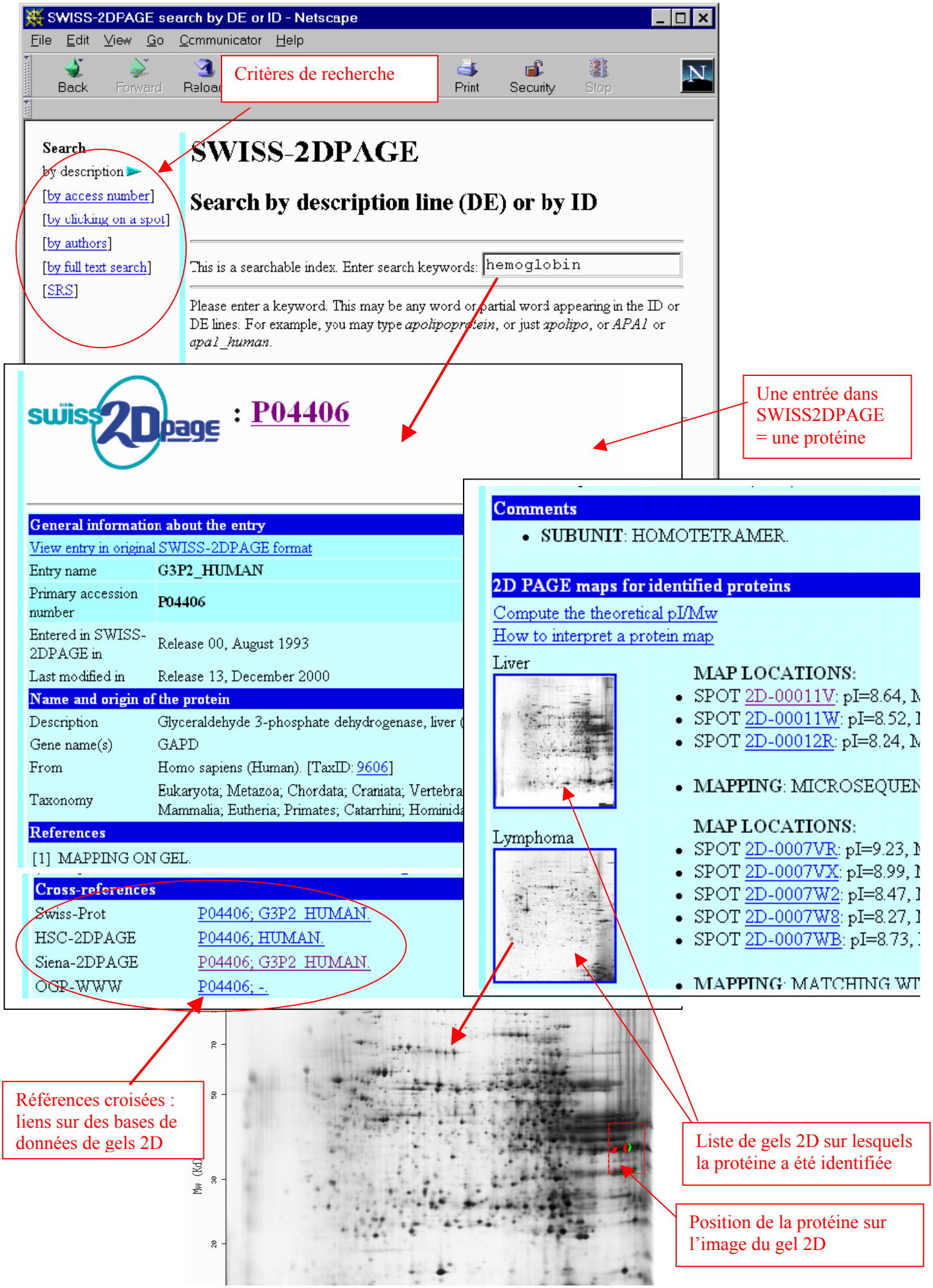
➢ Make2ddb

Une des références, en terme d'outil libre, est aujourd'hui la suite logicielle Make2ddb **[i8] [a2]** qui est mise à disposition de la communauté scientifique. Cet outil créé par l'Institut Suisse de Bioinformatique (SIB) représente un premier effort en matière d'organisation et de visualisation de fichiers d'images de gels 2D et des informations disponibles sur les spots. Il dispose d'un moteur de recherche permettant d'effectuer quelques requêtes pour obtenir l'identité d'un spot ou une liste de gels sur lesquels figure un spot spécifique. Ainsi l'utilisateur biologiste dispose d'une solution clé en main pour gérer ses résultats d'expériences d'électrophorèses. Cependant, ce système ne reposant pas sur un système de base de données de type SGBD mais sur des fichiers, l'outil s'en trouve limité. Il est difficile de stocker toutes les informations relatives à un spot telles que les conditions expérimentales, le type de numérisation des images de gels et les appariements entre les spots. Du fait de son architecture, il devient difficile d'assurer une bonne cohésion de ces données au vu de la quantité de données à stocker. Cependant la dernière version du package Make2ddb (Make2D-DB II package,version: 0.95) permet d'effectuer la migration des données textes vers le SGBD relationnel Postgres.

➢ SWISS-2DPAGE

Maintenue par l'Institut Suisse de Bioinformatique, la base de donnée SWISS-2DPAGE a été conçue grâce au système Make2ddb précédent. Il s'agit depuis 1993 d'une des ressources les plus importante en matière de base de données ressource en gel 2D et 1D **[i17] [a10]**. L'interrogation des données (**document 4**) s'effectue à travers une fonction de recherche permettant de disposer de critères d'interrogations variés tels que la recherche d'une protéine à partir d'un numéro d'accession; une description, un auteur. Une entrée dans SWISS-2D PAGE correspond à des données textuelles sur une protéine. Les informations sont structurées comme dans le format SWISSPROT, incluant les protocoles, des informations physiologiques et pathologiques, des données expérimentales et aussi des références bibliographiques. En plus de ces données textuelles, sur chaque entrée, SWISS-2DPAGE fournit les images des gels sur lesquels la protéine a été identifiée. Chacune de ces images miniatures fournit un lien hypertexte sur une image plus grande où est indiquée la position du (ou des) spots correspondant à la protéine en question. Une des caractéristiques les plus intéressante de cette base est de fournir des références de cette protéine dans un grand nombre de bases de données externes de gel 2D conçues grâce à l'outil Make2ddb.

Document 4 : La base de donnée SWISS2DPAGE



➢ **PROTICdb**

Le logiciel PROTICdb (PROTeome bioinformaTICS data base) **[i9]** permet de disposer d'une base de données et d'interfaces web permettant de contrôler, rechercher, questionner et éditer des données de protéomique. La base de données est conçue pour stocker l'ensemble des données issues d'études protéomiques, de la description de l'expérience à l'identification du spot ainsi que les variations quantitatives d'expression. L'originalité du système est d'inclure notamment la possibilité d'établir manuellement des relations entre les spots de plusieurs gels. L'application PROTICdb a été développée par l'équipe de Johann Joets (INRA/CNRS/UPXI/INA-PG) en partenariat avec le Centre de Bioinformatique de Bordeaux, et l'UMR BIOGECO 1202 de l'INRA. PROTIC peut fonctionner à la fois sous Oracle ou PostgreSQL et est disponible sur demande.

Les objectifs du projet PROTIC semblant suffisamment proches sur certains aspects du projet ProteomIs/GnpProt, nous avons discuté au démarrage du projet d'une possibilité de collaboration avec Johan Joets (responsable du projet PROTICdb). L'idée aurait été de partir sur la base de PROTICdb pour construire un seul outil qui soit capable de remplir à la fois les objectifs du projet ProteomIs/GnpProt et du projet PROTICdb. Cependant une divergence entre les deux outils se situait au niveau de la gestion des spots. Il n'était pas prévu dans ProteomIs de gérer des relations d'équivalence entre spots de la manière tel que le proposait PROTICdb. De plus le schéma relationnel de PROTICdb n'était pas conçu pour pouvoir être intégré au sein du système d'information de Génoplante. L'idée de collaboration fut donc abandonnée car le compromis était trop difficile à réaliser.

➢ **PARIS**

PARIS (Proteomic Analysis and Resources Indexation System) **[a30] [i10] [r4]** a été développé par Ju-hui Wang et Christophe Caron de l'unité BIA (Biométrie Intelligence Artificielle) de l'INRA de Jouy-en-Josas. Reposant sur une architecture 3-tiers, l'interface de l'application est entièrement en Java et téléchargeable sur un poste client grâce au protocole JavaWeb Start **[i103]** de Sun. Cet outil (utilisant un SGBD relationnel) gère essentiellement des données d'électrophorèse 2D. La particularité du système est de permettre d'effectuer des comparaisons entre les images des gels provenant de plusieurs projets, de poursuivre l'analyse de ces images et d'émettre des hypothèses quant à l'information biologique qu'elles contiennent (p.ex relation entre l'ensemble des protéines présentes sur un gel et leur appartenance à une voie métabolique **[G16]**). PARIS n'était pas encore disponible lorsque le projet ProteomIs/GnpProt a démarré. Cependant, nous avons eu par la suite l'occasion de recevoir Ju-hui Wang dans notre unité qui nous a fait une démonstration de son outil. Le module relatif à l'interprétation biologique des gels 2D pourrait à terme être adapté à notre outil.

4.2 Etat de l'art pour l'analyse des données protéomiques

Les traitements réalisés dans ProteomIs portent d'une part sur un contrôle qualité sur les données de séquences et d'autre part sur la caractérisation fonctionnelle des séquences. Dans cette partie nous traiterons seulement du deuxième aspect.

4.2.1 Les outils de comparaison de séquences

➢ **Intérêt de la comparaison de séquences**

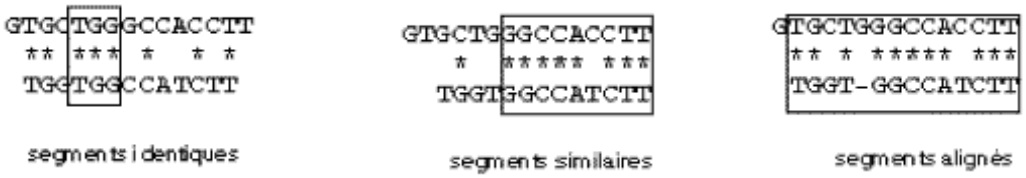
La comparaison de séquences est très largement utilisée en biologie au travers de logiciels spécifiques. L'objectif recherché dans notre projet est de pouvoir trouver une similarité entre une séquence requête de nature inconnue et toutes les séquences de la base de donnée ProteomIs.

Comme les séquences de protéines contenues dans la base ont des fonctions connues, le biologiste pourra ainsi déterminer la fonction de la protéine qu'il a voulu comparer. Ainsi l'information qui se trouve dans le texte des gènes ou des protéines devient accessible grâce à l'outil informatique.

➢ Les principes de base pour identifier la ressemblance entre deux séquences

- La recherche de segments et l'alignement :

Les programmes de comparaison de séquences ont pour but de repérer les endroits où se trouvent des régions identiques ou très proches entre deux séquences et d'en déduire celles qui sont significatives et qui correspondent à un sens biologique de celles qui sont observées par hasard. En général, les algorithmes fonctionnent sur des segments de séquences (on parle de fenêtres, de motifs ou de mots) sur lesquels on regarde s'il existe ou pas une similitude significative. On distingue pour cette catégorie deux classes précises de similitude : la ressemblance parfaite ou identité et la ressemblance non parfaite que l'on qualifie de similitude. La notion d'alignement, elle, suppose la recherche des positions auxquelles il est possible de faire des insertions (ajout d'une base) ou des délétions (suppression d'une base) afin d'optimiser le score d'une comparaison.



La plupart des programmes de comparaisons de séquences s'appuient sur une de ces trois notions (la recherche de segments identiques, de segments similaires ou d'alignements) pour faire ressortir des ressemblances entre séquences.

- Les différentes types d'alignements :

- Les **alignements globaux** : Un alignement global considère l'ensemble des éléments de chacune des séquences. Si les longueurs des séquences sont différentes, alors des insertions devront être faites dans la séquence la plus petite pour arriver à aligner les deux séquences d'une extrémité à l'autre. Cependant dans un alignement global, si uniquement de courts segments sont très similaires entre deux séquences, les autres parties des séquences risquent de diminuer le poids de ces régions. C'est pourquoi d'autres algorithmes d'alignements, dits locaux, basés sur la localisation des similarités sont nés.
- Les **alignements locaux** : Le but de ces alignements locaux est de trouver sans prédétermination de longueur les zones les plus similaires entre deux séquences. L'alignement local comporte donc une partie de chacune des séquences et non la totalité des séquences comme dans la plupart des alignement globaux.
- Les **alignements multiples** : Pour caractériser les régions conservées dans les séquences, il est souvent plus efficace d'utiliser plusieurs séquences et d'effectuer un alignement multiple.

➢ Programmes pour comparer une séquence contre un banque :

Dans cette section, nous présenterons BLAST et FASTA, deux programmes habituellement utilisés pour la comparaison d'une séquence contre une banque de données de séquences.

- Recherche d'alignements locaux avec BLAST :

L'outil de loin le plus populaire pour la comparaison de séquence avec une banque de données de séquences est le programme BLAST **[a11]** (BASIC Local Alignment Search Tool).

Il constitue le cœur de la plupart des serveurs de soumission de séquences. Ce logiciel possède en fait quatre programmes distincts de comparaison avec les bases de données : BLASTN (séquence nucléique contre base nucléique), BLASTP (séquence protéique contre base protéique), BLASTX (séquence nucléique traduite en séquence protéique contre base protéique), et TBLASTN (séquence protéique contre base nucléique traduite en séquence protéique). Le logiciel BLAST effectue des comparaisons de séquences en essayant de trouver les alignements optimaux locaux de meilleurs scores entre la séquence requête et la banque (Altschul et al. 1990). L'idée sous-jacente à l'algorithme est que les bons alignements doivent contenir quelque part des petits segments strictement identiques ou de score très important. Ces éléments sont des graines où l'alignement est ancré et à partir desquelles il s'étend tant que le score reste supérieur à un seuil donné (Altschul et al. 1990).

Il est possible d'utiliser le logiciel BLAST directement sur Internet à travers une interface (par exemple sur le site du NCBI **[i19]** pour faire des comparaisons sur la base Genbank) ou de l'installer localement pour effectuer des comparaisons sur notre propre base de données de séquences. Parmi les paramètres utilisés pour analyser les résultats du BLAST la E-values est le plus utilisée. La E-values donne des informations sur la significativité d'un alignement donné. Une E-value élevée (5 ou 10) indique que l'alignement est probablement dû au hasard et que la séquence requête a été alignée avec une séquence qui ne lui est pas apparentée dans la banque de données. Des E-values de 0,1 ou 0,05 sont utilisées typiquement comme valeurs de seuil lors des recherches dans des banques de données de séquences. Le recours à une E-value élevée lors d'une recherche dans une banque de données permet de trouver des correspondances plus distantes mais augmente le taux d'alignements équivoques.

- Recherche d'alignements locaux avec FASTA :

Un autre programme à base d'heuristique permettant l'alignement d'une séquence requête avec des séquences d'une banque de données, est le logiciel FASTA. FASTA a été développé avant BLAST et est encore activement maintenu par Pearson (University of Virginia). L'algorithme est basé sur l'identification rapide des zones d'identité entre la séquence recherchée et les séquences de la banque **[i20]**. Cette reconnaissance est primordiale car elle permet de considérer uniquement les séquences présentant une région de forte similitude avec la séquence recherchée. On peut ensuite, à partir de la meilleure zone de ressemblance, appliquer localement à ces séquences un algorithme d'alignement optimal. Le logiciel regroupe en fait deux programmes de recherche avec les banques de données. Le premier est le programme FASTA qui possède une version nucléique et protéique et le deuxième est le programme TFASTA qui recherche une séquence protéique avec les séquences d'une base nucléique traduite dans les 6 phases.

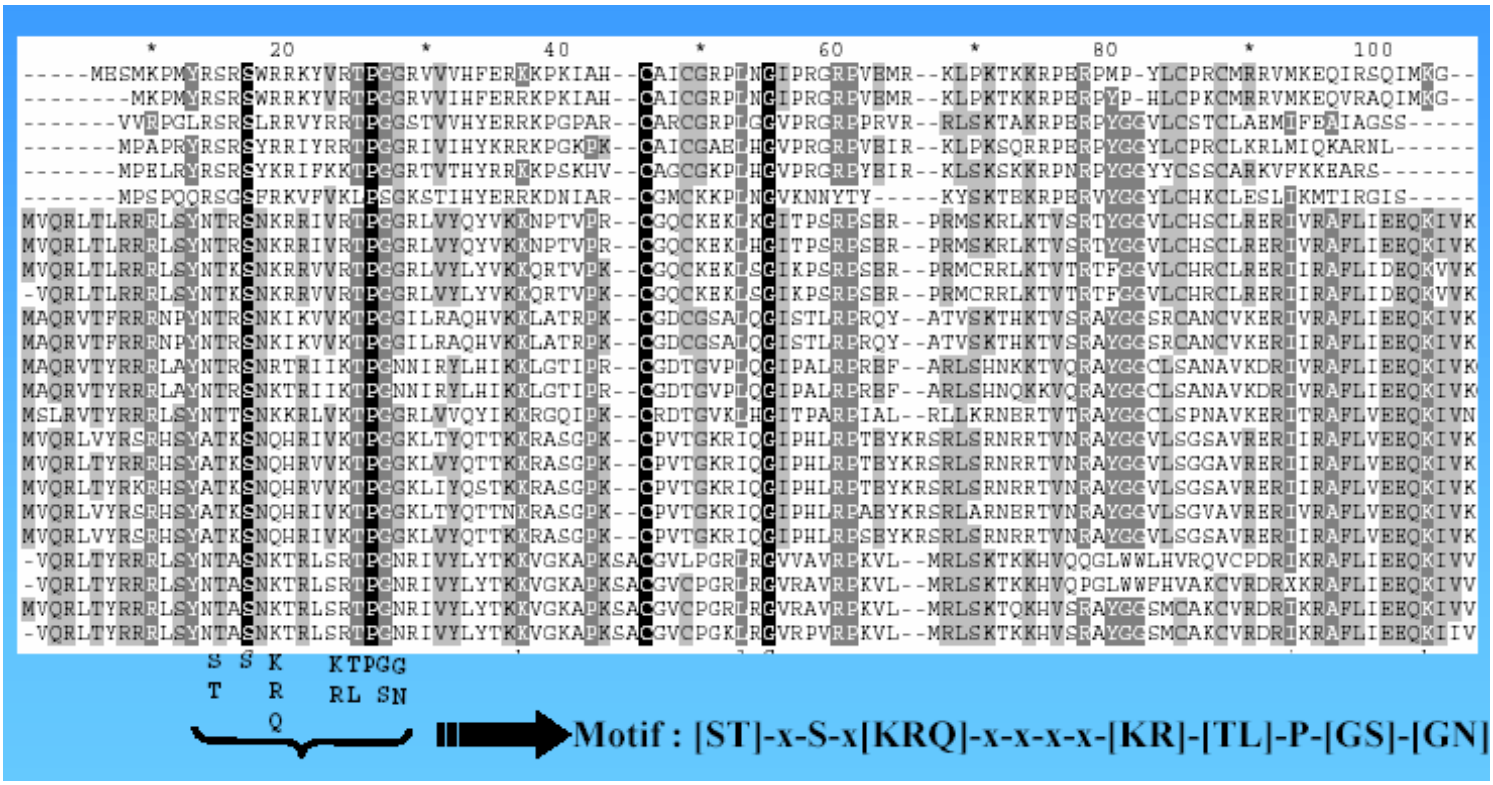
➢ Comparaison des logiciels BLAST et FASTA

Chacun des deux logiciels a des défauts et des qualités. En ce qui concerne BLAST, sa qualité première est la rapidité : une recherche classique effectuée avec BLAST sur la totalité des banques de données protéiniques ou d'ADN ne prend que quelques minutes. Cette caractéristique fait de BLAST le logiciel le plus utilisé dans l'ensemble de la communauté (10000 connexions / jour sur le serveur du NCBI). De plus, lorsqu'on fait une recherche avec les paramètres par défaut de BLAST et de FASTA, généralement BLAST a une plus grande sensibilité dans la mesure où il ne nécessite pas de similitude parfaite à la première étape de l'algorithme. Ensuite BLAST a accès à plus de bases de données intéressantes. D'un autre côté, l'inconvénient de BLAST est qu'il favorise la vitesse par rapport à la sensibilité. En effet pour la recherche sur des chaînes d'ADN, FASTA est alors la référence bien qu'il effectue les recherches en quelques heures contre 2 à 3 minutes pour Blast. BLAST reste le logiciel à utiliser pour une première recherche rapide, faisant apparaître les alignements possibles. Cet objectif étant celui recherché par ProteomIs, BLAST reste le logiciel le plus adapté.

4.2.2 Les outils pour la prédiction, recherche de motifs

➢ **Prédiction, recherche de motifs dans les séquences protéiques**

Toujours dans le cadre des fonctionnalités d'analyse de séquences de protéines à développer dans ProteomIs un autre volet est la recherche de motifs spécifiques dans ces séquences **[i96]**. Cette démarche faisant à nouveau intervenir l'outil informatique est également appelée prédiction de motif. Il convient dès lors de donner la définition de ce qu'est un motif dans une séquence protéique. Un motif est un élément structural que l'on retrouve dans tous les membres d'une famille de protéines. Il contient des acides aminés essentiels à une fonction conservée. Ces acides aminés ne sont pas nécessairement consécutifs (ils peuvent même être très éloignés dans la séquence), mais on s'attend à les trouver assez proches dans la structure 3D, car ils participent à la même fonction (par exemple formation d'un site actif). Afin de détecter des motifs dans une séquence, une des solutions est d'abord de réaliser un alignement multiples des séquences d'une même famille protéique (**document 5**).



Document 5 : Représentation des motifs

Pour réaliser cet alignement multiple il faut s'aider d'un logiciel comme ClustalW **[i40]**. Cet alignement va permettre de mettre en évidence des portions de séquences identiques entre les différents membres de la famille protéiques. C'est cette région conservée de la séquence que l'on va appeler motif.

Cependant il existe quelques écarts au niveau de la conservation des acides aminés entre les séquences de cet alignement. Pour cette raison un motif doit être décrit sous la forme d'une expression régulière **[G22]** (appelée aussi pattern) dont le formalisme est le suivant :

- [ST] signifie que l'acide aminé peut être soit S soit T à cette position du motif
- **x** : un acide aminé quelconque

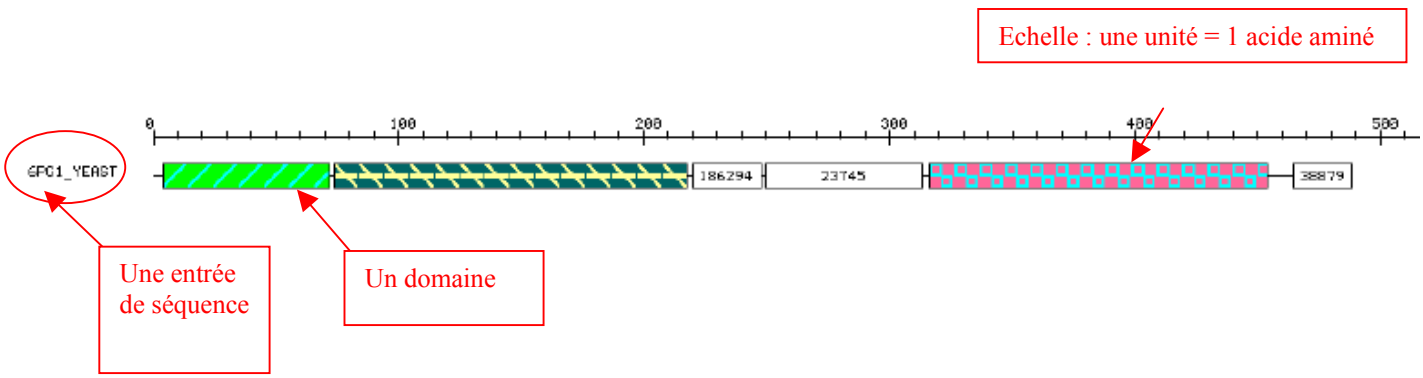
Ce formalisme est celui utilisé par la banque de motifs PROSITE (Bairoch, 1993) **[i21]**. PROSITE est une banque regroupant des motifs structuraux ou fonctionnels de protéines. Chacun des motifs identifié est associé si possible à une fonction biologique par les annotateurs de PROSITE.

Dans cette banque chaque entrée correspond à un motif spécifique d'une protéine. Dès lors un motif peut constituer une sonde efficace contre une banque de séquence protéique comme SwissProt pour retrouver toutes les protéines associées à la fonction associée au motif. Dans ce cas il faut utiliser le logiciel ScanProsite **[i22]**.

Tous ces outils (PROSITE, ScanProsite) font partie du site de l'ExPASy (Expert Protein Analysis System **[i23]**, maintenue par l'Institut Suisse de Bioinformatique. Ce site, reste le plus populaire et le plus complet dédié à l'analyse protéomique. En plus des liens sur la recherche de motif on peut par exemple y trouver également des outils relatif à l'analyse et la visualisation de structure 3D de protéines. Parmi les banques de motifs il faut également citer PRINTS (Protein Motif Fingerprint Database) **[i24]**. PRINTS est une base de données similaire à PROSITE, à l'exception qu'elle utilise des empreintes constituées de plusieurs motifs ou pattern (en anglais fingerprints) pour caractériser une séquence protéique entière

➢ Prédiction, recherche de domaines dans les séquences protéiques

Quelque fois les motifs dans une séquence peuvent être associée à d'autres motifs pour former un assemblage stable que l'on appelle « domaine ». A un domaine correspond généralement une structure tridimensionnelle spécifique. Cependant il n'existe pas forcément pour tous les domaines une relation univoque avec une fonction. Un domaine peut ne pas avoir de fonctions, et une fonction peut-être effectuée par plusieurs domaines. La base de donnée ProDom **[a12] [i77]** a pour but de collecter les familles de domaines homologues sur la base d'une analyse automatique des séquences de protéines de SWISS-PROT/TrEMBL. Il faut savoir que dans ProDom un numéro de domaine correspond à une entrée qui correspond elle-même à une famille de domaines homologues. Le serveur Web consacré à ProDom permet d'accéder à une représentation graphique de l'arrangement des domaines protéiques. Chaque protéine est représentée par une succession de boîtes utilisant un code conjuguant motifs et couleurs différents permettant d'identifier les domaines. Le **document 6** ci-dessous nous permet de mieux comprendre la signification de ces représentations graphiques. En fait sur ces représentations par rapport à l'échelle, une unité est égale à un acide aminé. D'après l'échelle, on peut alors déduire par exemple, que le premier domaine coloré en vert débute environ sur le 1er acide aminé et se termine sur le 70ème acide aminé.



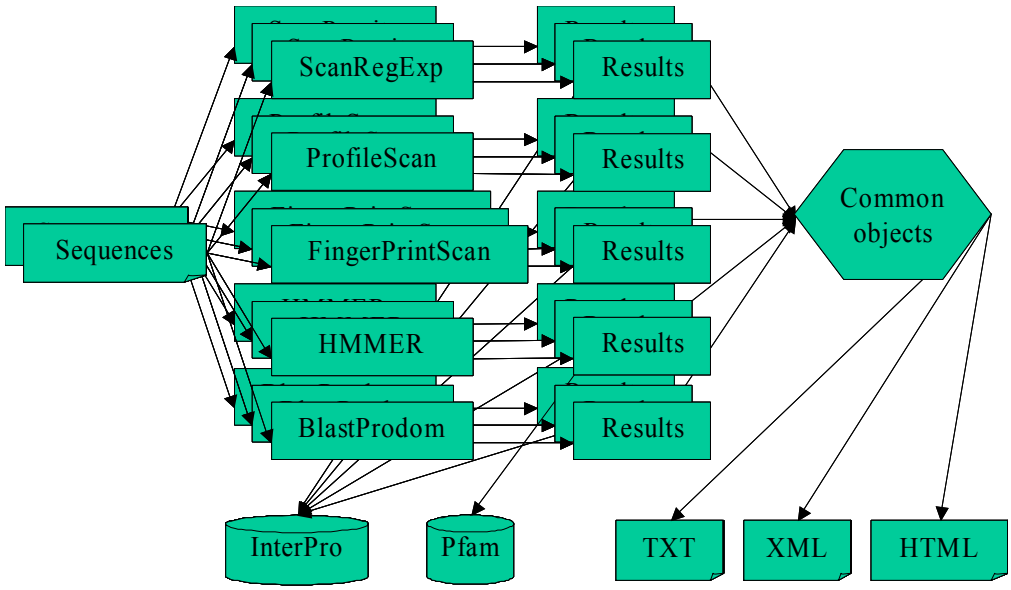
Document 6 : Représentation graphique des domaines protéiques dans ProDom

Parmi les bases de données de domaines on peut citer également Pfam **[i26]**. Pfam9 est construite à partir d'une version de SWISS-PROT/TrEMBL dans laquelle toutes les redondances ont été éliminées. Cette banque comprend en fait deux parties : Pfam-A et Pfam-B. La différence entre ces deux divisions tient au fait que les domaines de Pfam-A sont expertisés par des curateurs tandis que ceux faisant partie de Pfam-B sont générés par un programme partant des domaines ProDom.

➢ La base de donnée Interpro

Ainsi après avoir constaté que les bases de ressources de motifs et domaines sont nombreuses, on peut se demander quelles bases de motifs ou de profils utiliser pour analyser une nouvelle séquence ? On peut cependant mentionner l'initiative européenne autour du projet Interpro **[i25]** dont l'objectif est d'offrir un service qui intègre la recherche de motifs dans plusieurs bases en même temps telles que Pfam, PRINTS, ProDom et PROSITE (en 2001). Aujourd'hui il faut rajouter les ressources d'autres bases de motifs et domaines telles que SMART, TIGRFAMs et bientôt BLOCKS. Interpro est donc une ressource très complète de familles protéiques, domaines et sites fonctionnels. Régulièrement, les fichiers plats correspondant aux données des différentes bases sont récupérés par ftp. Puis ces données hétérogènes sont fusionnées et intégrées au sein d'un shéma relationnel qui est implémenté dans un SGBD Oracle. Une entrée Interpro rassemble alors les domaines et motifs en recouvrement en provenance des différentes sources. La première application de Interpro est l'annotation et la classification fonctionnelle de séquences non caractérisées. L'EBI a utilisé Interpro pour augmenter l'annotation automatique de TrEMBL.

Le site d'Interpro intègre également l'outil InterproScan **[i27]** qui sera à terme intégré dans ProteomIs. InterproScan est un outil qui combine différentes méthodes de reconnaissances de signatures protéique (motifs ou domaines) dans un seul package développé en langage Perl (**document 7**).



Document 7 : InterProScan (source Interpro documentation : **[i74]**)

Ces différents outils sont :

- ScanRegExp : utilisé pour la reconnaissance de pattern Prosite qui se limitent à des expressions régulières
- ProfileScan : utilisé pour la reconnaissance de profils Prosite (un profil est une représentation matricielle d'un motif). On recherche des profils plutôt que des patterns dans le cas où les séquences sont extrêmement divergentes.
- FingerPrintScan : utilisé pour la reconnaissance d'empreintes PRINTS.
- HMMER : ensemble de programmes permettant la construction et l'utilisation de profils HMM (Hidden Markov Models) pour la recherche de motifs dans les banques de données (utilisé par PFAM, SMART et TIGRFAMs)

- BlastProDom : processus automatique basé sur une utilisation récursive du programme de recherche d'homologie PSI-BLAST pour constituer la base de donnée ProDom.

Cet outil est téléchargeable à l'adresse : ftp://ftp.ebi.ac.uk/pub/databases/interpro/iprscan/ et fonctionne sous l'environnement UNIX. On peut donc l'utiliser sur nos propre jeux de séquences protéiques. Les sorties de cet outils peuvent être obtenu soit au format txt, HTML ou XML. Ces sorties sont composées d'une liste de liens InterPro pour chacune des séquences et permettent d'aider dans le processus de détermination de leur fonction et leur appartenance à des familles.

➢ Prédiction, recherche de motifs spécifiques : les sites de phosphorylations

L'unité de recherche en protéomique de Montpellier s'intéresse tout particulièrement à la recherche de motifs spécifiques associés à une modification physico-chimique que l'on appelle phosphorylation.

La phosphorylation est une modification que l'on qualifie de post-traductionnelle (définit partie **3.1**). Elle correspond à l'addition d'une molécule de phosphore sur un des 3 acides aminés suivants : sérine (S), thréonine (T) et tyrosine (Y). Cependant même si les phosphorylations s'effectuent sur un de ces 3 acides aminés, l'importance des acides aminés voisins est aussi déterminante pour cette phosphorylation. Pour cette raison on peut parler de motif pour un site de phosphorylation. La phosphorylation d'une protéine permet de réguler son comportement.

Il existe un grand nombre de bases de données spécialisées contenant des sites de phosphorylation prédits et annotés par des biologistes. Parmi elles, on peut citer Phospho.ELM **[i28]** qui contient une collection de sites de phosphorylation vérifiés expérimentalement. La base de donnée SWISSPROT, bien que non spécialisée dans ce domaine est également très riche en motifs annotés de ce type. Il suffit de faire une recherche à l'aide du mot clé Phosphorylation pour s'en apercevoir. En effet dans SWISSPROT chacune des entrées décrivant une protéine possède un champs **FT** (**F**eature **T**able) qui contient des informations et des annotations sur les motifs caractérisant la séquence. Une des originalités de cette base est la possibilité qu'elle offre de visualiser les motifs présents dans le champs FT à l'aide d'une applet Java (**document 8**) **[i29]**. En ce qui concerne les motifs de phosphorylation détectés informatiquement (prédits) mais non vérifiés, on peut se référer à la banque de données PROSITE **[i68]**.

<u>Document 8 :</u> L'applet de visualisation de motifs de SWISSPROT *(accession = P22612)*

Features

Feature table viewer

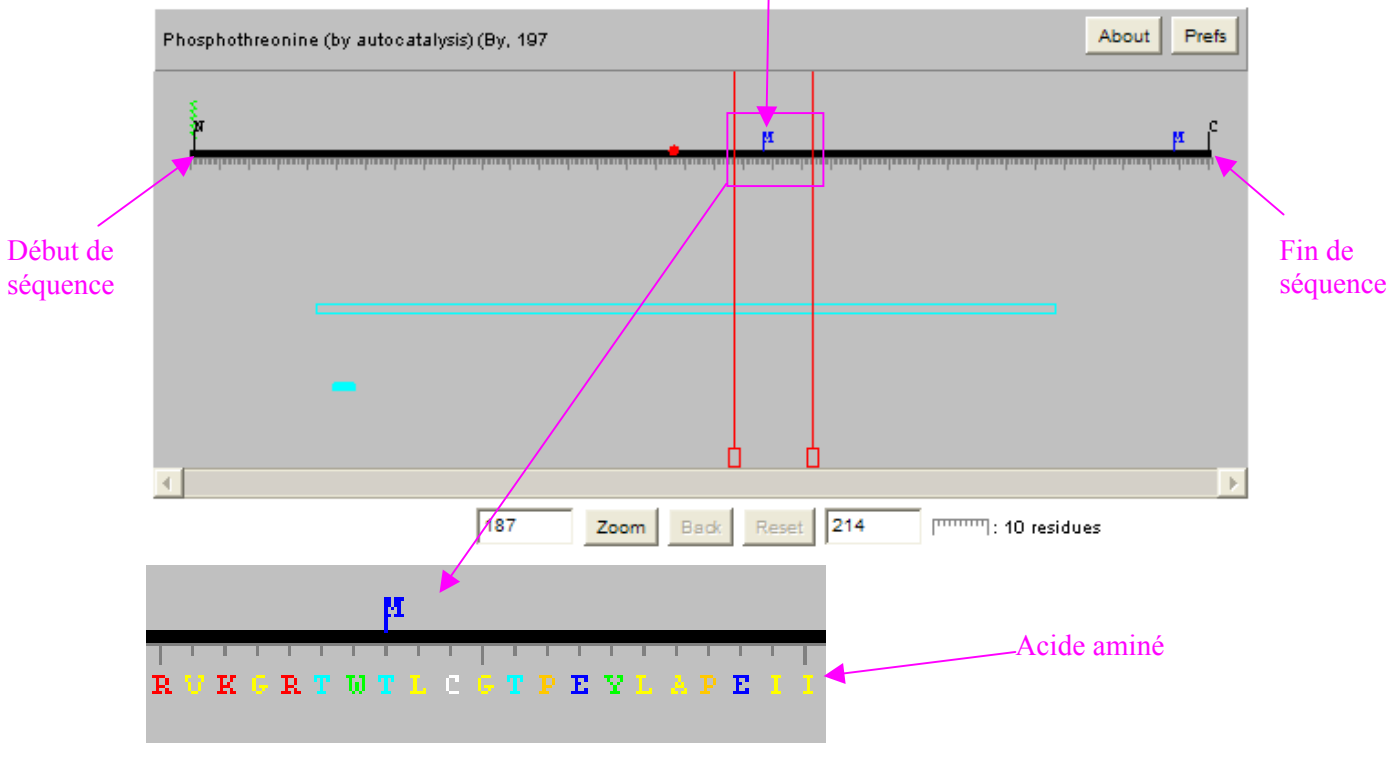
Key	From	To	Length	Description
INIT_MET	0	0		
LIPID	1	1		N-myristoyl glycine *(By similarity)*.
DOMAIN	43	297	255	Protein kinase.
NP_BIND	49	57	9	ATP *(By similarity)*.
BINDING	72	72		ATP *(By similarity)*.
ACT_SITE	166	166		Proton acceptor *(By similarity)*.
MOD_RES	197	197		Phosphothreonine (by autocatalysis) *(B*
MOD_RES	338	338		Phosphoserine (by autocatalysis) *(By s*
CONFLICT	267	267		H -> D (in Ref. 1).
CONFLICT	344	344		A -> P (in Ref. 3).

Sequence information

Length: **350 AA** | Molecular weight: **40303 Da** | CRC64:

```
        10         20         30         40         50         60
GNAPAKKDTE QEESVNEFLA KARGDFLYRW GNPAQNTASS DQFERLRTLG MGSFGRVMLV

        70         80         90        100        110        120
RHQETGGHYA MKILNKQKVV KMKQVEHILN EKRILQAIDF PFLVKLQFSF KDNSYLYLVM

       130        140        150        160        170        180
```

Site de phosphorylation



Début de séquence

Fin de séquence

Acide aminé

Zoom sur la séquence

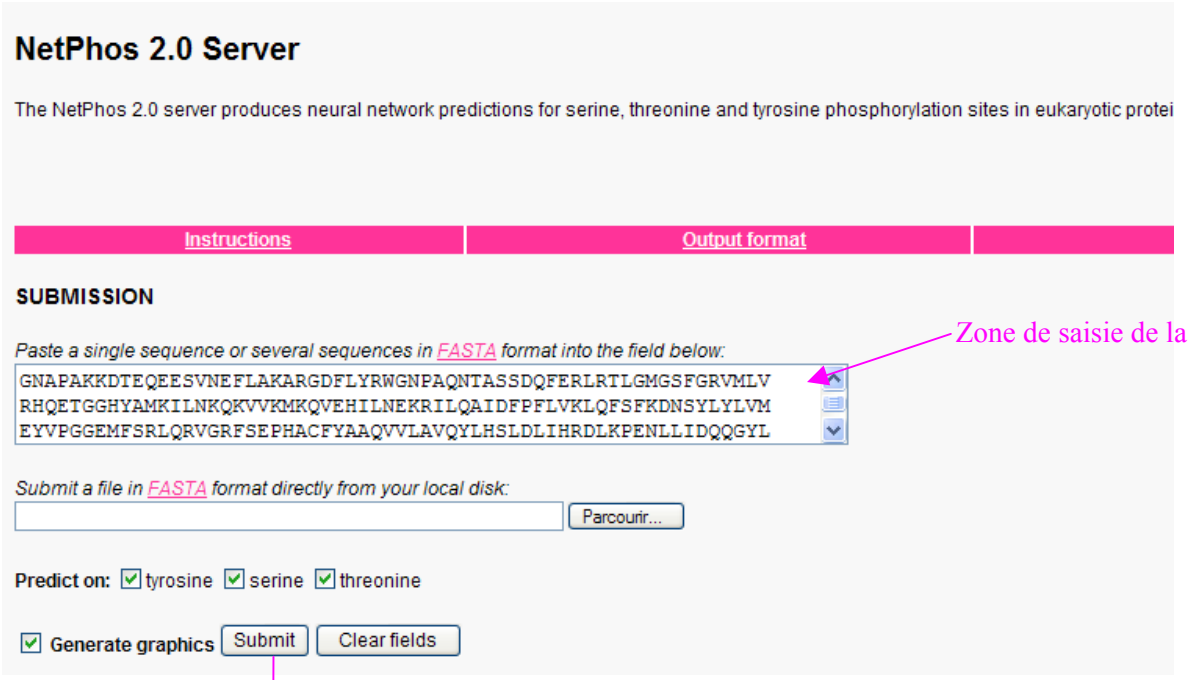
Intéressons nous maintenant aux outils permettant aux utilisateurs biologistes d'analyser une séquence afin de rechercher leurs propres sites de phosphorylation. Un nombre raisonnable de ces outils est présent sur le site d'Expasy à l'adresse suivante : http://www.expasy.org/tools/#ptm. Le site du cbs (http://www.cbs.dtu.dk/services) propose également de nombreux outils pour la prédiction de modifications post-traductionnelles. Parmi ces outils un des plus utilisé est NetPhos WWW server **[i30]**. Cette application à base de réseau de neurones multi-couches, permet de détecter (prédire) les sites potentiel de phosphorylation Sérine, Thréonine, et Tyrosine chez les protéines eucaryotes **[G10]**. Il effectue ces prédictions à partir de l'environnement en acides aminés des acides aminés Sérine, Threonine et Tyrosine. A partir de là il évalue la probabilité avec laquelle des modifications post-traductionnelles (PTM) pourraient se produire sur chacun de ces trois acides aminés. Ce sont donc des sites potentiels de phosphorylation qui sont mis en évidence et le résultat est alors basé sur une probabilité qui varie de 69% à 96% **[a13]**.

Pour utiliser ce logiciel il suffit d'aller sur le site http://www.cbs.dtu.dk/services/NetPhos/
On aboutie alors sur un formulaire HTML (**document 9**) permettant de saisir la séquence dans une zone de texte. Trois cases à cocher permettent de sélectionner les acides aminés sur lesquels on souhaite effectuer la prédiction. Après avoir appuyé sur le bouton Submit on obtient le résultat sur le **document 10** au format HTML. La séquence est reproduite et numérotée. En dessous, la séquence est reproduite en remplaçant tous les acides aminés non phosphorylés par des points. De cette façon, on connaît l'emplacement des sites de phosphorylation de la protéine. Dans l'exemple présenté nous avons analysé avec NetPhos une séquence de la protéine P22612 provenant de SWISSPROT. Si l'on se réfère maintenant au **document 8** précédent ; celui-ci montre que dans SWISSPROT le 197ème acide aminé correspondant à la Thréonine est phosphorylé. Ce résultat avait l'avantage d'être vérifié expérimentalement par des biologistes puisque présent dans SWISSPROT. Et bien nous pouvons constater que cet acide aminé (entouré en rose dans le **document 10**) a également été prédit comme phosphorylé dans NetPhos (la probabilité que ce résultat soit vrai étant de 96%). Cependant si on reprend le **document 8** on peut constater que dans SWISSPROT seul deux acides aminés ont été enregistrés comme phosphorylés ; alors que dans NetPhos 15 acides aminés sont considérés comme potentiellement phosphorylables ! Cela ne signifie pas que la majeure partie des résultats de NetPhos sont faux mais plutôt qu'un nombre restreint de sites de phosphorylation est découvert expérimentalement. Reste bien sûr à valider expérimentalement les prédictions de NetPhos.

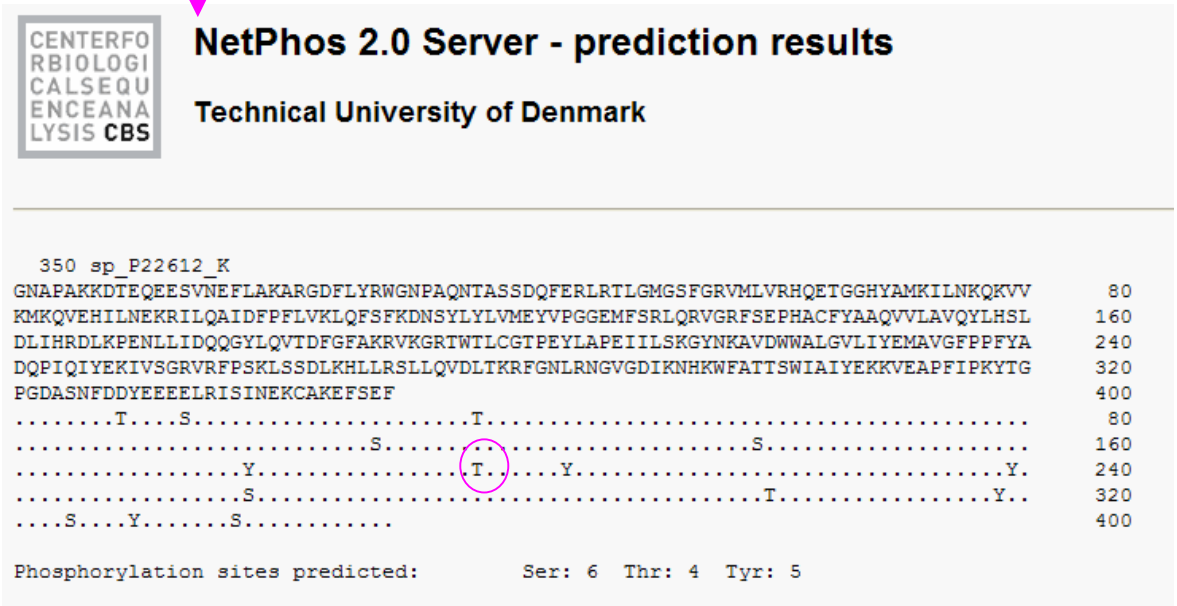
L'avantage de ce genre de logiciel est qu'il va « guider » le biologiste dans ses recherches.
Ainsi il faut considérer de manière générale que les résultats de prédiction sur des motifs à l'aide de l'outil bioinformatique sont complémentaires par rapport aux résultats expérimentaux. C'est le but de l'unité de recherche de Montpellier que d'associer résultats expérimentaux et prédiction bioinformatique sur les séquences protéiques de ProteomIs. Nous verrons dans la partie **5.2.3** de quelle manière cette association pourra être effectué toujours en utilisant l'outil informatique afin de faciliter le travail du biologiste.

Avant d'aborder ce sujet nous terminerons cette partie état de l'art en effectuant une présentation des solutions permettant d'assurer l'interopérabilité entre diverses sources de données biologiques. Ceci nous apportera les éléments de compréhensions nécessaires à la discussion (partie **5.3.1**) concernant la solution choisie pour assurer l'interopérabilité du module GnpProt avec les autres modules du système GpiIS mais aussi avec les autres bases de données protéiques disponibles sur le Web.

<u>Document 9 :</u> Fonctionnement du NetPhos 2.0 Server



<u>Document 10 :</u> Résultat d'analyse de NetPhos



4.3 Etat de l'art des solutions d'intégrations des données biologiques

L'intégration a pour objectif d'assurer à un utilisateur un accès à des sources multiples, réparties et hétérogènes, à travers une interface unique (accès transparent). Ceci est aujourd'hui nécessaire au biologiste qui a à sa disposition un nombre considérable et toujours croissant de ressources bioinformatiques que cela soit du traitement ou du stockage. Les approches génomiques que l'on pourrait qualifier d'intégratives nécessitent l'accès à de l'information distribuée dans plusieurs sources de données le plus souvent hétérogènes mais accessibles cependant pour la plupart via le web. En outre les questions biologiquement pertinentes portent pour certaines d'entre elles sur plusieurs sources de données à la fois. Idéalement il faudrait pouvoir poser une requête complexe que le système interrogé serait alors capable de décomposer en sous-requêtes adressables ensuite à chacune des sources de données concernées. La réponse globale serait alors reconstruite par composition des réponses partielles restituées par ces mêmes sources de données. Les problèmes essentiels qui se posent dans ce contexte, portent sur l'hétérogénéité des sources de données, et s'inscrivent dans la problématique actuelle du web sémantique.

Dans cette partie état de l'art nous donnerons en **4.3.1** les différents niveaux d'hétérogénéité à prendre en compte pour une bonne intégration des sources. Dans la section **4.3.2** nous évoquerons la nécessité de concevoir des ontologies et nous présenterons en **4.3.3** les solutions actuelles en termes de systèmes d'intégration. Les avantages et inconvénients de chaque solution seront ensuite discutés en **4.3.4**.

4.3.1 Problèmes lié à l'hétérogénéité des sources de données

Différents niveaux d'hétérogénéité peuvent être observés dans les données génomiques :

- l'hétérogénéité syntaxique : elle se manifeste au niveau des formats pour décrire le contenu de sources (CSV pour Entrez, ASN1 pour GenBank) et également par une diversité des modèles de données (relationnel pour Swiss-Prot, objet pour GUS **[a15]**)
- l'hétérogénéité sémantique : elle recouvre plusieurs aspects. Tout d'abord, chaque base se focalise sur un type d'objet biologique (p.ex., le focus de Swiss-Prot est la protéine, celui de GenBank le gène) et chacun de ces concepts clés peut-être représenté différemment en fonction des sources (GenBank représente un gène comme une annotation sur une séquence tandis que MGD **[i56]** représente un gène comme un locus qui confère un phénotype"). Ensuite, selon les bases, une même information n'est pas représentée avec le même niveau de détail : certaines bases on l'a vu (partie **3.2**) sont généralistes tandis que d'autres sont plus spécialisées. Le dernier aspect de l'hétérogénéité sémantique est relatif à la diversité du vocabulaires utilisé pour annoter les séquences. Par ailleurs, il existe pour une même entité (protéine, gène) plusieurs noms, et ce, à l'intérieur d'une même banque.
- l'hétérogénéité des outils informatiques : il s'agit de l'hétérogénéité des langages de requêtes (recherche par mots clés à l'aide du langage booleen, langage SQL, OQL …), des protocoles de rapatriement des données : CGI/http ou FTP etc

4.3.2 Les ontologies

Les ontologies sont très simplement des termes et des relations entre ces termes partagés par une communauté (pour plus de détail voir **annexe 8**). Elles apportent une réponse au problème de l'hétérogénéité sémantique existant entre les diverses sources de données biologiques. En étiquetant les différents objets biologiques à l'aide d'un vocabulaire contrôlé et en basant les recherches sur ce catalogue de termes standards on résout les problème des ambiguïtés de sens.

Des consortiums ont alors vu le jour **[a16]**, en vue d'établir une terminologie pour décrire les données et des hiérarchies pour classifier les concepts. Ainsi, le souci de standardisation de l'attribution de noms est pris en compte par le consortium HGNC (HUGO Gene Nomenclature Committee) **[a17]** qui propose une terminologie particulière pour les nouvelles séquences. Le projet GO (Gene Ontology) **[i57]**, de plus en plus utilisée par la communauté des biologistes, vise à fournir un ensemble structuré de vocabulaires pour des domaines biologiques spécifiques permettant de décrire des produits de gènes (protéines ou ARNs) dans un organisme donné. Des propositions pour permettre à la communauté biologique de spécifier et d'échanger des ontologies ont aussi vu le jour : standard OIL **[a28]**, méta-données (qui sont des données servant à décrire d'autres données) …

4.3.3 Les différents systèmes d'intégration

Depuis quelques années, de nombreuses solutions au problème de l'hétérogénéité des sources génomiques et à leur intégration ont été proposées. Certaines suivent une approche "non matérialisée" dans laquelle les données restent au niveau des sources tandis que d'autres suivent une approche "entrepôt" (Datawarehouse) dans laquelle les données sont extraites des différentes sources et combinées dans un schéma global **[i67]**. Nous présentons d'abord les solutions et projets permettant d'aboutir à une approche "non matérialisée" puis nous présenterons ceux suivant l'approche "entrepôt". Les avantages et inconvénients de chaque solution seront ensuite discutés.

a) L'approche non matérialisée/décentralisée/distribuée

➢ Les solutions à base d'hyperliens

Une solution consiste à intégrer sur une même page des liens sur différentes bases de données et outils. Il ne s'agit pas réellement d'une approche intégrative si ces systèmes ne sont pas interconnectés, cependant cette démarche à au moins le mérite de faciliter l'accès à ces outil. Ainsi des portails thématiques se sont multipliés sur Internet ces dernières années. Parmi les plus connus on peut citer le portail d'Infobiogen **[i58]**, le site Entrez qui permet d'accéder aux banques du NCBI ou celui d'Expasy **[i70]** construit autour de Swiss-Prot. Le site d'Expasy héberge notamment le World-2DPAGE **[a18] [i18]** qui est un index sur de nombreuses bases de données de gel 2D (SWISS-2PAGE présentée en **4.2.2** y est référencé). Sur ce site, les bases sont interconnectées grâce à des hyperliens basés sur leurs références croisées.

Cependant les liens hypertextes ne peuvent accepter qu'un nombre restreint de paramètres ce qui limite le nombre de requêtes entre bases. Afin d'exploiter les références croisées entre banques et permettre d'effectuer des recherches par mots clés, le système SRS **[a19]** a été développé par l'EBI (European Bioinformatic Institute) **[i59]**. Il permet à partir d'une interface commune (illustrée **annexe 9**) de diriger les requêtes sur diverses bases de données indexées par le système et reliées entre elles par les liens de références croisées qui font l'objet d'une indexation. C'est à l'utilisateur de choisir la/les ressource(s) qu'il souhaite interroger. Cependant son utilisation se limite à l'interrogation par mot clés de banques de données fournissant leurs données sous forme de fichiers à plats. Les interrogations ne peuvent pas se révéler aussi complètes que le permettrait un véritable langage de requête type SQL fournit par un SGBD.

➢ Les solutions à base de services Web

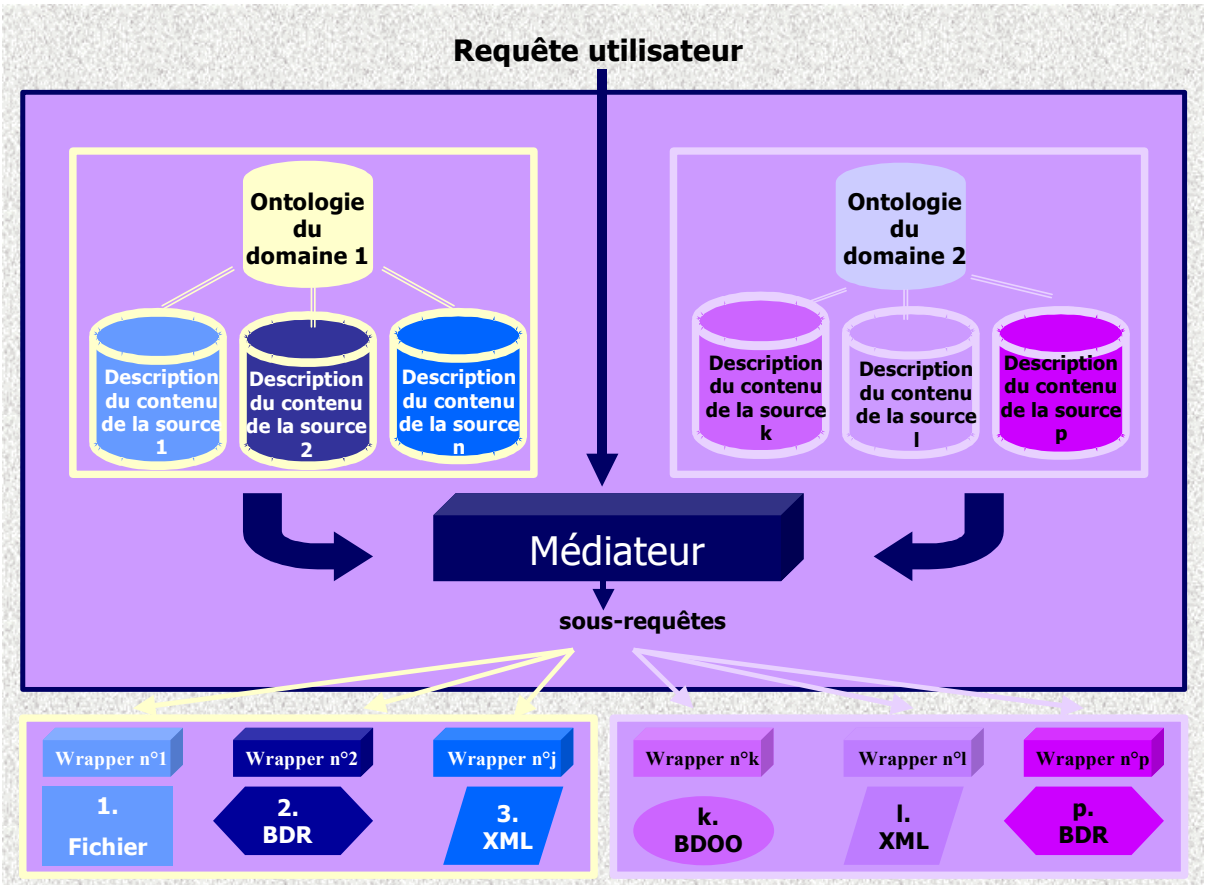
Les services Web peuvent être une réponse à ce genre de limitation. En effet, ils vont permettre d'utiliser un service permettant d'accéder aux informations d'une application distante sur Internet. Ce service peut être une méthode encapsulant une requête SQL sur cette application (*par exemple celle-ci peut demander la séquence au format FASTA dans SWISSPROT pour la protéine d'identifiant P04406).*

Ainsi il est possible d'effectuer des requêtes à distance beaucoup plus évoluées qu'avec les solutions à base d'hyperliens. Ensuite l'avantage avec les web services est que l'accès aux données (grâce à l'utilisation de SOAP et du XML) s'affranchit de l'hétérogénéité du format de ces données que ce soit fichier texte, bases de données relationnelles ou objets, XML, etc (p.ex. avec les hyperliens on est dépendant du HTML). Par là même on s'affranchit complètement du langage de requête associé à ces données puisque on ne fait qu'utiliser un service. Ces services sont en général déposés dans un annuaire de services (UDDI ou Universal Description Discovery and Integration) sur Internet par les fournisseur de services eux-même *(dans l'exemple de la requête précédente celui-ci peut-être SWISSPROT)*. En ce qui concernent leur caractéristiques, les services web se basent sur du XML pour représenter les données, le protocole HTTP est le protocole de transport de l'information et le standard SOAP (Simple Object Access Protocol) est le protocole d'invocation des messages envoyés aux applications distantes.

Parmi les projets se basant sur l'utilisation des services Web, il y'a le projet BioMOBY **[i38] [a20]**. Il a pour objectif essentiel de centraliser et rendre facilement accessibles les services disponibles dans le domaine de la biologie. Ce projet utilise notamment une ontologie (dite ontologie de services) pour les décrire afin de faciliter leur exploitation. Ensuite, le projet PlaNet **[i60] [a21]** est un projet plus spécialisé de l'EBI, qui se base sur l'utilisation de BioMoby pour garantir l'interopérabilité entre les différentes bases de données européennes sur Arabidopsis et d'autres plantes. Enfin, le projet MyGRID **[i61]** a pour objectif de concevoir une architecture basée sur Internet pour pouvoir proposer des services sur une grille informatique.

➢ L'approche médiateur (ou bases de données fédérées [a26])

Cette approche décrite originellement par Wiederhold **[a22]**, suggère une architecture standard d'un système construit au-dessus des sources à intégrer : ce système consiste en un module central (le médiateur) et des interfaces d'accès aux différentes sources, les wrappers (ou adaptateurs)(**<u>document 11</u>**).



<u>Document 11 :</u> Architecture Médiateur (source : **[i65]**)

L'utilisateur d'un tel système posera une requête au médiateur, lequel se chargera de traduire la requête en des sous-requêtes exécutables sur les sources locales qui peuvent alors être dans des formats hétérogènes. La réponse à la requête est construite à partir des réponses fournies par les différentes sources. Ce système résout ainsi le problème de l'hétérogénéité syntaxique. Dans les systèmes les plus évolués l'hétérogénéité sémantique est prise également en compte par l'utilisation notamment des ontologies.

Parmi les systèmes en biologie qui utilisent la notion de médiateur, on peut citer K2/Kleisli **[a23]**, **[a24]**, projet de l'université de Pennsylvanie, qui est une API qui permet d'interroger un ensemble de sources de données génomiques en utilisant un unique langage de requêtes, OQL, avec un modèle de données objet. Le projet DiscoveryLink **[a29]** est lui plus récent et propose la mise en forme des sources sous un schéma relationnel, leur interrogation en SQL et l'intégration d'une dizaine d'applications bioinformatiques. Un autre projet développé à l'Université de Manchester est TAMBIS (Transparent acces to multiple biological information sources) **[a25]**, qui utilise une démarche radicalement différente en dissimulant l'aspect requête derrière une ontologie qui va aider l'utilisateur à former ses requêtes. Cette dernière approche privilégie l'intégration sémantique par rapport à la manipulation des données.

Ces différents systèmes, dont beaucoup sont encore à l'état de projets recherche, offrent à priori tous les avantages d'une approche non matérialisée tout en gérant les problèmes d'hétérogénéité syntaxique et sémantique entre les différentes sources de données. Cependant leur niveau de complexité en font des outils difficiles à implémenter voir à utiliser comparé à des systèmes à bases d'hyperliens ou de web services. Il reste ici à évaluer l'utilisation de ces différents outils au sein de projets spécifiques nécessitant l'intégration de différentes sources de données biologiques. Dans ce contexte je peux citer le travail de Pierre Larmande qui effectue une thèse en bioinformatique au CIRAD de Montpellier et réalise une comparaison **[i64]** de l'approche médiateur du système LeSelect **[i79]** et de BioMoby pour permettre l'intégration de deux bases de données biologiques, dont une est relationnelle (MGIS **[i62]**) et l'autre objet (TropGene **[i63]**).

b) L'approche entrepôt de données/centralisées

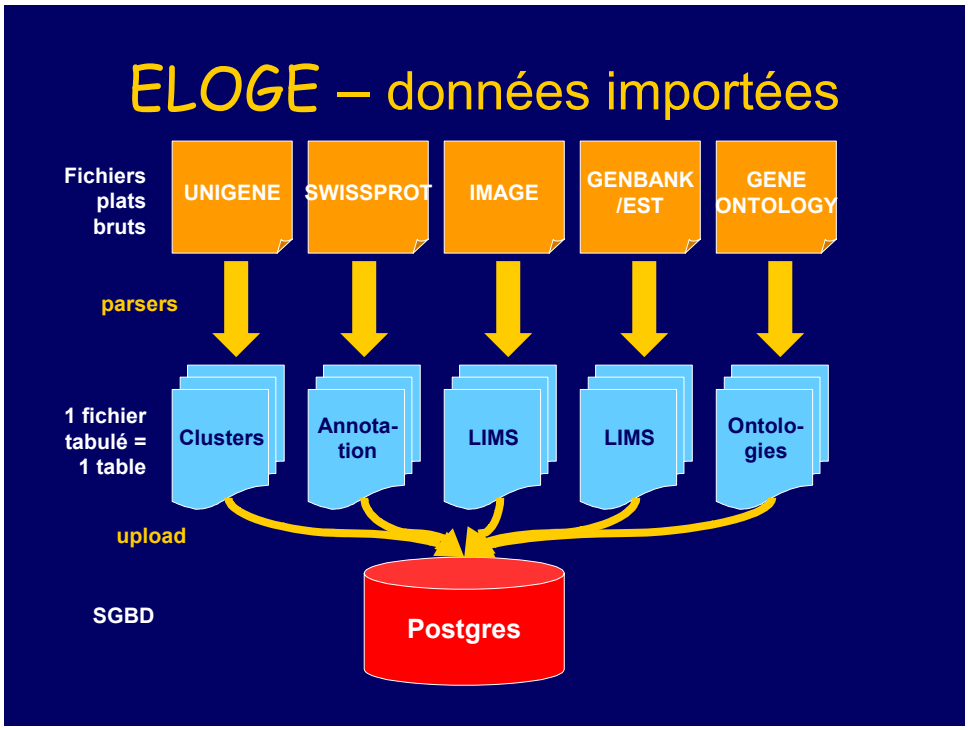
➢ Principe :

Dans une architecture de type entrepôt (appelée aussi Data Warehouse), les données sont extraites à partir des sources et stockées localement au sein d'une base de données unique. Le schéma global de l'entrepôt est constitué de l'union des schémas sources. Ce système peut proposer comme avec l'approche précédente une intégration à la fois au niveau syntaxique et sémantique, la différence étant que tout est centralisée. Le travail d'intégration syntaxique consiste ici à convertir l'ensemble des données sources (qui peuvent-être sous différents formats) dans le modèle unique choisi pour l'entrepôt (qui peut être aussi bien un modèle semi-structuré, relationnel, ou objet…). L'intégration sémantique consiste, elle, à établir une correspondance entre les schémas des sources et le schéma global intégrateur (ex : pour une table donnée, on fait correspondre la colonne *id* de la source 1, *ident* de la source 2 avec la colonne *identifiant* de l'entrepôt). Ensuite, l'intégration sémantique peut aller plus loin, en intégrant l'usage des ontologies dans le schéma local pour gérer les problèmes d'ambiguïtés de sens contenu dans les données.

➢ Exemples de projets type entrepôts de données

Je peux citer pour exemple le système ELOGE sur lequel j'ai travaillé de mars 2000 à septembre 2001 au laboratoire TAGC *(technologies avancées pour le génome et la clinique)* de Marseille. Eloge *(Environnement logiciel pour la génomique fonctionnelle)* **[i66]** permet l'archivage et l'analyse des données expérimentales issues des puces à ADN, de leur conception à leur utilisation.

La majeure partie des données génomiques de la base de donnée ELOGE est extraites de bases de données publiques : (UNIGENE, GENBANK, SWISSPROT, IMAGE, LOCUSLINK, GO) avec en supplément certaines données privées (librairies de clones, ESTs etc). Mensuellement une mise à jour de ces données est réalisée de façon automatique. Une partie de mon travail a consisté à réaliser des parsers **[G12]** qui sont des programmes permettant l'extraction des données des fichiers plats (bruts - sites ftp) des banques de données publiques en un format de fichier tabulé prêt à être chargé dans la base de données relationnelles d'ELOGE. Le **document 12** illustre la procédure utilisée pour importer les données dans la base locale de l'entrepôt de données.



Document 12 : Importation des données externes dans la base de donnée Eloge.

Cette expérience m'a permis d'appréhender les difficultés auxquelles tout concepteur d'entrepôt de donnée doit faire face pour intégrer des données de sources externes en local.

Un autre projet d'entrepôt de données est Chado **[i41]**, issu de l'initiative GMOD (Generic Model Organism Database Construction Set), qui propose une collection de schéma relationnel permettant d'accueillir des données en provenance de différentes thématiques biologiques. Enfin, on trouve également des entrepôt basés sur des modèles objets (projets GUS **[a67]** et GEDAW **[a27]**).

4.3.4 Comparaison des différentes approches

Les avantages et les inconvénients de chacune des approches décrites en **4.5.1 a)** et **4.5.2 b)** seront ici présentés.

➢ **Avantages de l'approche entrepôt de données/centralisée :**

Copier les données en local offre de nombreux avantages : cette démarche permet d'avoir un accès direct aux informations à la fois facile, rapide et **sécurisé**. La sécurité est un point crucial par exemple dans le domaine biomédical où les requêtes effectuées par les chercheurs ne doivent pas pouvoir être étudiées par d'éventuels laboratoires concurrents (recherche de brevets, découverte de nouveau gènes pouvant être impliqués dans une maladie, etc).

D'autre part, la copie locale des données offre la possibilité d'avoir un plus grand contrôle de l'information, on est propriétaire des données et on peut en **maîtriser les traitements**. Ainsi, on peut non seulement faire tourner sur les données des traitements complexes qui seraient trop lourds pour être lancés à distance mais aussi tester de nouvelles techniques de fouilles de données (*Data Mining*).

Enfin, et c'est souvent l'argument majeur qui est avancé par les constructeurs d'entrepôts de données, la copie locale des données permet un **ajout facile d'annotations** voire un réel nettoyage et tri de ces données, permettant une meilleure intégration des informations.

➢ Inconvénients de l'approche entrepôt de données/centralisée

Les inconvénients de cette approche correspondent aux difficultés liées à la construction et à la maintenance d'un entrepôt. Construire un entrepôt nécessite d'abord une étude des sources à intégrer pour en **dégager les informations *pertinentes*** à stocker puis une extraction des données des sources. Lors de l'insertion des données il faut veiller à supprimer les redondances et possibles divergences des données entre les différentes sources de données. Maintenir l'entrepôt impose, ensuite de **détecter les changements** dans les sources et mettre à jour régulièrement la base.

➢ Avantages de l'approche non matérialisée/décentralisée :

Ce type d'approche offre une réponse aux inconvénients de l'approche entrepôt de données. Ainsi avec l'approche non matérialisée on s'affranchit des problèmes de cohérences entre les données puisque tout est géré au niveau de la requête. Ensuite l'ajout et la mise à jour des données est prise en charge par les propriétaires des bases de données.

➢ Inconvénients de l'approche non matérialisée/décentralisée :

La construction de plans de requêtes n'est pas facilité dans l'approche non matérialisées de même que l'accès aux sources. Les requêtes sur les sources sont moins rapides que dans le cas de l'entrepôt de données. En effet, celles-ci doivent faire face au trafic sur le réseau et dans les banques de données publiques.

➢ Conclusion :

En conclusion nous pouvons constater que chacune des deux approches (centralisée/décentralisée) est complémentaire en terme d'avantages et inconvénients. Le choix d'une ou l'autre des solutions dépendra ensuite du contexte et des objectifs du projet. Nous présenterons dans la partie suivante en **5.3.1** les raisons qui ont poussé Génoplante à utiliser l'approche entrepôt de données pour concevoir le système d'information GpiIS.

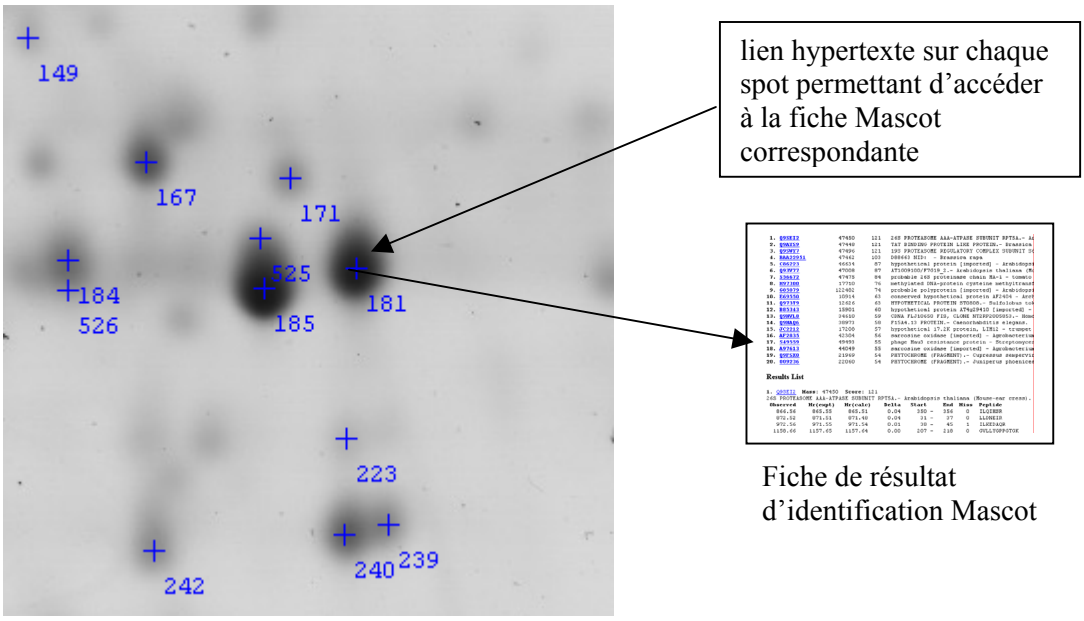
5 Analyse de l'existant et solution retenue

Dans cette partie nous ferons le bilan de l'existant au niveau de mon laboratoire pour chacun des grands volets du projet ProteomIs/GnpProt. Les grandes lignes du projet ProteomIs/GnpProt ont été définies dans la partie **2.2** de ce mémoire. Cependant lorsque cela sera nécessaire, nous apporterons des précisions sur ces objectifs notamment au niveau de l'analyse des données. Enfin dans chaque thème, nous présenterons la solution retenue en fonction des objectifs, des données de l'existant mais aussi de l'état de l'art (chapitre **4** précédent).

5.1 Gestion des données

5.1.1 Analyse de l'existant

Il existe déjà à l'unité de recherche en protéomique de Montpellier une application (appelé Click) permettant de gérer sommairement à la fois les résultats d'analyse de gels 2D et les résultats d'identifications de spectrométrie de masse (fiche Mascot). Cet outil développé en PHP par Thierry Hotelier permet notamment de visualiser des gels 2D sous la forme d'images cliquables via un navigateur (voir **document 13**). Ce programme avait été essentiellement réalisé dans un but de mise à disposition d'informations sur les différents postes de l'unité. Chacune des données sur un gel (image au format jpg, fichiers Mascot html) est contenue dans un répertoire portant le nom de ce gel. Par la suite, Thierry Hotelier a développé un programme (appelé aussi Parseur Mascot **[G12]**) capable d'extraire un certain nombre d'informations d'une fiche Mascot et de les présenter au format CSV **[G13]**.



Document 13 : Application Click développée à l'UR protéomique de l'INRA de Montpellier

5.1.2 Solution retenue

➢ **Un choix imposé**

L'organisation et le stockage des données associées au protéome sont à considérer tout particulièrement. Les données sont complexes, nombreuses et émanant de différents laboratoires. En outre, l'application à mettre en place doit pouvoir être opérationnelle de manière isolée comme de manière intégrée. Dans ce contexte, seule la solution de type "base de données relationnelle" peut être retenue.

En effet d'une part, le système GpiIS, dans lequel doit s'intégrer GnpProt, est prévu pour s'appuyer sur un SGBD relationnel (vraisemblablement Oracle ou Postgresql), et impose, de ce fait, que l'ensemble des modules, qui constituent le système, soient conçus à l'aide du modèle relationnel. D'autre part, l'application, à mettre en place, doit être particulièrement robuste et performante et les SGBD relationnels sont particulièrement reconnus pour leur robustesse (mécanismes de tolérance aux pannes), leur efficacité (langage de définition et de manipulation des données SQL) ainsi que pour les mécanismes mis en place pour garantir la sécurité et l'intégrité des données (droits utilisateur, contraintes d'intégrité, déclencheurs ou triggers...) Ainsi, nous ferons le choix d'analyser, concevoir et implémenter une base de données relationnelle. Nous allons cependant présenter une alternative possible déjà implémentée avec succès dans le monde de la protéomique.

➢ **Une alternative possible**

Nous avons déjà vu que les biologistes avaient une véritable culture de la banque de données et du fichier à plat. Une autre approche aurait été d'adopter une démarche de type collection de fichiers et de se doter d'un format de fichier spécifique et d'un moteur de recherche. L'outil Make2DB (présenté partie **4.1.2**) est une concrétisation de cette démarche et s'articule autour d'un système de gestion de fichiers et d'un moteur de recherche permettant l'accès aux informations sur les différents objets biologiques (protéines, gènes,...). Dans le contexte du projet, le recours à un moteur de recherche pourrait être une démarche complémentaire, par exemple, pour effectuer des recherches par mots clés sur des documents annexe de type référence bibliographique. A cet effet, nous pourrions utiliser des moteurs de recherche relevant du domaine public comme Htdig **[i68]** ou Webglimpse **[i69]**.

➢ **Les points clés**

Nous allons énumérer ci-dessous les points clés qui rendent une approche "base de données relationnelle" incontournable :
- confidentialité des données : dans le cadre du consortium Génoplante, les données biologiques ne sont pas toujours publiques et libres d'accès. Il faut donc pouvoir définir des rôles et des profils utilisateurs avec des mécanismes de droits appropriés.
- consistance des données : les données sont partagées par différents partenaires et donc accessibles en accès concurrents. Les mécanismes transactionnels sont donc nécessaires pour garantir la consistance des données lors d'accès concurrents.
- cohérence des données : la qualité des données est essentielle alors que même ces données sont nombreuses et complexes. Les contraintes d'intégrité (clés primaires, clés étrangères, contraintes de domaine) et les contraintes que l'on pourrait qualifier d'évènementielles (les déclencheurs par exemple) sont comme autant de gages de qualité.
- puissance du langage d'interrogation : les interrogations peuvent se révéler tout aussi complexes que les données et nécessitent de ce fait toute la puissance d'un langage de requête comme SQL (jointures, divisions, requêtes imbriquées, ...).

➢ **Réutiliser l'existant**

Les fonctionnalités de l'application Click ont été intégrées et améliorées dans ProteomIs/GnpProt. Ensuite, dans la partie **4.1.2**, nous avons étudié trois modèles relationnels supportant des applications associées au protéome, à savoir PROTICdb, PARIS et PPMDB. Le modèle de données de PPMDB s'est révélé suffisamment proche de notre perception et nous nous en sommes inspirés. Nous avons cependant fait largement évoluer ce modèle afin de pouvoir l'étendre (aux données issues des expériences de spectrométrie de masse) et de l'adapter aux contraintes imposées par l'intégration dans GpiIs.

➢ **Consultation des données contenues dans la base**

Plusieurs aspects sont à considérer pour tout ce qui concerne la restitution de l'information à l'utilisateur :
- une interface graphique : l'usager doit pouvoir interroger la base de données au travers d'une interface utilisateur adaptée. Il est nécessaire de rendre cette interface la plus flexible possible afin d'autoriser l'expression de requêtes complexes et d'avoir une visualisation adaptée des résultats.
Ensuite ses interfaces doivent être dotées de moyens supplémentaires tels que :
- facilités pour l'exportation des données :
Les données résultats doivent pouvoir être exportées sous différents formats : format de balisage XML, format tabulaire CSV, ... afin de pouvoir être traitées ultérieurement.
- facilités pour le rendu graphique de certains résultats :
Il est nécessaire, par exemple, pour les informations associées à un gel d'électrophorèse de proposer des présentations graphiques très spécifiques. ProteomIs/GnpProt devra par exemple être doté d'un visualiseur de gels d'électrophorèses.

➢ **Alimentation de la base de données**

L'objectif ici est de concevoir un outil permettant de saisir rapidement la somme importante des données accumulée au sein des différents laboratoires. Dans ce contexte deux solutions ont été étudiées : un fichier d'échange de données et des interfaces de saisie

Le fichier d'échange de données :

Celui-ci (appelé aussi format d'échange) se présente sous la forme d'un classeur Excel servant de support à la saisie. Chaque feuille correspond à une ou plusieurs tables de la base et les colonnes correspondent aux différents champs. Une fois les données saisies, chaque feuille est convertie au format CSV puis lu par un programme servant à importer les données dans la base. L'intérêt de cette solution est d'éviter d'avoir à ressaisir une quantité importante de données présentes dans des fichiers textes ou Excel. En effet, il est alors facile par des actions de copie de réorganiser ces données au sein du format d'échange.

Les interfaces de saisie :

L'idée est de concevoir des interfaces dotées de facilités d'importation des données contenues dans des fichiers textes. Ceci permettra comme dans la solution précédente de limiter au maximum la quantité de données à saisir manuellement. Par contre, on bénéficira en plus des avantages que peut procurer une interface à savoir : le contrôle des données saisies (devant répondre à des contraintes d'intégrités) et la mise à disposition de listes de choix. Toutes ces procédures facilitent la saisie des données et éliminent tout risque d'erreur. En effet les contrôles se faisant au moment de la saisie, l'utilisateur est immédiatement invité par un message à soumettre une nouvelle donnée si une erreur est détectée par l'interface. Au contraire avec la solution « format d'échange » le producteur de données n'est pas averti des erreurs de saisie dans le fichier Excel. Ce n'est qu'au moment de l'exécution du programme d'importation des données que les erreurs seront détectées. C'est alors au gestionnaire des données d'effectuer les corrections nécessaires dans le fichier Excel.

J'avais réalisé au début du projet un ensemble de maquettes d'interfaces de saisie sous Powerpoint qui avait été confronté à l'ensemble des partenaires du projet Génoplante. Les interfaces avaient été jugées satisfaisantes mais trop longues à développer. Les contraintes de délais étant fortes en ce qui concerne la saisie des données, le format d'échange a donc été préféré. Nous verrons cependant dans la partie **8.2 Perspective** qu'il est prévu de développer un programme permettant au producteur de données de vérifier les données qu'il saisit dans le format d'échange.

5.2 Analyse des données

5.2.1 Création de groupes non redondant de protéines (« clustering »)

a) Précision sur les objectifs

Les bases de données biologiques sont le reflet du travail de plusieurs centaines de laboratoires disséminés à travers le monde qui soumettent à intervalle régulier le fruit de leur travail, en l'occurrence les séquences biologiques. Une difficulté de cette politique de soumission réside dans le contrôle de la redondance. Rien, en effet, n'empêche la soumission de la même séquence biologique par des laboratoires différents. Les séquences peuvent être rigoureusement identiques ou présenter de très petites différences dans leur enchaînement. En outre, ces séquences ont bien souvent été analysées et annotées de façons multiples. Il s'agit alors d'éliminer l'information redondante tout en conservant l'information utile. Cette stratégie doit être appliquée à la base de donnée ProteomIs confrontée à ce genre de problème.

Nous donnons un exemple pour illustration : un laboratoire dépose dans ProteomIs une séquence protéique référencée à l'origine dans la base de donnée SWISSPROT. Cette protéine a par exemple pour identifiant le numéro d'accession Q9FG34. Maintenant, imaginons qu'un autre laboratoire dépose une autre protéine en provenance cette fois de la base TAIR avec le numéro d'accession At5g06730. Si l'on vérifie la nature des protéines Q9FG34 et At5g06730 respectivement dans les bases de données SWISSPROT et TAIR, on s'aperçoit qu'il s'agit dans les deux cas d'une enzyme Peroxydase. La base ProteomIs référencera alors cette protéine dans deux instances (ou entrées) différentes (**document 14**). Il y'a donc multiplication des entrées pour un même objet biologique ce qui n'est pas acceptable dans le cadre d'une gestion efficace de ces données. De plus cela peut poser des problèmes pour effectuer une recherche rapide et surtout synthétique. Par exemple, lorsque le biologiste effectuera une recherche sur l'enzyme Peroxydase à partir de l'identifiant At5g06730, il n'obtiendra pas les informations expérimentales complémentaires sur la Peroxydase enregistrée sous l'identifiant Q9FG34. Une recherche sur le nom exact ne garantit pas toujours non plus que l'on récupère les deux instances de la protéine car il peut exister différente façon de nommer le même objet biologique *(à moins de mette en place un système d'interrogation sur la base d'un vocabulaire contrôlé utilisant des ontologies)*. L'objectif sera donc, comme illustré dans le **document 14** de rassembler dans une seule instance de cette protéine les informations associées aux numéros d'accessions At5g06730 et Q9FG34. Après ce travail de regroupement (appelé aussi « **clustering** » en bioinformatique), nous qualifierons la base de données de « non redondante ».

Remarque sur le vocabulaire utilisé : Les termes de base de donnée « redondante » ou « non redondante » doivent être interprétés, dans le cadre de ce mémoire, comme un raccourci nous permettant de désigner l'état de redondance dans lequel se trouvent les séquences stockées dans ProteomIs.

Document 14 : Elimination de la redondance dans la base ProteomIs = clustering

Base ProteomIs REDONDANTE

1 identifiant SWISSPROT

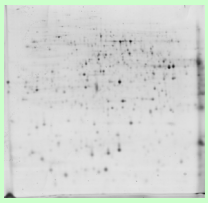
1 identifiant AGI

accession id : Q9FG34
database : SWISSPROT

protein name : Peroxydase 54 [Precursor]

Experiments :

• Gel 2D : G229
spot : 220
pi : 5.73
mw : 34

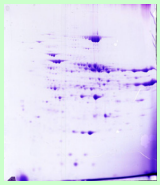


accession id : At5g06730
database : TAIR

protein name : Peroxydase

Experiments :

• Gel 2D : G300
spot : 150
pi : 5.24
mw : 33



= 2 instances du même objet biologique Peroxydase

CLUSTERING

Base ProteomIs NON REDONDANTE

1 identifiant unique AGI

= 1 instance de l'objet biologique Peroxydase

accession id : At5g06730
database : TAIR

protein name : Peroxydase

Cross-references :

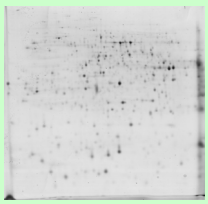
accession id : Q9FG34
database : SWISSPROT

Références croisées

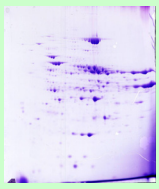
Experiments :

Synthèse des résultats expérimentaux

• Gel 2D : G229
spot : 220
pi : 5.73
mw : 34



• Gel 2D : G300
spot : 150
pi : 5.24



b) Etude des solutions :

Le problème de la solution à mettre en œuvre se pose alors pour être capable de déterminer si une protéine est équivalente à une autre dans la base de donnée ProteomIs afin de les regrouper. Face à ce genre de problème les grandes bases de données biologiques suivent plusieurs approches. Par exemple la solution envisagée par la base de donnée SWISSPROT réputée pour la qualité de ses informations, est de faire contrôler par un expert la nature des données saisies dans la base. Dans le cadre de ProteomIs, utiliser une personne pour faire ce travail n'était pas envisageable. Il fallait donc se tourner vers une solution automatisant le processus.

- Solution utilisant des logiciels de comparaison de séquences :

La première idée est d'utiliser le travail d'un logiciel de comparaison de séquences. Cette stratégie a été utilisé par l'Institut Suisse de Bioinformatique (SIB) pour éliminer la redondance au sein de sa base de données TrEMBL **[a8]**. Le logiciel alors utilisé est Lassap (Large Scale Sequence compArison Package) **[a9]** permettant, grâce à un langage de requêtes, de lancer des comparaisons de banque (ou sous-banque) contre banque. Ce logiciel ou un logiciel tel que BLAST ou FASTA (partie **4.2.1**) pourrait être utilisé pour effectuer des comparaisons entre toutes les séquences de la base ProteomIs.

Il resterait ensuite à analyser les résultats pour effectuer les regroupements sur la base des séquences homologues. On a vu que ces logiciels comparaient les séquences sur la base de leur composition en acides aminés. La correspondance n'étant pas toujours exacte mais approximative, le résultat par exemple dans le cas de BLAST est une liste de séquences ordonnées par ordre décroissant d'homologie avec la séquence candidate. Chacune des séquences résultats se voit alors attribuée plusieurs valeurs de score (score brut, e-value). Il se pose alors le choix d'une valeur de seuil à partir de ces valeurs pour retenir la séquence similaire. Cependant le choix n'est pas si simple. Les biologistes ont l'habitude de procéder à une évaluation de ces résultats en fonction de la connaissance qu'ils ont du domaine. Il arrive même souvent qu'une comparaison avec BLAST soit réitérée plusieurs fois en faisant jouer plusieurs paramètres de BLAST et ce jusqu'à ce que le résultat obtenu soit satisfaisant pour le biologiste. Ces paramètres peuvent être par exemple le masquage de certaines régions répétées dans les séquences ou le choix d'une matrice de comparaison. L'automatisation sur la base de l'exécution d'un logiciel de comparaison de séquences est donc risquée et nécessiterait un travail de vérification des résultats trop fastidieux.

- Solution basée sur l'utilisation des équivalences entre numéros d'accessions :

La solution envisagée pour le projet ProteomIs sera en fait basée sur l'utilisation des équivalences entre les numéros d'accessions des bases de données publiques. Pour cela on utilisera le champ références croisées des bases de données publiques. On peut ainsi facilement construire des groupes de protéines dont on est certain qu'elles sont équivalentes puisque le rapprochement a déjà été effectué par des experts.

L'objectif du projet de **clustering** sera d'automatiser cette démarche afin qu'elle soit réalisable sans intervention humaine. Les résultats devront être organisés afin qu'ils soient exploitables pour mettre ensuite à jour la nouvelle version non redondante de la base de donnée. Nous allons maintenant présenter la démarche technique envisagée pour mettre en œuvre cette solution.

c) Solution technique envisagée pour mettre en place le clustering

La redondance des séquences provenant de la multiplication des types d'identifiants dans ProteomIs, la mise en œuvre du projet de clustering impose d'abord que l'on définisse une **nomenclature unique** au niveau des identifiants des protéines, comme c'est le cas dans les bases de données publiques. Le choix se portera sur la nomenclature **AGI [G15]** qui a été crée pour nommer les séquences des gènes ou protéines d'Arabidopsis Thaliana.

La majeure partie des protéines de la version redondante de ProteomIs étant déjà identifiée sous la nomenclature AGI, l'objectif sera de trouver l'équivalence AGI des séquences identifiées sous une autre nomenclature (p.ex SWISSPROT). Ensuite, les séquences possédant le même accession AGI pourront être regroupées en cluster sous un identifiant unique (AGI) et la redondance éliminée.

Voici, maintenant, à travers un exemple, les différentes étapes prévues pour aboutir à ce résultat :

Il nous faut dans un premier temps construire la liste des numéros d'accession non AGI présents dans la base et obtenir pour chacun leur équivalence en AGI. Sur le **document 15** est détaillée la procédure qui sera utilisée en prenant pour exemple l'accession SWISSPROT Q9FG34 dont on souhaite retrouver l'accession AGI équivalent.

La solution est de lire la fiche SWISSPROT correspondant à l'accession Q9FG34 ; ceci en interrogeant directement la base SWISSPROT sur son site internet. La référence AGI At5g06730 peut alors être récupérée dans le champ « Références croisées » et venir compléter la liste des équivalences entre accessions AGI et NON AGI.

Les accessions NON AGI de cette liste seront conservés pour pouvoir construire ultérieurement le champ « Références croisées » de la base ProteomIs comme sur le **document 28** précédent. Ces références permettront ainsi d'accéder à des informations complémentaires dans les banques où cette protéine est décrite. Le champ référence croisée est une notion importante dans les bases de données publiques car elle permet d'établir des passerelles entre ces différentes bases. Ce champ existe dans beaucoup de bases de données biologiques comme par exemple SWISSPROT qui est très riche en références de ce genre (voir le format SWISSPROT partie **3.2.1**).

En ce qui concerne les accessions NON AGI pour lesquels on n'aurait pas trouvé de correspondance automatique avec la nomenclature AGI, le biologiste aura la possibilité d'utiliser l'interface de comparaison par BLAST pour compléter cette fois manuellement les équivalences. Le gestionnaire de données devra ensuite exploiter ces résultats pour supprimer cette fois manuellement la redondance restante dans la base de donnée ProteomIs.

Remarque : Pour retrouver les équivalences entre accessions, une alternative à l'exploitation du champs référence croisée des bases de données publiques aurait pu être l'utilisation de l'application AliasServer **[i53] [a51]**. Cette application développée par le CBiB *(Centre de Bioinformatique de Bordeaux)* permet, à partir d'un (ou plusieurs) accessions requêtes, de retrouver automatiquement la liste des accessions équivalents dans les bases de données. Cet outil n'a pas été utilisée car la publication trops récente de cet outil dans la revue Bioinformatic est apparue après que notre application de clustering soit développée.

Document 15 : **Utilisation des équivalences entre numéros d'accessions pour le clustering**

1 entrée SWISSPROT = 1 Protéine = 1 identifiant = 1 numéro d'accession

NiceProt View of Swiss-Prot: Q9FG34

[Entry info] [Name and origin] [References] [Comments] [Cross-references]

Note: most headings are clickable, even if they don't appear as links. They link to the user manual or other documents.

Entry information	
Entry name	PER54_ARATH
Primary accession number	Q9FG34
Secondary accession number	P93729
Entered in Swiss-Prot in	Release 41, February 2003
Sequence was last modified in	Release 41, February 2003
Annotations were last modified in	Release 46, February 2005
Name and origin of the protein	
Protein name	**Peroxidase 54 [Precursor]**
Cross-references	
EMBL	AP002032; BAB09807.1; -. [EMBL / GenBank / DDBJ] [CoDingSequence] AK118827; BAC43417.1; -. [EMBL / GenBank / DDBJ] [CoDingSequence] BT008584; AAP40411.1; -. [EMBL / GenBank / DDBJ] [CoDingSequence] AY088509; AAM66044.1; -.[EMBL / GenBank / DDBJ] [CoDingSequence] Y11794; CAA72490.1; -. [EMBL / GenBank / DDBJ] [CoDingSequence]
HSSP	Q42578; 1PA2. [HSSP ENTRY / PDB]
GeneFarm	1908; 61
TAIR	At5g06730; Q9FG34.

Références croisées

INTERROGATION

INTERNET

GET

ProteomIs

Liste des accessions SWISSPROT, NCBI … :

NON AGI

Q9FG34
AAG21494
P54609
AAC04902
AAD25855
CAC35872
AAD31573
AAD26480
AAC64298
AAC20727
AAD25640

Fichier d'équivalences entre accessions

NON AGI	AGI
Q9FG34	At5g06730
AAC04902	At2g33210
AAD25855	At2g14170
CAC35872	At5g08670
AAD31573	At2g36880
AAD26480	At2g31390
AAC64298	At2g43090
AAC20727	At2g30930
AAD25640	At2g05710

Liste des accessions dont l'équivalence n'a pas été trouvée :

NON AGI

AAG21494
P54609

5.2.2 La comparaison de séquences

a) Rappel sur les objectifs :

L'objectif est d'installer en local une application permettant de comparer n'importe qu'elle séquence avec les séquences contenues dans ProteomIs. Comme les séquences de protéines contenues dans la base ont des fonctions connues, le biologiste pourra ainsi déterminer la fonction de la protéine qu'il a voulu comparer et accéder aux annotations et résultats expérimentaux de protéomique qui ont servi à caractériser cette protéine.

b) Solution retenue :

Nous avons vu dans la partie **4.2.1** qu'il existait déjà des logiciels spécialisés dans la comparaison de séquence. De plus ces logiciels ont l'avantage d'être disponibles en libre téléchargement à partir d'Internet où inclus dans des packages de logiciels spécialisés en bioinformatique (EMBOSS **[i31]**, GCG **[i32]**). D'après la comparaison des différents logiciels de comparaison de séquence effectuée dans la partie **4.2.1**, BLAST reste le plus adapté aux besoins du projet. BLAST étant disponible pour être installé en local ; il pourra ensuite être configuré pour pouvoir effectuer des comparaisons contre la base de donnée ProteomIs. BLAST étant le logiciel de comparaison de séquence le plus utilisé dans la communauté des biologistes, l'avantage est aussi que toutes les procédures d'installation sont décrites dans de nombreux tutoriaux **[i67]**. Il est ensuite possible d'utiliser ce logiciel en ligne de commande mais cela ne le met pas à la portée de l'utilisateur biologiste. Pour cette raison, une interface homme machine (IHM), permettant de faciliter l'utilisation de BLAST, doit être développée.

5.2.3 La recherche de motifs

a) Précision sur les objectifs :

L'objectif est de mettre en place un système pipeline (ou chaîne de traitement) constitué d'un « pool » de logiciels bioinformatiques capables de détecter des motifs et domaines sur les séquences protéiques de ProteomIs. La priorité est de se concentrer sur l'étude d'un motif spécifique appelé phosphorylation (voir partie **4.2.2**). La démarche préconisée par le laboratoire, pour effectuer cette étude est d'associer des résultats expérimentaux aux résultats des logiciels. Cependant ce travail est long et fastidieux. C'est pourquoi un des objectifs du projet ProteomIs est de développer une application permettant de faciliter le travail des biologistes à ce niveau. Dans un deuxième temps cette application devra être adaptée pour analyser de manière automatique les séquences de la base ProteomIs.

b) <u>Analyse de l'existant :</u>

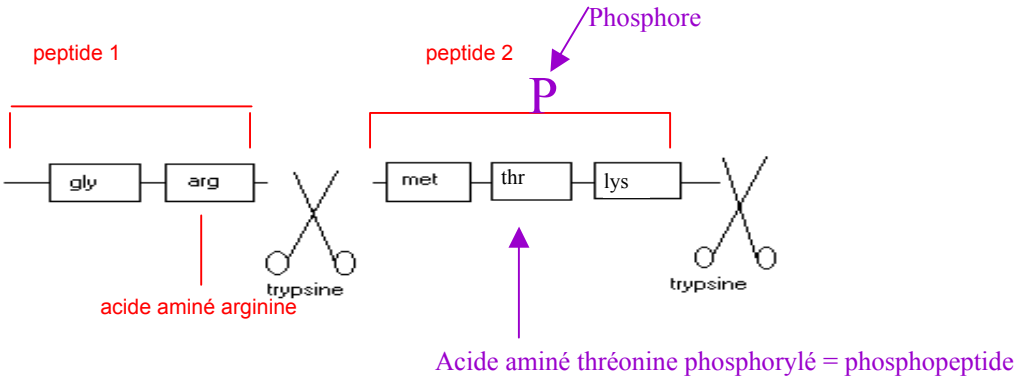
Pour construire un outil facilitant le travail du biologiste, il convient d'avoir une bonne connaissance de ses habitudes de travail. La présentation qui suit va nous permettre d'évaluer les phases qu'il sera nécessaire d'automatiser à l'aide de l'outil informatique.

➢ Démarche expérimentale

Nous allons donc commencer par examiner comment les biologistes procèdent pour identifier expérimentalement la position des sites de phosphorylation dans la séquence des protéines :

La démarche présentée ici se base sur la mesure de la masse des peptides (p.ex en utilisant un spectromètre de masse MALDI-TOF). Les 3 premières étapes ont été présentées dans le **document 3** de la partie **3.4**. Pour rappel, il s'agit tout d'abord de séparer les protéines d'un extrait protéique à l'aide de la technique d'électrophorèse bidimensionnelle. Les spots contenant différentes protéines sur le gel sont ensuite prélevés sur le gel et les protéines digérées par une enzyme nommée trypsine. La digestion des protéines par la trypsine permet de couper la protéine contenue dans le gel en plusieurs peptides qui la composent. En effet, la trypsine coupe les liaisons peptidiques après deux acides aminés (a.a) connus (la lysine et l'arginine).

De manière schématique, la trypsine agit de cette manière :



Cela va nous permettre d'obtenir une combinaison (empreinte) de peptides caractéristiques de la protéine. La lettre P en violet symbolise un phosphore qui est posé sur l'acide aminé thréonine du peptide 2. Seul les acides aminés sérine (S), thréonine (T) et tyrosine (Y) sont capables d'accepter une molécule de phosphore. On dit alors qu'ils sont phosphorylés et par extension, le peptide possédant cet acide aminé l'est également. On parle alors de phosphopeptide.

Peptide 1	Peptide 2	Peptide 3	Peptide 4	Peptide 5

Dans notre exemple le peptide 2 en violet est donc phosphorylé. Pour compléter notre exemple nous admettrons que le cinquième peptide également en violet est phosphorylé.

Pour le biologiste qui ne connaît pas les propriétés physico-chimique de sa séquence, l'objectif est de déterminer si cette protéine contient un ou plusieurs peptide phosphorylé.

La démarche est la suivante :

A l'aide d'un procédé appelé IMAC **[G11],** le biologiste va pouvoir récupérer uniquement les peptides phosphorylés de sa protéine digérée. L'ensemble des peptides phosphorylés de la protéine va ensuite être analysé en spectrométrie de masse.

Au final, le biologiste va obtenir une liste de masses avec par exemple les valeurs suivantes dont on sait qu'elles appartiennent uniquement à des phosphopeptides :

Masses expérimentales :

700
800

Maintenant que l'on sait que la protéine contient des phosphopeptides, l'objectif est de savoir à quels peptides dans la protéine appartiennent ces masses.

Le pré-requis indispensable pour faire ce rapprochement est de disposer de la séquence d'origine de la protéine analysée. Pour cela, la protéine de départ qui est de nature inconnue devra être au préalable identifiée grâce à la spectrométrie de masse. Une fois reconnue par le logiciel Mascot, la séquence de la protéine peut alors être extraite par interrogation dans les bases de données biologiques. La solution est ensuite de calculer, à partir de la séquence de cette protéine, la masse théorique de tous les peptides obtenus après digestion par la trypsine. Cependant il faut savoir qu'un peptide phosphorylé possède une masse qui est supérieure à celle d'un peptide non phosphorylé. Il faudrait donc inclure dans notre liste de masses celle de tous les peptides après phosphorylation (c'est-à-dire ceux possédant une sérine (S), thréonine (T) et tyrosine (Y) capables d'accepter une molécule de phosphore). Dans ce cas, on pourrait avoir par exemple pour notre séquence la liste suivante. Ceci en imaginant que pour les peptides 2 et 5 de cette liste, leur masse aurait été calculée avec et sans phosphorylation :

Masses théoriques :

Peptide 1 : 600
Peptide 2 : 620 (calcul de la masse sans phosphorylation)
Peptide 2 : 700 (calcul de la masse avec phosphorylation)
Peptide 3 : 850
Peptide 4 : 900
Peptide 5 : 735 (calcul de la masse sans phosphorylation)
Peptide 5 : 800 (calcul de la masse avec phosphorylation)

Il serait alors possible de dire que nos deux peptides phosphorylés de départ dont la masse expérimentale est 700 et 800 correspondent respectivement aux peptides 2 et 5 dans la liste des masses théoriques.

Masses théoriques calculées	Masses expérimentales
Peptide 1 : 600	
Peptide 2 : 620 (calcul de la masse sans phosphorylation)	
Peptide 2 : 700 (calcul de la masse avec phosphorylation)	Peptide 2 phosphorylé : 700
Peptide 3 : 850	
Peptide 4 : 900	
Peptide 5 : 735 (calcul de la masse sans phosphorylation)	
Peptide 5 : 800 (calcul de la masse avec phosphorylation)	Peptide 5 phosphorylé : 800

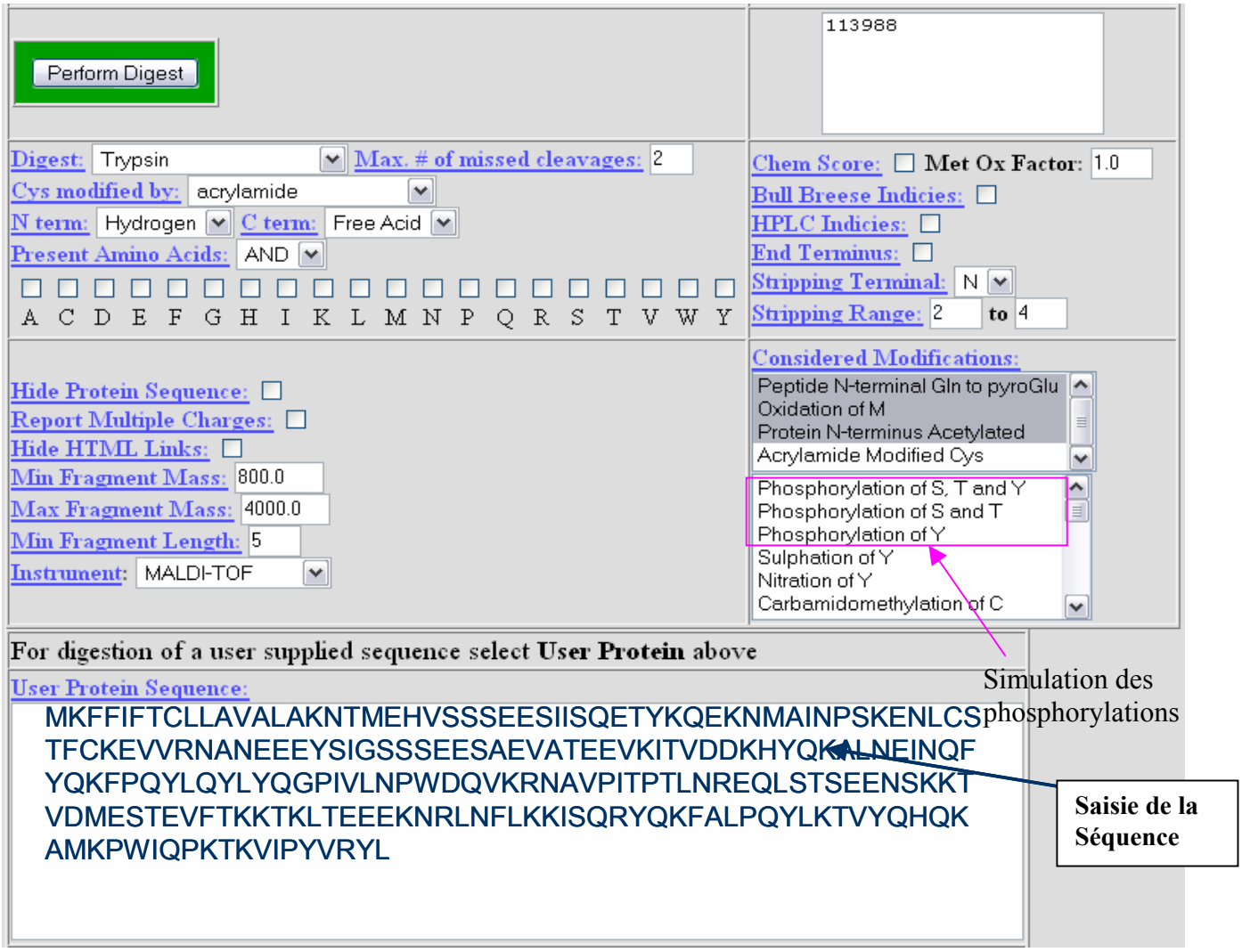
➢ Utilisation du logiciel MSDigest

Le logiciel MSDigest utilisé au laboratoire permet d'obtenir la liste des masses théoriques afin de pouvoir faire ce rapprochement.

En effet, le logiciel MSDigest est capable de réaliser une digestion virtuelle de la séquence d'une protéine et de calculer la masse de tous les peptides alors obtenus. La simulation de la digestion est possible car on sait que dans la nature, l'enzyme coupe la séquence après les acides aminés lysine (L) et arginine (A). Cependant il faut tenir compte des phosphorylations dans les séquences pour obtenir la liste complète des masses des tous les peptides de la séquence. Dans MSDigest la masse d'un peptide est calculée en réalisant la somme des masses des acides aminés qui le compose ; chaque acide aminé ayant une masse qui lui est propre (**annexe 25**). Lorsqu'un acide aminé reçoit l'ajout d'une molécule de phosphore, il voit sa masse augmenté de la valeur 79,9799 Dalton (1 Dalton = 1,660 540 2.10^{-27} kg) **[r3]**.

Exemple pour un peptide dont la séquence est la suivante : ASMDQPR
Masse du peptide non phosphorylé : 785
Masse du peptide phosphorylé : 865 (en admettant que l'acide aminé phosphorylé est la Sérine)
Pour cette raison avant de lancer la digestion virtuelle par le logiciel, il faut préciser dans les paramètres de son interface (**document 16**) que le calcul des masses des peptides après phosphorylation doit aussi être réalisé.

Document 16 : Interface de MSDigest



Document 17 : Résultat d'analyse de MSDigest

Combinaison de peptides

Number	m/z (mi)	m/z (av)	Modifications	Start	End	Missed Cleavages	Database Sequence
1	826.4228	826.9138	1PO4	215	220	0	(K)VIPYVR(Y)
1	828.3392	828.7931	1PO4	168	173	0	(K)LTEEEK(N)
1	874.4457	875.0411		40	47	0	(K)NMAINPSK (E)
1	890.4406	891.0405	1Met-ox	40	47	0	(K)NMAINPSK(E)
1	903.4688	904.0199		197	203	0	(K)TVYQHQK (A)
1	904.5369	905.0949		174	180	1	(K)NRLNFLK (K)
1	922.5110	923.0665		182	188	1	(K)ISQRYQK (F)
1	954.4120	955.0210	1PO4	40	47	0	(K)NMAINPSK(E)
1	970.4069	971.0204	1Met-ox 1PO4	40	47	0	(K)NMAINPSK(E)
1	975.5991	976.2148		213	220	1	(K)TKVIPYVR (Y)
1	977.5155	978.0942		166	173	1	(K)TKLTEEEK (N)
1	979.5617	980.2028		189	196	0	(K)FALPQYLK (T)
1	983.4352	983.9998	1PO4	197	203	0	(K)TVYQHQK(A)
1	1002.4774	1003.0464	1PO4	182	188	1	(K)ISQRYQK(F)

Masse des peptides

Phosphorylations simulées

Au final après avoir analysé une séquence de protéine avec le logiciel, les biologistes synthétisent les résultats du logiciel (**document 17**) dans un tableau. On pourrait par exemple avoir pour une séquence quelconque le résultat suivant :

Position du premier acide aminé du peptide dans la séquence

Position du dernier acide aminé phosphorylé dans la séquence

Position de l'acide aminé phosphorylé dans la séquence

Peptide	Séquence	Start	Stop	Masse	Phospho sites
1	AQYSR	1	5	789,3245	None
2	MQQK	6	9	678,9762	None
3	OMTSK	10	14	896,8654	None
4	AQYSR	1	5	815,3245	4
5	AQYSR	1	5	815,3245	3
6	AQYSR	1	5	895,3245	3 : 4
7	MQQK	6	9	693,9762	None
8	OMTSK	10	14	911,8654	None
9	OMTSK	10	14	976,8654	12
10	OMTSK	10	14	976,8654	13
11	OMTSK	10	14	1056,8654	12 : 13

Tableau 1 : Synthèse des résultats de MSDigest

En rouge, ce sont les acides aminés qui ont été phosphorylés dans la simulation. On voit bien que comme dans la nature ce sont les trois acides aminés S(Serine)T(threonine) et Y(tyrosine) qui sont dans certains cas phosphorylés. Le logiciel a simplement calculé toutes les combinaisons possibles dans ce sens et déduit les masses théoriques des peptides correspondants.

Récapitulatif :

Ainsi avec ce logiciel, il est possible d'obtenir la liste des masses théoriques de tous les peptides sans et avec phosphorylation pour les peptides qui sont potentiellement phosphorylables.

On obtient donc la liste que l'on voulait au départ pour faire le rapprochement avec nos masses expérimentales. On peut donc déterminer quels sont réellement les peptides phosphorylés dans la séquence que l'on avait au départ analysés expérimentalement :

Séquences analysée :

Peptides phosphorylés

Peptide 1	Peptide 2	Peptide 3	Peptide 4	Peptide 5

Masses théoriques calculées par MSDigest

Masses expérimentales

Peptide 1 : 600
Peptide 2 : 620 (calcul de la masse sans phosphorylation)
Peptide 2 : 700 (calcul de la masse avec phosphorylation)
Peptide 3 : 850
Peptide 4 : 900
Peptide 5 : 735 (calcul de la masse sans phosphorylation)
Peptide 5 : 815 (calcul de la masse avec phosphorylation)

Peptide 2 phosphorylé : 700

Peptide 5 phosphorylé : 815

➢ **Utilisation du logiciel NetPhos**

Dans certains cas l'utilisation du logiciel MSDigest n'est pas suffisante pour déterminer la position exacte de l'acide aminé qui est phosphorylé sur le phosphopeptide de sa séquence expérimentale. En effet, dans certains cas l'information est ambiguë. Prenons par exemple le cas du Peptide 5 et admettons qu'il ait la séquence suivante :

AQYSR

Sachant que seul les acides aminés S(Serine), T(threonine) et Y(tyrosine) peuvent être phosphorylés ont obtient avec MSDigest les 4 combinaisons suivantes pour cet acide aminé :

Résultats de MSDigest :

	Séquence	Masse
Combinaison 1 : Peptide 5	AQYSQR	735
Combinaison 2 : Peptide 5	AQYSQR	815
Combinaison 3 : Peptide 5	AQYSQR	815
Combinaison 4 : Peptide 5	AQYSQR	895

En rouge est représenté l'acide aminé phosphorylé dans le peptide. A chaque fois qu'un acide aminé est phosphorylé dans la séquence, il voit sa masse augmentée d'environ 80. Dans la combinaison 2 et 3, le nombre d'acides aminés phosphorylés est le même et donc la masse du peptide obtenue est à chaque fois identique (on a deux fois 815). Si l'on reprend notre comparaison de masses théoriques et expérimentales pour la partie concernant le peptide 5, on se retrouve alors dans la configuration suivante :

Masses théoriques calculées par MSDigest — Masses expérimentales

	Séquence	Masse	
Combinaison 1 : Peptide 5	AQYSQR	735	
Combinaison 2 : Peptide 5	AQYSQR	815	Peptide 5 phosphorylé : 815 ?
Combinaison 3 : Peptide 5	AQYSQR	815	Peptide 5 phosphorylé : 815 ?
Combinaison 4 : Peptide 5	AQYSQR	895	

Le biologiste n'est alors plus en mesure de déterminer la position de l'acide aminé phosphorylé dans la séquence. Le logiciel NetPhos (voir partie **4.2.2**) est utilisé au laboratoire pour résoudre ce problème. En effet il est capable de prédire, à partir de l'environnement en acides aminés des acides aminés S, T et Y, à partir d'une valeur seuil, s'il est possible que ces acides aminés soient phosphorylés dans la séquence. En utilisant le logiciel NetPhos on pourra par exemple prédire que la combinaison 2 est peu probable. Notre peptide 5 correspond donc à la combinaison 3. Grâce à NetPhos le biologiste aura vérifié que c'est l'acide aminé Y(Tyrosine) et uniquement celui-ci qui est phosphorylé dans sa séquence. Si l'on reprend notre analyse sur le Peptide 5 en intégrant les résultats de NetPhos, on obtient le tableau récapitulatif suivant :

Résultats théoriques de MSDigest			Résultats théoriques de NetPhos	Résultats expérimentaux
Combinaison :Peptide	Séquence du peptide	Masses théoriques	Validation par NetPhos des résultats de MSDigest	Validation expérimentale des résultats de MSDigest
Combinaison 1 : Peptide 2	AQYSQR	735	No	No
Combinaison 2 : Peptide 2	AQYSQR	815	No	No
Combinaison 3 : Peptide 2	AQYSQR	815	Yes	Yes
Combinaison 4 : Peptide 2	AQYSQR	895	Yes	No

Tableau 2 : Comparaison des résultats expérimentaux et bioinformatiques

Ainsi en reliant les résultats expérimentaux et les résultats théoriques calculés de MSDigest, et NetPhos, les biologistes parviennent à obtenir une information complète. La position des acides aminés phosphorylés (ou sites de phosphorylation) dans la séquence des protéines peut ainsi être déterminée en limitant les ambiguïtés. On obtient là l'exemple d'une complémentarité parfaite entre la démarche expérimentale biologique et le travail à l'aide des outils bioinformatiques. Cependant ce travail est long et fastidieux pour le biologiste et l'objectif est maintenant d'automatiser ce travail.

c) Solution retenue

Nous venons de voir que la démarche du biologiste intègre l'outil bioinformatique avec notamment l'utilisation du logiciel MSDigest et NetPhos. Cependant l'utilisation de ces logiciels est manuelle. Elle se fait à travers l'utilisation d'interfaces Web. Cette démarche est tout à fait acceptable lorsque le biologiste a quelques séquences à analyser. En effet lorsque l'on utilise le formulaire de soumission de MSDigest ou Netphos (partie **4.2.2**) il n'est possible d'analyser qu'une seule séquence à la fois. Ainsi la tâche devient très vite insurmontable si le nombre de séquences est élevé. Dans le cadre du projet ProteomIs l'objectif est d'analyser toutes les séquences de la base de données. Si toutes les séquences de la base doivent être analysées une par une, on parle alors, dans le jargon bioinformatique, d'une annotation manuelle des séquences. Cette situation n'étant pas satisfaisante je choisirai donc de construire une chaîne de traitements (ou pipeline) permettant l'annotation automatique des séquences contenues dans ProteomIs à l'aide de l'outil bioinformatique.

La chaîne de traitement qui est en fait un programme informatique devra inclure l'utilisation des logiciels MSDigest et NetPhos et automatiser leur travail sur toutes les séquences de ProteomIs. Le programme devra ensuite être capable d'accepter en entrée les résultats expérimentaux pour construire le tableau de synthèse vu précédemment dans l'étude de l'existant (**Tableau 2 : Comparaison des résultats expérimentaux et bioinformatiques**). Enfin les résultats d'analyses au final stockées dans ProteomIs devront pouvoir être accessible facilement par le biologiste.

Pour cela, on choisira de développer dans un premier temps des interfaces fournissant les résultats des analyses bioinformatiques sous la forme de tableaux. Dans un deuxième temps et de manière complémentaire, je choisirai de développer une interface graphique permettant de visualiser de manière schématique la position des acides aminés phosphorylés dans la séquence (partie **8.2 Perspectives**).

La solution envisagée jusque là fonctionne uniquement à partir des données de séquence stockées dans la base. Il serait également intéressant pour l'utilisateur de pouvoir utiliser la chaîne de traitements sans passer par la base de données. L'idée serait de permettre à l'utilisateur de saisir directement une ou plusieurs séquences dans un formulaire à partir duquel on pourrait activer la chaîne de traitement. Le formulaire devrait pouvoir également accepter en entrée les données expérimentales associées à ces séquences. Au final le résultat obtenu après validation du formulaire serait accessible dans le même format que prévu précédemment c'est-à-dire sous la forme d'un tableau de synthèse.

Cette manière de procéder est courante dans le domaine des applications bioinformatique d'analyse de séquence. Prenons l'exemple du logiciel InterproScan décrit dans la partie **4.2.2** et librement disponible en téléchargement. C'est également une chaîne de traitement pour la recherche de motif qui peut-être utilisée de deux manière :

- Il est d'abord possible de l'utiliser en ligne de commande pour rechercher des motifs à partir d'un grand jeu de données de séquences qui peuvent provenir d'une base de donnée. Ce logiciel a par exemple été utilisé pour réaliser l'annotation automatique de la base de donnée TrEMBL.
- Nous avons ensuite vu dans la partie **4.2.2** qu'il était possible d'utiliser InterproScan à l'aide d'une interface autorisant la saisie manuelle ou l'importation à partir d'un fichier des séquences à analyser.

5.3 Solution pour l'intégration des sources de données

5.3.1. Solution retenue

La façon dont le module GnpProt interagit avec les autres modules de GpiIS a été contraint par la solution retenue par Génoplante pour implémenter GpiIS. Nous commenterons ce choix avant de centrer la problématique sur GnpProt.

➢ **Stratégie intégrative de GpiIS : approche centralisée**

Rappelons l'objectif prioritaire du système GpiIS (décrit partie **2.2**) qui est l'intégration de sources de données provenant de thématiques biologiques différentes (transcriptome, génomique, protéome …). La solution choisie par Génoplante pour mettre en relation ces différentes données est de les intégrer au sein d'une base de donnée relationnelle (la base de GpiIS) constituée d'une collection de modules sous-jacents interconnectés et spécialisés chacun sur une thématique. Ainsi les données sont centralisées dans le système GpiIS qui est implanté sur le serveur de Génoplante Info. C'est donc la solution entrepôt de données (décrite en **4.5.3**) qui a été retenue pour concevoir le système GpiIS.

Plusieurs raisons précises ont conduit à effectuer le choix d'une gestion centralisée des données plutôt qu'à un accès direct des données sur les sites distant à partir d'un seul point d'entrée (approche non matérialisée/décentralisée/distribuée décrite en **4.5.3**). Tout d'abord, les différentes données ne sont pas organisées et stockées dans de véritables bases de données au sein des différents laboratoires. Ces données sont le plus souvent organisées dans des fichiers aux formats très hétérogènes : formats tabulaire (liste de gènes sous Excel), formats d'images (gel), formats html (résultats de spectrométrie de masse). Il est donc impossible d'imaginer pouvoir interroger et mettre en corrélation ces différentes sources de données au sein d'une interface unique sans concevoir en arrière plan un système d'information unique centralisant et gérant toutes ces données (approche entrepôt/centralisée) ou bien plusieurs systèmes d'information distribuées sur les différents sites et accédés à partir d'un seul point d'entrée (approche non matérialisée/décentralisée/distribuée).

La solution la moins coûteuse dans ces conditions est bien évidemment de concevoir un seul système d'information avec tous les avantages que cela comporte (voir partie **4.3.4**). Ensuite la politique de Génoplante est d'organiser la collecte des données provenant des différents laboratoires au travers d'un protocole de contrôle visant à garantir la qualité des données. Enfin un aspect très important concerne la confidentialité et la sécurité. En effet, certaines données sont privées et centraliser ces données, au sein d'un même entrepôt, garantit un niveau de sécurité au niveau des transactions (qui se feront via Secure Sockets Layers **[G8]**) plus élevé que lors de transactions entre des modules distribués sur Internet.

Le choix de la centralisation étant fait, en ce qui concerne l'aspect technique de la mise en place du système, la solution entrepôt de données a impliqué la mise en place de procédures garantissant une certaine homogénéité dans la conception des différents modules constituant le système GpiIS. L'utilisation du modèle relationnel a par exemple été adopté pour l'ensemble des modules (dont fait partie GnpProt). Dans la partie **6.3.2** nous présenterons les connexions qui seront effectués entre ces modules (liens entre tables, tables communes).

➢ **Stratégie intégrative de GpiIS : approche décentralisée**

Le système GpiIS dans sa forme centralisée n'offre pas de solution pour faciliter la gestion et l'exploitation des résultats expérimentaux au sein des différents laboratoires, désireux également de conserver le contrôle sur une partie de leurs données.

Pour pallier cet inconvénient, les différents modules en plus d'être intégrés dans le système GpiIS doivent pouvoir fonctionner de manière **autonome** au sein des différents laboratoires désireux de posséder une instance locale du module associé à leur thématique. C'est selon ce principe qu'une instance du module GnpProt est installée au sein des unités protéomiques de l'INRA de Montpellier et de Nantes.

Un autre des avantages du système GpiIS est de faciliter l'accès aux ressources de GpiIS notamment aux autres bases de données spécialisées en génomique végétale au travers de **services web** (voir partie **4.3.3**). Ces services web font l'objet, dans le service de Génoplante Info, d'un projet spécifique nommé PlaNet (voir partie **4.3.3**). Les versions locales de ProteomIs/GnpProt installées au sein des différentes unités devraient également pouvoir profiter de ces services web afin de pouvoir accéder aux informations biologiques des autres modules de GpiIS et les comparer avec les données protéomiques.

Ainsi le système GpiIS, en essayant de concilier les avantages d'un système centralisé et distribué, offre un bon compromis et une certaine souplesse et cohérence dans l'organisation de l'information au sein de la communauté de Génoplante.

➢ **Connexion de ProteomIs/GnpProt avec les bases de données publiques**

Une autre facette du projet GpiIS est de prévoir de mettre en corrélation les données expérimentales produites par les laboratoires de Génoplante avec les données contenues dans les bases de données publiques. Cette interopérabilité avec des bases de données externes doit être assurée au niveau de chaque module en fonction des thématiques correspondantes.

Dans le cas du module GnpProt qui nous intéresse plus particulièrement, la priorité est de pouvoir accéder au niveau d'une interface unique à des informations complémentaires sur les protéines dans les banques de données publiques. L'interopérabilité devra dans un premier temps être garantie avec les bases de données généralistes (SWISSPROT, Uniprot, Genbank, GO) et un certain nombre de bases de données spécialisées sur la plante Arabidopsis (TAIR, MatDB, Aramemnon, TIGR, PlaNet, FlagDB++). Il nous faut donc réfléchir sur la stratégie d'intégration à adopter (approche centralisée ou décentralisée) pour assurer l'intéropérabilité de ProteomIs/GnpProt avec ces bases de données publiques.

Si l'on choisit l'approche centralisée il faut extraire les données des bases de données publiques pour les importer dans la base GpiIS. L'interrogation des données reste alors locale. Maintenant si l'on choisit l'approche décentralisée, l'interrogation des données se fait à l'aide de requêtes croisées entre les données du système GpiIS et les données des banques de données publiques qui restent sur leur site distant d'origine. Les avantages et les inconvénients de chacune des deux approches ont été décrits dans la partie **4.3.4**.

Etant donné le nombre important de sources externes auxquels on veut accéder, il paraît peu envisageable de rapatrier et gérer l'ensemble des données de toutes les banques en local. L'objectif est donc de favoriser autant que possible l'accès à distance de ces données. Dans la partie **4.3.3** de ce mémoire nous avons listé un certain nombre de solutions :

- les solutions à base d'hyperliens
- les solutions à base de services web
- l'approche médiateur

Le choix dépendra de la nature des requêtes que l'on souhaite effectuer sur chacune des bases externes afin d'en évaluer la complexité et choisir la solution la plus simple et la plus adaptée. Dans le cadre de GnpProt la question prioritaire est : « Quel sont les informations disponibles pour la protéine X dans la base de donnée Y ». Pour permettre de répondre à cette question biologique on a vu dans la partie **4.3.3** de ce mémoire que la plupart des bases de données publiques sur Internet offraient la possibilité d'interroger leurs données par l'intermédiaire d'un lien hypertexte acceptant en paramètre l'identifiant du gène ou de la protéine recherchée dans la banque.

Exemple pour la protéine d'identifiant At2g14750 de ProteomIs/GnpProt le lien sur la base de donnée TAIR sur Arabidopsis est :
http://www.arabidopsis.org/servlets/Search?type=general&name=At2g14750&action=detail&method=4&sub_type=protein

Ce lien est possible car ProteomIs/GnpProt et la base TAIR se basent sur la même nomenclature qui est AGI (Arabidopsis Genome Initiative) **[G15]** au niveau des identifiants. C'est un des objectifs du programme de clustering (voir partie **5.2.1**) que de normaliser les identifiants de ProteomIs/GnpProt afin de pouvoir faire le lien sur les bases de données d'Arabidopsis qui utilisent toutes cette nomenclature AGI. Ensuite, pour l'accès aux banques de données publiques (Genbank, SWISSPROT et UNIPROT) autres que Arabidopsis et donc n'utilisant pas la nomenclature AGI, l'identifiant spécifiques de ces banques sera récupéré dans le champ référence croisée de la base ProteomIs/GnpProt.

En ce qui concerne l'accessibilité aux différents liens une section spécifique intitulée « Liens externes » est prévue dans chacune des interfaces qui renseigneront une protéine. Le résultat une fois implémenté est visible en **annexe 11** et en **annexe 12** en prenant pour exemple l'accès à FlagDB++. Dans la partie **7.2 document 34** de ce mémoire est décrit le lien sur la banque de donnée du NCBI.

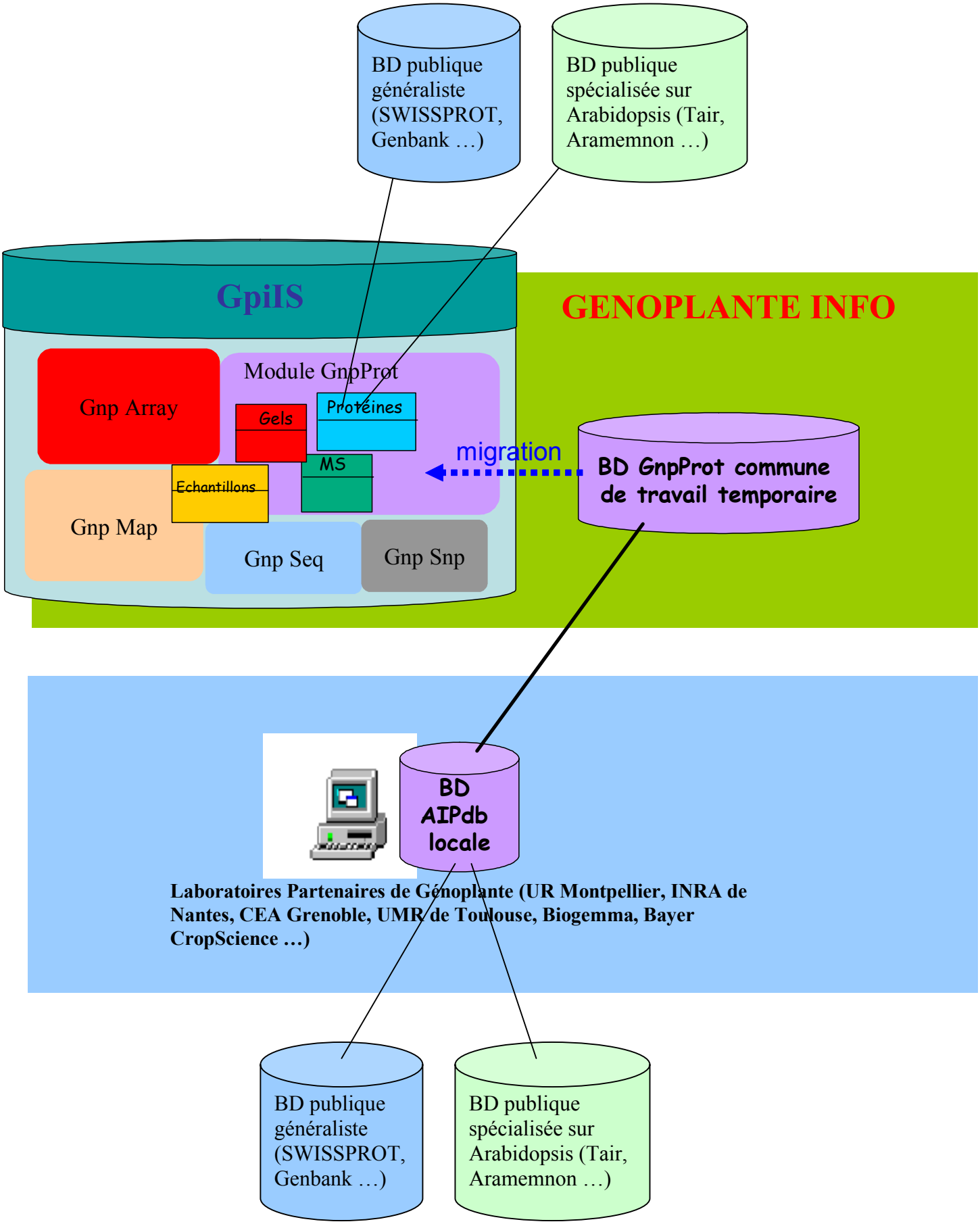
Dans la partie **8.2 Perspectives** de ce mémoire nous verrons que d'autres projets d'interopérabilité de ProteomIs/GnpProt avec des ressources externes sont prévus. Ces perspectives concernent la mise en correspondance des données sur les motifs de phosphorylation, InterproScan et SWISSPROT.

5.3.2. Organisation prévue au niveau du déploiement

La base de donnée protéome sera hébergée sur le serveur privé sécurisé de Genoplante Info sous le nom de GnpProt à Evry. Les données des différents partenaires y seront regroupées et consultables via le réseau par une ligne sécurisée par SSL **[G17]**. Pour disposer d'un accès à la base sur ce site privé **[i137]** les utilisateurs doivent disposer d'un compte à Génoplante qui leur fournit un login et un password. Néanmoins certains partenaires pourront bénéficier d'une version locale de la base sous Postgres pour leurs propres données uniquement (**document 18**). Sur le site privé de Genoplante Info il sera possible d'utiliser toutes les ressources bioinformatiques disponibles.

Après un temps d'hébergement de 6 mois sur la base GnpProt du site privé, les données doivent ensuite être transférées sur une instance de la base GnpProt qui sera placé sur le site public de Génoplante **[i138]** ouvert à tout l'internet. L'objectif à terme est que les données protéomiques soient transférées sur la base intégrative GpiIS (voir partie **2.2**) comprenant à la fois la base GnpProt (sous forme de module protéomique GnProt) mais aussi les autres modules de Génoplante (GnpSeq, GnpMap, GnpArray, GnpSNP). Concernant la base protéome aussi bien pour la version intégrée (GnpProt) que non intégrée (ProteomIs) les données devront pouvoir être croisées avec celles des banques de données publiques externes.

Après avoir présenté l'existant et une vue synthétique des objectifs de notre projet la suite de ce mémoire sera consacrée aux aspects analyse et conception de ProteomIs/GnpProt. Il s'agira dans la partie conception de présenter et justifier les choix d'implémentation et architecturaux, ainsi que les technologies à mettre en place pour chacune des fonctionnalités applicatives prévues dans l'analyse.



<u>Document 18 :</u> Intéropérabilité de ProteomIs/GnpProt avec les autres modules de GpiIS et les banques de données publiques

6 Analyse et conception

6.1 Choix du processus de développement

➢ La problématique du développement d'un logiciel :

La construction d'un logiciel est complexe car elle met en œuvre de nombreuses ressources humaines, matérielles et technologiques.

Bien que le nombre de personnes travaillant sur le projet ProteomIs/GnpProt soit assez réduit, le projet est suffisamment complexe pour qu'il soit nécessaire de suivre un processus de développement bien défini correspondant au cycle de vie d'un logiciel dans lequel il faudra :

• prévoir et planifier les travaux ;
• coordonner les activités de conception, de fabrication, de validation, …
• réagir à l'évolution des objectifs.

Il faut également utiliser une méthode rigoureuse basée sur des modèles. Ces modèles étant des représentations sémantiques simplifiées mais justes du système visant à l'analyser et le comprendre pour mieux le concevoir.

➢ Le choix du processus de développement

L'un des plus importants débats à propos des processus est celui qui oppose le style en cascade et le syle itératif. La différence essentielle entre les deux réside dans la façon de décomposer un projet.

Le **style en cascade** décompose un projet en phases distinctes sur le principe du non-retour : analyse des besoins, conception, codage et tests. L'inconvénient de cette démarche est que le contrôle significatif survient en fin de projet, et, à ce moment là, si l'utilisateur s'aperçoit que le système ne répond pas correctement aux besoins exprimés, il est souvent trop tard pour recommencer.

Le **style itératif** décompose un projet en extrayant des sous-ensembles de fonctionnalités. On décomposera par exemple un projet d'un an en itérations de trois mois. La première itération traitera un quart des exigences et verra se dérouler la totalité du cycle de vie du développement logiciel : analyse, conception, codage et tests. A la fin de cette première itération, on dispose d'un système qui assure un quart des fonctionnalités requises. Puis on effectue une deuxième itération : au bout de six mois, le système répond à la moitié des besoins.

On peut très bien ne pas mettre le système en production à la fin de chaque itération, mais celui-ci doit être de qualité équivalente. Néanmoins, il arrive souvent que l'on puisse mettre le système en production à intervalles réguliers, ce qui est une bonne chose : on crée ainsi de la valeur ajoutée plus tôt, et on obtient un feed-back de meilleure qualité. Dans ce cas, on dit souvent que le projet a plusieurs **versions,** chacune étant réalisée en plusieurs **itérations**.

C'est ce processus de développement qui a été utilisé dans le cadre du projet ProteomIs/GnpProt à travers l'utilisation de certains principes tirés de la méthode UP (Unified Project). Le processus unifié est un processus de développement logiciel itératif, centré sur l'architecture, piloté par des cas d'utilisation et orienté vers la diminution des risques.

De manière conjointe le projet a été modélisé à l'aide de la notation UML qui est intégrée dans UP. UML étant simplement une notation et non une méthodologie d'analyse, UP a été utilisé pour guider la modélisation.

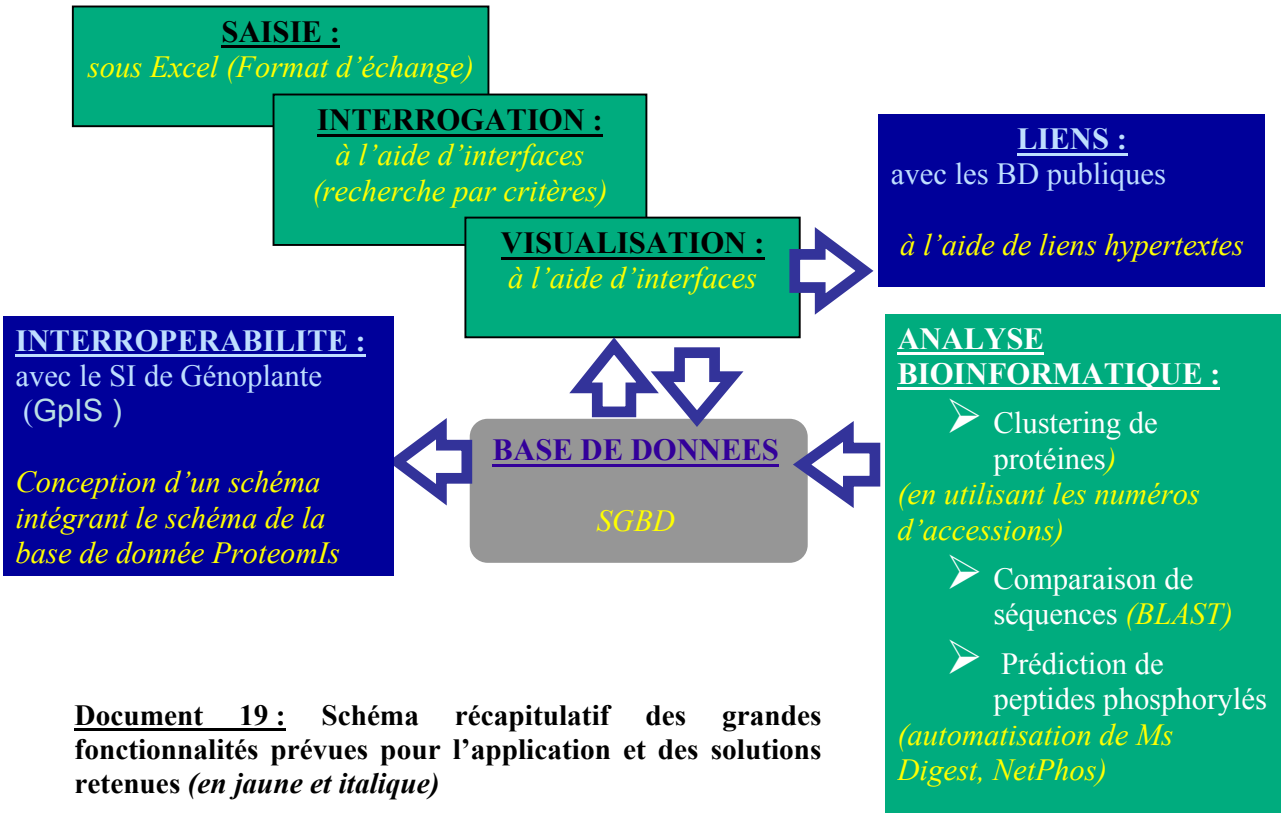
6.2 L'analyse des besoins au niveau global

6.2.1 Récapitulatif des besoins (ou cahier des charges)

Nous proposons de préciser ici les grandes lignes du cahier des charges de l'application. Les objectifs du projet (décrit en **2.2**) stipulent que l'application doit pouvoir être déclinable sous deux formes :

- La première correspond à un module dédié à la gestion des protéines, GnpProt, à intégrer au sein du système d'information global GpiIS.
- La seconde correspond à une application de gestion locale nommée ProteomIs dotée de facilités d'installation et de maintenance pour les laboratoires partenaires.

Le chapitre **5** précédent nous a donné une idée des principales fonctionnalités attendues et des solutions retenues rappelées dans le **document 19** ci-dessous.



Document 19 : Schéma récapitulatif des grandes fonctionnalités prévues pour l'application et des solutions retenues *(en jaune et italique)*

➢ **La saisie des données :**

Avant d'être importées dans la base, les données devront être saisies dans un classeur Excel appelé aussi « Format d'échange ».

➢ **L'interrogation et la visualisation des données :**

L'interrogation et la visualisation des données s'effectueront à travers des interfaces connectées à une base de données ; la gestion des données se faisant à l'aide d'un SGBD. A partir de ces interfaces, des liens devront permettre de retrouver les informations correspondantes sur les protéines dans les banques de données publiques. Une option supplémentaire permettra à l'utilisateur d'exporter les données contenues dans certaines interfaces de visualisation. L'exportation devra pouvoir se faire à la fois au format CSV et dans Excel afin que les biologistes puissent facilement retravailler ces données.

➢ **L'analyse bioinformatique des séquences de protéines :**

Les séquences de protéines contenues dans les banques de données publiques doivent être importées régulièrement dans la base. C'est sur la base de ces séquences qu'il pourra être possible d'effectuer un certain nombre d'analyses.

Ces analyses pourront aller de la simple **comparaison de séquences** à l'aide du logiciel BLAST jusqu'à la **recherche de motif** de phosphorylations à l'aide des logiciels MSDigest et NetPhos.

Toujours au niveau de l'analyse des données, il est question de créer des groupes non redondants de protéines (« **clustering** »). Il s'agit là d'effectuer un regroupement des protéines similaires contenues dans la base de données en se basant sur les équivalences entre numéros d'accessions. Un identifiant unique devra être attribué à ces groupements (« clusters ») de protéines, cet identifiant étant basé sur la nomenclature AGI pour les protéines d'Arabidopsis.

➢ **L'interopérabilité :**

Enfin au niveau du module GnpProt, l'interopérabilité avec le système d'information (GnpIS) développé à Génoplante est garantie grâce à la conception d'un schéma intégré. De plus la mise en place de la nomenclature AGI pour identifier de manière unique les protéines d'Arabidopsis va contribuer à cette interopérabilité en permettant de faire le lien avec les gènes décrits dans le système GpiIS et identifiés également par des numéros d'accessions AGI.

Enfin en terme d'interopérabilité, grâce à la mise en place de la nomenclature AGI, il sera possible d'accéder à des informations complémentaires sur les protéines dans les banques de données publiques sur Arabidopsis qui utilisent également cette nomenclature AGI (TAIR, Aramemnon, Matdb, …).

Bien sûr, ce cahier des charges préliminaires a été discuté dans le détail avec les futurs utilisateurs. En **annexe 15** est présenté l'exemple d'un questionnaire qui a permis de dégager les besoins des différents laboratoires partenaires du projet (présenté en **2.1.3**).

L'envoi de ces questionnaires a précédé une réunion générale qui a lieu à Evry en juin 2002 et qui rassemblait les différentes personnes concernées afin de valider les grandes lignes du cahier des charges de l'application.

6.2.2 Spécifications des exigences d'après les cas d'utilisation « le QUOI »

Dans la suite, nous utiliserons les concepts UML fondamentaux pour la spécification des exigences, à savoir les acteurs et les cas d'utilisation. Nous allons les identifier à partir de l'expression initiale des besoins de notre étude (partie **6.2.1** précédente).

➢ Identification des acteurs :

Les acteurs sont spécialisés en acteurs humains et acteurs électroniques. Pour ajouter une information supplémentaire on distinguera les acteurs externes au système des acteurs internes au système. Trois acteurs humains ont été répertoriés :

Acteurs externes :

- Le producteur de données : C'est le scientifique réalisant les analyses protéomiques et chargé de saisir des données dans la base. Pour simplifier on considèrera que cette personne est également propriétaire des données contenues dans la base. Il y aura en fait plusieurs producteurs de données puisque la base est susceptible d'accueillir des données en provenance de plusieurs laboratoires de recherche
- L'utilisateur du système

Acteurs internes :

- Le gestionnaire de données : c'est la personne en charge de la gestion des données dans l'ensemble de l'application. Elle peut intervenir aussi bien sur la mise à jour des données dans la base que sur le lancement de procédures d'analyses bioinformatiques des données. Elle sera également responsable de l'accès aux données.

Huit acteurs électroniques ont été ensuite définis :

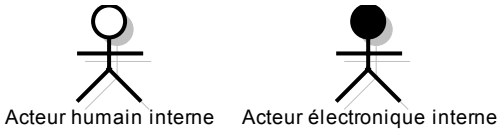
Acteurs externes :

- Banques de données publiques
- GpiIS (le système d'information de Génoplante)

Acteurs internes

- SGBD
- Interfaces
- Programmes : peut représenter tout programme informatique qui n'est pas destiné à la réalisation d'une interface mais plutôt à un traitement spécifique de l'information biologique.
- Logiciels bioinformatiques (BLAST, Netphos …).

Nous utiliserons une couleur différente dans les diagrammes pour distinguer les acteurs électroniques des acteurs humains :



➢Identification des cas d'utilisation :

Pour chaque acteur identifié précédemment, il convient de rechercher les différentes interactions qu'il peut avoir avec le système ; ces interactions étant modélisées par les cas d'utilisations qui représentent également les services rendus par le système.

Commençons par l'acteur le plus important : l'utilisateur. Ses cas d'utilisations principaux ont été bien mis en évidence par l'expression des besoins préliminaires, à savoir :

- *« Consulter les données »*
- *« Procéder à des analyses bioinformatiques »*
- *« Mise à jour des données* »

Le cas d'utilisation *« Mise à jour des données* » est en relation avec le producteur de données externe qui va saisir les données dans le format d'échange. Le gestionnaire des données pourra intervenir sur ce cas d'utilisation en procédant à des mises à jour ou des suppressions d'enregistrements dans la base. C'est également lui qui lancera le programme d'importation des données contenues dans le format d'échange dans la base. Le gestionnaire des données pourra intervenir également sur le cas d'utilisation *« Consulter les données »* car il fournira l'autorisation d'accès aux données. Il interviendra également sur l'analyse bioinformatique des données car il a pour tâche de lancer certains traitements : clustering de protéines, importation des séquences …

➢Relations entre cas d'utilisation :

Les deux cas d'utilisation de l'utilisateur (*« mise à jour des données »* et *« analyses bioinformatiques des données »*) sont assez naturellement reliés par une relation d'inclusion au troisième cas d'utilisation *« consultation des données »*. Une relation d'inclusion formalisée par un mot-clé *« include »* spécifie que le cas d'utilisation de base en incorpore explicitement un autre de façon obligatoire. L'*« analyse bioinformatique des données »* est également dépendante du cas d'utilisation *« mise à jour des données »*.

➢Présentation du diagramme de cas d'utilisation général préliminaire :

A ce stade nous pouvons présenter sur le **diagramme 1**, le diagramme de cas d'utilisation préliminaire qui exprime les concepts énoncés précédemment. On peut voir également apparaître les interactions supplémentaires entre les acteurs électroniques et le système. Nous avons réalisé un seul diagramme d'utilisation qui prévaut à la fois pour GnpProt et ProteomIs. La seule restriction concerne l'interaction du système avec l'acteur GpIS non valide dans le cadre du module GnpProt. Enfin nous avons représenté les trois cas d'utilisation avec trois couleurs différentes :

- jaune pour le cas d'utilisation « Mise à jour des données »
- bleu pour le cas d'utilisation « Consulter les données »
- vert pour le cas d'utilisation « Analyse bioinformatique des données »

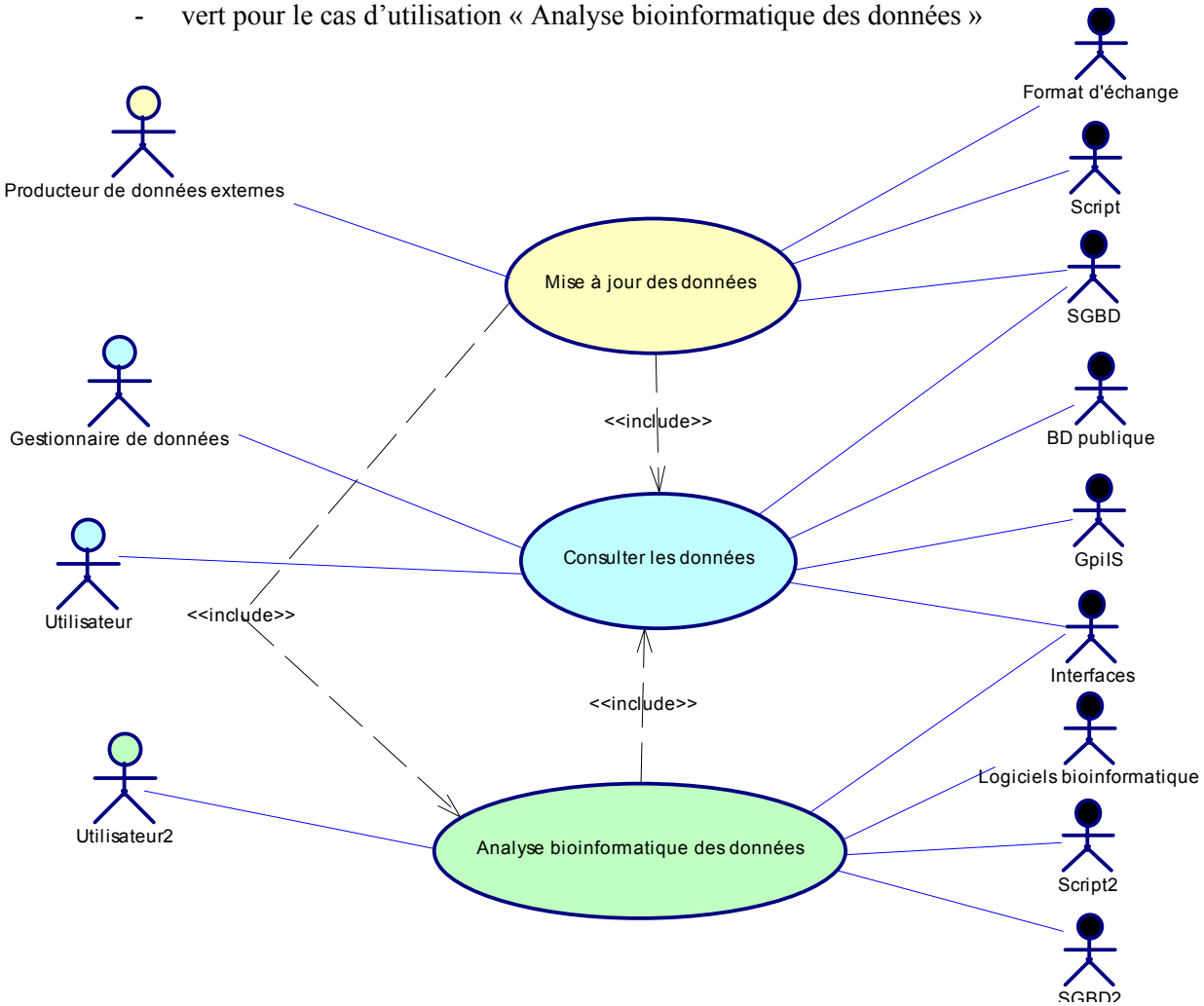


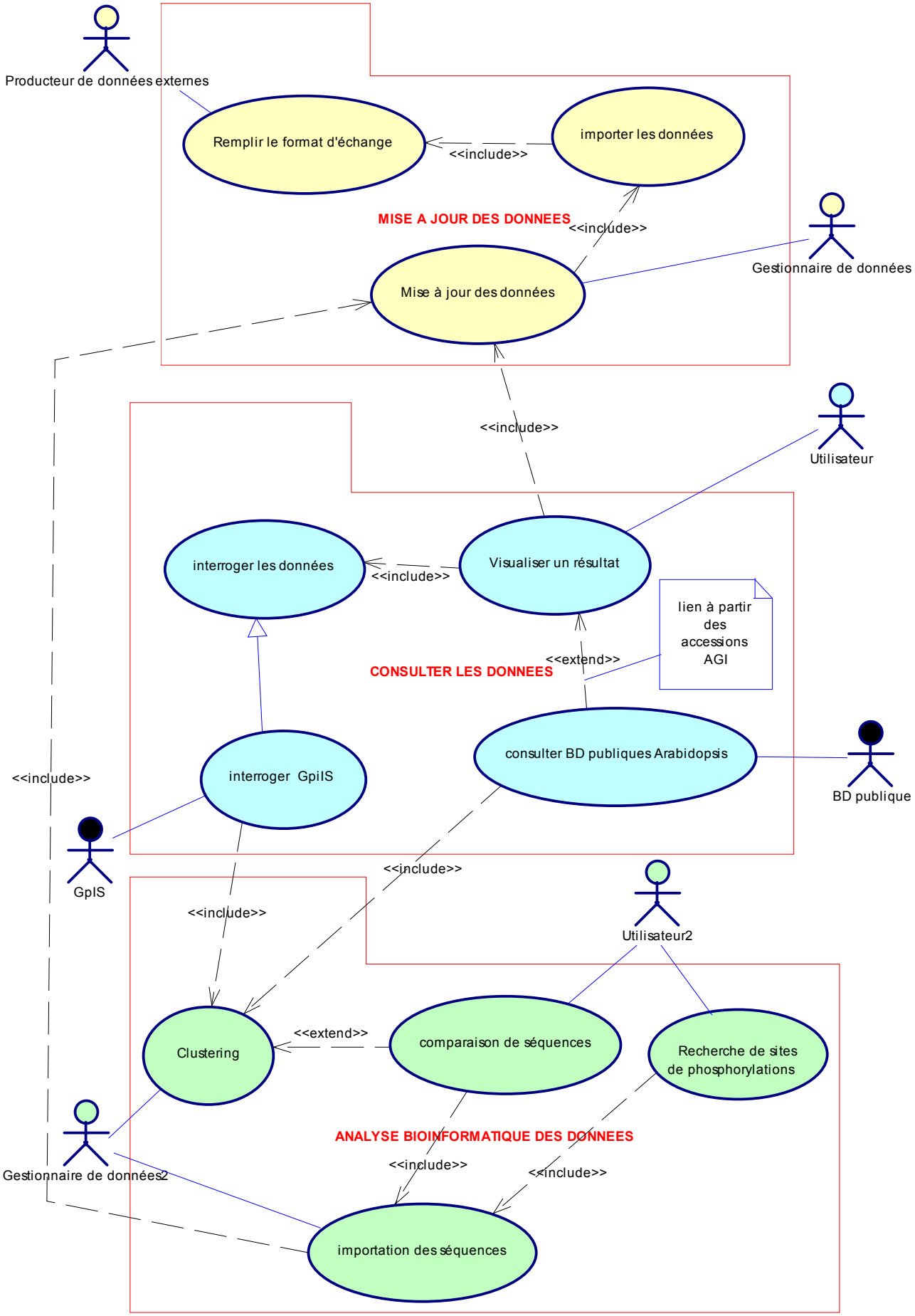
Diagramme 1 : diagramme de cas d'utilisation préliminaire

Ce code couleur sera utilisé par la suite pour mettre en correspondance les diagrammes UML correspondant à une analyse de ces trois cas d'utilisation. Les acteurs humains seront colorés de la même façon que le cas d'utilisation avec lequel ils interagissent.

➢Présentation du diagramme de cas d'utilisation général détaillée :

Nous détaillons le modèle précédent dans le **diagramme 2**, sans en faire apparaître les acteurs internes machines, par souci de lisibilité.

Diagramme 2 : **Diagramme de cas d'utilisation général détaillé**



Dans ce nouveau modèle chacun des trois cas d'utilisation du diagramme précédent est décomposé. Pour améliorer notre modèle, nous avons organisé et regroupé les cas d'utilisations obtenus en ensembles fonctionnels cohérents. Pour ce faire, nous utilisons le concept général d'UML qui s'appelle le « *paquetage » (ou package en anglais)*. Nous avons 3 packages correspondant pour chacun au détail d'un des cas d'utilisation du diagramme précédent. On va faire apparaître sur ce diagramme les relations d'extension qui apportent un niveau de détail supplémentaire. Une relation d'extension est représentée par son stéréotype *« extends »* : le cas d'utilisation de base en incorpore implicitement un autre, de façon optionnelle. Par exemple sur notre diagramme le cas d'utilisation *« clustering de protéines »* peut être complété par l'utilisation du logiciel BLAST de comparaison de séquences.

Le niveau de détail présenté dans ce diagramme correspond à celui spécifié dans le cahier des charges (en **6.2.1**). L'avantage est qu'il détaille les relations de dépendances entre les trois cas d'utilisation principaux : *« saisie des données »*, *« consultation des données »* et *« analyse bioinformatique des données »*.

A noter que le cas d'utilisation *« consulter des informations complémentaires sur les protéines dans GpIS »* n'est valable que pour la version intégrée de ProteomIs (module GnpProt).

➢Analyse par cas d'utilisation

Cette partie de l'analyse apporte un niveau de détail supplémentaire. Elle est présente en tant que document technique en **annexe 1**.

Dans cette partie, conformément au cadre de la méthode UP pilotée par les cas d'utilisation, nous avons poursuivi notre analyse en nous limitant à chaque fois au domaine représenté par chacun des trois cas d'utilisation présentés dans le diagramme de cas d'utilisation général préliminaire.

Ces trois cas d'utilisation qui ont guidé guideront notre analyse sont :

- le cas d'utilisation « Saisie des données »
- le cas d'utilisation « Consulter les données »
- le cas d'utilisation « Analyse bioinformatique des données »

Pour chacun de ces cas d'utilisation nous avons procédé de la manière suivante :

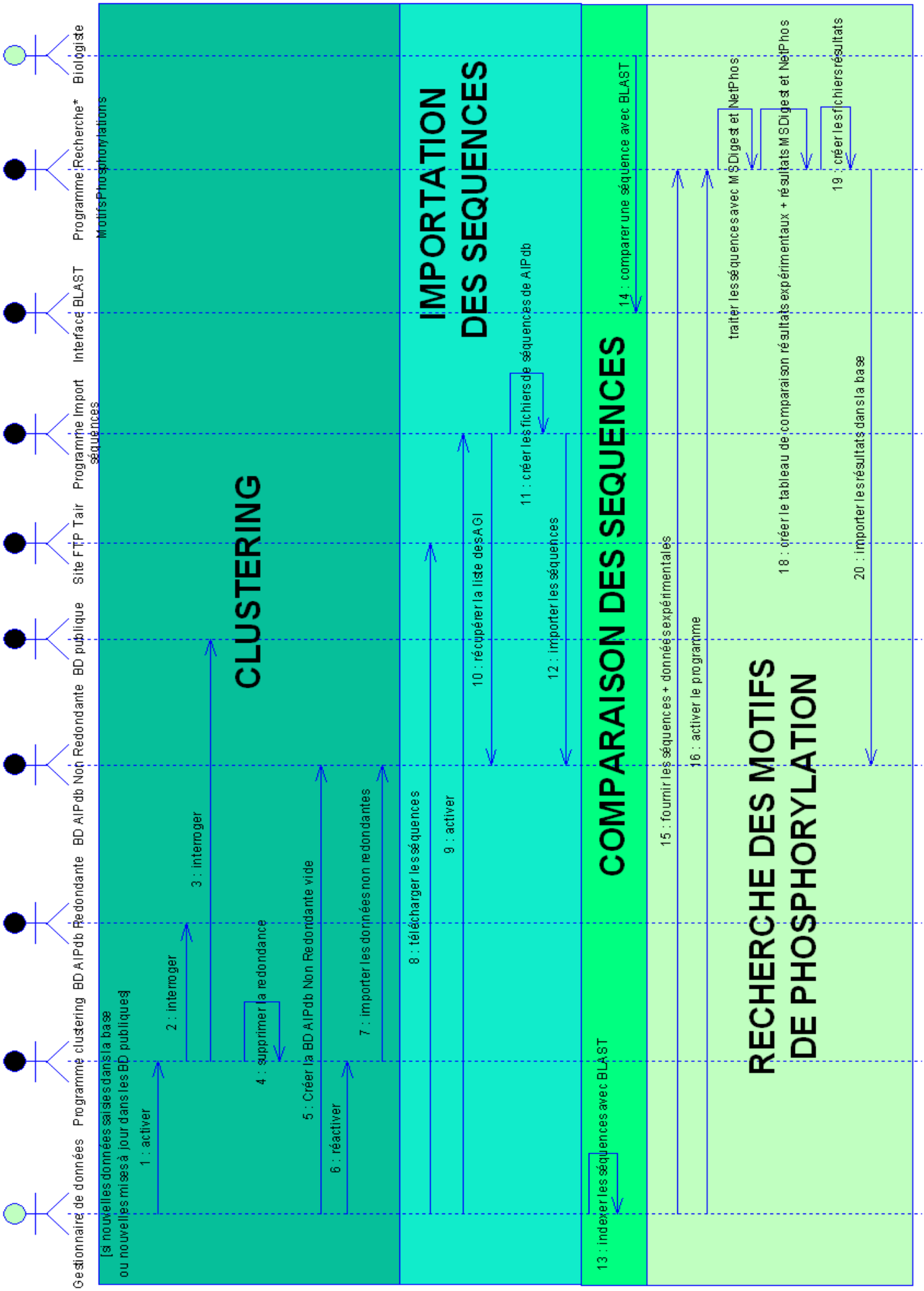
- l'expression des besoins a été affinée de manière textuelle et illustrée par un ou plusieurs diagrammes de cas d'utilisation
- ce travail a été complété par une vue « dynamique » du système représentée par un graphique UML très utile : le diagramme de séquence « système ».

Les résultats de l'analyse statique (partie **6.3** du Mémoire) ont alors été très utiles.

6.2.3 Synthèse et chronologie des différentes étapes de l'analyse des données

On se propose de réaliser ici la synthèse des principaux traitements bioinformatiques mis en évidence dans l'étape précédente. Pour apporter une vision chronologique des différentes étapes on utilise un diagramme de séquence (**diagramme 3**). Nous n'avons pas représenté les flèches de retour des actions pour ne pas encombrer le diagramme qui doit fournir une vue synthétique.

Diagramme 3 : **Diagramme de séquence récapitulatif : Analyse bioinformatique des données**



Voici quelques commentaires concernant les principales étapes représentées dans ce diagramme :

❶ clustering :

La partie réservée au clustering décrit toute la procédure permettant de faire migrer les données d'une version redondante de la base ProteomIs vers une version non redondante de cette base. Les différentes phases de la procédure seront automatisées au travers d'un programme informatique. Le gestionnaire de données est responsable de l'exécution de ce programme que l'on appellera programme de clustering.

Le gestionnaire de données devra également lancer l'exécution du programme chaque fois que la base ProteomIs aura été mise à jour avec de nouvelles données des producteurs de données. En fait lorsque la base passe dans l'état non redondant, elle devient à nouveau une base redondante à chacune des mises à jour par les données du format d'échange.

❷ importation de séquences :

Le programme de clustering précédent nous a permis de normaliser l'ensemble des numéros d'accession des protéines de ProteomIs au format AGI. Pour chacun de ces numéros d'accessions, l'objectif va maintenant être d'importer la séquence correspondante au format FASTA dans la base de donnée ProteomIs. Les séquences ainsi rapatriées en local pourront être accessibles par les programmes de comparaison de séquences et recherche de motifs.

Ces séquences peuvent être récupérées dans la banque de données TAIR via le site FTP : ftp://ftp.arabidopsis.org/home/tair/. Le format FASTA des séquences est un des plus répandus dans le monde de la bioinformatique. Il s'agit d'un format textuel simple qui comprend une ligne d'en-tête identifiable par le symbole « > » et qui précède les informations caractéristiques de la séquence. Sur cette ligne, se trouve le numéro d'accession AGI qui permet d'identifier la séquence de manière unique. Ce numéro d'accession est extrêmement utile puisqu'il va permettre d'établir des correspondances avec les séquences contenues dans ProteomIs.

Les lignes suivantes sont dévolues à l'enchaînement protéique de la séquence. Un exemple de séquence protéique au format FASTA issu du site FTP de Tair est donné ci-dessous :

```
>At1g01010.1 68414.m00001 no apical meristem (NAM) family protein contains Pfam PF02365: No
apical meristem (NAM) domain; similar to NAC domain protein NAM GB: AAD17313 GI:4325282 from
[Arabidopsis thaliana]
MEDQVGFGFRPNDEELVGHYLRNKIEGNTSRDVEVAISEVNICSYDPWNLRFQSKYKSRD
AMWYFFSRRENNKGNRQSRTTVSGKWKLTGESVEVKDQWGFCSEGFRGKIGHKRVLVFLD
GRYPDKTKSDWVIHEFHYDLLPEHQRTYVICRLEYKGDDADILSAYAIDPTPAFVPNMTS
SAGSVVNQSRQRNSGSYNTYSEYDSANHGQQFNENSNIMQQQPLQGSFNPLLEYDFANHG
```

Ce sont ces séquences qui seront importées dans la base ProteomIs pour mettre à jour la table séquence (voir partie **6.3.2 Le modèle de données**).

❸ comparaison de séquences :

Les séquences importées dans la base ProteomIs vont ensuite servir de base de confrontation pour les comparaisons de séquences avec l'outil de recherche de similarité BLAST.

Cette collection de fichiers devra être prétraitée au préalable par des techniques d'indexation. En effet l'outil BLAST utilise des structures à base d'index pour rendre les recherche de similarité quasi-instantanées. Nous conservons la banque de données à la fois sous sa forme originelle (collection de fichiers FASTA) qui nous sera précieuse pour d'autres traitements (recherche de motifs par exemple) et sous sa forme indexée (uniquement pour BLAST).

❹ recherche de motifs :

D'après le cahier des charges, la priorité est de se concentrer sur la recherche de motifs correspondant aux sites de phosphorylation. La solution retenue pour cette fonctionnalité a été présentée en **5.2.3**. En résumé deux versions de ce programme ont été envisagée :

- première version du programme orientée base de donnée : Elle a pour objectif d'analyser toutes les séquences contenues dans la base ProteomIs et d'importer les résultats dans la base. C'est le fonctionnement de cette version qui est décrit dans le diagramme de séquence précédent. Il faut préciser au niveau du diagramme que la recherche de motif ne dépend pas des résultats du BLAST et nécessite uniquement les étapes préalables de clustering et d'importation des séquences.
- deuxième version du programme orientée utilisateur : Elle se présente sous la forme d'un formulaire capable d'accepter en entrée les séquences au format FASTA que lui fournit l'utilisateur. Ces séquences seront analysées de la même manière que dans la première version du programme.

6.2.4 Calendrier des réalisations et personnel impliqué : "le QUI et le QUAND"

Je me propose dans cette partie de décrire :

- l'ordre dans lequel les différentes tâches aboutissant à l'implémentation des cas d'utilisation ont été effectuées
- comment en tant que chef de ce projet, j'ai pu répartir le travail entre les différents membres de l'équipe
- dans quelle mesure je me suis investi sur les différents aspects du projet.

Le **tableau 3** illustre ces différents aspects.

Mon affectation sur ce projet au sein de l'unité protéomique de l'INRA de Montpellier a pris effet le 1er avril 2002. A mon arrivé dans le laboratoire, j'ai été chargé de faire **un état de l'existant** que cela soit au niveau du laboratoire ou au niveau des projets, ressources de bioinformatiques ou de la littérature scientifique.

J'ai également pris connaissance de la démarche expérimentale utilisée par les techniciens et ingénieurs du laboratoire de protéomique. Ma position au sein de ce laboratoire et ma double compétence biologie-informatique me permettait une approche privilégiée dans la définition des besoins.

J'ai pu ainsi réaliser un **cahier des charges** (présenté en **6.2.1**) et faire l'inventaire des données (partie **6.3.1**). J'ai alors réalisé un **modèle conceptuel des données** de la base de données qui sera présenté dans la partie suivante en **6.3.2**. J'ai également constitué une **maquette** interactive des interfaces sous PowerPoint (présentée en **6.4.2**) en vue d'une présentation aux usagers potentiels.

Les différentes versions de ces documents ont été confrontées aux futurs utilisateurs répartis dans les laboratoires de Montpellier mais aussi dans les laboratoires de Nantes, Toulouse et Grenoble. Etant donné l'éloignement géographique, les échanges se sont fait le plus souvent par téléphone ou par mail avec cependant deux réunions qui ont eu lieu à Evry en juin 2002 et fin 2002, réunions au cours desquelles j'ai pu présenter les différentes versions de mes documents. Au niveau de chaque laboratoire, la discussion s'est faite la plupart du temps par l'intermédiaire d'interlocuteurs privilégiés représentant chacun des laboratoires et présentés dans la partie **2.1.3**. Mon interlocuteur le plus proche pour l'évaluation des besoins fut Michel Rossignol.

Ma collaboration avec les informaticiens de Génoplante Info fut également indispensable pour concevoir le schéma de manière à ce qu'il réponde aux contraintes d'interopérabilité avec les autres modules du système de GpIS de Génoplante.

Les informaticiens de Génoplantes Info à Evry avec lesquels j'ai été amené à étudier cette problématique sont : Delphine Samson et Delphine Grando.

J'ai ensuite été impliqué dans la réalisation du **fichier Excel de soumission des données**. Le développement des scripts Perl et de chargement des données dans la base a ensuite été entièrement réalisé par Thierry Hotelier.

Parallèlement j'ai entrepris le développement de **l'interface de visualisation des gels** puis celui des autres **interfaces de consultation des données**. J'ai été la principale personne investie sur les aspects conception et implémentation de ces interfaces. J'ai eu cependant des contacts réguliers avec Guillaume Albini (ancien CDD informaticien à Génoplante) puis Farid Chatouani qui est ingénieur informaticien à Génoplante et responsable des aspects interfaces du système GpiIS.

Une aide m'a également été apporté par Arnaud Nemrod que j'ai encadré dans le cadre de son stage d'IUP. Arnaud a travaillé du 15 avril 2003 au 30 août 2003 sur :

- le développement des interfaces de ProteomIs : design des interfaces et implémentation
- l'étude des différentes solutions existantes en matière de persistance dans la couche d'accès aux données

La première version de ProteomIs/GnpProt a pu être livrée le 4 avril 2004 à Génoplante permettant la saisie et la consultation des données.

La deuxième étape consistait alors à développer les **outils d'analyse des séquences protéiques** contenues dans la base.

Pour développer ces outils j'ai bénéficié de l'aide de deux stagiaires :

1) De juin 2004 à septembre 2004, Mohamed Ndiaye (étudiant en DESS « Informatique Appliquée aux Organisations » à l'Université Montpellier 2), a travaillé sur :

- la problématique du « **clustering** des protéines »
- la **comparaison de séquences** avec BLAST

Le stage a été suivi en parallèle par Isabelle Mougenot.

2) De juin à août 2004, Cyril GENIN (étudiant en école d'ingénieur à EPSI à Montpellier) a travaillé sur la **recherche de motifs**.

Une deuxième version de ProteomIs/GnpProt a été livrée le 12 décembre 2004 à Génoplante. Dans cette version, les nouveautés dans cette version concernaient essentiellement des compléments techniques permettant notamment d'améliorer la montée en charge de l'application (voir partie **7.4**).

<u>Tableau 3 :</u> Diagramme de GANTT des tâches/réalisations et répartition des moyens humains autour du projet ProteomIs/GnpProt

Réunion 1 Evry *(présentation cahier des charges aux partenaires)* ▽ — Réunion 2 Evry *(travail sur le MCD)* ▽ — Réunion 3 Evry *(présentation maquettes aux partenaires)* ▽

Tâches/Réalisations	auteurs	acteurs	Avril 02	Mai 02	Juin 02	Juil 02	Août 02	Sept 02	Oct 02	Nov 02	Dec 02
veille technologique et sensibilisation aux techniques expérimentales	C.Bouttes										
cahier des charges, analyse	C.Bouttes	M.Rossignol									
conception MCD et implémentation BD	C.Bouttes	D.Sansom, D.Grando									
maquette des interfaces	C.Bouttes										
format d'échange	C.Bouttes, T.Hotelier										
scripts d'importation des données du format d'échange	T.Hotelier										
interface de visualisation des gel	C.Bouttes										

Réunion 4 Evry *(présentation architecture)* ▽

Tâches/Réalisations	auteurs	acteurs	Janv 03	Fev 03	Mars 03	Avr 03	Mai 03	Juin 03	Juil 03	Août 03	Sept 03	Oct 03	Nov 03	Dec 03
scripts d'importation des données du format d'échange	T.Hotelier													
interfaces de consultation des données	C.Bouttes, A. Nemrod	Guillaume Albini				↔ stage Arnaud Nemrod								

Version 1 ProteomIs/GnpProt ▽ — Version 2 ProteomIs/GnpProt ▽

Tâches/Réalisations	auteurs	acteurs	Janv 04	Fev 04	Mars 04	Avr 04	Mai 04	Juin 04	Juil 04	Août 04	Sept 04	Oct 04	Nov 04	Dec 04
interfaces de consultation des données	C.Bouttes	F.Chatouani												
programme de clustering et comparaison de séquences	C.Bouttes, M. Ndiaye							↔ stage M.Ndiaye						
programme de recherche de motifs de phosphorylation	C.Bouttes, C.Genin							↔ stage C.Genin						

Installation version 2 de ProteomIs/GnpProt sur le site de Génoplante ▽

Tâches/Réalisations	auteurs	acteurs	Janv 05	Fev 05	Mars 05
interfaces de consultation des données	C.Bouttes,	F.Chatouani			

6.3 Aspect statique : conception du modèle de données

Nous décrirons ici la démarche qui nous a permis d'aboutir à la construction du modèle conceptuel de données à savoir les classes, les associations et les attributs à partir desquels nous implémenterons les tables de notre future base de données.

6.3.1 Démarche utilisée

Resituons l'activité de modélisation des données dans l'ensemble du processus d'analyse. L'expression préliminaire des besoins a donné lieu assez directement à une modélisation par les cas d'utilisation. Il s'agissait là principalement d'une description fonctionnelle assez générale des besoins par opposition à la description maintenant structurelle, statique, du système que nous allons devoir réaliser avant de concevoir la base de données.

Les cas d'utilisation présentés précédemment et bien que préliminaires vont nous aider à cerner les limites du domaine à modéliser, en ce sens, que c'est le modèle conceptuel des données qui va servir de support à l'ensemble des traitements associés à ces différents cas d'utilisations.

Si l'on reprend maintenant le diagramme de cas d'utilisation général préliminaire (**diagramme 1**), l'acteur producteur de données est l'élément majeur autour duquel la base de données sera construite. En effet, c'est lui qui alimentera principalement en informations cette base et qui pourra définir le mieux les entités qui doivent en ressortir.

Ainsi, j'ai dû comprendre et assimiler son travail en vue de modéliser sa méthodologie par un diagramme d'activité (**diagramme 4**). Ce dernier va nous permettre de cerner au mieux une grande partie des données à stocker dans la base.

J'ai donc assisté à l'ensemble du processus d'une expérience : la préparation des échantillons, la coloration et le coulage des gels, l'analyse d'images, la découpe des spots, la digestion trypsique et l'extraction des protéines, l'identification des protéines par spectrométrie de masse et la recherche des protéines dans les bases de données à l'aide du logiciel Mascot (partie **3.4 L'analyse protéomique**).

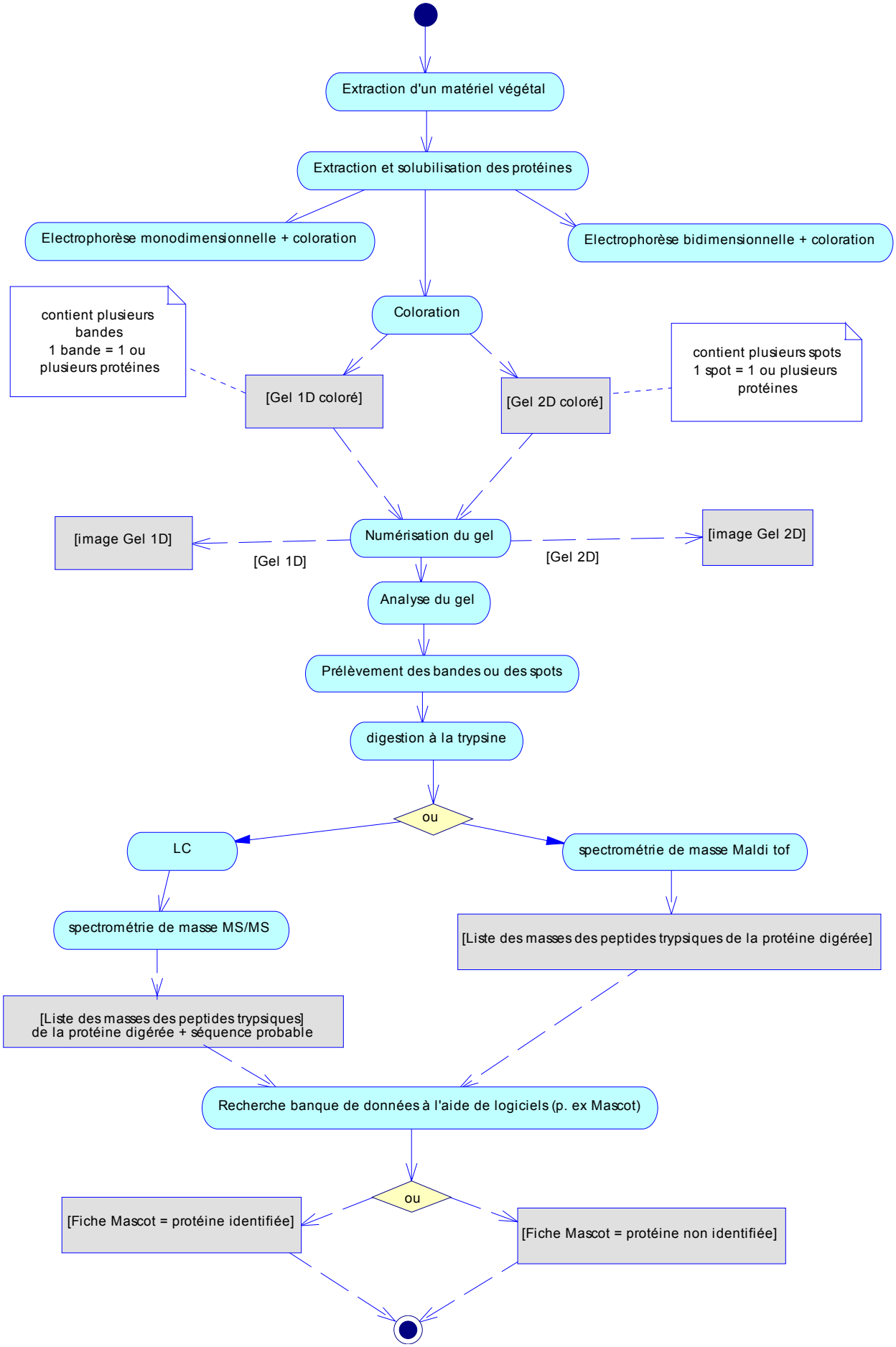
L'ensemble de ces activités génère une certaine quantité de données dont j'ai fait l'inventaire. Cependant toutes les données ne sont pas à archiver d'autant plus que ProteomIs a pour vocation notamment d'être une base de gestion des résultats expérimentaux et non pas d'être un outil permettant de gérer l'ensemble des données de laboratoire comme un LIMS (partie **4.1.1 Les LIMS**).

Afin de cibler parfaitement la nature des données que les biologistes souhaitaient archiver, un questionnaire a alors été préparé et envoyé aux personnes concernées dans les différents laboratoires impliqués dans le projet. En **annexe 14** est présenté pour exemple le questionnaire qui a été rempli et renvoyé par Mme Elizabeth Jamet de l'UMR 5546 de Toulouse.

L'étape suivante de l'analyse est typiquement orientée objet. Il s'agit de décomposer le domaine d'intérêt en classes conceptuelles représentant les entités significatives de ce domaine. On va simplement créer une représentation visuelle des objets du monde réel dans un domaine donné.
Si l'on emploie la notation UML, le modèle du domaine (ou modèle conceptuel des données) est constitué d'un ensemble de diagrammes de classes dans lesquelles aucune opération n'est définie.
Ce modèle peut représenter les éléments suivants :

- Les classes conceptuelles ou les objets du domaine ;
- Les associations entre classes conceptuelles ;
- Les attributs des classes conceptuelles.

<u>Diagramme 4 :</u> Diagramme d'activité général « Démarche expérimentale utilisée en protéomique »



A partir de l'inventaire des données précédent, nous avons donc conçu les classes de notre modèle. Certaines données ont pu être regroupées pour constituer les attributs de classes tandis que d'autres ont conduit directement à des classes qu'il a fallu ensuite détailler.

L'observation des modèles de données de bases préexistantes en protéomique (partie **4.1.2)** a également été une bonne source d'inspiration pour la conception du modèle. J'ai notamment pu obtenir le schéma de la banque de données protéomique PPMDB **[a3]** en faisant une demande auprès des personnes responsables de ce projet. Je me suis également inspiré du modèle Pedro qui fut la première initiative pour représenter, sous un modèle standard, les méthodes utilisées et les données générées par les expériences de protéomique (voir **annexe 16 Les initiatives de normalisation**)

Par ailleurs, la conception du schéma a nécessité une concertation avec les responsables de Génoplante et les équipes en charge du développement des schémas des autres modules du système GpiIS dans lequel la version intégrative de ProteomIs doit s'insérer. Il a fallu définir ensemble les classes et relations de ProteomIs qui devaient être communes à celle des autres schémas. Nous ferons dans la partie **6.3.2** l'inventaire des classes et relations retenues pour être partagées par tous les modules de GpIS.

J'ai ensuite présenté et proposé le modèle réalisé aux futurs utilisateurs des différentes équipes partenaires du projet qui l'ont validé. Le modèle une fois validé fut cependant par la suite affiné, notamment lors de la phase de maquettage des interfaces. En effet pour que le prototype de l'application soit implémentable, certaines réalisations ou modifications effectuées côté interfaces ont conduit nécessairement à des ajustements au niveau du modèle. Cette phase de validation des données par les traitements est une étape très importante qui fait souvent évoluer les modèles. De plus, il est évident que des besoins nouveaux trouvés par la suite lors de l'analyse approfondie et la conception des différents cas d'utilisations ont obligé à effectuer des modifications sur le Modèle Conceptuel des Données. (comme ce fut le cas à la suite de l'analyse approfondie du cas d'utilisation *« Clustering des protéines »)*.

6.3.2 Le modèle de données

6.3.2.1. Le modèle conceptuel de données

Le modèle conceptuel de données de ProteomIs/GnpProt a été conçu sous la forme d'un diagramme de classes. Ce diagramme de classe est divisé en cinq paquetages représentés sur le **diagramme 5**. Nous décrivons ici le rôle de chacun des paquetages ainsi que l'ensemble des classes qu'ils contiennent. La modélisation a été réalisée grâce au logiciel Rational Rose.

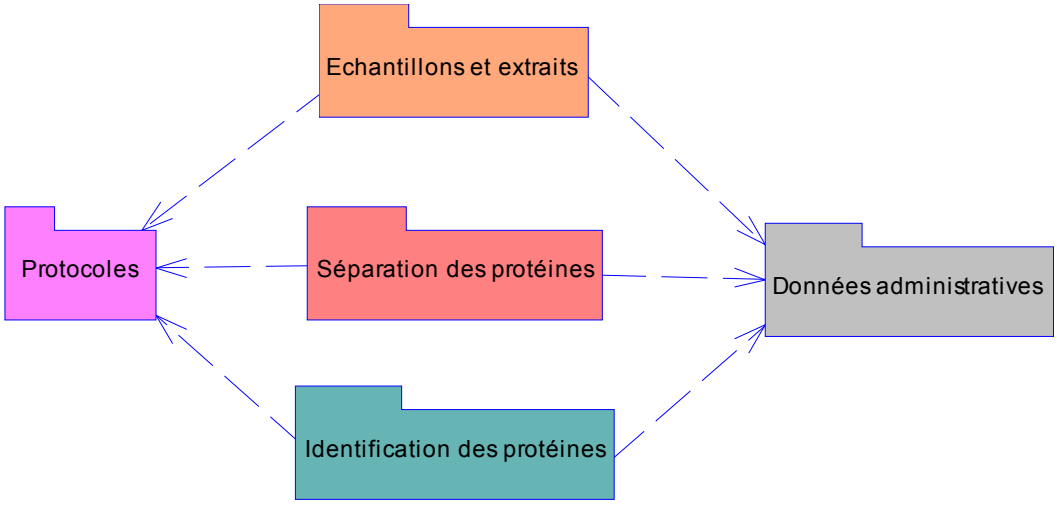
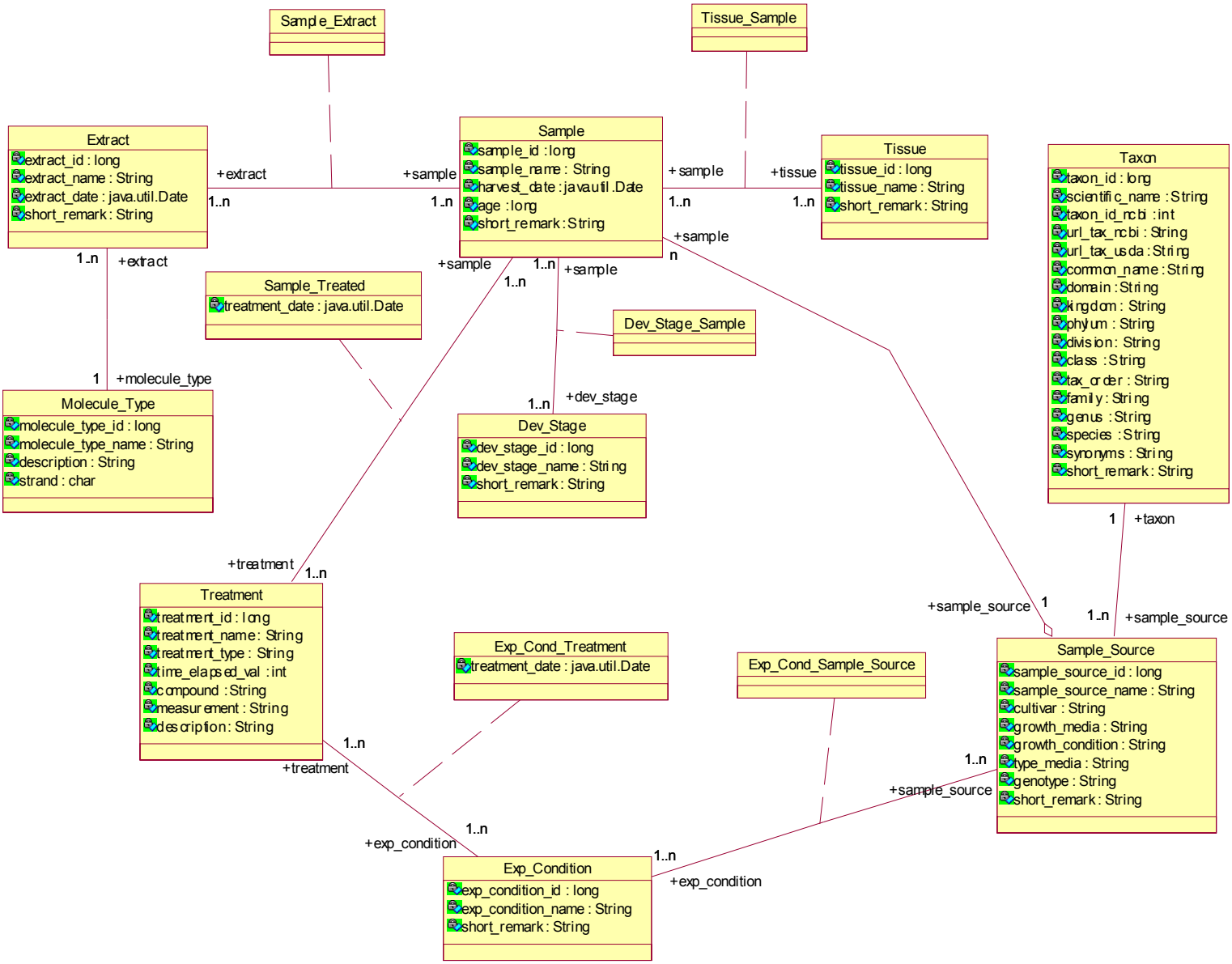


Diagramme 5 : Diagramme de paquetages de ProteomIs/GnpProt

➢ **Paquetage « Echantillons et extraits » (<u>diagramme 6</u>)**

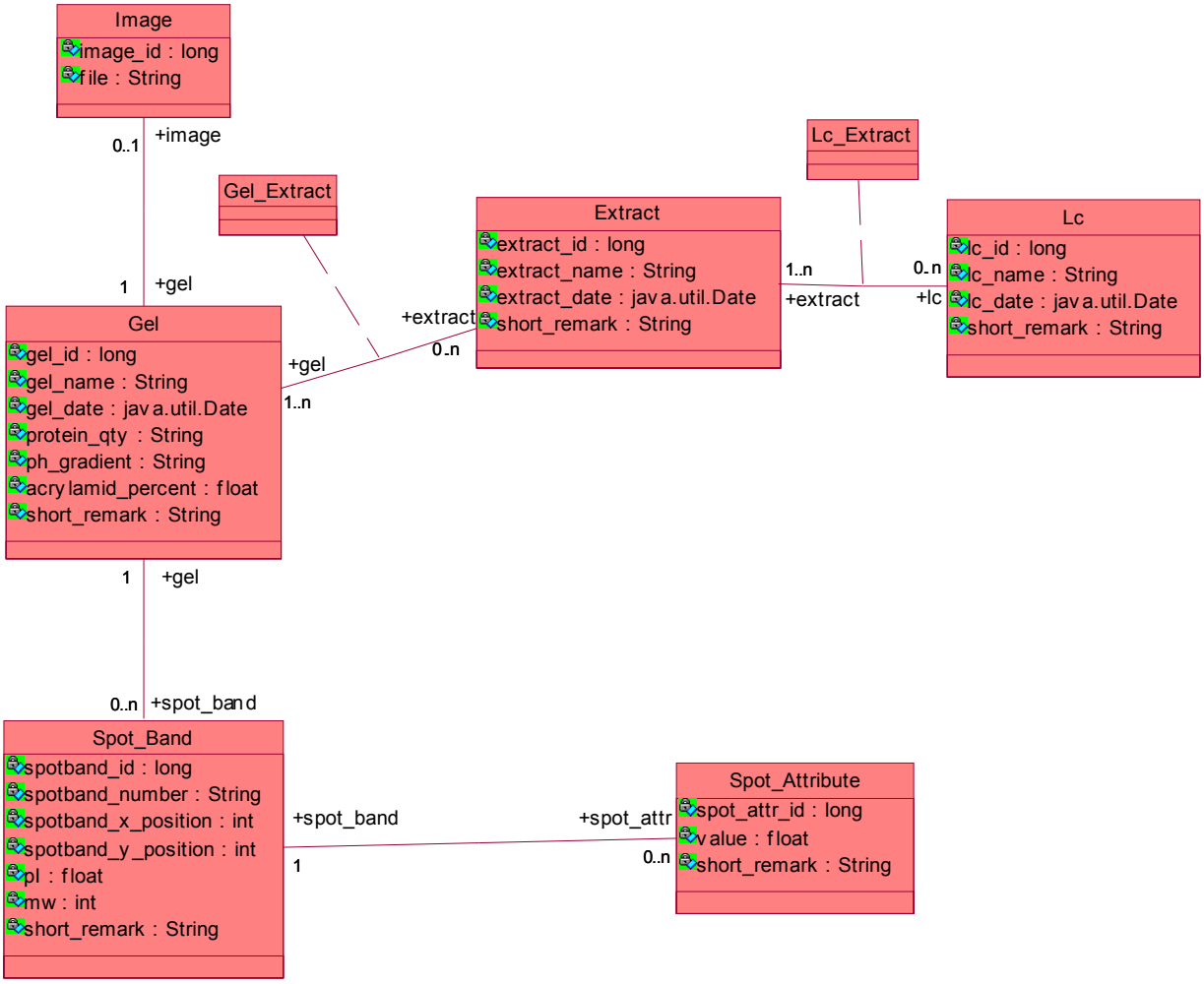
Les classes de cette section décrivent le matériel biologique qui est manipulé. Un extrait protéique (*classe extract*) est décrit par l'échantillon biologique d'origine (*classe sample*). A cet échantillon sont associées des informations relatives à l'espèce d'origine (*classe sample_source*), à sa taxonomie (*classe taxon*), au tissue d'origine (*classe tissue*) et au stade de développement de l'organisme (*classe dv_stage*) au moment du prélèvement. L'échantillon biologique peut alors subir un ensemble de traitements (*classe treatment*) et faire référence à une ou plusieurs conditions expérimentales (*classe exp_condition*).



<u>Diagramme 6 :</u> Diagramme de classe « Echantillons et extraits »

➢ **Paquetage « Séparation des protéines » (<u>diagramme 7</u>)**

On retrouve dans cette partie tout ce qui concerne la séparation des protéines contenues dans un extrait protéique. Les protéines peuvent être séparées par la technique d'électrophorèse monodimensionnelle (gel 1D) ou bidimensionnelle (gel 2D) (*classe gel*), et également par la technique de chromatographie en phase liquide (*classe LC*). Chaque gel est ensuite scanné avec pour résultat une image (*classe image*). Cette image est analysée par un logiciel d'analyse d'image afin d'identifier les spots (dans le cas des gel 2D) ou les bandes (sur les gel 1D). C'est dans ces spots/bandes représentés par la classe *spot_band* que sont concentrées les protéines.



<u>Diagramme 7 :</u> Diagramme de classe « Séparation des protéines »

➢ **Paquetage « Identification des protéines par spectrométrie de masse » (diagramme 8)**

Cette partie concerne l'analyse en MS ou MS/MS (*classe ms_analysis*) des protéines contenues dans le produit de séparation des gels (les spots/bandes) ou des expériences de chromatographie en phase liquide. Cette étape aboutie à la génération d'un fichier de résultat d'identification des protéines (*classe ms_analysis_result*) obtenu grâce aux logiciels d'interrogation en banques. Un fichier peut contenir plusieurs protéines (*classe protéine*). A chacune des protéines est associée des informations sur ses modifications post-traductionnelles (*classe PTM*), sa séquence (*classe sequence*), son numéro d'accession (*classe acc*) et la base de donnée d'origine où elle a été identifiée (*classe DB*).

Un diagramme de classes a également été envisagé pour les protéines qui seraient identifiées dans les spots/bandes par des techniques d'immuno-détection. Cependant il n'y a toujours pas de données de ce type dans la base (voir diagramme **annexe 17**).

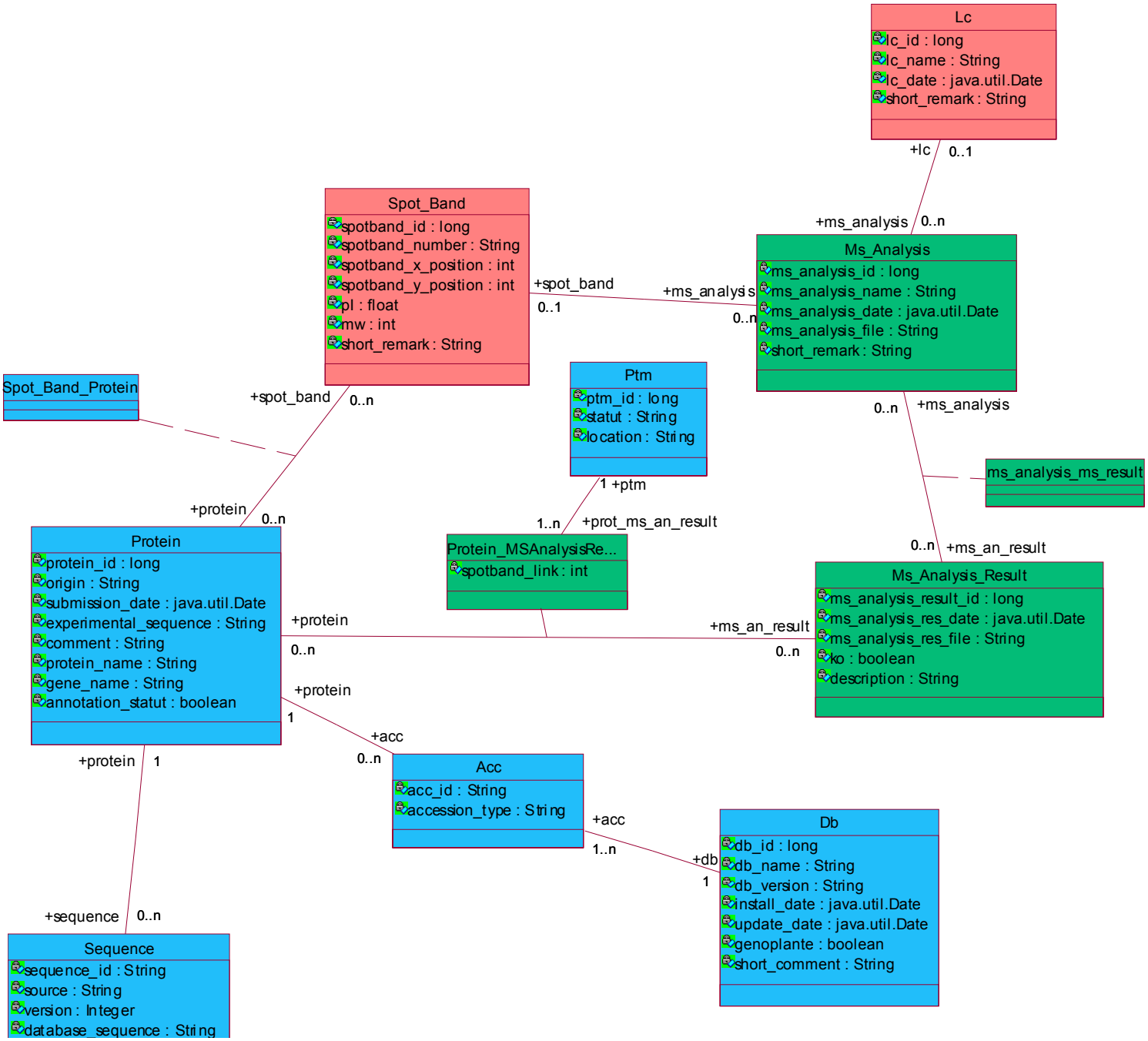
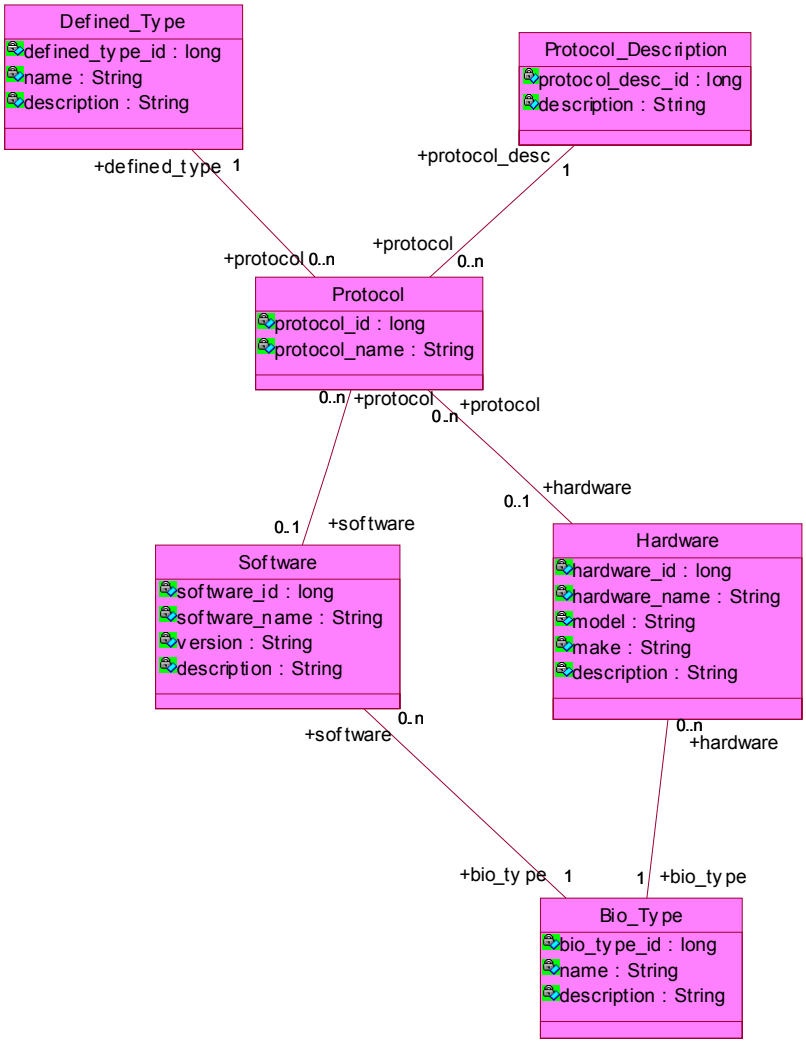


Diagramme 8 : Diagramme de classes « Identification des protéines par spectrométrie de masse »

➢ **Paquetage « Protocoles » (<u>diagramme 9</u>)**

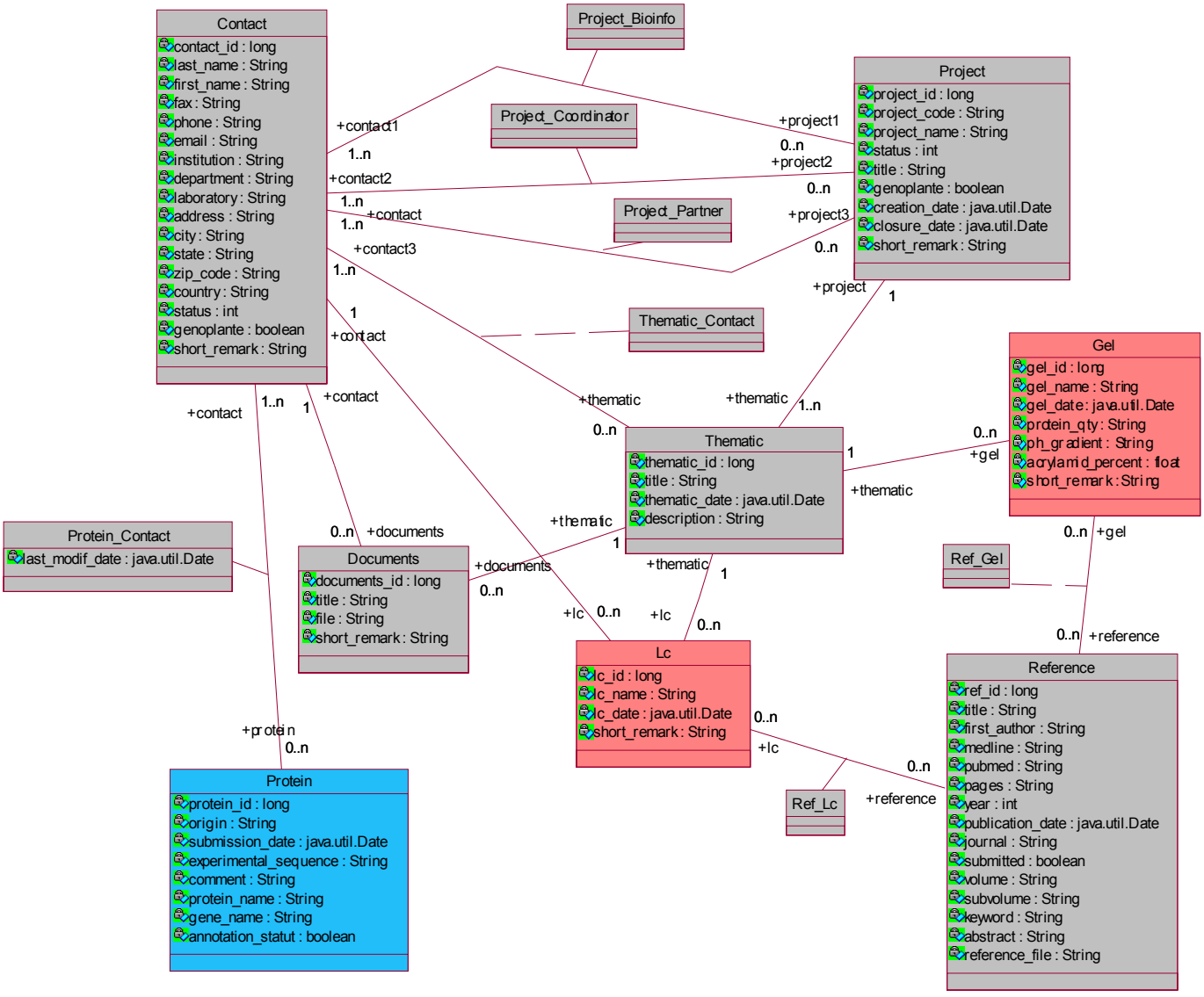
La classe *protocole* permet d'instancier l'ensemble des protocoles associés aux différentes manipulations expérimentales (préparation des extraits, gels, analyse d'image, expériences de chromatographie en phase liquide, spectrométrie de masse, interrogation en banque). Chaque protocole est défini par une description générale (*classe protocol_description*). Un lien a été mis en place vers du matériel (*classe hardware*) ou des logiciels (*classe software*) intervenant dans certains protocoles. La classe *defined_type* décrit le type du protocole et la classe *bio_type* contient la description des autres types généraux que l'on peut trouver dans la base ProteomIs/GnpProt.



<u>Diagramme 9 :</u> Diagramme de classe « Protocoles »

➢ **Paquetage « Données administratives » (<u>diagramme 10</u>)**

La base peut stocker un ensemble de projets (*classe project*) auxquels sont reliés les personnes (*classe contact*) qui sont impliquées : coordinateur du projet (*association project coordinator*), bioinformaticiens dans le projet (*association project_bioinfo*), partenaires du projets (*association project_partner*). Chacun des projets est également associé à une ou plusieurs thématique (*classe thématique*) renseignée éventuellement par des documents (*classe documents*). Les publications (*classe reference*) sur les gels et les expériences de chromatographie en phase liquide sont également archivées dans la base.



<u>Diagramme 10 :</u> Diagramme de classe « Données administratives »

6.3.2.2 Contraintes au niveau interopérabilité

Nous établissons ici la liste des classes du modèle conceptuel de données de ProteomIs/GnpProt partagés avec les autres modules de GpiIS :

➢ **Classes partagées au sein du paquetage « Echantillons et extraits :**

classe Sample
classe Tissue
classe Sample_source
classe Dv_stage
classe Treatment
classe Exp_condition
classe Taxons

➢ **Classes partagées au sein du paquetage « Protocoles » :**

classe Protocol
classe Protocol_description
classe Defined_type
classe Software
classe Hardware
classe Bio_type

➢ **Classes partagées au sein du paquetage « Données administratives » :**

classe Contact
classe Project

6.3.2.3 Conception du modèle logique de données

Tout MCD peut être transformé en un MLD (Modèle Logique de Données) (présenté **annexe 18** directement exploitable par une base de données. On représente sur ce modèle l'ensemble des tables avec leurs champs dans une base de données. Ces tables et ces champs correspondent aux classes avec leurs attributs dans le MCD. Il faut prendre en compte les règles de transformation du MCD réalisé selon un modèle objet pour passer à un MLD de type relationnel.

Par rapport au modèle conceptuel, une table a été rajouté dans le modèle logique : il s'agit de la table spotband_protein. Cette table correspond en fait à une relation supplémentaire entre la table spotband et la table protein. Elle permettra de faciliter les requêtes croisées entre ces deux tables.

6.4 Phase de maquettage

Afin de réaliser une maquette de l'application nous allons d'abord faire l'inventaire de l'ensemble des interfaces du système. En même temps chacune des interfaces sera décrite sommairement en terme de fonctionnalités, ceci en s'écartant de toute préoccupation technique. Nous présenterons ensuite une seule des maquettes d'interface homme-machine (IHM) réalisées. Enfin nous modéliserons les liens entre les interfaces.

6.4.1 Inventaire des différentes interfaces de l'application :

Interface de Login :

C'est la première interface qui sera présentée à l'utilisateur. Dans le cadre de la version locale de ProteomIs, pour accéder à cette interface l'utilisateur devra saisir un login et un mot de passe initialisé dans la base de donnée. Dans le cadre du module GnpProt, l'utilisateur doit disposer d'un compte pour se connecter sur le site de Génoplante et accéder au système d'information GpiIS de Génoplante. L'authentification se fait alors toujours via un login et mot de passe fournit par Génoplante à la création du compte. Cependant la connexion se fait ensuite via SSL (Secure Sockets Layers) **[G17].**

Interface d'accueil :

Sur cette interface est décrit succinctement le contexte du projet ProteomIs/GnpProt (objectifs et partenaires impliqués).
On pourra également faire un bilan des données disponibles et fournir un accès sur la documentation de l'application. Cette interface fournira le lien sur l'interface de recherche avancée de l'application.

Interface d'interrogation :

Sur cette interface, des options de recherche avancées devront être mis à la disposition de l'utilisateur.

Interface Liste de résultats :

Description :

C'est une interface présentant le résultat d'une requête d'interrogation sous la forme d'une liste d'éléments lorsque le nombre de résultats renvoyés par la requête est > 1. Sur chacun de ces éléments est présent un lien permettant d'accéder à une interface de visualisation des informations textuelles sur l'élément en question.

Interfaces de visualisation :

Les interfaces que l'on a regroupées ici ont simplement pour fonction de **visualiser** des informations sur un objet identifié de manière unique dans la base de données (p. ex un projet, une personne, une expérience …). Pour cela, on a choisi par convention d'appeler ces interfaces : Interfaces de visualisation. Par rapport à l'analyse elles répondent aux besoins du cas d'utilisation « Visualiser un résultat ». L'information apportée par ces interfaces est majoritairement textuelle sauf dans le cas de l'interface de navigation dans l'image d'un gel.

Voici la liste des interfaces de visualisation :

Interface de visualisation Projects : Interface informative sur les projets.

Interface de visualisation Thematic : Interface informative sur les thématiques associées aux projets.

Interface de visualisation Contact : Interface informative sur les personnes impliquées dans les projets.

Interface de visualisation Documents : Interface informative sur divers documents associés aux thématiques.

Interface de visualisation Extract : Interface informative sur les extraits protéiques analysés par les différentes expériences de protéomique.

Interface de visualisation Sample : Interface informative sur les échantillons organiques à partir desquels on a réalisé des extraits protéiques *(Extract)*.

Interface de visualisation Sample_source_name : Interface décrivant la plante d'origine à partir duquel on a prélevé des échantillons de tissus *(Sample)*.

Interface de visualisation Treatment : Interface décrivant les traitements effectués sur les plantes où les échantillons prélevés sur ces plantes.

Interface de visualisation LC : Interface décrivant les expériences de chromatographie en phase liquide (Liquid Chromatography) ayant permis de séparer les protéines contenues dans les extraits *(Extract)*.
Cette interface contient notamment la liste des protéines identifiées dans le produit de cette séparation ; cette identification se faisant par spectrométrie de masse MS/MS.

Interface de visualisation Spot : Interface décrivant les spots contenus dans les gels ainsi que les protéines identifiées par spectrométrie de masse dans ces spots.

Interface de visualisation Proteine : Cette interface décrit de manière précise la nature de la protéine identifiée. Ensuite des liens doivent permettre de retrouver les informations correspondantes sur la protéine décrite dans les banques de données publiques. Enfin, en plus des précisions biologiques, un récapitulatif des expériences qui ont permis de l'identifier est fournit sur cette interface. Cette interface sera présentée en détail après implémentation dans la partie **7.2**.

Interface de visualisation Db : Interface décrivant les différentes bases de données à partir desquelles les protéines de ProteomIs ont été identifiées.

Interface de visualisation Gel : Interface rassemblant des informations textuelles sur un gel. Une image miniature clickable d'un gel doit permettre d'accéder à l'interface suivante.

Interface de navigation dans l'image d'un gel : Il s'agit de concevoir ici une interface permettant d'explorer facilement l'image d'un gel et donc accéder facilement aux informations sur les spots identifiés sur le gel. Ses fonctionnalités ont été spécifiés dans le cas d'utilisation « Visualiser un gel » (**annexe 1**).

6.4.2 Réalisation des maquettes

Les maquettes des IHM ont été construites à l'aide du logiciel PowerPoint en proposant un certain niveau d'interactivité. Pour cette phase de maquettage, un autre logiciel tel que Zope **[i14]** aurait pu permettre de fabriquer des maquettes aux fonctionnalités plus étendues. Cependant Powerpoint était dans notre cas suffisant pour créer rapidement une maquette de l'application incorporant des liens hypertextes qui permettent de simuler la navigation à travers les interfaces.

Le fichier PowerPoint de la maquette a été présenté à plusieurs reprises aux futurs utilisateurs et testé ensuite par ces derniers. Les remarques de ceux-ci ont permis plusieurs améliorations de la maquette jusqu'à obtenir une version finale. Nous présentons ici la démarche qui nous a permis d'aboutir à la réalisation de la maquette de l'interface de navigation des gels.

En **annexe 2** est détaillée :

- la maquette de l'interface d'interrogation
- la maquette de l'interface Liste de résultats
- la maquette de l'interface de visualisation des informations textuelles sur un gel

Réalisation de la maquette de l'interface de navigation dans l'image d'un gel :

Inventaire des données à visualiser :

La première étape a consisté à préciser l'ensemble des données devant être visualisée par l'interface de navigation des gels. Ces données correspondent à celles encadrées en jaune dans le **diagramme 11** qui est extrait du modèle conceptuel de données de ProteomIs/GnpProt. C'est en faisant l'inventaire de ces données que la maquette de l'interface va pouvoir être construire. Cette démarche nous a permis d'identifier de nouvelles données que les biologistes voulaient faire apparaître sur les interfaces de visualisation. Le modèle conceptuel de données devait alors être complété. Ainsi cette phase de maquettage constitue en même temps une phase de validation du modèle de données.

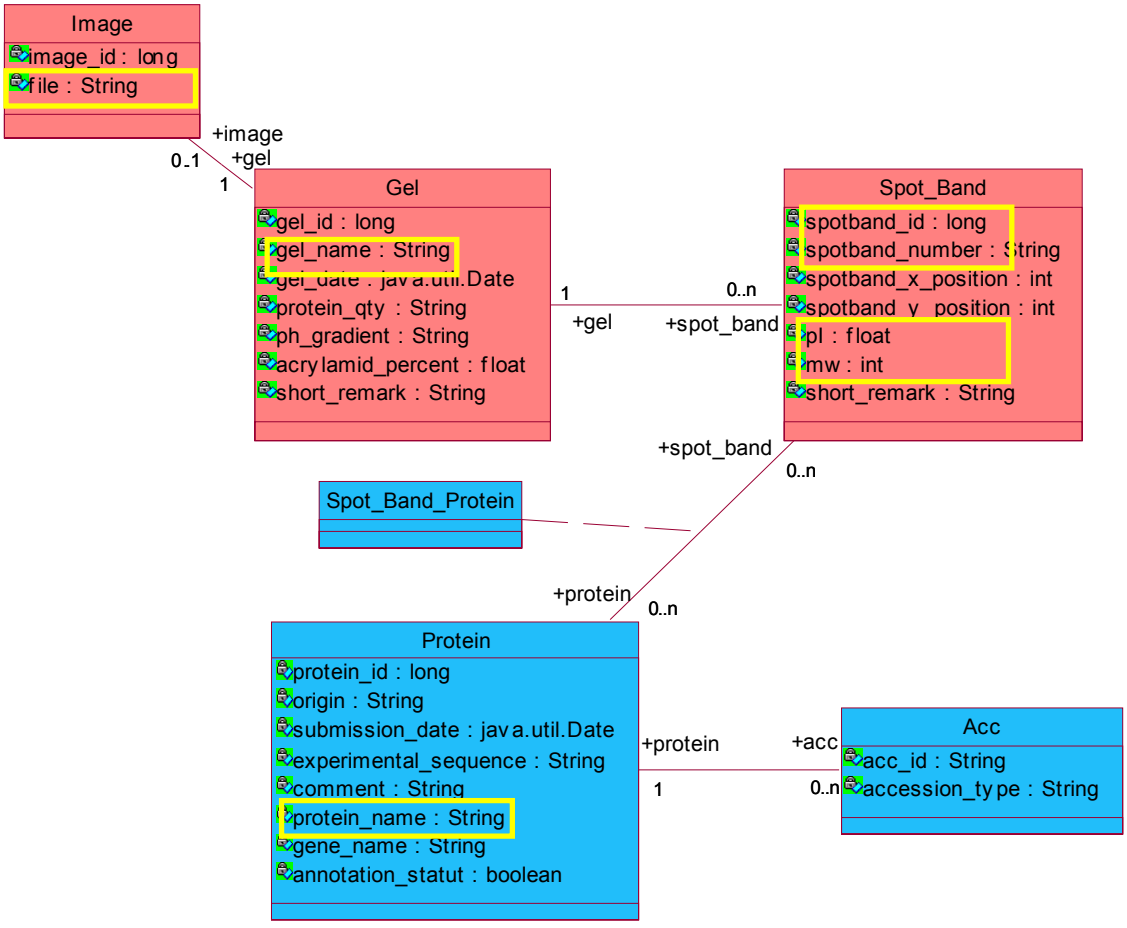
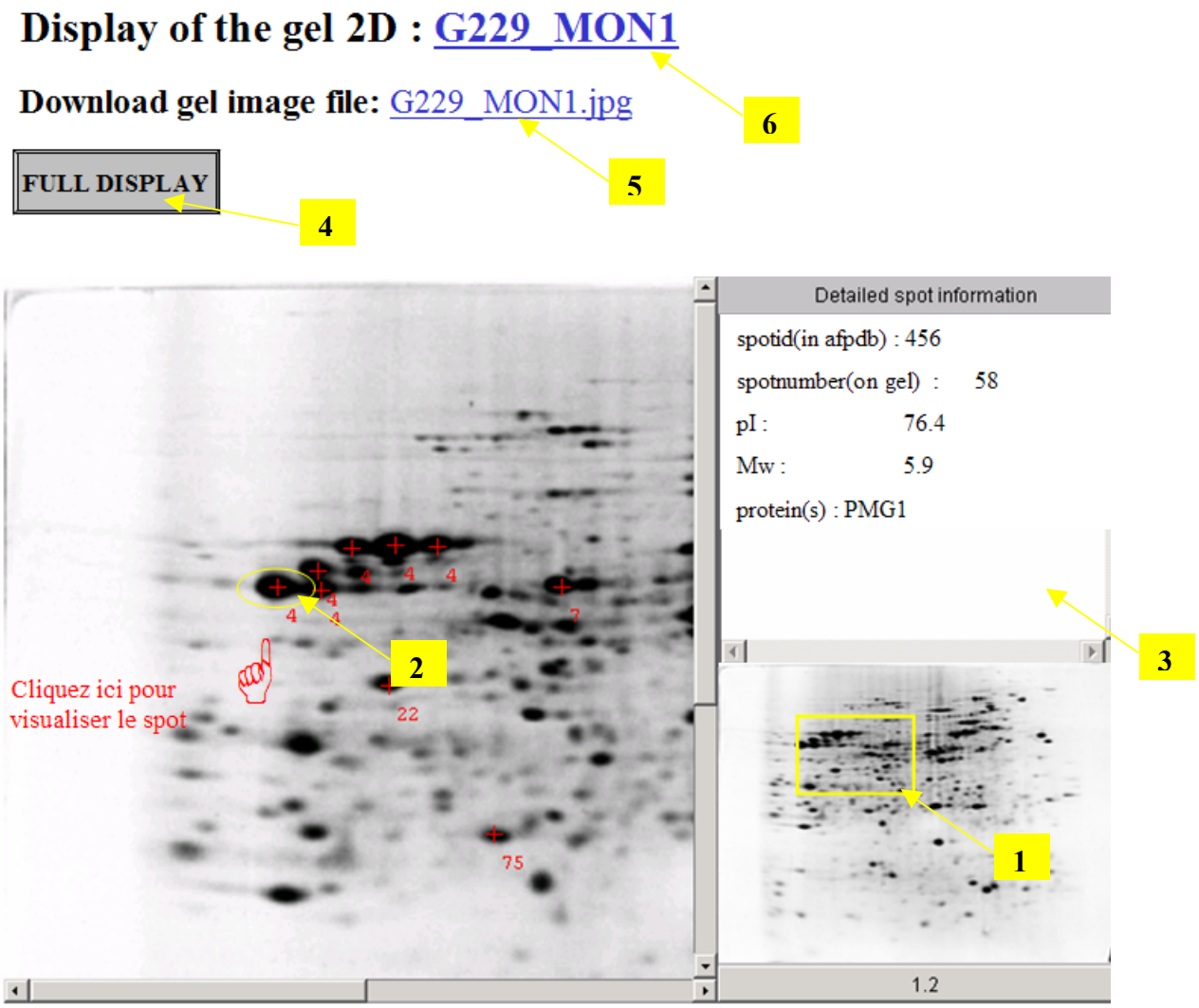


Diagramme 11 : Domaine du MCD couvert par l'interface de navigation des gels

Maquette :

A partir de là il a été possible de concevoir la maquette de cette interface. Celle-ci est présentée dans le **document 20**. Cette interface permet de visualiser les spots sur un gel. On a en bas à droite une vue réduite de l'image complète. Le déplacement manuel du rectangle jaune (flèche jaune 1) permet de faire déplacer l'image zoomée dans le cadre de gauche. Si l'on pointe le curseur de la souris sur un spot marqué par une croix rouge (flèche jaune 2), on peut afficher les infos sur ces spots dans la fenêtre en haut à droite. Si un spot est rattaché à plusieurs protéines, dans la ligne protein(s) de cette fenêtre, les noms de protéines apparaîtront séparés par un caractère &.
Si l'on clique sur un spot, on affiche l'interface de visualisation du spot correspondant qui permettra d'accéder à des informations plus détaillées sur le spot en question.
Le bouton « FULL DISPLAY » (flèche jaune 4) ouvre l'interface en mode plein écran pour autoriser une zone visible sur le gel plus importante. Le lien G229_MON1.gif (flèche jaune 5) permet de télécharger l'image du gel tandis que le lien G229_MON1 (flèche jaune 6) redirige vers l'interface de visualisation des informations textuelles d'un gel.



Document 20 : Maquette Powerpoint « Interface de navigation dans l'image d'un gel »

6.4.3 Modélisation des liens entre les interfaces

Nous avons fini de concevoir les maquettes à partir des cas d'utilisations identifiés et décrit dans la partie **6.2.2**. Il s'agit maintenant de réaliser l'ensemble des diagrammes dynamiques représentant de manière formelle l'ensemble des chemins possibles entre les différentes interfaces proposées à l'utilisateur. UML nous offre la possibilité de modéliser cela au moyen d'un diagramme dynamique appelé diagramme d'activités.

Dans ce modèle nous allons nous servir d'un certain nombre d'éléments standard mais aussi de conventions particulières. Un diagramme d'activités de base contient un nombre restreint d'éléments à savoir :

- des activités
- des transitions entre activités, pouvant porter des conditions ;
- des branchements conditionnels ;
- un début et une ou plusieurs terminaisons possibles.

Pour aller plus loin, nous nous servirons du concept d'activités pour modéliser plusieurs concepts différents, grâce aux conventions graphiques suivantes :

- une interface (« interface ») ;
- une action simple, telle qu'un téléchargement de fichier, etc. (« action ») ;
- une erreur ou un comportement inattendu du système (« exception » avec un niveau de gris intermédiaire) ;
- une liaison vers un autre diagramme d'activités, pour des raisons de structuration et de lisibilité (« connector » avec un niveau de gris soutenu).

Afin de ne pas surcharger les diagrammes avec trop de flèches, certaines des liaisons entre interfaces seront écrites dans l'activité « interface ».

Nous décomposons le diagramme d'activités en deux diagrammes. Tout d'abord un diagramme d'activités *« Se loguer et rechercher un élément »* (**diagramme 12**) avec une liaison vers un deuxième diagramme d'activité *« Liens entre les interfaces de visualisation »* (**diagramme 13**) au niveau de l'activité « connector » intitulé « Interface de visualisation d'un élément ». Ce deuxième diagramme d'activités est particulier car il n'a pas de point d'entrée et de terminaison. En effet à partir de l'activité « action » intitulé « Recherche » et décrite dans le premier diagramme il est possible d'aboutir sur n'importe laquelle des activités « interface » décrite dans le deuxième diagramme.

Le diagramme d'activité « Liens entre les interfaces de visualisation » rappelle la démarche expérimentale classique utilisée en protéomique aboutissant à partir d'un extrait organique à l'identification de la nature des protéines contenues dans cet extrait (voir **partie 3.4 l'analyse protéomique**).

On peut constater que l'interface de visualisation «protéine » est bien l'objet central à partir duquel on peut remonter aux expériences qui ont permit de l'identifier :

- le gel d'électrophorèse (interface gel) et la chromatographie en phase liquide (interface LC) qui sont les techniques de séparation des protéines qui ont permis d'isoler les protéines à partir des extraits protéiques (interface extract) issues des échantillons végétaux (interface sample)
- l'interface de navigation dans les gels où sont localisés les spots qui contiennent une ou plusieurs protéines et à partir duquel on peut accéder aux interfaces de visualisation des spots

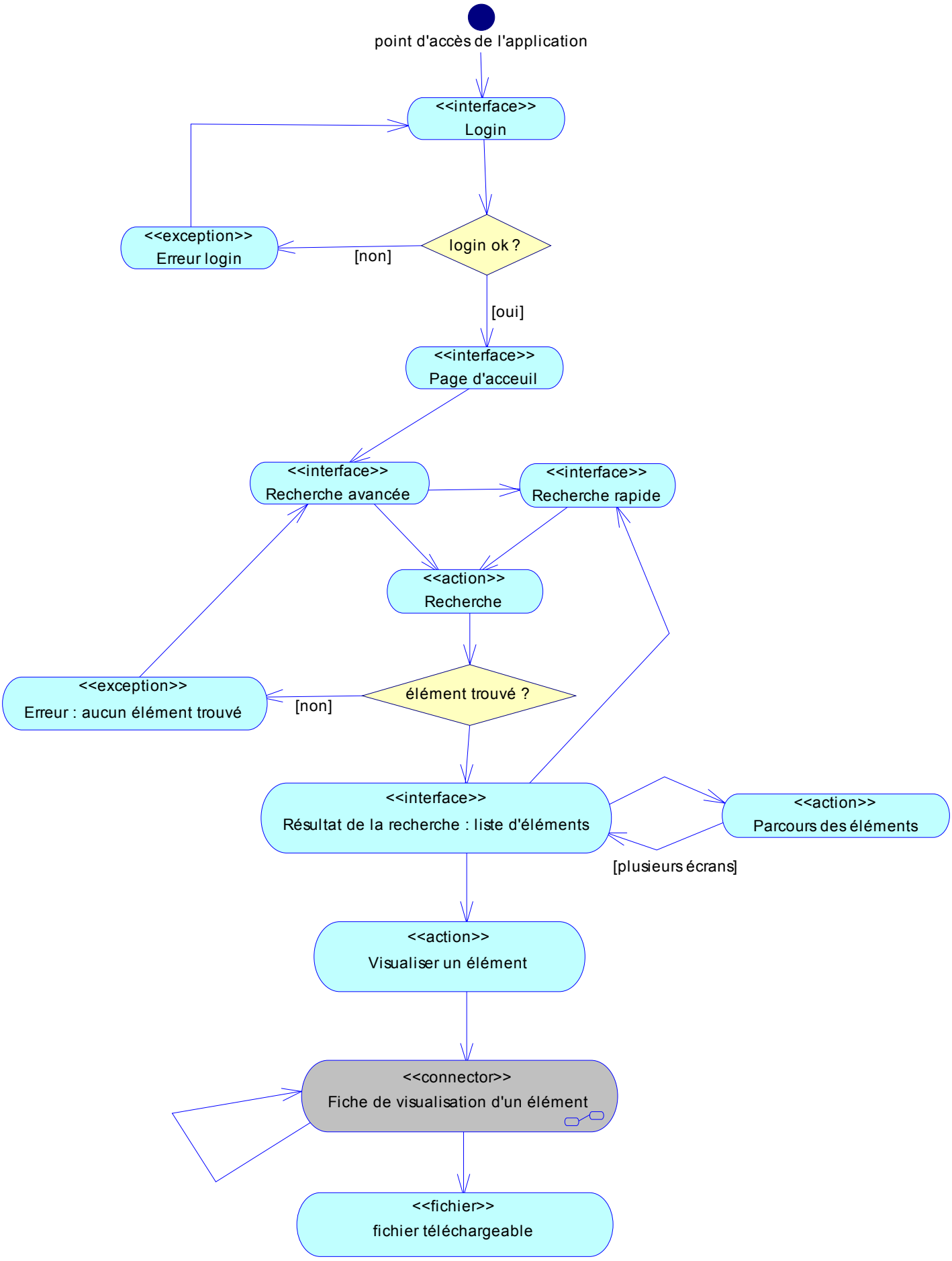
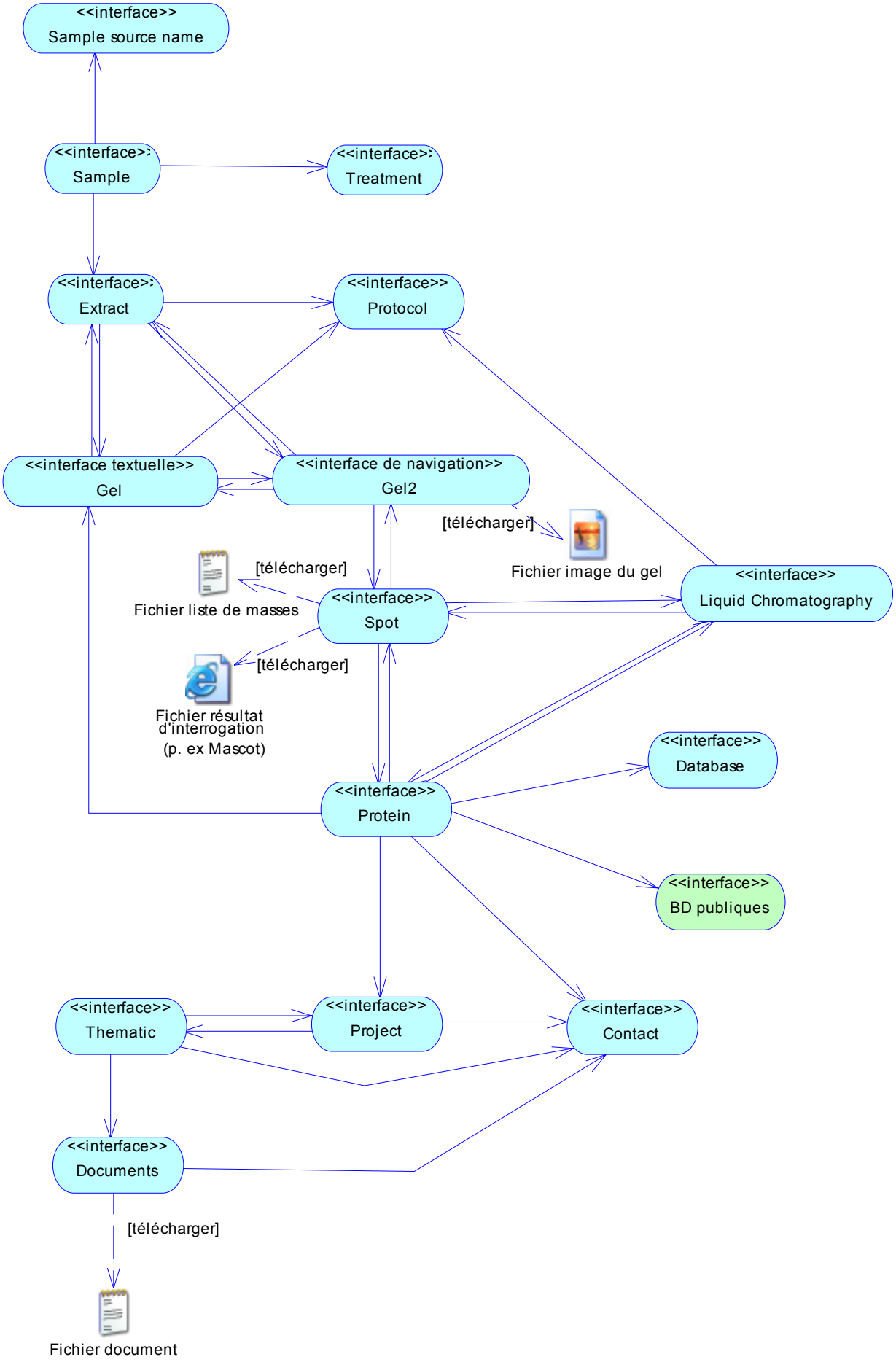


Diagramme 12 : Diagramme d'activités « Se loguer et rechercher un élément »



<u>Diagramme 13 :</u> Diagramme d'activités « Liens entre les interfaces de visualisation»

6.5 Conception détaillée et choix d'implémentation par cas d'utilisation : "le COMMENT"

Après avoir découvert "Quoi coder" (avec les spécifications et l'analyse), on se demande ici "Comment le coder". En conception détaillée il sera toujours question de modéliser. Toutefois, les diagrammes réalisés pourront être enrichis d'éléments appartenant aux choix technologiques et d'implémentation effectués. Chacun des cas d'utilisation suivant : l'interrogation, la visualisation et l'analyse bioinformatique des données sera approfondie dans ce sens. Pour le cas d'utilisation « saisie des données » seul le résultat final d'implémentation sera présenté au **chapitre 7**.

6.5.1 Le choix du SGBD

La base de données ProteomIs/GnpProt devait pouvoir être implémentée sous un SGBD open source. Cependant, il est prévu à terme que le projet ProteomIs/GnpProt puisse être supporté par le SGBD commercial Oracle **[i75]**. Dans le domaine du logiciel libre, les SGBD qui émergent aujourd'hui sont MySQL **[i73]** et PostgreSQL **[i74]**. Le choix s'est porté sur PostgreSQL qui est celui qui se rapproche le plus de Oracle en terme de fonctionnalités et du respect du standard SQL.
Remarque sur la gestion des fichiers : Parmi les données à gérer dans ProteomIs/GnpProt se trouvent des objets binaires : fichiers images de gel au format jpeg ou gif, fichiers textes décrivant les protocoles, fichiers de résultats d'interrogation au format html, fichiers pdf pour les publications, documents word …
Voici les deux possibilités qui ont été envisagées pour la gestion des fichiers :
- stockage direct dans la base sous le type BLOB (Binary Large Object),
- stockage au sein du serveur dans des répertoires et référence du nom des fichiers dans la base de données.

C'est la deuxième solution qui a été retenue afin de ne pas prendre le risque d'affecter les performances du SGBD avec des types de données trop lourds à gérer.

6.5.2 Architecture et conception pour l'interrogation et la visualisation des données

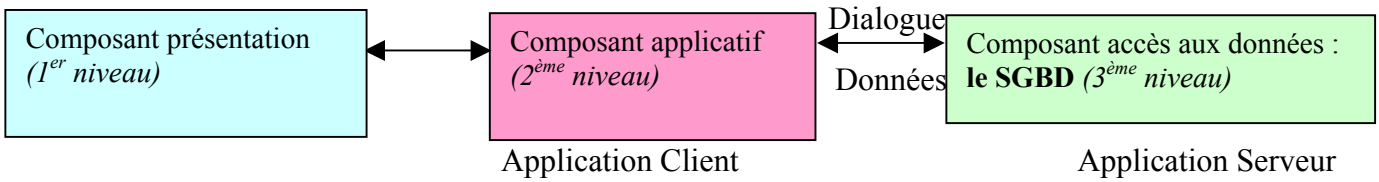
L'expression initiale des besoins nous a enseigné que l'application devrait à terme être consultable par la communauté scientifique sur le site Internet de Génoplante. On parle alors d'applications Web. Cette application doit être capable, dans un environnement multi-utilisateurs, de restituer les données à partir d'un SGBD, qui, dans le cadre du module GnpProt, devra obligatoirement être hébergé sur le serveur de Génoplante. Afin d'effectuer un choix technique adapté à ces contraintes, nous commencerons par effectuer un inventaire des modèles d'architectures les plus courants adaptés au contexte de notre projet. Les outils retenus devront être gratuits et offrir un maximum de portabilité ; ces conditions ayant été imposées par Génoplante.

6.5.2.1 Architecture des applications Web

a) Les modèles architecturaux

Le nombre impressionnant de produits et de technologies liées à l'Internet disponibles actuellement a suscité des architectures d'applications Web multiples et variées. Cependant, en règle générale une architecture applicative se découpe en trois niveaux d'abstraction distincts : le niveau présentation, le niveau applicatif et le niveau d'accès aux données. Dans le cadre de notre projet, les données sont gérées par le SGBD. Ainsi, on obtient une séparation distincte entre le 2ème niveau (le composant applicatif) et le 3ème niveau (le composant accès aux données représenté par le SGBD).

Le composant applicatif interrogera le SGBD à l'aide du langage SQL qui lui restituera alors les données afin de mettre en forme la couche présentation. Nous sommes donc dans une configuration client-serveur.



Ces trois niveaux peuvent être imbriqués ou répartis de différentes manières entre plusieurs machines physiques. Cependant dans le cadre de notre projet l'application et le SGBD doivent se situer sur le même serveur à Génoplante. Le noyau de l'application étant en fait composé de la logique de l'affichage (présentation) et de la logique des traitements (applicatif), le découpage et la répartition de ce noyau permettent de distinguer les architectures applicatives suivantes :

- l'architecture 2-tiers *(ou poste client lourd)*,
- l'architecture n-tiers *(ou distribués)*.
- l'architecture 3-tiers HTML/Web *(ou client léger passif)*
- l'architecture mixte *(ou plate-forme de développement)*

Nous décrirons ici succinctement ces différents modèles.

➢ L'architecture 2-tiers *(ou poste client lourd)*

Dans ce modèle toute la partie applicative (composant présentation et traitement) est sur le poste client. On parle alors de poste client lourd. Ce type d'architecture permet de tirer partie de la puissance des ordinateurs déployés en réseau pour fournir à l'utilisateur une interface riche (graphique, dynamique …), tout en garantissant la cohérence des données, qui restent gérées de façon centralisée par le SGBD sur le serveur dédié. Cependant le poste client est fortement sollicité, il devient de plus en plus complexe et doit être mis à jour régulièrement pour répondre aux besoins des utilisateurs. Néanmoins la contrainte du déploiement est aujourd'hui moins forte que par le passé, grâce à l'apparition de protocole (e.g Java Web Start de Sun **[i87]**) facilitant le téléchargement et la mise à jour des programmes clients via le web.

➢ L'architecture n-tiers *(ou modèle distribué)*

Dans ce modèle l'application est découpée en n-niveaux. Il est possible de répartir les traitements de la couche applicative (en rose) entre le poste client et un serveur d'applications contenant par exemple des objets java et communiquant via le protocole RMI. Ensuite, si l'on veut encore alléger la charge sur le serveur d'applications on peut distribuer les traitements sur plusieurs machines généralement en parallèles (*cluster*). On passe alors d'un modèle de 3 niveaux à n niveaux possibles.

➢ L'architecture 3-tiers HTML/WEB *(ou client léger passif)*

L'architecture 3-tiers HTML/WEB repose sur l'utilisation du serveur Web en tant que middleware. En d'autres termes, toutes les requêtes de l'utilisateur passent par le serveur Web et celui-ci les redirige vers le code applicatif adéquat. Le navigateur agit alors comme un simple afficheur (ou viewer) de pages HTML et n'exécute aucun traitement (on parle alors de client passif). L'avantage est que toute la maintenance des programmes est centralisée sur le serveur. Ces programmes écrits pour s'exécuter uniquement côté serveur d'application utilisent des technologies très spécifiques (CGI, ASP, PHP, servlets/JSP en Java …) permettant de générer efficacement du HTML dynamique. En dehors du HTML, il faut cependant souligner qu'il est possible d'intégrer dans le navigateur l'utilisation du langage JavaScript et des applets java qui peuvent venir apporter un certain niveau d'interactivité qu'une simple page Web ou un formulaire HTML ne peuvent apporter.

➢ **L'architecture mixte (ou plate-forme de développement)**

Les plates-formes de développement se proposent d'unifier les modèles d'architectures client serveur en éliminant le critère du mode de connexion. Elles sont en mesure de prendre en charge à la fois des architectures 3-tiers de type HTML/Web (fonctionnant en mode connecté, avec une session émulée) et des architectures 2 tiers ou n-tiers (fonctionnant généralement en mode connecté au dessus de TCP) **[L13]**. Les plates-formes les plus importantes sont à l'heure actuelle .NET (qui remplace Windows DNA) et J2EE (Java 2 Enterprise Edition) **[i88]** qui proposent une infrastructure globale, capable de couvrir la plupart des besoins d'un projet d'architecture ouvert sur Internet.

b) La solution choisie :

➢ **Pourquoi la plate-forme J2EE n'a pas été retenue pour le développement de ProteomIs/GnpProt**

On pourrait penser qu'en utilisant une plate-forme de développement telle que .NET et J2EE le problème du choix d'architecture se trouve résolu. NET étant la plate-forme de développement de Microsoft, cette solution propriétaire ne peut en aucun cas être adoptée pour le projet ProteomIs/GnpProt. J2EE semble donc à priori la solution idéale. De plus, J2EE inclut l'utilisation du langage java possédant de nombreux avantages adaptés aux objectifs de notre projet :

- excellente portabilité grâce à l'utilisation d'une machine virtuelle
- richesse des librairies graphique (AWT, Swing) qui font de java un langage de tout premier choix pour le développement d'interfaces élaborées
- le JDK de Sun fournissant une API performante de connexion aux bases de données relationnelles : JDBC.

Cependant le principal intérêt de l'utilisation de J2EE se justifie surtout dans le cadre d'applications distribuées avec l'usage notamment des EJB **[G3]** pour la couche métier. Cependant l'utilisation des EJB n'est pas justifiée dans le cadre de ProteomIs/GnpProt. De plus les EJB et donc l'utilisation de J2EE dans son ensemble sont des technologies lourdes et complexes à mettre en œuvre. De ce fait, il parait plus simple d'utiliser de manière isolée les API java qui pourrait nous être vraiment utiles (JSP/Servlets, Swing, JDBC).

La simplicité des outils fut pour moi un argument de poids dans le choix des solutions de développement choisies. Lorsque j'ai commencé à développer les interfaces, je n'avais qu'une expérience limitée du langage java. J'étais donc débutant et il me paraissait plus judicieux de commencer par utiliser des techniques simples. De plus, j'étais le seul investi dans le développement des interfaces et ne disposait pas de l'aide directe de personnes compétentes en java. Mon objectif était donc d'avoir suffisamment de recul sur ces technologies afin de ne pas tomber dans le piège de l'utilisation d'outils trop complexes et pas vraiment nécessaires.

➢ **Le choix du HTML dynamique pour les interfaces de visualisation d'informations textuelles**

Afin de faire un choix judicieux parmi les solutions offertes, faisons un retour sur les besoins de l'utilisateur. Lors de la phase de maquettage (partie **6.4**) nous avons réalisé l'inventaire des interfaces d'interrogation et de visualisation. Nous avons décrit des interfaces présentant des éléments d'interactivité *(l'interface de navigation dans les gel et le formulaire d'interrogation)* et d'autres (pour la majorité) ayant simplement un aspect d'affichage d'informations statiques *(les interfaces de visualisation d'informations textuelles)*.

En ce qui concerne les interfaces statiques, de nombreuses raisons ont permis d'axer mon choix vers une implémentation en HTML généré dynamiquement. Faisons le point sur les avantages du HTML :

- Les traitements sont sur le serveur, la charge est donc allégée sur le client
- Le HTML est portable sur tous les navigateurs installés par défaut sur les postes clients
- Le HTML possède un fort potentiel pour le rendu de l'aspect graphique
- Il est facile d'extraire l'information textuelle d'une page en HTML
- L'utilisation de liens hypertextes facilite les liens entre les interfaces de visualisation mais également avec d'autres bases de données externes dont le contenu est pour beaucoup en HTML

De plus, il est plus facile de développer des programmes côté serveur tel que des CGI, PHP, JSP/Servlets pour générer des écrans statiques en HTML plutôt que d'implémenter une application tout en java pour le même résultat. Le navigateur propose également tout un ensemble de fonctionnalités auquel est habitué l'utilisateur et qui lui permettent notamment de se déplacer facilement les différents écrans en HTML.

➢ Le choix du couple HTML-JavaScript pour l'implémentation des formulaires

Le problème concerne le formulaire d'interrogation et l'interface de navigation dans les gels qui exige un niveau d'interactivité plus élevé avec l'utilisateur. Dans ce cas là l'usage du HTML ne semble pas adapté. Commençons par le formulaire d'interrogation présenté dans le **document 21**. Un utilisateur souhaite accéder aux informations sur une protéine à partir de son numéro d'accession. Pour interroger la base de donnée, il dispose de trois listes déroulantes :

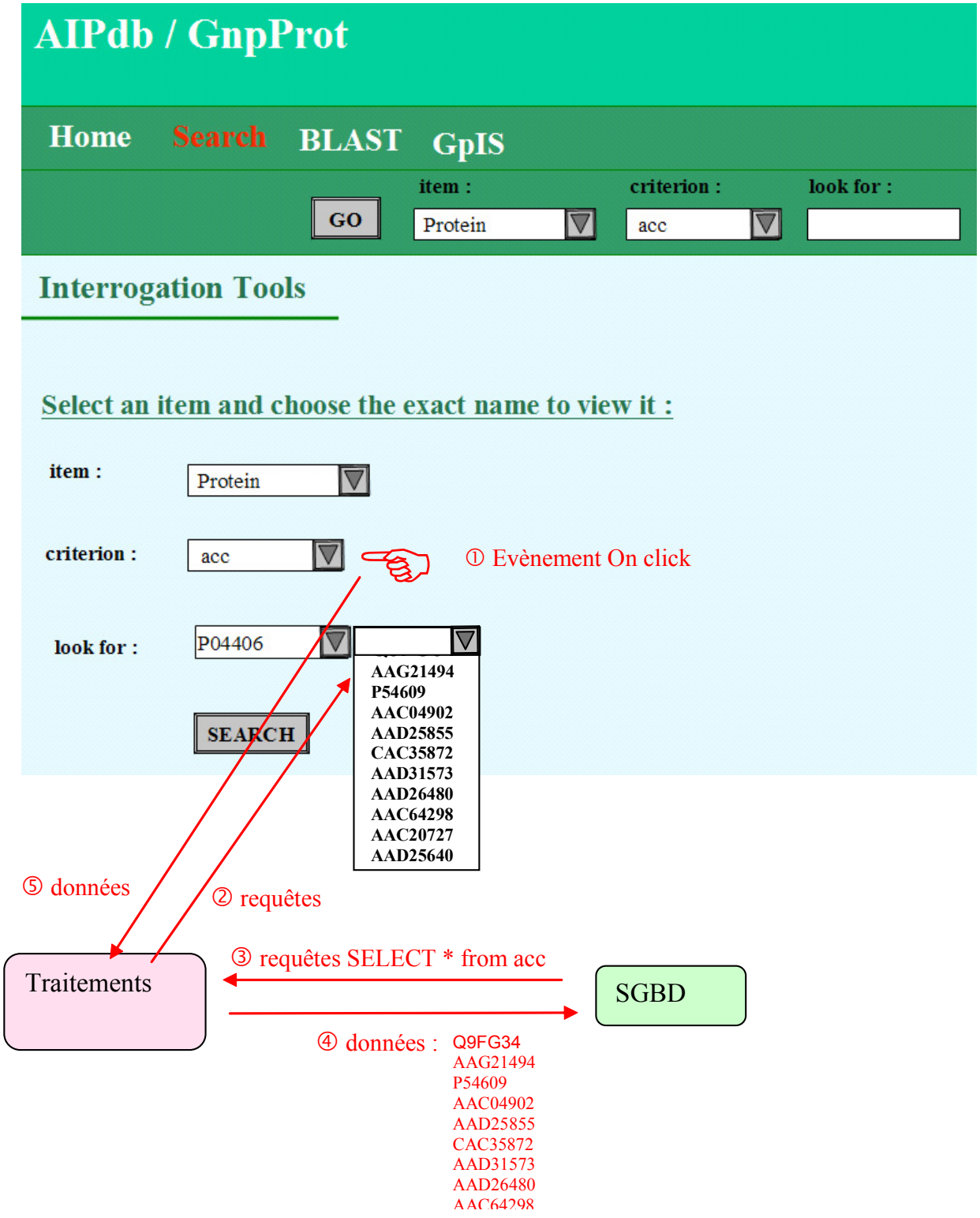
- la liste Item : lui permet de sélectionner le type d'objet recherché (ici la protéine)
- la liste Criterion : lui permet de sélectionner le critère numéro d'accession (acc)
- La dernière liste est la liste Look for qui a pour objectif de présenter à l'utilisateur la liste des accessions disponibles dans la base pour cette protéine

Pour réaliser cette fonctionnalité, l'interface doit être équipée d'un système de gestion des événements. Lorsque l'utilisateur « clique » (événement On click) sur la liste déroulante acc l'interface doit réagir et lancer l'exécution d'une procédure qui ira interroger la base de donnée afin de récupérer les accessions qui doivent alimenter la liste. L'objectif est donc d'intégrer un applicatif proposant l'interactivité nécessaire tout en étant idéalement intégré au sein du navigateur pour être facilement accessible.

Il est possible dans ce cas d'utiliser les applet java pour apporter le plus d'interactivité à une application devant être intégrée à du HTML. Le langage java intègre parfaitement la gestion des évènements dans ses librairies AWT, Swing de composants graphiques pour interfaces et semble donc être à priori une solution plus adaptée. Le problème, c'est qu'une page Web contenant une applet est plus importante en taille et met donc plus de temps à être chargée par le navigateur qu'une simple page HTML. De plus lors du premier accès, une applet doit d'abord être compilée avant d'être chargée dans la page. Tout ceci n'est pas très pratique dans le contexte d'un accès fréquent et rapide au formulaire. Enfin il est beaucoup plus complexe de développer une applet qui devrait communiquer côté serveur avec des programmes qui permettent de générer le code HTML des pages de résultats.

La solution alternative utilisée sera celle du HTML associé au JavaScript. En effet, si l'on décide de créer un formulaire en HTML, on est obligé d'y ajouter une procédure en Java Script qui déclenchera les traitements sur action de l'utilisateur. Cependant le Java Script n'a aucun moyen de communiquer avec les programmes côté serveur car son champ d'action se limite aux objets du navigateur.

La technique sera donc dans ce cas là, d'activer le bouton SUBMIT (éventuellement caché) d'un formulaire dans la page qui pourra lancer l'exécution d'un programme côté serveur.



Document 21 : Interface d'interrogation

➢ Le choix de l'utilisation d'une applet pour l'interface de navigation dans l'image des gels

Le choix de la combinaison HTML+JavaScript étant fait pour le formulaire, il reste maintenant à examiner la solution technique à utiliser pour l'interface de navigation dans l'image d'un gel. Dans ce cas là, l'utilisation d'une applet java était satisfaisante. En effet je disposais au préalable d'une applet java de visualisation (viewer) de gel qui m'avait été fourni par les concepteurs de la base de donnée PPMDB. Dans le **document 22** je montre en trait pointillé vert la partie de l'applet viewer qui a été construite à partir de l'interface de PPMDB. Il s'agit de la fonction permettant d'afficher les informations contextuelles sur les spots dans le panneau de droite et aussi d'ouvrir l'interface de visualisation des spots à partir des croix hypertexte des spots en rouge.

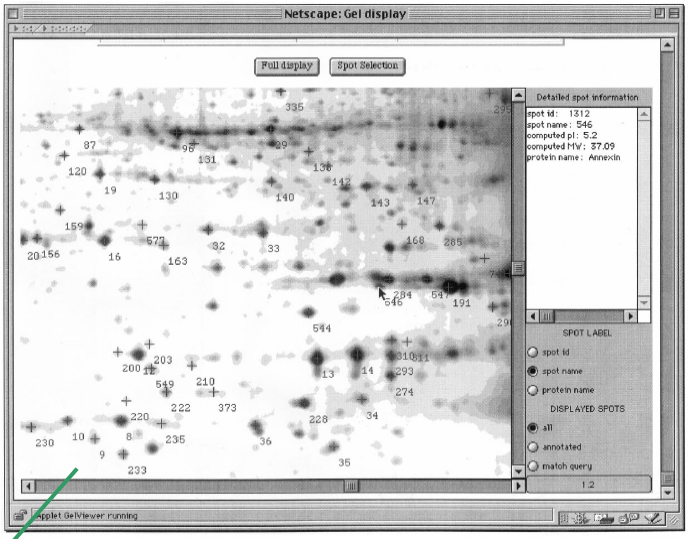
Les ascenseurs encore visibles sur la maquette seront supprimés dans la version finale de l'application ; ceci au profit du mode de déplacement à partir du rectangle jaune dans le cadre en trait pointillé rose. Le rectangle jaune définit la partie visible de l'image sur le panneau de gauche. En faisant glisser ce rectangle sur l'image réduite, l'image de gauche sera déplacée de manière synchronisée. Ainsi on peut se déplacer en ayant une vue précise de la position dans laquelle on se trouve sur l'image.

Le code source de cette fonctionnalité a pu être téléchargé sur le site : http://www.serve.com/wizjd/java/JDScrollingImagemap/index.html. Ce code y était notamment utilisé pour faciliter l'exploration de cartes astronomiques.

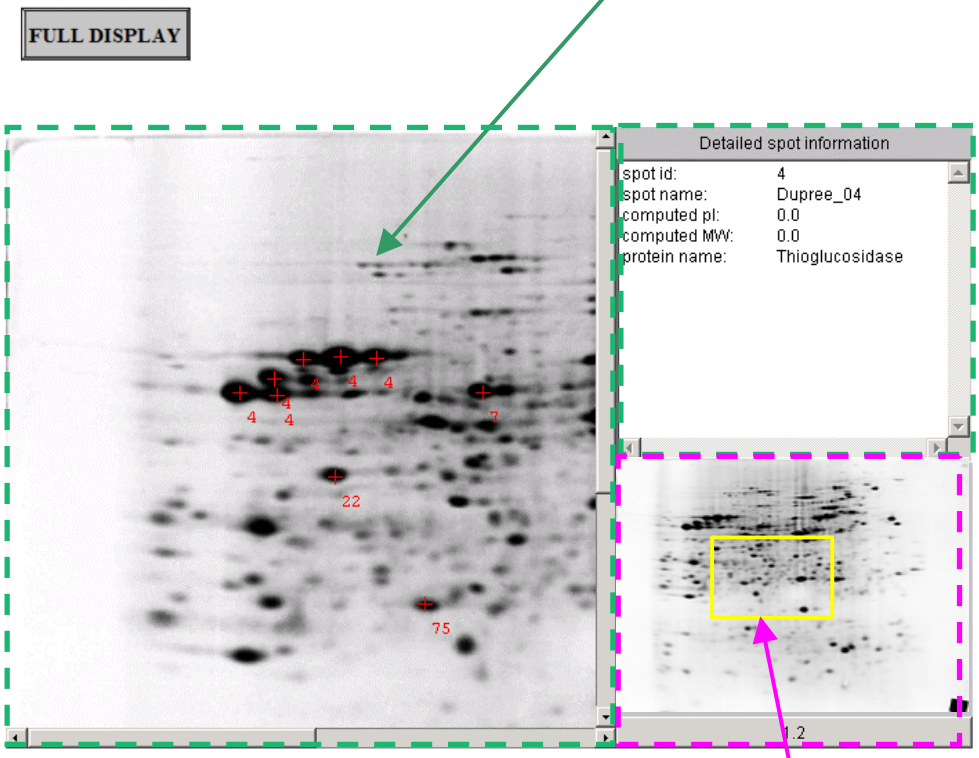
Le langage java étant objet il était alors pratique de réutiliser et assembler ces différents composants pour implémenter l'interface de navigation dans les gels *(en **annexe 19** est placé le diagramme des classes du programme)*. De plus, étant débutant en java, étudier le code source d'autres applications fut pour moi très instructif. Bien sûr il a fallu apporter certaines modifications pour adapter les différents composants et j'ai dû ensuite, ajouter d'autres fonctionnalités prévues comme le positionnement automatique sur un spot ou une protéine à l'ouverture de l'applet (fonctionnalité décrite dans la partie **7.2**). Une des difficultés fut de passer en paramètre à l'applet la liste des spots qui devaient être dessinés par des croix sur l'image du gel. La valeur des paramètres d'une applet est en fait spécifiée dans le fichier HTML. Ces valeurs sont ensuite récupérées depuis le programme Java avec la méthode `getParameter()`. La solution était de générer dynamiquement la page HTML contenant la liste des paramètres en faisant appel à un programme (CGI, PHP ou JSP …) côté serveur ; ce programme devant pour cela récupérer la liste des spots pour le gel dans la base de données grâce à une requête SQL : `SELECT * from spot where gel_id = x`

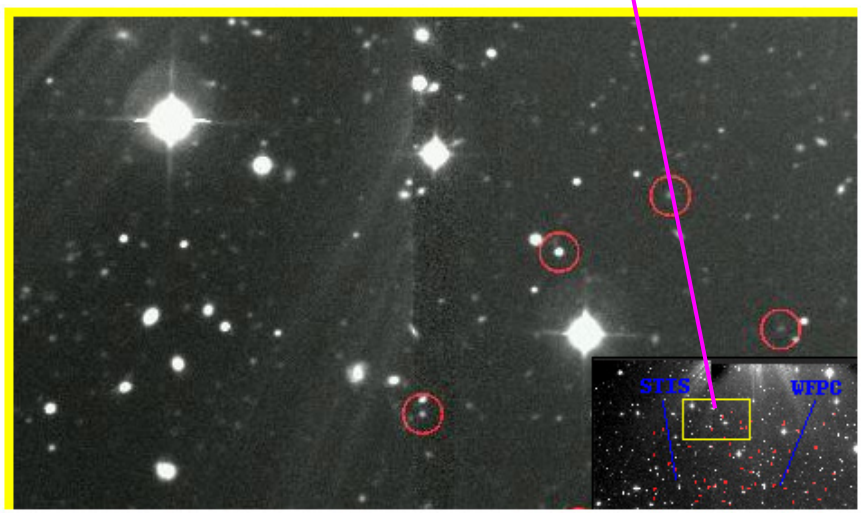
Afin de faciliter la programmation de l'applet Java, j'ai utilisé un environnement de développement intégré (Integrated Development Environment en anglais). Il s'agit du logiciel JBuilder développé par Borland, dont une version est disponible en version d'évaluation sur le site http://www.borland.fr/jbuilder/

Document 22 : Réutilisation de composants pour le développement de l'interface de navigation dans l'image des gels



« Viewer » de gel de PPMDB (source : [a11])





Applet permettant d'explorer une carte astronomique
(source : http://www.aao.gov.au/hdfs/Redshifts/hdfmap.html)

➢ **Récapitulatif sur l'architecture choisie : le modèle 3-tiers HTML/Web**

Au terme de cette discussion nous pouvons dire que le choix de l'architecture logicielle de ProteomIs/GnpProt s'est porté sur le modèle **3-tiers HTML/Web** avec :

- utilisation de programmes de génération dynamique de page HTML (CGI, PHP ou JSP/Servlets) pour construire les interfaces de visualisation et les formulaires (login, interface d'interrogation)
- utilisation dans certains cas du JavaScript pour la gestion des évènements
- utilisation d'une applet java pour construire l'interface de navigation dans les gels

De plus, nous verrons dans la partie **8.2 Perspectives** qu'il est prévu d'implémenter une application en java permettant de visualiser graphiquement les motifs dans les séquences de protéines de ProteomIs/GnpProt. Même si ce dernier point est à l'état de projet, il reste néanmoins un prérequis qui permet de nous orienter vers une architecture suffisamment modulaire dont l'objectif premier est de faciliter l'intégration et la réutilisation de différents composants. Nous verrons dans la suite de ce mémoire, comment le modèle architectural retenu sera perfectionné pour répondre à ces deux derniers critères.

6.5.2.2 Le choix des JPS/Servlets pour la génération de pages HTML dynamique

La problématique est la suivante : il s'agit de créer des pages HTML dynamiques, c'est à dire des pages Web crées en réponse à la demande des utilisateurs, pages dont la forme est fixe et le contenu relatif à celui de la base de données est variable. Par conséquent, ce système permet d'adapter le contenu aux critères de recherche de l'utilisateur. Nous avons vu en présentant le modèle architectural 3-tiers HTML/Web qu'il existait un nombre assez important de technologies réalisant ces traitements côté serveur. Nous nous proposons ici de justifier le choix de la solution JSP/Servlets par rapport aux autres technologies.

➢ **Les Servlets, avantage sur les scripts CGI**

Avec le langage Java est apparue la technologie des servlets **[i89] [i90]**. Ce sont des programmes Java s'apparentant aux CGI **[i91] [i92]** et fonctionnant côté serveur. Les servlets utilisent le package *javax.servlet* et *javax.servlet.http* fournit dans le JSDK. Hormis le fait que les servlets possèdent tous les avantages inhérents au langage java, les servlets corrigent certaines faiblesses des CGI. Par exemple, le langage Perl qui est le plus souvent utilisé dans les CGI, est interprété à chaque appel, ce qui requiert beaucoup de ressources en comparaison de l'exécution du code compilé d'une servlet.

Ensuite, après leur création les servlets restent activent en mémoire grâce à des threads, tandis qu'avec les langages de scripts traditionnels comme Perl un nouveau processus est crée pour chaque requête http. Les performances sont donc améliorées avec les servlets au niveau du serveur. Enfin, le fait que les servlets restent en mémoire permet de conserver l'état du serveur entre les requêtes (utilisation de l'objet HttpSession). Ceci peut être utile par exemple, dans le cas de pages successives aboutissant à la préparation d'une commande sur un site d'achat en ligne. Au contraire, avec les CGI on doit développer notre propre prise en charge de session avec par exemple l'utilisation de cookies ou d'url codées, ceci pour mémoriser l'état d'un utilisateur entre deux requêtes (http étant un protocole non connecté).

➢ **Les Java Server Pages, avantage sur les Servlets et la technologie PHP**

Les Java Server Pages, connues sous le nom de JSP **[i93] [i84]**, constituent une technologie crée par Sun Microsystems et font partie intégrante du JSDK. Les JSP sont en fait une amélioration des servlets Java en permettant au développeur d'inclure du code Java dans des documents HML.

Ceci évite comme avec les Servlets ou les CGI, d'écrire dans les programmes des lignes de code pour afficher chaque ligne en langage HTML, ce qui permet de créer ainsi rapidement des pages web dynamiques. En **annexe 21** est présenté la comparaison entre un programme écrit en Servlet et le même programme écrit à l'aide de JSP.

En PHP, le code est également directement inclu dans la page HTML (technique de scripting). Cependant JSP est plus avantageux car il permet l'emploi de balises spécifiques ce qui facilite grandement la maintenance de l'aspect graphique des pages HTML. Grâce à l'utilisation d'accesseurs fournis par ses balises spécifiques, JSP peut être utilisé pour accéder aux méthodes de composants réutilisables comme les JavaBeans **[G2]**. Il est alors possible de séparer les aspects présentation (directement codées en HTML dans la JSP) de la logique applicative (traitements) contenue dans les JavaBeans.

➢ **Conclusion sur la solution retenue**

Au final la technologie retenue pour générer des pages web dynamiques est l'association des JSP/Servlets avec les JavaBeans. La possibilité avec JSP de séparer l'aspect présentation des traitements est un des arguments qui m'a le plus convaincu. Ayant auparavant programmé en Perl CGI pendant 18 mois, mon expérience m'a permis de me rendre compte combien il était difficile de maintenir et réutiliser des composants de programmes mélangeant les deux aspects.

Cependant l'investissement fut conséquent car je découvrais une nouvelle façon de coder des applications web. Afin de faciliter la programmation, j'ai utilisé un environnement de développement intégré. Il s'agit du logiciel Netbeans développé par Sun, et diffusé gratuitement en open-source sur le site www.netbeans.org

6.5.2.3 Choix du modèle MVC2 et utilisation du Framework Struts

➢ **Présentation du design pattern MVC**

Nous avons vu précédemment comment un des intérêts forts de l'utilisation des JSP était de pouvoir séparer la couche présentation de la couche traitement en combinant l'utilisation des JSP avec celle de JavaBeans. Dans le cadre de la conception d'une application importante comme ProteomIs/GnpProt il était intéressant d'utiliser le design pattern MVC (Modèle Vue Contrôleur) qui faciliterait le maintient de cette séparation présentation-traitement au niveau de l'architecture logicielle.
Dans le modèle MVC on distingue trois entités dont le rôle est défini de la façon suivante :

- le Modèle décrit les données manipulées par l'application et définit les méthodes d'accès ;
- la Vue définit l'interface utilisateur et la présentation ;
- le Contrôleur prend en charge la gestion des événements de synchronisation pour mettre à jour la vue ou le modèle.

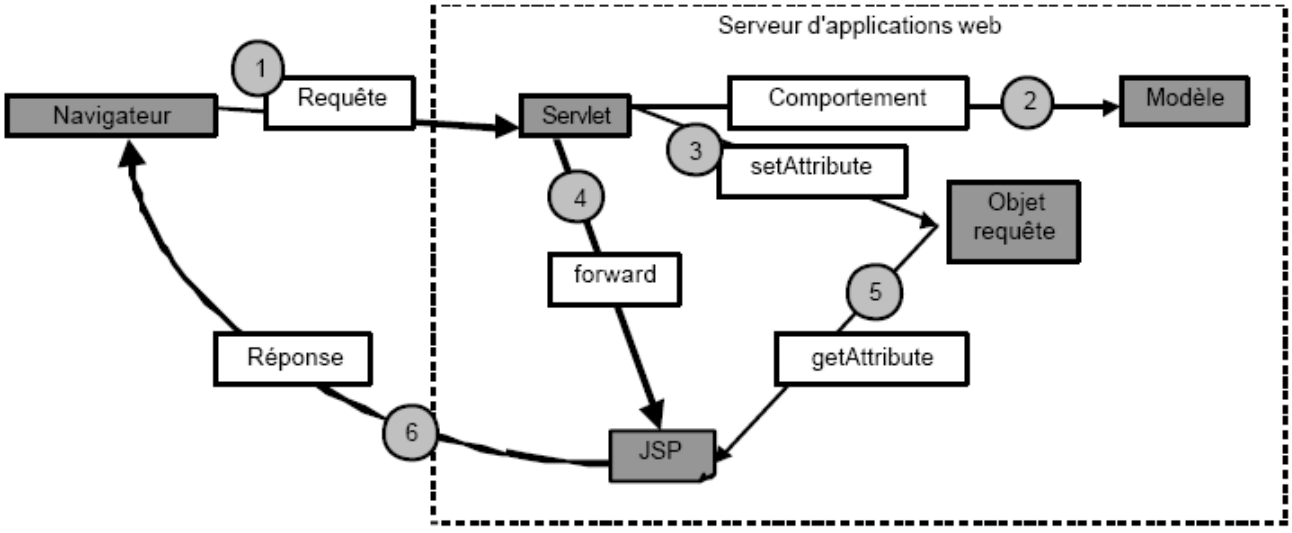
Cette décomposition permet de minimiser l'impact des modifications, généralement fréquentes, de la Vue sur le Modèle, assurant ainsi une certaine stabilité du logiciel.

➢ **Utilisation du modèle MVC dans les applications Web et passage au modèle MVC2**

Le modèle MVC est une avancée importante en terme d'architecture d'applications web. Elle permet entre autre de faire collaborer une équipe à consonance infographie sur les aspects design des pages HTML et une équipe à consonance informatique sur la partie traitement.

Le **document 23** **[i36]** présente une implémentation de MVC avec les technologies Servlet et JSP :

1. un navigateur lance une requête sur un serveur d'applications web. La réception de cette requête est assurée par un servlet.
2. Le servlet sous-traite la partie traitement de cette requête au modèle. Le modèle est assuré par un ensemble de classes java. Ces classes java accèdent au système d'information afin de récupérer les données requises par le traitement de la requête.
3. Le servlet récupère les données du modèle qui devront être renvoyées à l'utilisateur et les dépose dans un objet accessible par le jsp (généralement un JavaBeans).
4. Le servlet sous-traite l'affichage des données en provenance du modèle à un JSP.
5. Le JSP récupère les données en provenance du servlet dans un objet (le JavaBeans).
6. Le HTML généré par le servlet est renvoyé au navigateur



Document 23 : MVC classique dans les applications Web J2EE.

Cette configuration n'est cependant pas encore idéale : elle oblige à écrire une multitude de servlets, qui sont autant de points d'entrée dans l'application.

➢ Les framewoks MVC2 : le choix de Struts

Pour palier à cet inconvénient, des frameworks **[G4]** ont été développés. Ces frameworks (**annexe 20**) sont composés d'un seul servlet contrôleur sont regroupés sous l'étiquette "Model 2" encore appelé "MVC2". Ainsi un de ces framework MVC2 sera choisit pour l'implémentation du modèle MVC dans le projet ProteomIs/GnpProt. Il s'agit du framework Struts **[i49]**. Struts comprend les composants suivants :

- Un contrôleur facilement configurable permettant d'associer des actions (méthode d'un objet Java) à des requêtes HTTP.
- Des librairies de tags spécifiques pour créer facilement une vue.
- Un Digester, permettant de parser un fichier XML et d'en récupérer seulement les informations voulues.
- Des utilitaires permettant de remplir automatiquement des champs et de créer des applications supportant plusieurs langages.

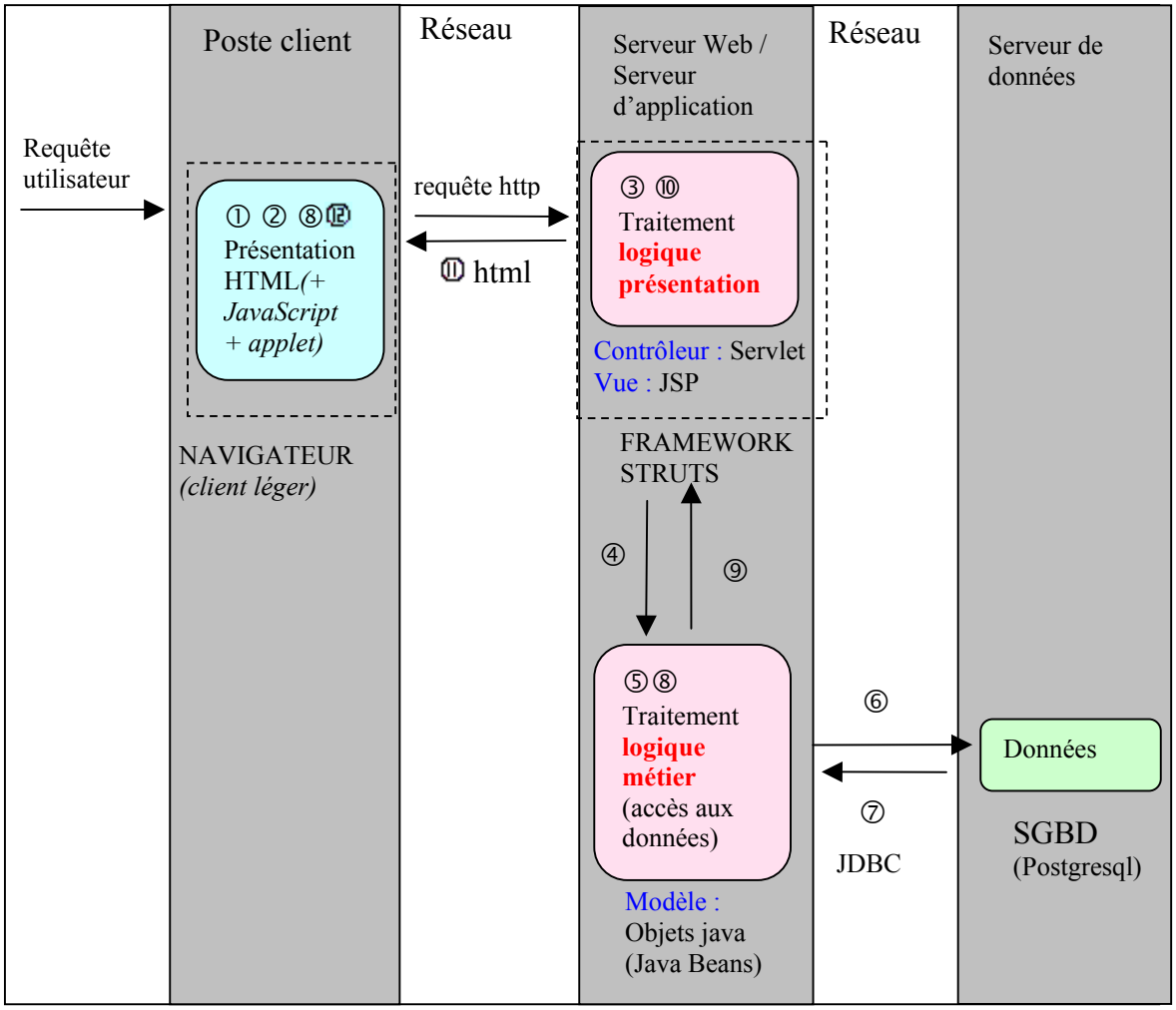
Struts a été choisi comme framework MVC2 pour pour plusieurs raisons :

- il s'agit d'un projet Open Source développé par la communauté Jakarta d'Apache, ce qui garantit sa pérénité
- sa popularité fait qu'il est très documenté et permet d'accéder à de nombreuses ressources
- Le framework Struts est utilisé pour le développement des autres modules du système GpiIS de Génoplante. Ce framework ne m'a pas été imposé par Génoplante mais la possibilité de pouvoir bénéficier de l'expérience positive d'une autre équipe de développement sur le sujet a été pour moi une motivation importante.

6.5.2.4 Conception détaillée et implémentation

Les interfaces d'interrogation et de visualisation de ProteomIs/GnpProt ont donc été développées en combinant JSP, Servlets et Java Beans à l'aide du framework Struts. Grâce à cette implémentation du modèle MVC2 , le modèle architectural 3-tiers choisi pour ProteomIs/GnpProt s'est perfectionné. Il nous est possible comme, montré sur le **document 24**, de scinder la partie applicative (en rose) en deux couches :

- une couche applicative orientée vers la gestion de la présentation (en utilisant le couple JSP/Servlets à l'aide de Struts)
- une couche applicative orientée vers la logique métier (en utilisant des objets Java comme les JavaBeans pour faciliter l'accès aux données)



Document 24 : L'architecture de ProteomIs/GnpProt

Cette logique va nous permettre de favoriser la maintenance et la réutilisation des composants. Par exemple la couche métier pourra être réutilisée pour l'application java de visualisation des motifs décrite dans la partie **8.2 Perspectives**. Cependant dans notre architecture le framework Struts ne prend en charge que l'implémentation de la couche traitement orientée présentation. La couche traitement orientée métier pour l'accès aux données restait donc à développer.

Dans la partie suivante nous allons présenter la conception détaillée et l'implémentation des différentes composantes de notre application à savoir le contrôleur, le modèle et la vue. Nous commencerons par décrire le fonctionnement du contrôleur Servlet de Struts dans le contexte de l'implémentation du Modèle MVC2 dans ProteomIs/GnpProt. Ensuite, nous présenterons l'implémentation du Modèle (la couche d'accès aux données) dans ProteomIs/GnpProt en insistant au préalable sur les solutions qui m'étaient offertes dans le contexte du projet. Enfin, nous décrirons la création des Vues à travers l'usage des JSP et de balises spécifiques.

a) Le contrôleur – fonctionnement général de l'architecture

Dans l'exemple du **document 25**, le contrôleur est appelé par l'interface d'interrogation qui se présente sous la forme de trois listes déroulantes permettant d'interroger la base sur des critères spécifiques. Dans notre exemple, l'utilisateur recherche un projet à l'aide de son code. Toutes les requêtes http ① vont alors aboutir au contrôleur dont le point central dans Struts est un servlet de la classe *ActionServlet.* Celui-ci va alors choisir la servlet *Action* adaptée au type de la requête.

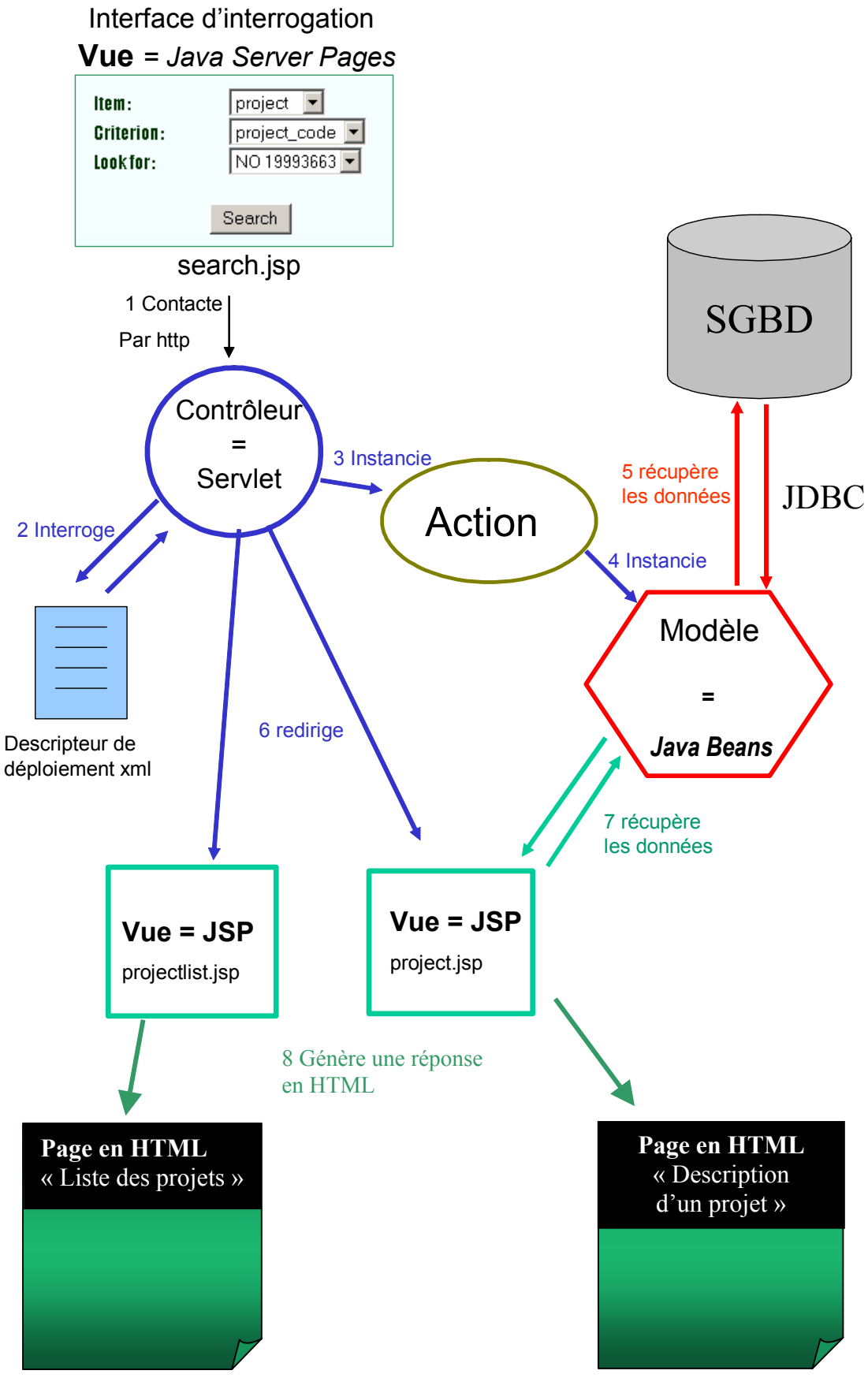
En effet dans ProteomIs/GnpProt les requêtes peuvent soit provenir de la validation d'un formulaire soit de l'activation d'un lien hypertexte. Chacune des actions de l'utilisateur est prévue pour être traitée par une servlet différente appelée servlet *Action*. Grâce au descripteur de déploiement *strutsconfig.xml* ②, le contrôleur sait quel objet « *Action* » il doit invoquer pour traiter la requête.

L'objet *Action* une fois instancié ③ va permettre d'instancier les Java Beans ④ nécessaires à la construction de la vue. Ces Java Beans ont pour objectif de jouer le rôle de conteneur des données contenues dans la base. Ils devront pour cela être initialisés au préalable à l'aide d'une requête SQL sur la base de données ⑤.

Ensuite le contrôleur va jouer un rôle d'aiguilleur en orientant la réponse vers telle ou telle jsp ⑥ en fonction du nombre de ligne de résultats récupérés par les Java Beans.

Si par exemple il existe plusieurs projets dans la réponse à la requête, le contrôleur va orienter directement sur la JSP *projectlist.jsp*, tandis que si il n'existe qu'un seul projet, le renvoie se fera sur la JSP *project.jsp* qui ne représente plus une liste de projets mais la totalité des informations d'un projet.

La JSP sélectionnée est alors compilée en servlet et le contenu de la page HTML⑧ générée à partir des données ⑦ contenues dans les JavaBeans. C'est grâce à l'utilisation de balises spécifiques que la JSP peut lire les propriétés de ces objets. Ces propriétés sont en fait des méthodes appelées aussi accesseurs (méthode *get()* et *set()*) qui permettent d'accéder aux données des JavaBeans.



<u>Document 25 :</u> Implémentation de MVC2 dans ProteomIs à l'aide de Struts

b-1) Les objets d'accès aux données

Il s'agit maintenant de concevoir la couche métier d'accès aux données. Celle-ci aura simplement pour objectif d'accéder aux données grâce à l'utilisation de JavaBeans. Il n'y aura pas d'opération de suppression ni de modification dans les données, ces opérations se faisant par l'intermédiaire du format d'échange. Nous décrirons tout d'abord l'approche classique utilisée pour implémenter les beans d'accès aux données. Nous décrirons ensuite plusieurs méthodes pouvant être utilisé pour initialiser ces beans avec les données de la base. La difficulté majeure étant ensuite de générer le code de tous ces beans, nous présenterons plusieurs alternatives à cela (ORM de Génoplante, Hibernate). Enfin nous expliquerons pourquoi ces solutions ont été abandonnées au profit d'une autre architecture que nous détaillerons par la suite.

➢ Approche classique pour l'implémentation des beans d'accès aux données

L'objectif est ici de créer des beans permettant d'accéder aux données de la base grâce à des méthodes *set()* qui initialiseront les JavaBeans. Ces objets vont en quelque sorte jouer le rôle de conteneur de données pour alimenter les JSP. Pour accéder aux données des JavaBeans, les JSP utiliseront les méthodes *get()* de ces objets par l'intermédiaire des balises spécifiques d'accès (voir **annexe 21** l'exemple de la JSP *compteur.jps* utilisant ce type de balise).

Pour créer ces beans, l'approche classique est de créer une classe pour chaque table existante dans la base de données **[L2]**. Ensuite on crée les méthodes d'accès adéquates prenant autant de paramètres que la table possède de champs. Prenons l'exemple de la table project (en vert ci-dessous) présente dans la base ProteomIs/GnpProt :

project		
project_id	numeric(38,0)	<pk>
project_code	varchar(15)	
project_name	varchar(200)	
status	numeric(2,0)	
title	varchar(255)	
genoplante	numeric(1,0)	
creation_date	date	
closure_date	date	
short_remark	varchar(1000)	

Une telle table pourrait avoir la classe équivalente Project (en rouge ci-contre) dans la couche d'accès aux données.

Ensuite chacun de ces beans possède respectivement :

- une méthode d'accès aux données

(exemple : *getproject_code()* qui renvoie le code du projet)

- une méthode d'initialisation des données

(exemple : *setproject_code()* qui initialise le code du projet)

La méthode *get()* est très simple à programmer puisqu'il s'agit simplement d'accéder à la données contenue dans l'attribut *project_code* :

```
public String getproject_code() {
      return project_code ;
}
```

En ce qui concerne la méthode *set()*, le code le plus simple pour l'implémenter devrait être le suivant :

```
public String setproject_code(String project_code) {
      this project_code = project_code ;
}
```

Dans ce cas le paramètre project_code de cette méthode doit avoir été initialisé auparavant.

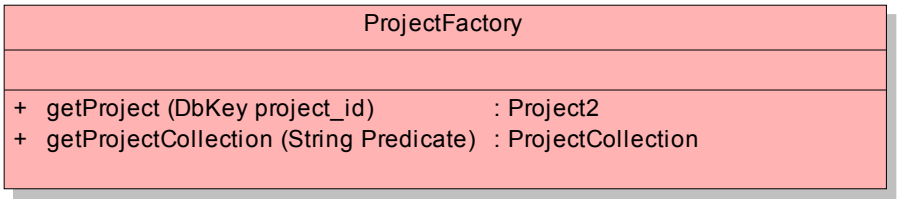
➢ Méthodes d'initialisation des beans d'accès aux données : utilisation d'une fabrique d'objet

Nous présenterons deux approches pour initialiser les beans à partir des données de la base. Dans la première approche les beans pourrait être directement initialisé à partir de la Servlet *Action*. On obtiendrait alors le code suivant :

```
conn = dataSource.getConnection() ;
Stmt = conn.createStatement() ;
rs = stmt.executeQuery(" select * from project where
project.project_id="+project_id);
project = new Project ();
project.setproject_code(rs.getString("project_code");
```

L'exécution de la requête SQL se fait donc ici dans la servlet *Action*. Ceci n'est pas satisfaisant car le rôle essentiel de la servlet Action est de jouer le rôle de contrôleur. On n'a alors plus de séparation entre la logique de présentation réservée aux servlets et jsp avec Struts et la logique métier d'accès aux données.

Dans la deuxième solution envisagée le beans *Project* sera initialisé par l'intermédiaire d'une autre classe afin d'obtenir clairement cette séparation. Cette classe, portant souvent le nom de **fabrique d'objet** (souvent appelée ObjectFactory), va permettre de créer une instance du bean *Project* pour chacune des lignes de la table project. Cette classe va alors jouer en quelque sorte le rôle d'adaptateur entre les beans et la source de données : le SGBD. L'utilisation du design patterns Fabrique abstraite (Abstract Factory) et Fabrication (Factory Method) peut éventuellement être utilisé pour faciliter la réalisation d'une « Fabrique d'objets » **[L6]**. Ces fabriques d'objets portent généralement le nom de DAO (Data Access Objects) **[i44]**. Voici à quoi pourrait ressembler la classe fabrique d'objet ProjectFactory générant les objets Project :



La méthode *getProject ()* de cette classe permet de récupérer simplement l'instance correspondante d'un beans *Project* en lui passant un identifiant project_id. Le beans récupéré contient alors les données de l'enregistrement dans la table project ayant pour clé primaire project_id. La méthode *getProjectCollection()* elle a pour rôle de renvoyer une classe collection contenant un ensemble d'instances de la classe *Project*. La récupération des données dans le SGBD peut-être réalisée à l'aide de requêtes SQL exécutées par l'intermédiaire de ces méthodes. Les données des beans générés par la fabrique d'objets deviennent ensuite accessibles par les balises des JSP faisant référence aux méthodes d'accès de ces beans. Nous obtenons ainsi comme prévu une couche d'accès aux données indépendante de la logique de présentation de Struts.

Nous avons vu cependant que pour accéder aux données de la table project il fallait créer les classes Project, ProjectCollection et la classe fabrique ProjectFactory. Le problème est maintenant de coder toutes ces classes pour chacune des tables de la base.

Sachant que dans le modèle conceptuel de ProteomIs/GnpProt il y'a 54 tables, cela demande un travail trop important pour une seule personne. Ceci est d'autant plus fastidieux que pour initialiser chaque objet concerné il sera nécessaire d'écrire le code SQL adéquat dans d'autres classes utilitaires qui pourraient être utilisées par les classes fabriques. De plus il n'est pas toujours évident de créer une structure de données objet calqué sur celui d'une base de données relationnelle. La problématique est donc maintenant de générer le code des beans d'accès aux données. Une alternative consiste à utiliser des outils appelés ORM facilitant l'implémentation des concepts que nous venons de décrire.

➢ Discussion autour de la solution ORM commerciale proposée par Génoplante

Un ORM (outil commercial de mapping objet-relationnel ou object-relationnal mapping en Anglais) est un outil visant à simplifier la création d'une couche d'accès aux données, voire à en automatiser le fonctionnement ou la génération. Il met en correspondance bidirectionnelle les données situées dans une base de données relationnelle et des objets au niveau du code (mapping objet-relationnel **[i137]**), en se basant sur une configuration et en exécutant des requêtes SQL (le plus souvent dynamiques) sur la base de données **[i45]**.

Afin de faciliter la réalisation de la couche d'accès aux données de ProteomIs/GnpProt, Génoplante m'a proposé d'utiliser un ORM commercial. Celui-ci avait été développé par la SSII SYSRA **[i46]** pour les besoins de Génoplante, afin de concevoir la couche métier d'accès aux données des modules de GpiIS. Cet outil se base sur l'utilisation d'une fabrique d'objet comme décrit précédemment. Cependant celui-ci utilise un générateur de code programmé dans le logiciel Rational Rose. Le générateur , à partir du diagramme de classes correspondant au Modèle Conceptuel de Données de la base, est capable d'automatiser 90 % de l'écriture du code. Le travail de mise correspondance entre les objets et les tables est ainsi automatisé.

Ayant accepté d'étudier la proposition de Génoplante, le code source de la couche métier d'accès aux données d'un des modules de GpiIS (le module carto) m'a alors été fournit. Ce code était accompagné d'une documentation succincte (présenté **annexe 22**) décrivant dans les grandes lignes le système. Cependant après avoir étudié ce code source pendant quelques jours j'ai décidé de ne pas utiliser la solution proposée par Génoplante. pour plusieurs raisons :

La raison principale est la complexité du système qui m'était offerte. La quantité de code source généré par le générateur était de manière tout à fait logique très importante. Cela me semblait à priori très fastidieux d'imaginer seul pouvoir maintenir facilement une telle quantité de code source. De plus cela m'était très difficile d'établir une cohérence entre les nombreux composants de l'applicatif. Un autre problème est que je ne disposais pas dans mon laboratoire du logiciel commercial Rationnal Rose **[i55]** pour générer mes classes. J'aurais donc été obligé de solliciter Génoplante pour générer toutes les classes de mon application après chacune des modifications sur le modèle conceptuel de données.

J'ai donc décidé d'implémenter ma propre couche métier d'accès aux données en m'orientant vers une approche plus générique conduisant à produire beaucoup moins de code.

➢ Une alternative : l'ORM open source Hibernate

Avant de décrire la couche d'accès aux données, nous présenterons l'ORM open source Hibernate **[i47]**. Bien que cette solution n'ait pas pu être envisagée car découverte trop tardivement, j'explique ici en quoi cet outil aurait pu représenter une alternative intéressante au développement de ma propre couche d'accès aux données. La première raison est que cet outil très populaire offre une quantité importante de documentations permettant de faciliter son utilisation. Il est soutenu par une grande communauté de développeurs qui contribuent à maintenir et faire évoluer le système.

D'autres éléments sont intéressants, comme par exemple la possibilité de pouvoir configurer, non plus dans le code mais dans un fichier de mapping XML, tous les aspects servant à relier les tables avec les objets Java beans. Hibernate prend également en charge tous les aspects de pagination (navigation dans les pages de résultats) et également l'algorithme d'alimentation d'un cache objet (que nous verrons plus loin) ; différents aspects que j'ai dû coder dans ma couche d'accès aux données.

Hibernate dispose ensuite de son propre langage de requêtage HQL adapté au monde objet et dispose d'une couche d'abstraction qui permet une totale indépendance vis à vis du SGBD avec lequel il doit accéder. Dans le cadre de ProteomIs/GnpProt cela aurait été un avantage sachant que la couche d'accès aux données devra être capable à terme de communiquer à terme avec Oracle en plus de PostgresSQL.

Hibernate peut également être couplé avec un autre framework appellé Spring **[i48]** permettant entre autre de gérer tous les aspects transactionnels, les relations entre les beans d'accès aux données. Le couplage de ces différents framework peut constituer une alternative intéressante aux EJB de J2EE pour la conception de couches d'accès aux données très évoluées devant répondre à des contraintes spécifiques de transaction, gestion de la concurrence, persistance …

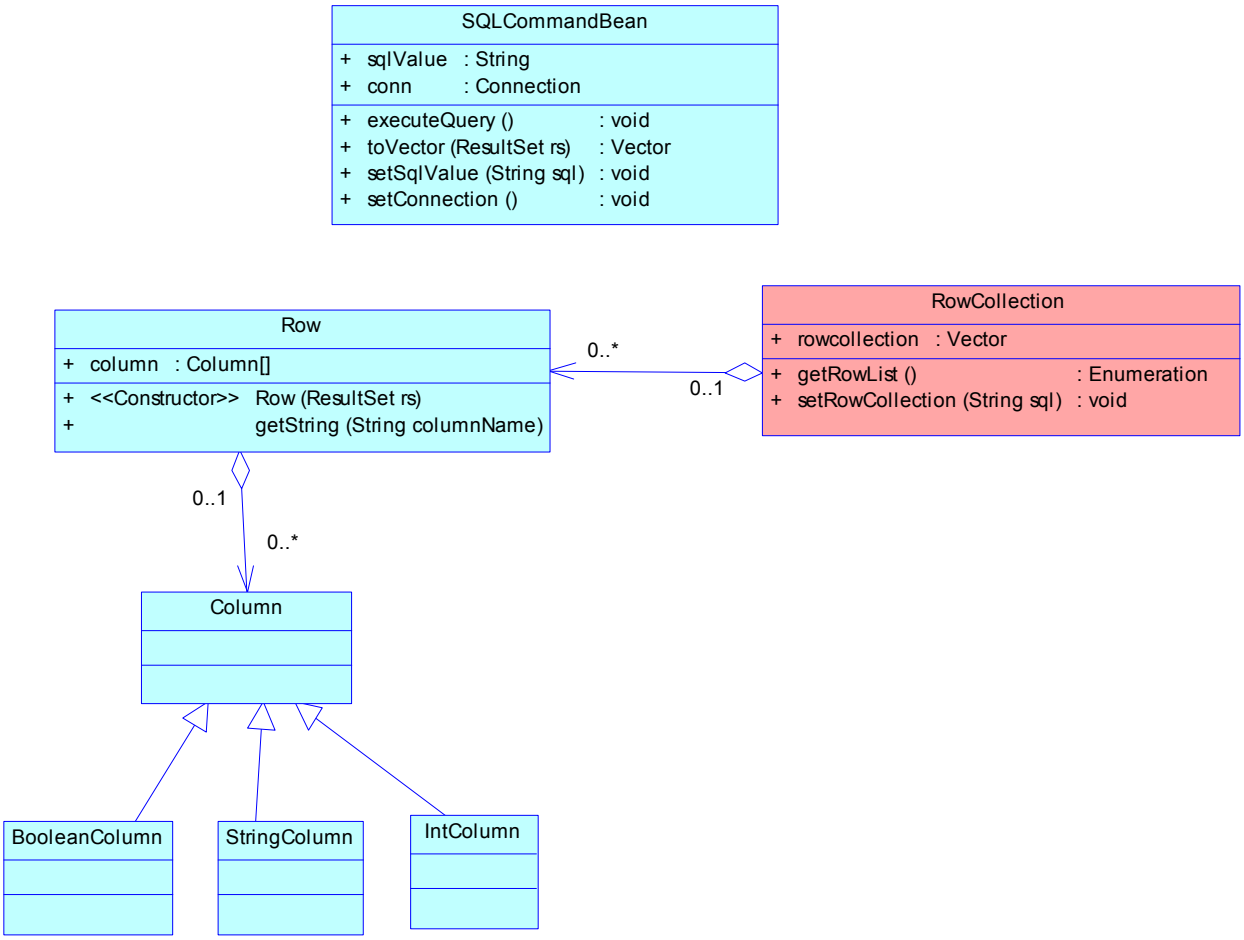
Il existe également d'autres ORM open source comme OJB (*ObJectRelationalBridge)* **[i50]** et quelques autres possédant chacun des spécificités qui peut orienter le choix vers un de ces outils plutôt qu'un autre **[i45]**. Cependant Hibernate est sans doute à l'heure actuelle l'un des ORM les plus populaires et les plus évolué ce qui en fait un outil très largement utilisé dans la conception des couches d'accès aux données.

Génoplante a notamment récemment abandonné l'usage de son ORM commercial pour adopter l'utilisation de Hibernate. Le logiciel Rational Rose a lui aussi été remplacé au profit d'une solution open source : c'est maintenant le logiciel ArgoUML **[i54]** qui permet la génération des classes mais aussi la génération du fichier de mapping XML d'Hibernate.

➢ La solution retenue : implémentation d'une couche d'accès aux données basée sur l'utilisation de beans génériques

Nous allons voir maintenant de quelle manière j'ai implémenté ma propre couche d'accès aux données tout en respectant les concepts du modèle MVC. La solution choisie fut d'utiliser une classe capable, à partir d'une requête SQL de générer automatiquement des instances d'un seul type de bean capable de contenir les données de n'importe quelle table de la base. Pour cette raison on qualifiera ces beans de générique. Ainsi, au lieu d'avoir comme dans les ORM classiques un type de bean par table, on a cette fois un seul type de bean pour toutes les tables de la base. J'ai trouvé le code source de cet objet dans un package de classes téléchargeables sur le site de la première édition du livre : JavaServer Pages de chez O'Reilly écrit par Hans Bergsten *(édition 2001)* **[L1] [i78]**. Sur le **diagramme 14** est représenté le diagramme UML des classes que j'ai utilisé pour implémenter la couche d'accès aux données de ProteomIs/GnpProt.

Diagramme 14 : Diagramme de classe : couche d'accès aux données



Seules les principales méthodes ont été représentées. Le diagramme de collaboration, sur le **diagramme 15**, récapitule les interactions entre ces classes. Les classes en vert ont été écrites par Hans Bergsten. Elles ont été adaptées et modifiées par la suite pour les besoins du projet.

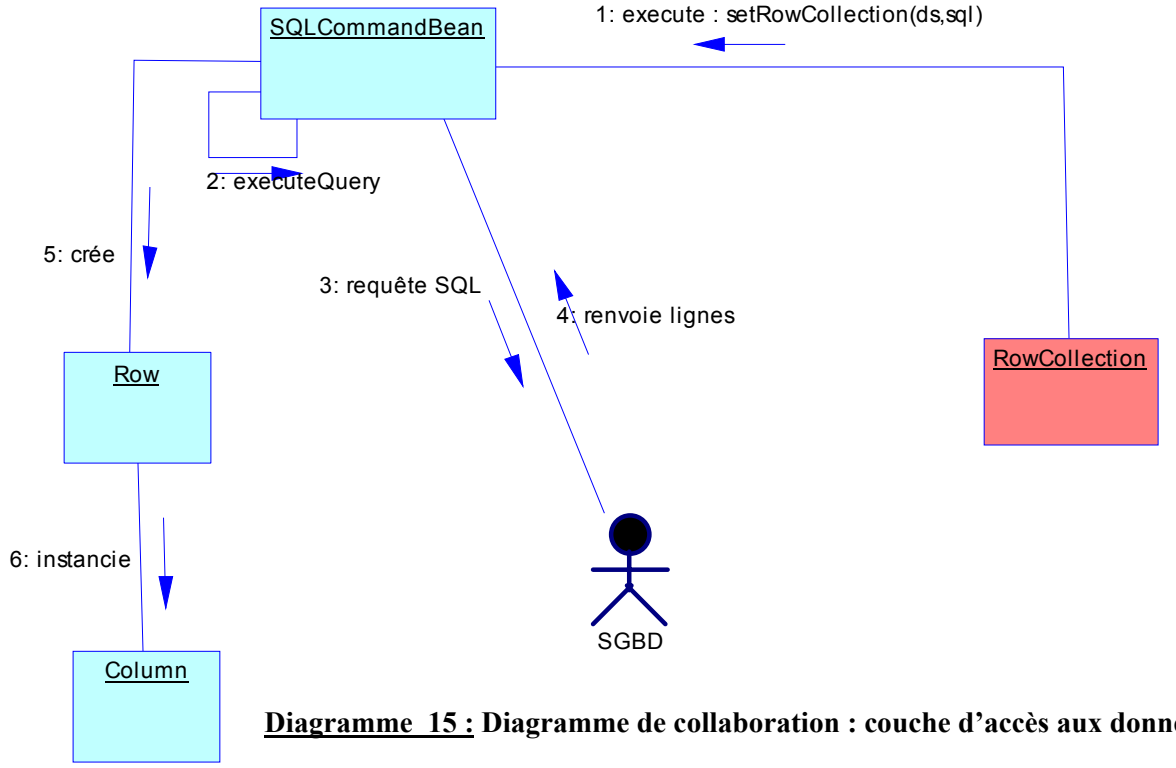
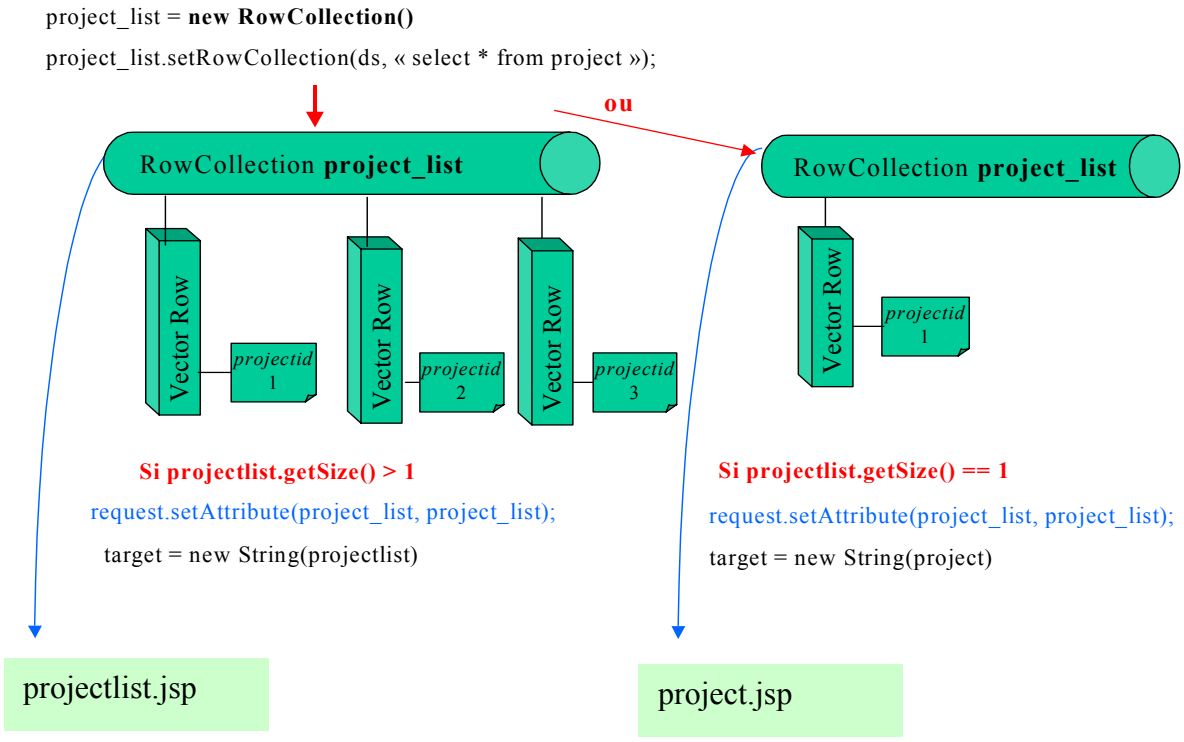


Diagramme 15 : Diagramme de collaboration : couche d'accès aux données

La classe qui permet d'initialiser le processus est *RowCollection*. Elle accepte une requête SQL en paramètre et un objet *DataSource*. Elle va utiliser *SQLCommandBean* par l'intermédiaire de la méthode *setRowCollection(ds,sql)*. La classe centrale est donc *SQLCommandBean*. C'est elle qui va générer les objets beans. C'est l'objet *RowCollection* qui aura initialisé *SQLCommandBean* avec un objet JDBC de connexion à la base de données et la chaîne de caractère contenant la requête SQL. Ensuite *SQLCommandBean* exécute cette requête SQL à l'aide de la méthode *executeQuery()*. Chacune des lignes retournées par l'instruction SELECT sert alors à créer un objet *Row* contenant lui-même plusieurs objets *Column* pour chaque colonne de la ligne. La liste des objets *Row* retournés sert alors à alimenter la classe *RowCollection* qui constitue alors une liste d'objets *Row*. Ce sont ces objets *Row* qui vont constituent les beans génériques qui pourront être accessibles à partir des JSP. C'est la méthode *getString()* qui permet d'accéder aux informations contenues dans ces objets.

Sur le **document 26**, on peut voir maintenant que l'utilisation de l'objet *RowCollection* se fait au niveau de la servlet Action : *ResultListAction* qui joue le rôle de contrôleur. Admettons qu'un utilisateur souhaite récupérer tous les projets dans la base de données ProteomIs/GnpProt. La requête est alors transmise au contrôleur *ResultListAction*. L'objet *RowCollection* **project_list** est alors crée et contient une collection d'objets *Row*. Ensuite, en fonction du nombre d'objets contenus dans le *RowCollection project_list* la servlet Contrôleur orientera vers la bonne JSP. Parallèlement l'objet *RowCollection* contenant les objets *Row* sera transmis en même temps en porté de requêtes pour être exploitable par les JSP. Le mécanisme est ici simplifié car il ne prend en compte qu'une requête. Il faut cependant effectuer plusieurs requêtes pour alimenter en beans la construction d'une JSP. Dans ce cas là la fabrication des objets beans sera encapsulée dans des objets spécialisés qui seront sollicités à partir de la servlet contrôleur. Chacun de ces objets sera ainsi spécialisé dans la fabrication de l'ensemble des beans de telle ou telle JSP.



Document 26 : La servlet Action contrôleur ResultListAction

Le **diagramme 16** récapitule l'ensemble du processus dans un diagramme de collaboration. La classe *ProjectJSPbeans* représente la fabrique d'objet beans pour la JSP *project.jsp*. Ainsi on obtient bien une séparation entre la vue (les jsp), le modèle (le package logique métier) et le contrôleur (la servlet *ResultListAction*).

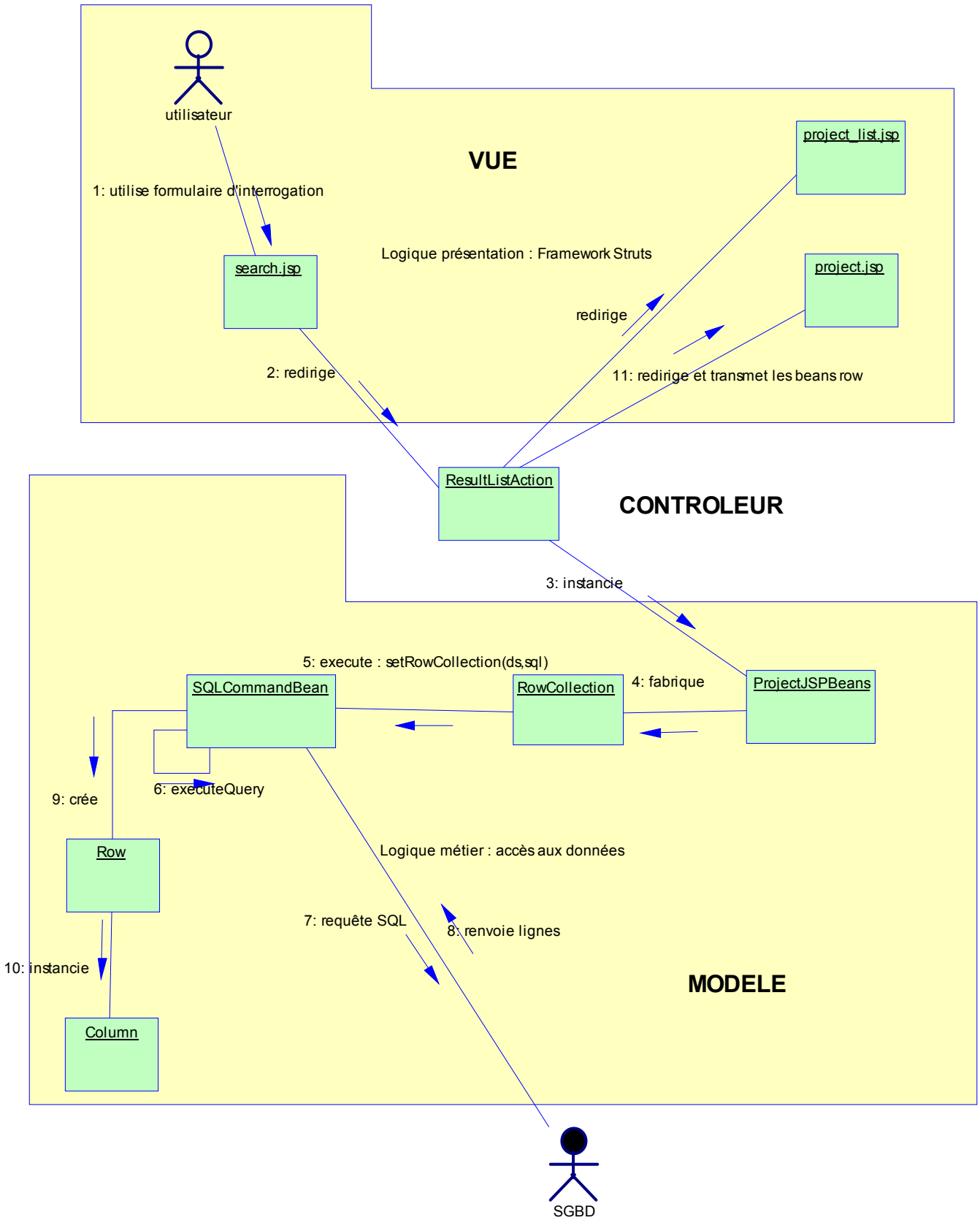


Diagramme 16 : Diagramme de collaboration du modèle MVC dans ProteomIs/GnpProt

b-2) La gestion de la persistance

➢ Les objectifs d'un framework de gestion de la persistance

Nous avons présenté précédemment des framework appelés ORM visant à simplifier la création d'une couche d'accès aux données. Plusieurs de ces ORM dont Hibernate prennent en charge ce que l'on appelle la gestion de la persistance dans l'accès aux données **[i121] [i51]**. On dit que les données sont persistantes lorsqu'elles survivent à l'exécution des programmes. Dans un framework gérant la persistance, lorsqu'un objet bean conteneur de données est crée par la couche d'accès aux données, celui-ci peut être placé en mémoire vive ou stocké sur le disque (on dit aussi sérialisé **[G13]**). Ainsi le nombre d'instances de beans conteneurs de données n'existe qu'une fois dans un système persistant, ce qui économise la mémoire de la machine. Ce principe permet d'améliorer sensiblement les performances de montée en charge des applications. Ce critère devait être pris en considération dans le cadre de l'application ProteomIs/GnpProt. En effet, la couche d'accès aux données doit être capable de supporter des requêtes dans un environnement multi-utilisateur sur le serveur du site privé **[i52]** (accès authentifié) de Génoplante Info.

D'après les informaticiens de Génoplante le nombre d'accès moyens au serveur reste assez restreint. Cependant, l'objectif à terme est que ProteomIs/GnpProt soit transférée sur le site public de Génoplante **[i2]** (partie **5.3.2**). L'accès étant alors ouvert à tout l'Internet, le nombre de connexions augmentera très sensiblement et de manière incontrôlé. L'architecture de la couche d'accès aux données de l'application doit donc être conçue pour supporter une élévation de la montée en charge tout en n'épuisant pas les ressources du serveur de Génoplante-Info où sont hébergées d'autres applications.

➢ Solution retenue pour la gestion de la persistance : le cache de Hibernate

Les autres modules du système GpiIS déjà implantés sur le serveur de Génoplante Info avaient déjà pris en compte ce critère de montée en charge en intégrant la gestion de la persistance dans leur couche d'accès aux données. Dans les premières versions de ces modules (p. ex le module carto), c'est l'ORM commercial qui, par l'intermédiaire d'un système appelé cache objet, gérait la persistance dans la couche d'accès aux données (voir **annexe 22**). Le framework Hibernate utilisé à l'heure actuelle par Génoplante intègre également l'utilisation d'un cache objet très performant appelé ehcache **[i53]**. Un cache objet est une structure qui permet de stocker temporairement et de restituer sur demande des objets en mémoire vive, mais aussi sur le disque pour les caches les plus élaborés (les objets doivent alors être sérialisables).

Les informaticiens de Génoplante m'ont alors recommandé l'utilisation d'un cache objet dans la couche d'accès aux données de ProteomIs/GnpProt afin de gérer la persistance des beans génériques conteneurs de données (les objets *Row*) générés par les requêtes SQL. Deux solutions m'ont alors été proposées : le cache utilisé à la base dans l'ORM commercial de Génoplante et le cache de Hibernate disponible en libre téléchargement sous la forme d'un fichier jar. Après avoir testé les deux solutions, le cache de Hibernate s'est finalement révélé plus performant. Le cache étant simplement un conteneur d'objets persistants évolué, la principale difficulté fut alors d'écrire un algorithme performant permettant d'alimenter le cache en objets tout en leurs associant une clé unique permettant de les récupérer.

➢ Description de l'utilisation du cache dans la couche d'accès aux données

Le diagramme de collaboration sur le **diagramme 17** illustre l'utilisation du cache dans la couche d'accès aux données de ProteomIs/GnpProt. En fait, suite à l'exécution d'une requête, une clé est générée en concaténant les valeurs des identifiants plus les noms des colonnes contenues dans chacune des lignes de résultats d'une requête *(message 5)*. Cette clé identifie alors de manière unique chacune de ces lignes.

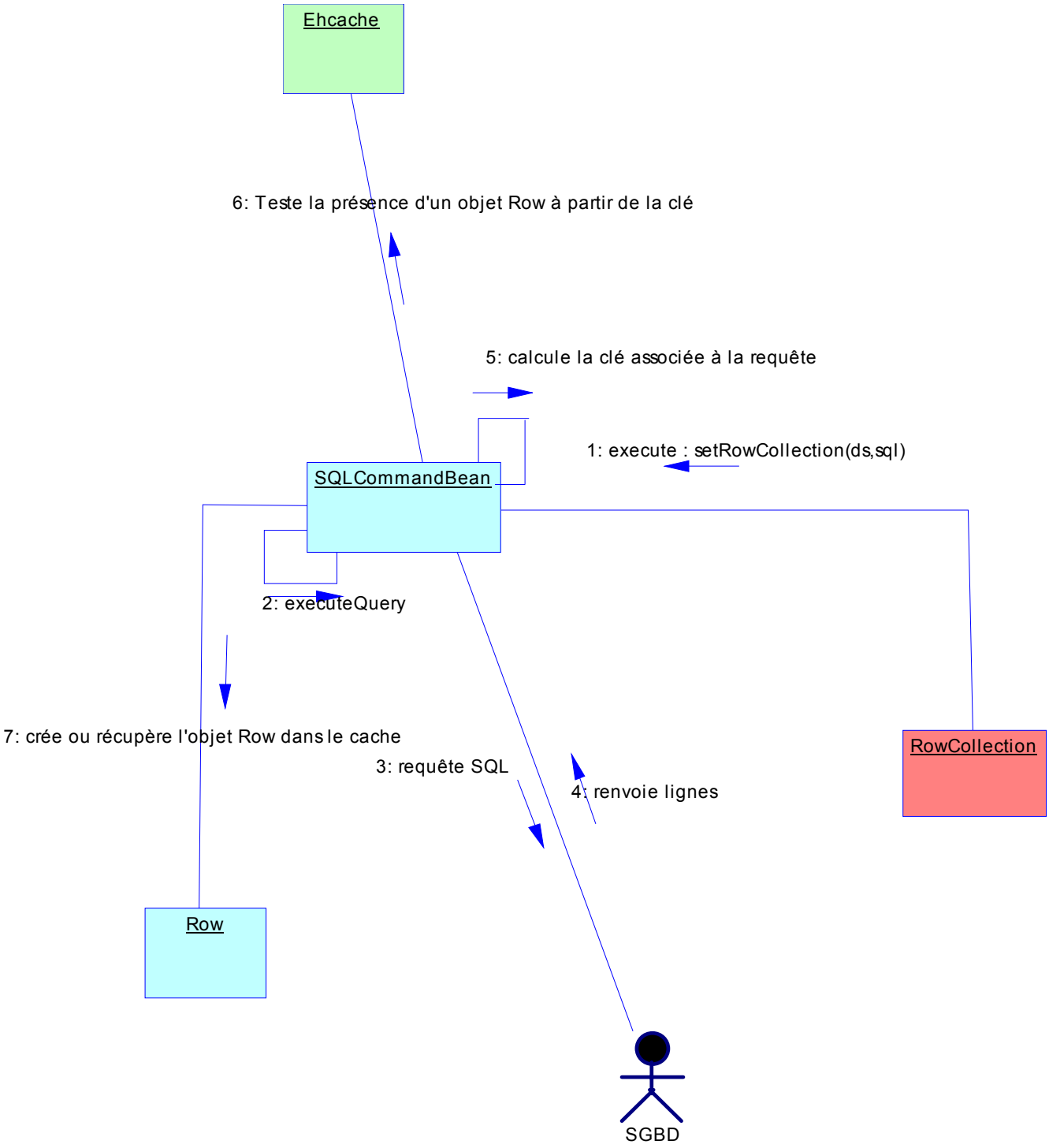


Diagramme 17: Diagramme de collaboration gestion de la persistance

La classe SQLCommandBean teste ensuite si un objet Row est associé à cette clé dans le cache *(message 6)*. Si le test est vrai alors *(message 7)* l'objet Row correspondant à cette clé est récupéré (méthode `cache.get(String key)` de ehcache). Par contre si le test est faux un objet Row est crée et déposé avec la clé dans le cache (méthode `cache.put(element)` de ehcache). J'ai ensuite optimisé cet algorithme de manière à ce que ce soit une prérequête plus légère que la requête d'origine qui interroge la base afin de générer une clé. Exemple de prérequête : `SELECT gel_id FROM gel` (générée à partir de la requête d'origine `SELECT * FROM gel`).

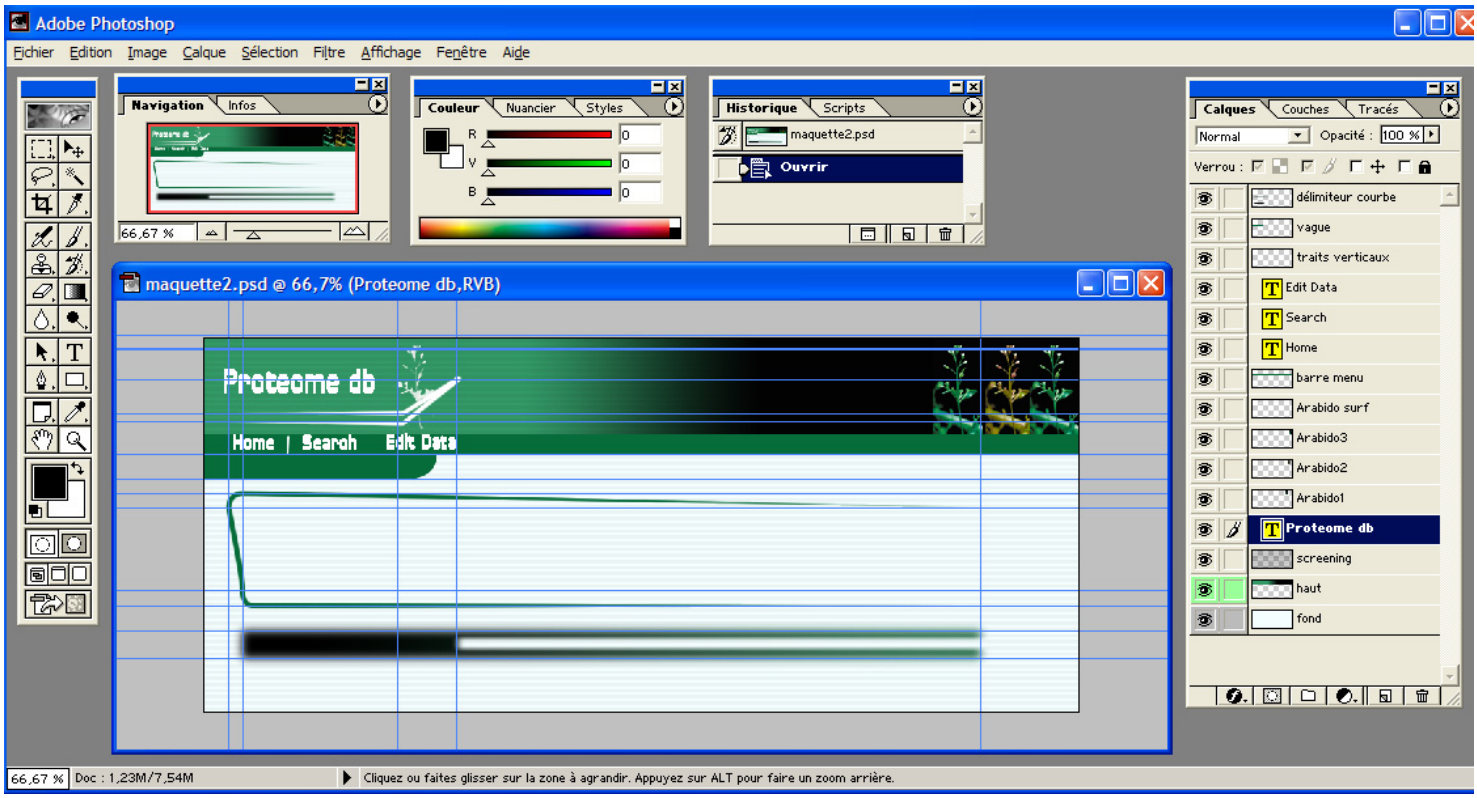
L'interrogation `SELECT gel_id` se fait uniquement sur des champs indexés ce qui est plus rapide. De plus, le nombre de colonnes renvoyées par la prérequête est moins important qu'avec le SELECT *, ce qui permet d'accélérer le parcours des lignes de résultats. Si la clé obtenue à partir de la prérequête ne permet pas de récupérer d'objet *Row* dans le cache alors un nouvel objet *Row* est crée cette fois à partir de la requête d'origine et déposé dans le cache.

Au terme de ce travail, la couche d'accès aux données de ProteomIs/GnpProt intégrait la gestion de la persistance. Une amélioration sensible de la montée en charge pouvait alors être vérifiée suite à l'utilisation du cache. Les résultats de ces tests seront présentés dans la partie **7.4**.

c) Les vues

➢ Création des aspects graphiques

La première étape dans la réalisation des jsp de ProteomIs/GnpProt fut de travailler sur les aspects statiques graphiques. En fait les différents composants de la page furent d'abord réalisés sous photoshop (**document 27**).

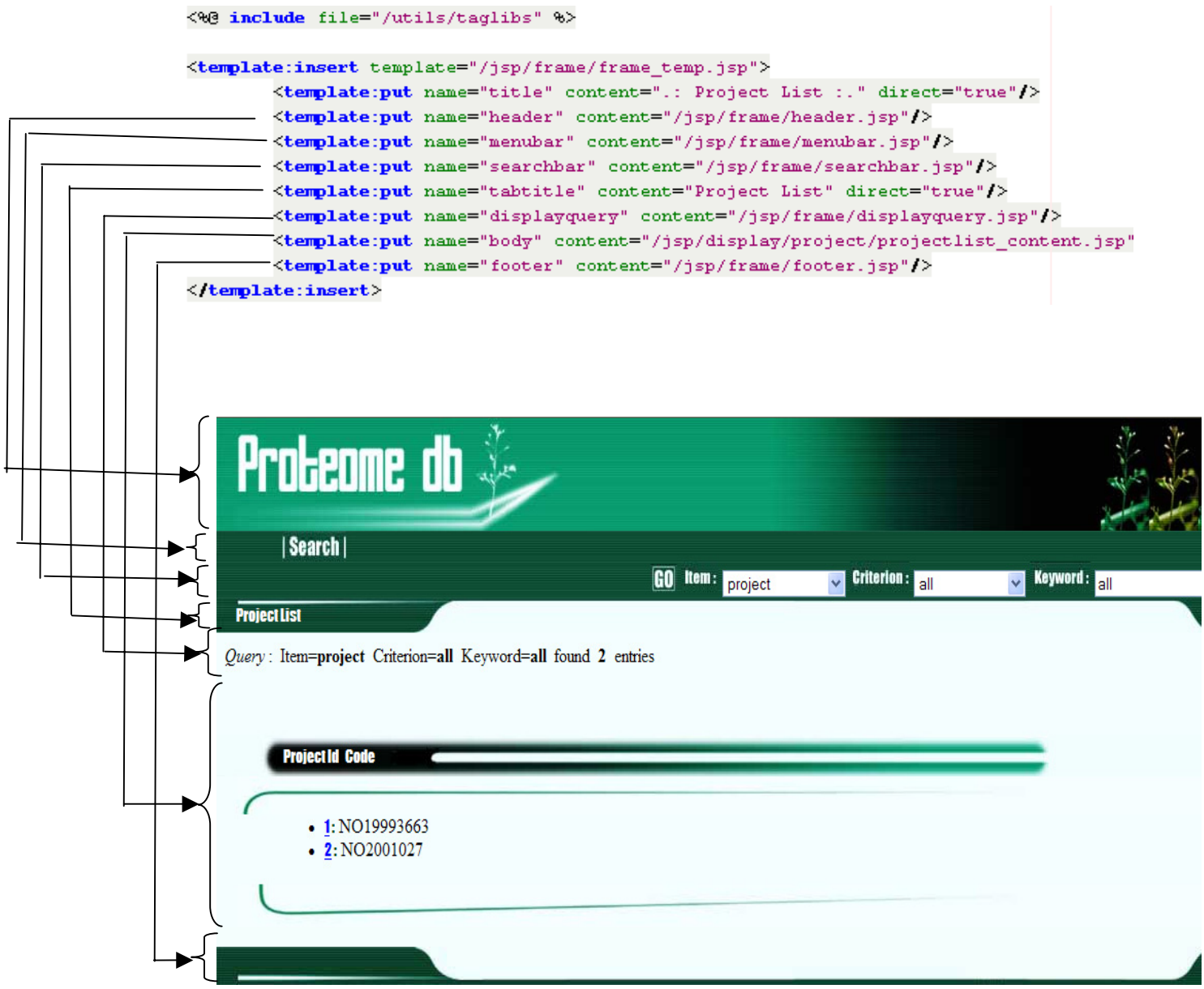


Document 27 : Conception du design des pages avec photoshop

A partir de la maquette, le rendu final de l'interface a été travaillé et dessiné sur un fichier image. Puis l'image a été découpée à l'aide de l'outil tranche de photoshop. Les différentes images obtenues ont alors été incorporées au HTML pour composer la page.

L'utilisant des *template* dans les jsp à permis de rendre réutilisable les différents composants HTML obtenues. Ces composants qui sont en fait des éléments de présentation de la page sont : le header, la barre de menu, la barre de recherche, la barre du titre Project List, la barre d'affichage du nombre de résultats de la requête, le corps contenant les données qui doivent être générées dynamiquement et le bas de page.

Le rôle de chaque template qui fait référence à une jsp est illustré dans le **document 28**.



Document 28 : Utilisation des templates

Un des avantages des JSP est aussi de pouvoir construire des balises personnalisées. Une de ces balises personnalisées est par exemple la balise <cb:blockdata/> qui contient des éléments HTML pour la mise en forme des données. Son rôle est d'afficher le composant ci-dessous :

Enfin les jsp dans ProteomIs/GnpProt intègrent l'utilisation conjointe des "Cascading Style Sheet" (CSS) appelées aussi feuilles de style. L'avantage des feuilles de style est que l'on va déclarer toute la mise en forme des pages : la couleur de fond, les polices de caractère, leurs couleurs, etc. Celle-ci sera liée à chaque page html.

Ainsi, lorsqu'on en modifiera un élément, cela se répercutera immédiatement sur toutes les pages web. Ainsi tous ces outils (template, balises personnalisées, css) utilisés conjointement dans les jsp, ont permis de faciliter la création et la maintenance des aspects graphiques statiques des pages.

➢ Création du HTML dynamique à partir des données contenues dans les JavaBeans

La problématique est cependant maintenant de créer des pages HTML dynamiques intégrant dans le corps du document les données ci-dessous récupérées dans la base de données :

- 1: NO19993663
- 2: NO2001027

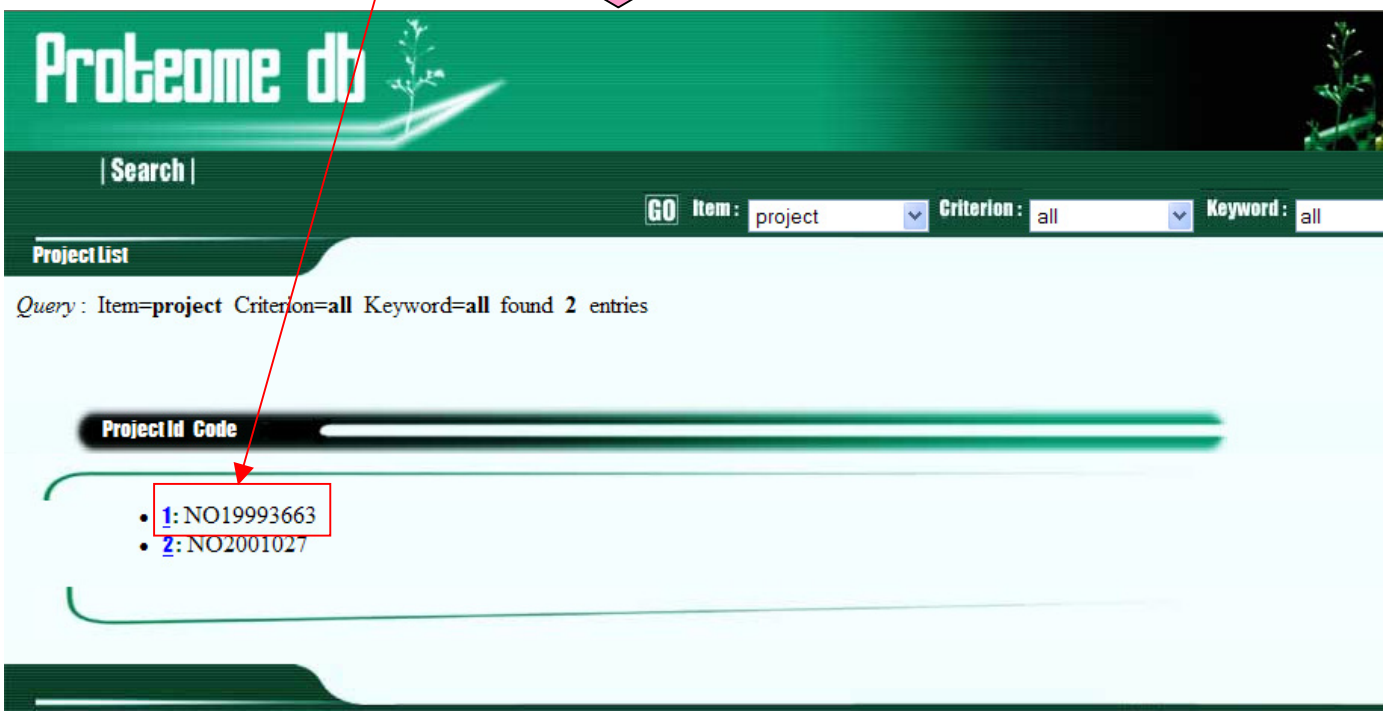
L'objectif avec les JSP est de réduire au maximum le code applicatif java dans les pages grâce à l'emploie des balises spécifiques permettant d'encapsuler les traitements référencés dans les beans. Nous avons vu que Struts proposait l'utilisation de 4 librairies de tags ou *taglib* spécifiques : *html*, *bean*, *logic* et *template*. Dans le cadre de ProteomIs/GnpProt d'autres balises furent également utilisées provenant du code source des exemples fournis avec le livre : JavaServer Pages de chez O'Reilly **[L14]**. Une de ces balises développées par Hans Bergsten est la suivante : <ora:loop>

Voyons maintenant l'emploie de cette balise à travers un exemple. Dans le **document 29,** est présenté le code source de la jsp représentant les projets : project.jsp. Le résultat après implémentation est placé en dessous. La balise <ora:loop> permet d'effectuer une itération sur la liste des objets *Row* contenus dans un objet *RowCollection.* Cet objet *RowCollection* a auparavant été placé dans la portée de requête de la JSP sous le nom *project_list* par la servlet Contrôleur. L'attribut name de la balise <ora:loop> permet de le récupérer à partir son nom. L'attribut property indique à la balise, la méthode de l'objet *RowCollection* qui permettra d'en extraire les beans *Row.* Le bean courant extrait est alors mis à disposition de la jsp sous le nom *project* fournit par l'attribut loopid de la balise. Les informations contenues dans l'objet *Row* peuvent alors être récupérées grâce à la méthode *getString()* en lui passant en paramètre un nom de colonne. Ces noms de colonnes correspondent exactement aux noms des champs de la table project de la base de donnée ; les objet *Row* représentant une instance d'enregistrement de cette table.

NO19993663⇦ project.getString("project_code") ⇦

Table project		
project_id	project_code	…
1	NO19993663	…
2	NO2001027	…

```
<tr>
 <td WIDTH="1000" HEIGHT="47" COLSPAN="12">

  <!-- List of Project(s) -->
  <cb:blockdata striptitle="Project Id   Code" />

      <ora:loop name="project_list" property="rowList" loopId="project" className="com.ora.jsp.sql.Row">

            <li><span class='dataprez'>
            <a href="Urlid.do?project_id=<%=project.getString("project_id")%>"><%=project.getString("project_id")%></
             :</span> <%=project.getString("project_code")%></li>

       </ora:loop>

    <!--Fin block List of Project(s) -->
    <%@include file="/jsp/frame/bottom_blockdata.jsp"%>

   </td>
</tr>
```



Document 29 : Fonctionnement de la balise <ora :loop> et récupération des informations contenues dans le bean générique *Row*

Ainsi nous pouvons vérifier sur ce dernier exemple que les principes du Modèle MVC ont bien été respectés. Nous avons vu que les avantages sont nombreux comme par exemple la possibilité de faire collaborer une équipe à consonance infographie et une équipe à consonance informatique.

Cette répartition des tâches a pu être mise en application lors du développement de ProteomIs/GnpProt. J'ai en effet encadré un stagiaire d'IUP Arnaud Nemrod qui a travaillé lors de l'été 2003 notamment sur le design du site **[r1]**. Cette personne ayant des compétences dans le graphisme et l'utilisation de logiciels tel que photoshop, mon travail pouvait alors se concentrer sur la partie traitement.

6.5.3 Conception de l'analyse bioinformatique des données

Nous présenterons dans cette partie les détails de conception et d'implémentation concernant trois des principaux cas d'utilisation de l'analyse bioinformatique des données (clustering, comparaison de séquence et recherche de motifs) identifiés dans la partie **6.2.2**. Nous ne détaillerons pas la conception du script d'importation des séquences qui est en cours de finalisation. Je précise simplement que ce script est implémenté en Perl et utilise le module `Bio::seqIO` de la librairie BioPerl **[i33]**. Le module `Bio::seqIO` est un objet développé en langage Perl permettant de faciliter la manipulation des séquences au format FASTA. Le langage Perl et la librairie Bioperl constituent des outils essentiels pour l'analyse bioinformatique de ProteomIs/GnpProt. Nous commencerons par justifier les raisons de ce choix avant de détailler la conception du programme de clustering et des autres programmes de bioinformatique écrits également en Perl.

6.5.3.1 Création de groupes non redondants de protéines (« clustering »)

a) <u>choix technologiques</u>

➢ **Le choix du langage Perl et de la librairie BioPerl**

<u>Présentation du langage Perl :</u>

Perl signifie « Practical Extraction and Report Language » que l'on pourrait essayer de traduire par « langage pratique d'extraction et d'édition ». Il a été créé en 1986 par Larry Wall. Le langage Perl n'est pas un langage compilé, comme Java, mais un langage interprété. Perl peut également faire office de langage hôte dans le contexte des bases de données. Ainsi des programmes externes écrits en Perl peuvent intégrer des ordres SQL et interagir avec la base de données, ceci grâce à l'utilisation d'un module Perl appelé DBI (DataBase Interface).

<u>Le choix du langage Perl :</u>

Perl est un langage qui s'applique bien à la réalisation d'applications bioinformatiques; en témoigne par exemple l'écriture du livre : « Introduction à Perl pour la bioinformatique **[L5]**». Les applications de Perl dans ce contexte peuvent aller de l'analyse de séquence à l'extraction d'informations (parsing **[G23]**) dans les fichiers plats des banques de données qui seront des tâches effectuées dans les applications bioinformatiques de ProteomIs/GnpProt. Ces tâches nécessitent en effet le traitement de chaînes de caractères et Perl a de grandes facilités pour cela. En effet Perl était destiné initialement aux traitements de chaînes de caractères sous Unix (avec l'utilisation des expressions régulières **[G9] [i37]**). Il est donc avantageux de l'utiliser lorsqu'il s'agit de développer des parseurs **[G5]**.

<u>Utilisation de la librairie Bioperl :</u>

Une des ressources les plus intéressantes pour notre projet est le module Bioperl disponible sur le site <u>http://www.bioperl.org</u>. Bioperl est une librairie de modules Perl développés spécialement pour traiter des données biologiques avec aisance. Les données habituelles de bioinformatique sont représentées comme des objets de Bioperl. Ainsi, de multiples taches bioinformatiques sont facilitées. Des modules sont optimisés pour le traitement et l'exploitation des résultats obtenus en sortie de logiciels couramment utilisés comme Blast, d'autres pour la manipulation de fichiers au format *Fasta.* Au niveau des bases de données de fichiers plats comme Genbank l'accès direct à certaines valeurs de chaque fiche comme le numéro d'accession, la séquence devient ainsi possible.

Pour ces raisons j'ai choisi d'utiliser le langage Perl et la librairie BioPerl pour implémenter l'ensemble des applications bioinformatiques de ProteomIs/GnpProt. Lors de son stage **[r2],** Mohamed Ndiaye, a pu implémenter rapidement une grande partie des fonctionnalités des aspects « Création de groupes non redondants de protéines » et « Comparaison de séquences » de ProteomIs/GnpProt.

b) conception détaillée

Le travail de conception est représenté dans le **diagramme 18** sous la forme d'un diagramme de collaboration (ce diagramme suit la logique présentée partie **5.2.1**, **6.2.3** et détaillé dans le diagramme de cas d'utilisation et diagramme de séquences : *« Suppression de la redondance dans la base ProteomIs/GnpProt (« clustering ») »* présentée partie **3.1** de l'**annexe 1**). Nous apportons ici la notion d'interactions entre les différents composants de l'application et surtout les détails sur la solution technique utilisée.

Le programme est prévu pour être lancé en ligne de commande par le gestionnaire de données *(message 1 dans le diagramme)*. Ce programme est en fait composé de trois scripts. Le processus démarre avec le premier script Perl nommé `redondance.pl` et s'achève avec la création de la base non redondante. Le script `redondance.pl` a pour objectif de créer la liste de correspondance AGI – NON AGI des accessions de la base redondante *(message 6)*. Cela est possible en interrogeant d'abord la base redondante *(message 2)* pour obtenir la liste des accessions contenues dans la base. L'interrogation de la base se fait à l'aide du module Perl DBI. Ensuite il faut également analyser le champ « référence croisées » dans les pages HTML des accessions NON AGI des banques de données publiques. Pour cela les pages sont rapatriées en local à l'aide de la commande GET *(message 4)*.

La méthode GET permet de charger dans une variable ou un fichier le contenu d'une page au format HTML correspondant à un URL précis. L'extrait de code ci-dessous montre comment à partir d'une url la méthode GET est appliquée pour récupérer la page correspondant à une accession dans la base de donnée PIR **[i43]** ; cet accession étant contenu ici dans la variable $pir (en rouge) :

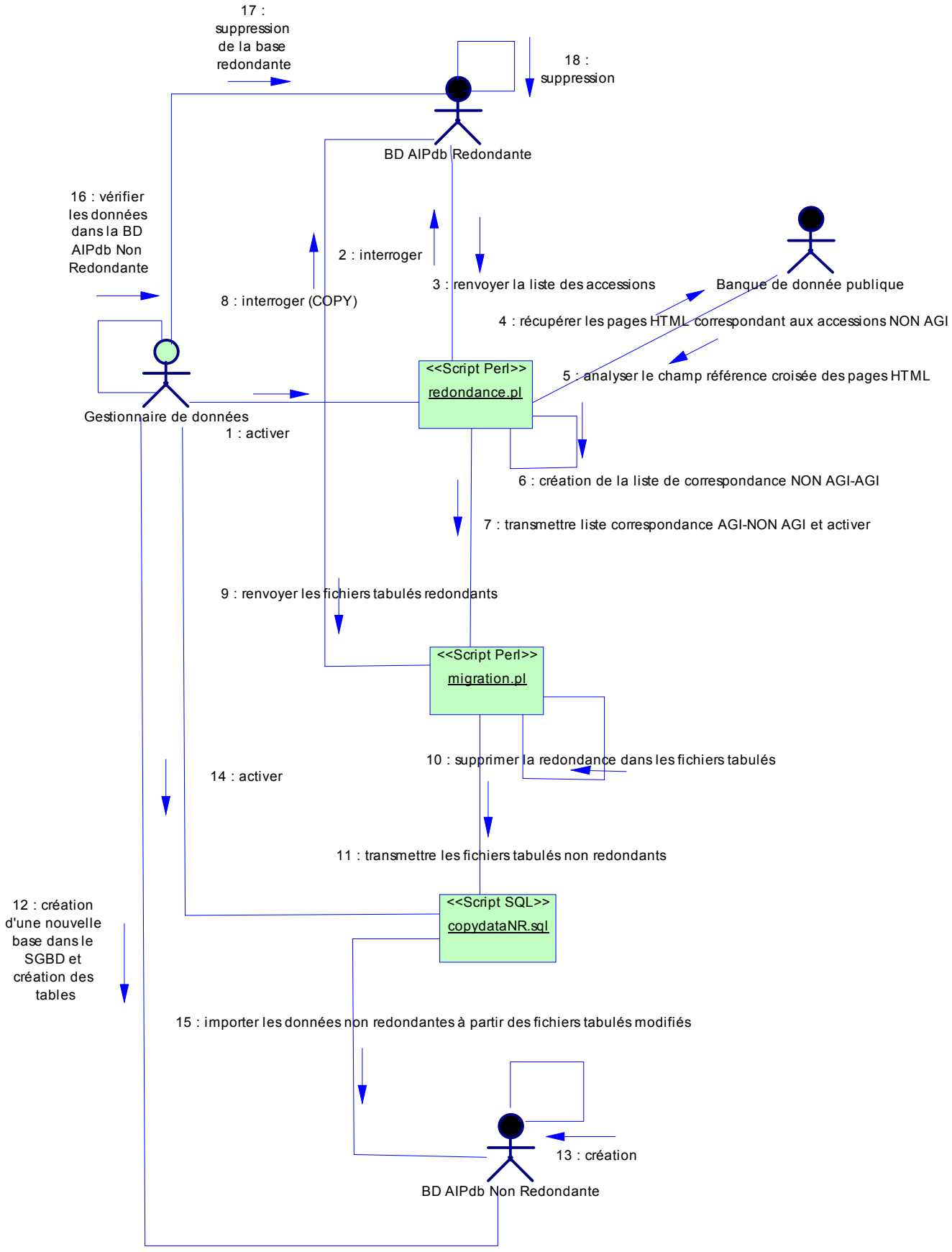
```
`GET "http://us.expasy.org/cgi-bin/get-entries?DR=$pir &view=full"`;
```

Nous exploitons ici le fait que généralement les bases de données publiques laissent la possibilité d'interroger leurs données en permettant de passer plusieurs types de paramètres dans une url (accession, nom de la protéine …), leur objectif étant de faciliter ainsi la création de liens inter-base. Les pages HTML des accessions NON AGI ainsi rapatriées, il est ensuite facile de les analyser à l'aide des expressions régulières de Perl afin d'extraire les accessions AGI correspondant du champ « Reference croisées » (message 5). La liste de correspondance NON AGI-AGI (message 6) est ainsi construite par le programme `redondance.pl`. Le deuxième script `migration.pl` a lui ensuite pour objectif de récupérer les données de la base ProteomIs/GnpProt redondante sous forme de fichiers tabulés (message 8 et 9) et de supprimer la redondance dans ces données (message 10). Pour récupérer les données aux formats tabulés on a choisi d'utiliser la commande **COPY** par l'intermédiaire du module Perl DBI. COPY déplace les données entre les tables du SGBD et les fichiers standard Unix. La commande envoie l'instruction au SGBD d'écrire ou de lire vers un fichier.

```
# COPY table TO 'filename' USING DELIMITERS 'delimiter'
$dbh->do("COPY molecule_type TO
'/home/root/AFPDB gpi/Tuples/molecule type.out' USING DELIMITERS '\t' WITH
NULL AS 'NULL'") or die "$DBI::errstr \n";
```

On pourra récupérer dans des fichiers standards, le contenu des 48 tables de la base de données ProteomIs/GnpProt redondante. Le script `migration.pl` modifiera ensuite ces fichiers de manière à supprimer la redondance dans les données en se basant sur la liste d'équivalence AGI-NON AGI transmise par le programme `redondance.pl` *(message 10)*. Le gestionnaire de données doit ensuite intervenir pour créer manuellement une nouvelle base de donnée à l'aide du SGBD et créer les tables *(message 12 et 13)*. Puis le gestionnaire de données doit lancer l'exécution du script SQL `copydataNR.sql` *(message 14)*. Ce script va importer à l'aide de la commande COPY les données non redondantes dans la nouvelle base non redondante *(message 15)*; et ce à partir des fichiers tabulés fabriqués par le script `migration.pl`.

<u>Diagramme 18 :</u> Diagramme de collaboration : Suppression de la redondance dans la base ProteomIs « clustering »



6.5.3.2 Comparaison de séquences

a) choix technologiques

Les choix technologiques concernant l'interfaçage du logiciel BLAST étaient déjà prédéterminés car j'ai pu mettre à profit une interface pour BLAST développée par l'ingénieur bio informaticienne, Cécile Fizame. L'interface permettant de saisir la séquence et les paramètres pour utiliser le logiciel BLAST est écrite en HTML. Le programme qui permet de vérifier les données saisies, exécuter BLAST en ligne de commande et mettre en forme les résultats, a été écrit en Perl CGI.

b) conception détaillée

La mise en place de l'application de comparaison de séquences a été établie en trois étapes:
❶ L'installation de blast en local.
❷ Quelques adaptations sur le document HTML contenant le formulaire qui permettra au biologiste de saisir des données ou d'accéder à une aide : ce sont les pages blast_proteomis.html et blast_help.html.
❸ Le développement d'options supplémentaires (p. ex. vérification des données saisies) dans le programme Perl CGI de lancement du Blast et d'affichage des résultats : blast.cgi.

6.5.3.3 Recherche de motifs

Dans le cadre de ce mémoire, on présentera uniquement l'implémentation de la deuxième version du programme de recherche de motif permettant d'analyser les séquences à partir d'un formulaire (présenté partie **6.2.3**). Cette fonctionnalité était plus urgente pour les biologistes du laboratoire que l'implémentation de la première version du programme permettant de rechercher automatique des motifs de phosphorylation dans les séquences contenues dans la base de données ProteomIs/GnpProt. L'implémentation de cette deuxième version du programme sera réalisée par la suite au cours du deuxième semestre 2005 (voir **chapitre 8.2 Perspectives**).

a) choix technologiques

La première étape pour l'implémentation de la chaîne de traitement était de récupérer les logiciels MSDigest et NetPhos à partir de leur site respectif d'hébergement. Le problème est que le site où les biologistes ont l'habitude d'interroger MSDigest **[i34],** n'offrait pas de version de cette application en libre téléchargement. Une autre solution fut envisagée qui consistait à interroger à distances le site de MSDigest à partir de notre application de recherche de motif (en utilisant par exemple le module LWP **[i35]** de Perl). Notre application enverrait alors autant de requêtes au formulaire de MSDigest qu'il y'a de séquences dans la base, les pages de résultats HTML étant ensuite rapatriées en local. Il faut donc que le serveur interrogé soit capable de supporter un nombre relativement important de requêtes. Afin de ne pas pénaliser le fonctionnement du site de MSDigest, une autorisation pour cela fut demandée au responsable du site. Cette autorisation fut alors refusée.

Il ne restait plus dès lors qu'à envisager de développer les fonctionnalités qui nous intéressaient dans le logiciel MSDigest. Ces fonctionnalités ont été décrites dans la partie **5.2.3**. Il s'agit principalement de fonctionnalités du type « traitement de chaînes de caractères » puisqu'il est question d'analyser le texte décrivant la séquence d'une protéine. Grâce aux expressions régulières **[G22]** de Perl parfaitement adaptées à ce genre de traitement les fonctionnalités intéressantes de MSDigest ont pu être développé en moins d'un mois par Cyril Genin **[r3]**.

En ce qui concerne le logiciel NetPhos les développements furent plus rapides. Bien que le site de NetPhos ne fournisse pas de version en libre téléchargement, la popularité de ce logiciel a conduit la communauté des bioinformaticiens à développer un module en Perl **[i39]** permettant d'interroger à distance le site de NetPhos. De plus ce module dispose d'accesseurs qui permettent de récupérer directement les données contenues dans les pages HTML de résultats de NetPhos.

Ainsi nous disposions d'un outil facilitant l'utilisation à large échelle de NetPhos. Ce module appartenant à la librairie BioPerl **[i82]** et disponible en open source sera utilisé pour notre projet. Le travail de prédiction de NetPhos pourra ainsi être automatisé. Le choix des outils (Perl et Bioperl) pour l'implémentation de la chaîne de traitement étant effectué, il reste maintenant à définir le choix du langage permettant de réaliser les interfaces permettant d'utiliser le programme. Ce choix ne s'est pas porté sur les JSP/Servlet mais sur la technologie Perl CGI. En effet Cyril Genin ne connaissant pas JSP/Servlet et ayant effectué une grande partie de son stage en utilisant le langage Perl, il ne disposait pas du temps suffisamment nécessaire pour apprendre l'utilisation des JSP/Servlets.

b) conception détaillée

Les résultats présentés ici sont le fruit du travail de Cyril Genin. Nous ne rentrerons pas dans les détails du programme. Le travail de conception est représenté sous la forme d'un diagramme de collaboration (**diagramme 19**).

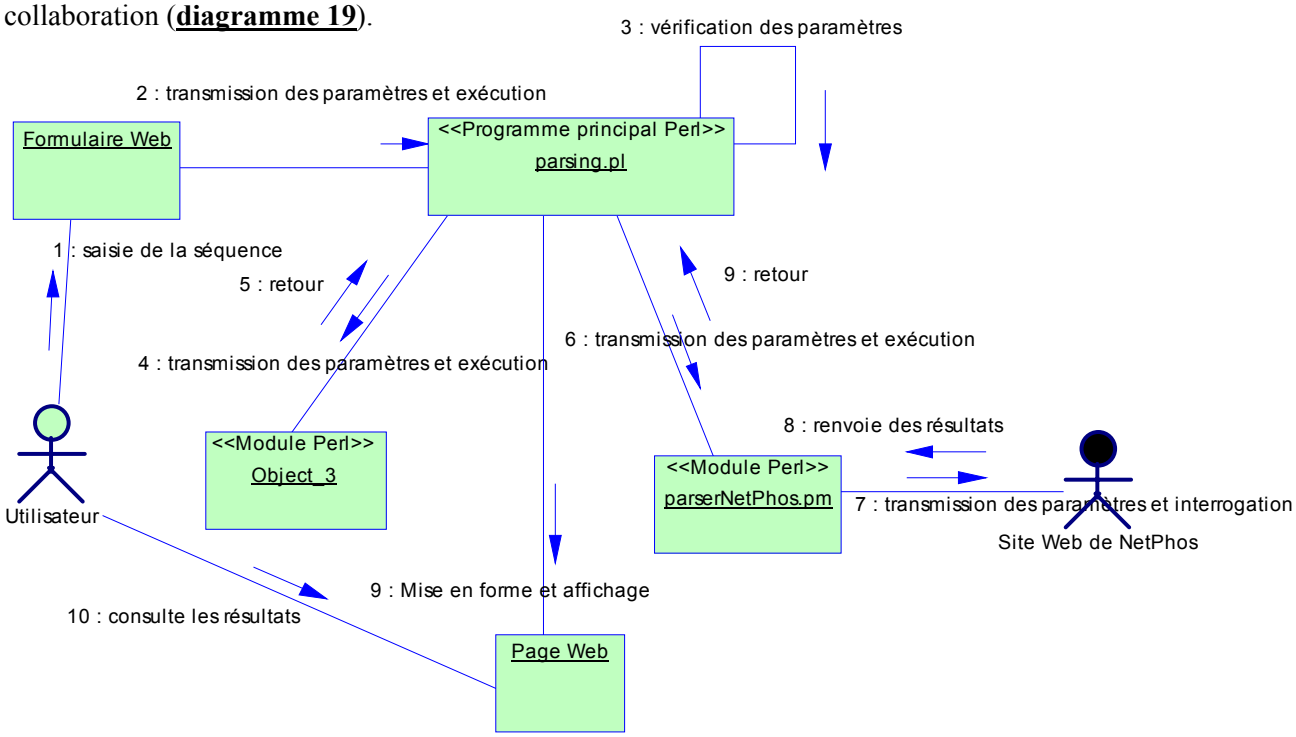


Diagramme 19 : Diagramme de collaboration : Recherche des motifs de phosphorylation à partir de l'interface utilisateur

Le programme est prévu pour être lancé grâce à un formulaire Web écrit en Perl CGI. Cette page fait appel au programme principal `parsing.pl`. Ce programme est un script Perl qui rassemble les paramètres (p. ex. la séquence) donnés par l'utilisateur et qui vérifie leur intégrité. Le formulaire Web sera présenté dans la partie **7.3.3**. Ce script utilise un module Perl `digest.pm` reproduisant le travail du logiciel MSDigest. Le module passe ensuite la main au programme principal qui exécute le module `parserNetPhos.pm` qui interroge le site web de NetPhos à partir des paramètres qui ont été récupérés dans le formulaire. Les résultats sont ensuite imprimés en HTML dans le navigateur par le programme principal. Des copies d'écrans seront présentées dans la partie suivante.

7. Présentation des résultats d'implémentation

Dans cette partie, sera présenté le résultat de l'implémentation des trois cas d'utilisation : saisie des données, interrogation et visualisation des données et analyse bioinformatique des données. Je présenterai à chaque fois une vision d'ensemble des différentes options applicatives en justifiant, quand c'est nécessaire, les choix ergonomiques et fonctionnels.

7.1 Saisie des données

a) Le format d'échange

Rappelons tout d'abord que le format d'échange est la procédure choisie pour saisir les données dans la base de ProteomIs.

Ce format d'échange est composé :

- d'un classeur Excel comportant plusieurs feuilles (voir **annexe 23** la liste des feuilles) dont le contenu sera importé dans la base

- d'un ensemble de sous dossiers dans lesquels seront associés les fichiers suivant : fichiers décrivant les protocoles, fichiers images de gel, fichiers résultats d'analyse spectro et immuno, fichiers de résultats d'interrogation ….

Sur le **document 30** est représenté une copie d'écran de la feuille Spot, de la feuille Gel et de la feuille List. Comme on peut le voir la feuille List va servir de feuille de référence pour remplir les autres Feuilles du format d'échange.

Sur ces feuilles sont illustrées différentes options permettant de rendre plus facile la saisie. La ligne en rouge est une ligne exemple dans chaque feuille. Les 4 premières lignes sont figées.

Les deux premières lignes en bleu référencent l'entité et le champ correspondant dans le dictionnaire des données, si l'on souhaite des explications sur le champ en question.

Le dictionnaire des données est un classeur Excel fourni avec le format d'échange et dans lequel sont décrits les différents champs de la base : description, nature du champs, contrainte, etc …

En **annexe 13**, est représenté une copie d'écran d'une partie du dictionnaire des données.

Document 30 : Le format d'échange

Lien hypertexte qui dirige vers la feuille Gel dans laquelle on doit choisir le nom du gel parmi ceux contenus dans la colonne gel_name

	A	B	C	D	E	F	G
1	SPOT_BAND	SPOT_BAND	SPOT_BAND	SPOT_BAND	SPOT_BAND	SPOT_BAND	GEL
2	spotband_numeric	spotband_x_position	spotband_y_position	pi	mw	short_remark	gel_nan
3							
4							gel
5	**spotband_number**	spot_x	spotband_y	spot_pl	**spotband_Mw**	spotband_comment	**gel_na**
6	1	501	7	4.70246	289		G222
7	3	708	109	4,93	191		G229
8	4	977	125	5,27	179		G229
9	5	994	125	5,3	179		G229

list / Reference / Taxon / Thematic / DocThematic / Contact / Protocole / Treatment / Sample_source / Extract_Sample

Feuille Spot

	A	B	C	D	E	F	G	
1	GEL	GEL	GEL	GEL	GEL	BIO_TYPE	GEL	BIO_
2	gel_name	gel_date	protein_qty	pH_gradient	acrylamid_percent	type_name	short_remark	type_
3								
4								list->
5	**gel_name**	**gel_date**	gel_protein_qty	gel_pH_gradient	gel_acrylamid_percent	gel_coloration_type	gel_comment	**gel_t**
6	G222	22/08/2002		4 à 7		Ag	18 cm	2D
7	G229	17/10/2001	500µg	4to 7	12	Colloidal coomassie blue	18cm	2D
8								
9								

list / Reference / Taxon / Thematic / DocThematic / Contact / Protocole / Treatment / Sample_source / Extract_Sample

Liste déroulante

Feuille Gel

	A	
1		
2	(Tous)	
3	(10 premiers...)	**protocol_type**
4	(Personnalisé...)	plant_culture_protocole
5	ACC	sample_protocol
6	DATABASE	extract_protocol
7	DOCUMENT	gel_protocol
8	DV_STAGE	LC_protocol
9	EXP_CONDITION	image_protocol
10	GEL	spectro_protocol
11	HARDWARE	immuno_protocol
12	IMMUNO_ANALYSE	query_protocol
13	LC	
14	MATURATION	**gel_type**
15	MEDIA	2D
16	MS_ANALYSE	1D
17	PROJECT	western
18	PROTEIN	
19	PROTOCOL	
20	SOFTWARE	
21	TISSUE	

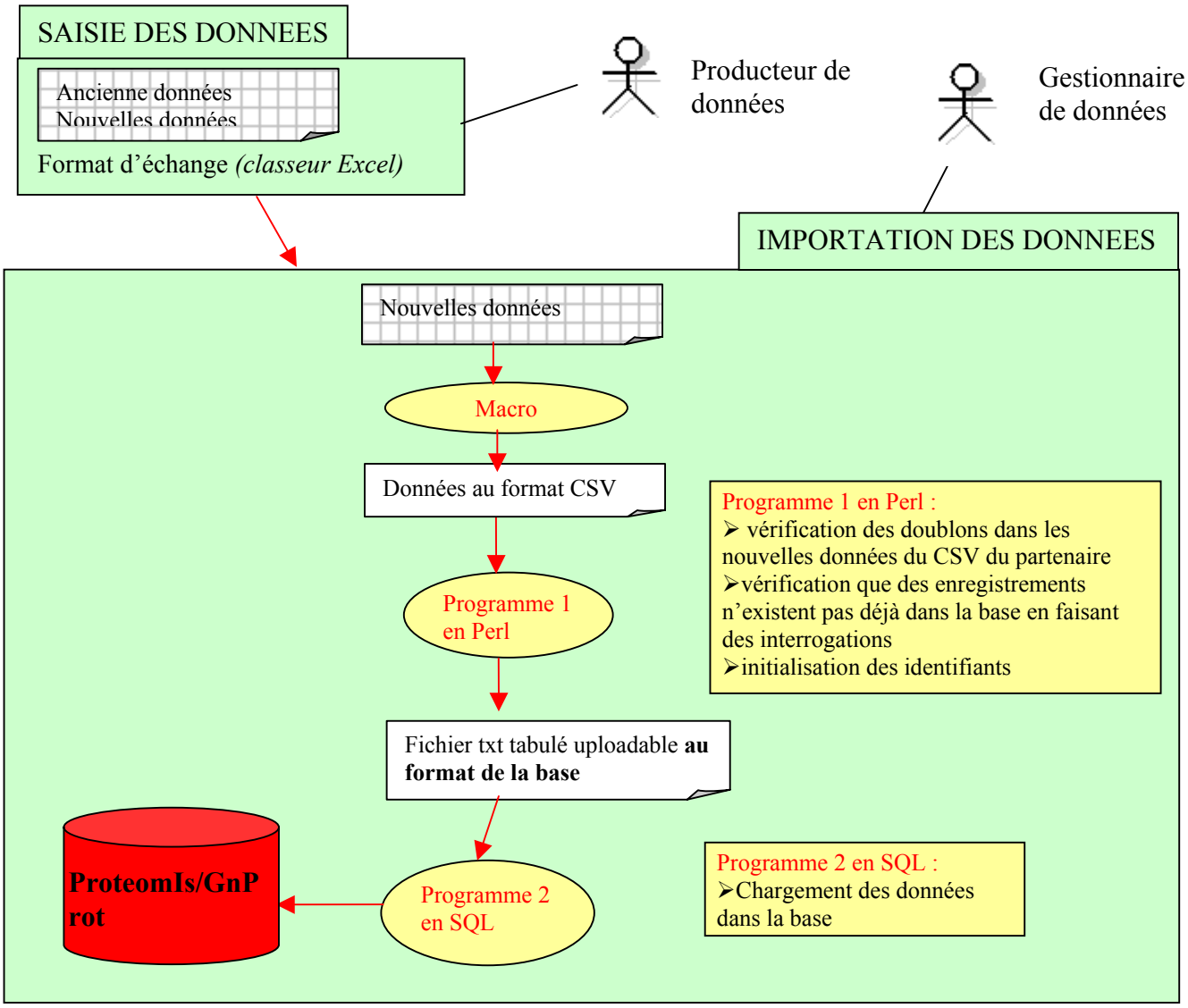
list / Reference / Taxon / Thematic / Doc

Ce lien oriente vers la feuille list dans laquelle on doit choisir le type du gel (1D, 2D).
L'intitulé list-> GEL après la flèche est le critère à sélectionner dans la liste déroulante de la feuille list afin d'aboutir sur la liste des termes disponibles pour le critère gel_type

b) La procédure d'importation

Sur le **document 31** est récapitulée la procédure utilisée pour importer les données dans la base à partir du format d'échange. Les données sont saisies dans un fichier contenant les données précédemment saisies servant de référence. Le producteur de données doit ensuite copier les nouvelles données dans un autre fichier Excel. C'est ce fichier Excel contenant exclusivement les nouvelles données qui est ensuite envoyé au Gestionnaire de données de Génoplante Info ou du laboratoire (pour une importation en local). Le gestionnaire de données est ensuite chargé de lancer les différentes étapes permettant d'aboutir au chargement des données dans la base ProteomIs/GnpProt à savoir :

- Etape 1 : Exportation des différentes feuilles du fichier Excel au format CSV. On utilise pour cela une macro Excel qui automatise la création des 19 feuillets au format CSV.
- Etape 2 : Lecture par un premier programme en Perl chargé d'éliminer les doublons et de vérifier les contraintes sur les champs 'not null'. La vérification des doublons se fait d'abord entre les données du fichier. Ensuite le programme vérifie qu'il n'y ait pas des données qui existent déjà dans la base. Enfin le programme complète la valeur des champs identifiants (clés primaires et étrangères) qui devront être attribués dans les différentes tables.
- Etape 3 : On obtient au final autant de fichiers tabulés au format texte qu'il y a de tables dans la base. Les données de ces fichiers sont ensuite importées directement dans la base de données à l'aide d'un deuxième programme en SQL.



Document 31 : Schéma récapitulatif de la procédure utilisée pour la saisie des données

7.2 Interrogation et visualisation des données

Nous allons dans cette partie présenter certaines copies d'écrans parmi les interfaces d'interrogation et de visualisation implémentées. Afin de rendre cette partie plus vivante, nous allons organiser cette présentation sous la forme d'un scénario organisé autour d'une des fonctionnalités les plus utilisées dans la base de donnée : la consultation de protéines et la navigation dans le gel où cette protéine a été identifiée.

1 – Connexion à l'application :

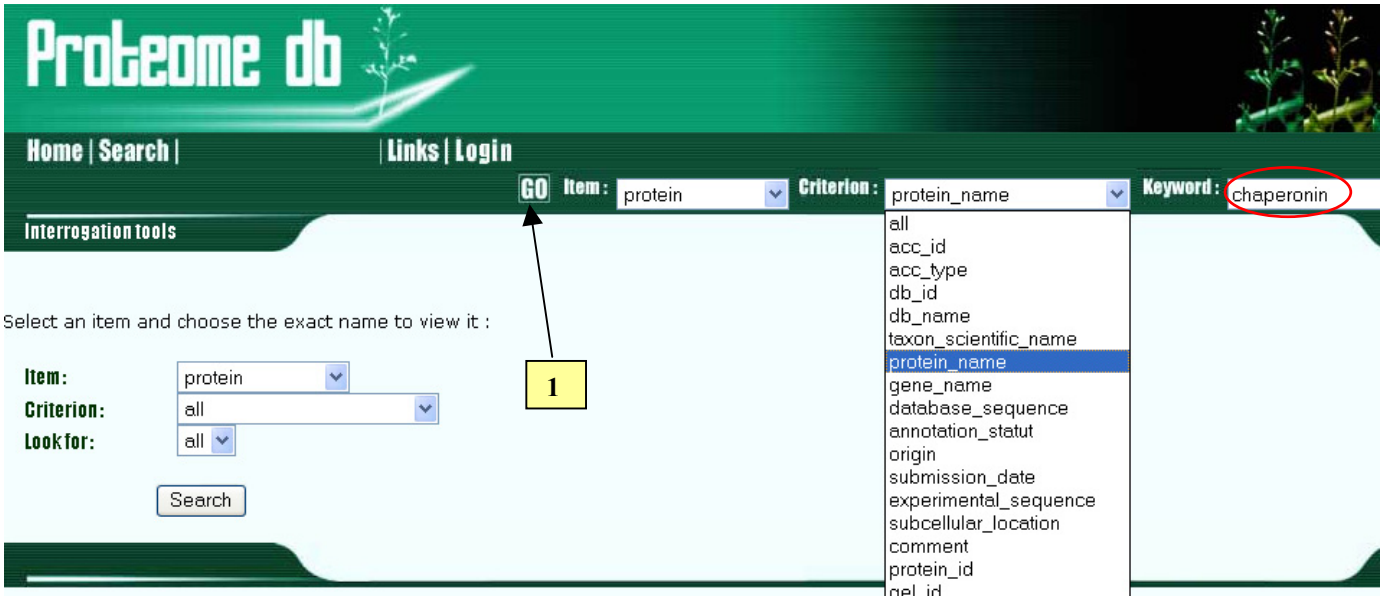
Pour utiliser l'application l'utilisateur doit d'abord se connecter en saisissant un login et un password via une interface de Login. Dans le cas de la version locale de ProteomIs/GnpProt l'interface de login se présente sous la forme d'une interface web. L'application compare le login et le password saisis avec ceux enregistrés dans la base de donnée et une session de login est crée pour l'utilisateur. Cette session empêche qu'une personne non autorisée puisse accéder à l'application en jouant sur les url. Dans la version de ProteomIs/GnpProt qui sera installée à Génoplante Info le niveau de sécurité est beaucoup plus élévé. En effet la connexion se fait toujours à travers une interface de login mais par l'intermédiaire d'une liaison SSH cryptée.

2 - Recherche de la protéine (**document 32**) :

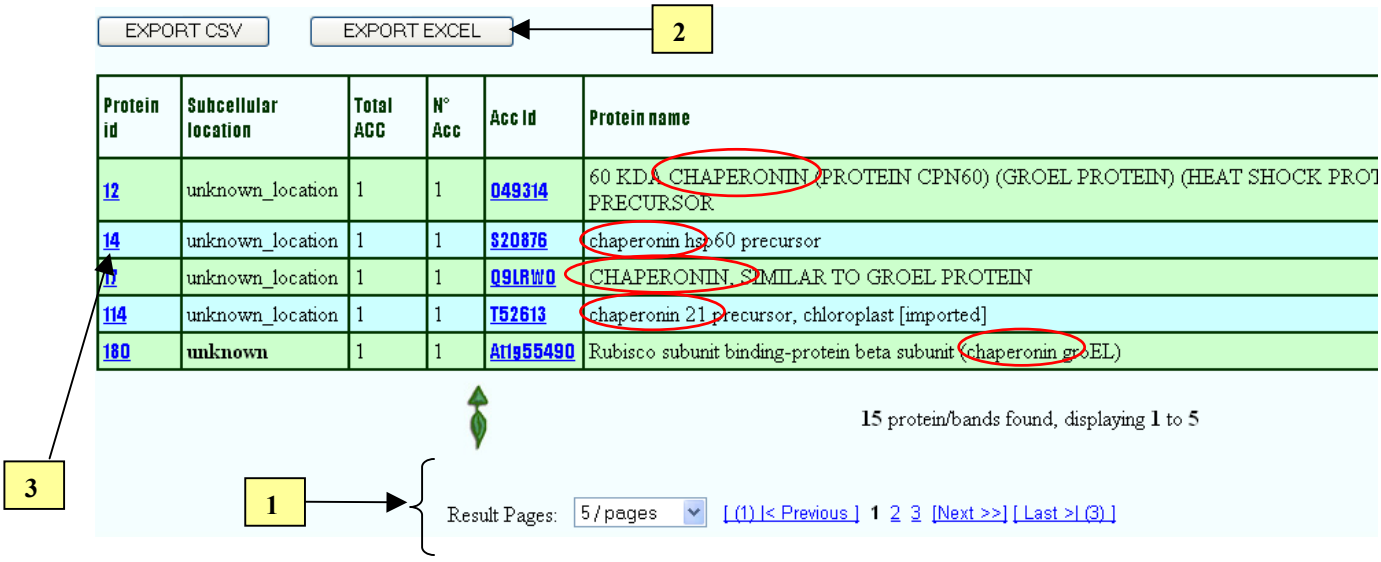
Une fois connecté à l'application, l'utilisateur peut accéder au formulaire de recherche avancée de l'application. Imaginons maintenant que l'utilisateur veut rechercher la liste des protéines contenant dans leur nom le mot clé « chaperonin ». Pour ce type de requête il doit utiliser la barre de recherche rapide. Cet outil se situe, juste sous le menu, et est accessible à partir de toutes les pages de l'application. L'utilisateur peut sélectionner dans la liste le critère de recherche « protein name » et doit taper dans la zone de saisie le mot « chaperonin ».

3 – Parcours de la liste des résultats :

Le résultat de la requête précédente est un tableau comportant la liste des protéines comportant dans leur nom le mot clé « chaperonin » (**document 33**). Chaque ligne du tableau correspond à un couple bien distinct : proteine + son numéro d'accession. Chaque colonne du tableau apporte ensuite une information précise sur la nature de la protéine. La colonne protein_id renseigne sur l'identifiant qui identifie de manière unique la protéine dans la base de donnée de ProteomIs *(cet identifiant sera basé sur la nomenclature AGI lorsque le programme de clustering sera finalisé)*. Cet identifiant est aussi un lien hypertexte qui dirige vers la page Web de visualisation de la fiche détaillée de la protéine correspondante. La colonne subcellular_location apporte une information sur la localisation cellulaire de la protéine dans l'organisme. La colonne Total ACC précise le nombre de numéros d'accessions qui ont été associé à cette protéine par les biologistes lors du processus d'identification. La colonne ACC id renseigne sur ces numéros d'accessions. Enfin, la dernière colonne renseigne sur le nom de la protéine. L'alternance des couleurs est là pour apporter une meilleure lisibilité. La couleur change lorsque l'on passe d'une protéine à une autre. Quand une nouvelle protéine a été isolée par les biologistes mais qu'elle n'est pas référencée dans les banques et de nature inconnue, elle apparaît en rose. Il est à noter que les protéines sont d'abord triées par ordre de protein_id croissant et ensuite groupées par localisation subcellulaires. Le style passe de gras à non gras lorsque l'on change de type de localisation subcellulaire. En bas de la liste des protéines, un système de navigation page/page (flèche jaune 1) permet de naviguer à travers l'ensemble des résultats de la requête. Par défaut l'ensemble des lignes affichées à l'écran est limité à 50. Cependant il est possible de modifier le nombre résultats / page autorisé grâce à une liste déroulante. Dans notre exemple on a choisi d'afficher 5 lignes par pages.



Document 32 : Recherche de protéines à l'aide de la barre de menu « recherche rapide »



Document 33 : Liste des protéines obtenues à partir du mot clé « chaperonin »

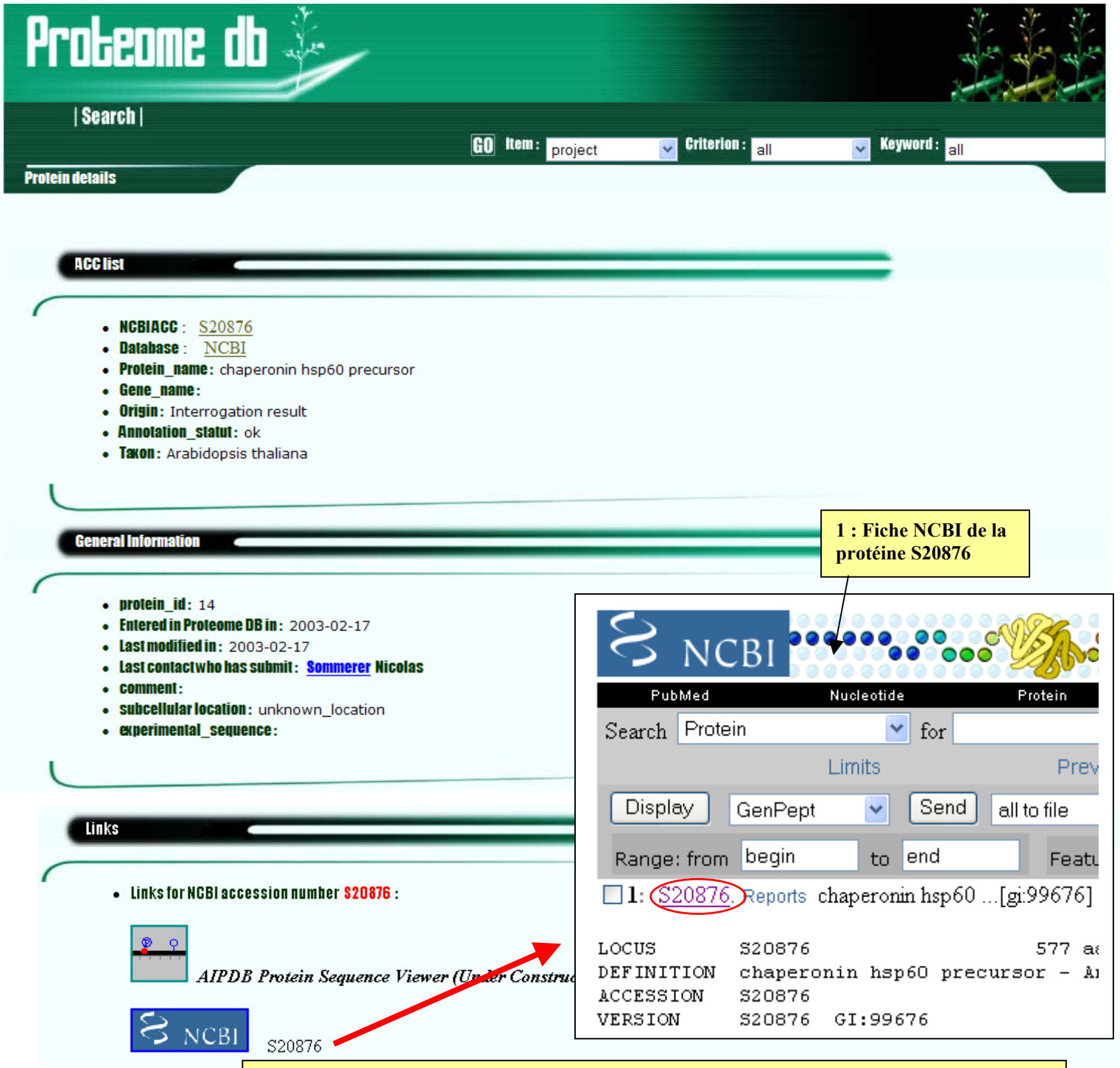
4 – Exportation de la liste des résultats :

Le bouton « EXPORT CSV » permet d'obtenir un format d'exportation dit CSV **[G6]** dont le séparateur est le point virgule. Le bouton « EXPORT TO EXCEL » (flèche jaune 2) exporte la fiche (également au format CSV) directement dans EXCEL.

5 – Visualiser un résultat :

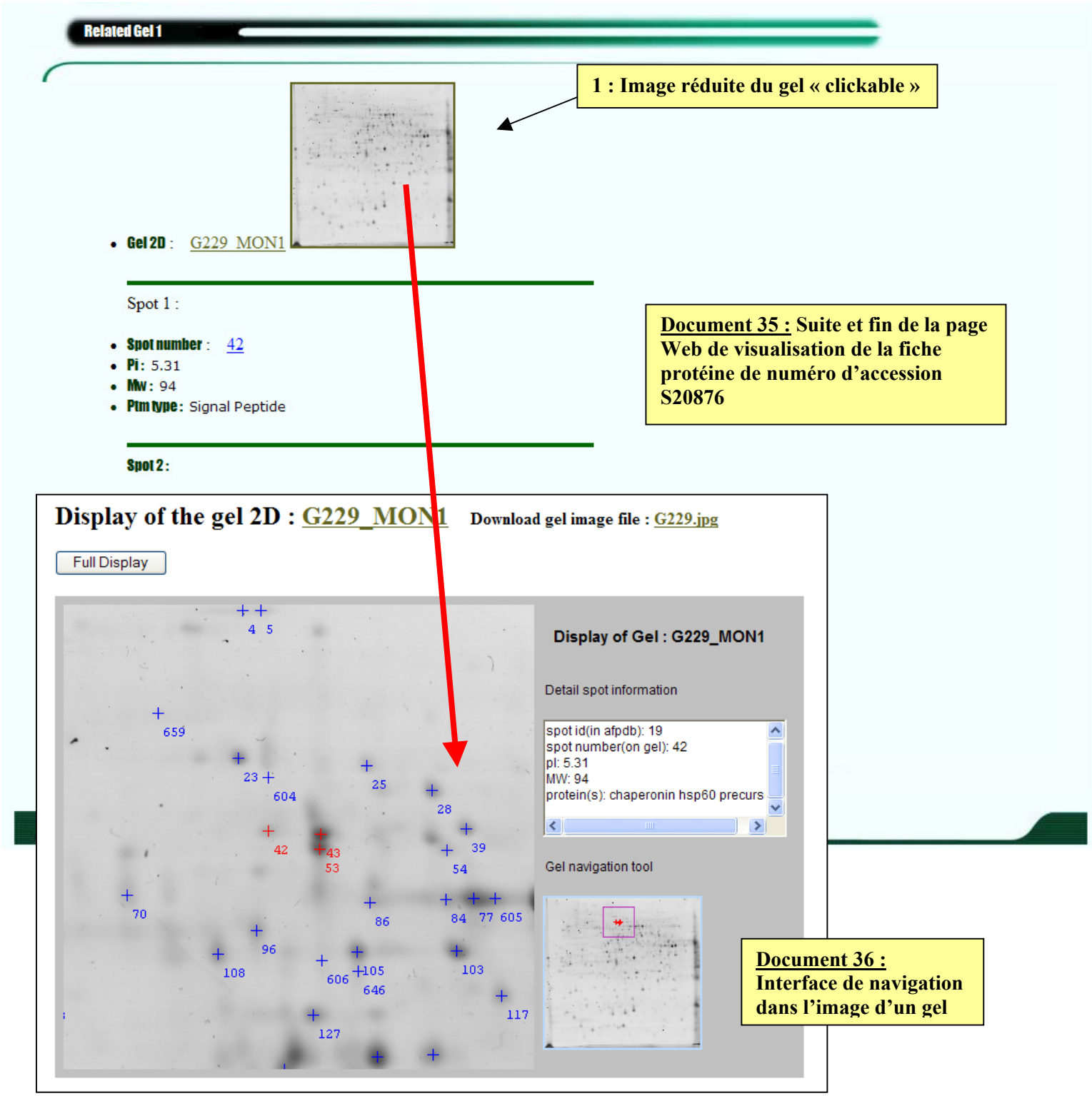
Après avoir « cliqué » sur le protein_id 14 (voir flèche jaune 3), l'utilisateur peut consulter la page Web de visualisation de la fiche détaillée de cette protéine. Cette page propose une structure en section d'informations spécifiques. La première section intitulée « Acc list » concerne l'identité de la protéine avec notamment la liste des numéros d'accessions qui ont permis d'identifier la protéine dans les banques de données publiques suite à l'utilisation de logiciels d'analyses des résultats de spectrométrie de masse comme Mascot. Dans notre exemple nous avons un seul numéro d'accession qui est *S2076* (entouré en rouge).

La deuxième section intitulée « General Information » concerne des informations assez générale sur la protéine. La troisième section intitulée « Links » permet à l'aide d'images se comportant comme des hyperliens d'atteindre la fiche de cette protéine dans un certain nombre de banques de données publiques où elle est susceptible d'être enregistrée. Cet hyperlien se base sur le numéro d'accession *S20876* qui caractérise la protéine étudiée sur la fiche ProteomIs/GnpProt. Nous avons illustré (flèche rouge sur le **document 34**) le lien hypertexte qui permet d'accéder à la fiche correspondante dans la banque de donnée NCBI. Lorsque la protéine identifiée sur la fiche possède un numéro d'accession AGI il est possible de proposer des liens hypertextes sur les banques de données publiques sur Arabidopsis qui utilisent également cette nomenclature AGI (voir **annexe 11** les liens sur les banques de données sur Arabidopsis et **annexe 12** les liens sur l'application FlagDB++ de Génoplante).



Document 34 : Première partie de la page Web de visualisation de la fiche protéine

Ensuite, les prochaines sections présentes sur cette fiche vont fournir un récapitulatif des expériences qui ont permis d'isoler la protéine (voir **document 35**). L'intérêt est de pouvoir regrouper sur une même fiche les résultats expérimentaux des différents laboratoires associés à l'étude d'une même protéine. Notre protéine *S20876* a pu, elle, être isolée par le laboratoire de protéomique de Montpellier grâce à la technique d'électrophorèse bidimensionnelle. Pour cette raison, nous avons une section intitulée « Related Gel » sur notre fiche protéine qui contient une image réduite du gel (flèche jaune 1) à partir duquel la protéine a pu être isolée. En fait, sur ce gel trois spots (42, 43 et 53) correspondent à cette protéine. Si l'on clique sur l'image réduite du gel on accède à l'interface (applet java) qui va nous permettre de naviguer dans le gel (**document 36**).



Document 35 : Suite et fin de la page Web de visualisation de la fiche protéine de numéro d'accession S20876

Document 36 : Interface de navigation dans l'image d'un gel

6 – Naviguer dans l'image d'un gel :

En cliquant sur l'image iconifiée du gel à partir de la fiche protéine, on est positionné directement sur l'ensemble des spots correspondant à cette protéine lors de l'ouverture de l'applet. Les spots concernés par cette protéine apparaissent alors en rouge. Les fonctionnalités de l'applet ont déjà été décrites lors de la phase de maquettage (partie **6.4.2**). En **annexe 24** est présentée la page Web de visualisation de la fiche Spot à laquelle il est possible d'accéder en cliquant sur la croix (hyperlien) qui localise un spot. La fiche Spot permet d'accéder à des informations plus détaillées sur le spot en question. Notamment, la section « Ms_analysis » renseigne sur l'analyse effectuée en spectrométrie de masse pour analyser la composition en protéines du spot prélevé sur le gel. La section « Ms_analysis_result » renseigne sur le résultat de la manipulation d'interrogation qui a permis d'identifier la (ou les) protéine(s) contenues dans le spot. Dans cette section il doit être possible d'accéder au fichier html de résultat d'interrogation (p.ex Mascot).

7.3 Analyse des données

7.3.1 Clustering de protéines

En terme de résultats, le programme de clustering est fonctionnel. Une nouvelle version de ProteomIs non redondante a été crée à partir de ce programme. Lorsque les tests ont été effectués, la base ProteomIs redondante contenait 530 numéros d'accessions de protéines. Parmi ces numéros d'accessions, il existait 376 numéros d'accessions AGI et 154 numéros d'accessions NON AGI comprenant ceux présentés en **annexe 7** et de type Genbank, PIR et SWISSPROT. A la fin du traitement, le programme a trouvé les équivalences AGI pour 124 accessions NON AGI. Il reste donc 30 accessions NON AGI qui n'ont pas d'équivalent AGI et qui sont susceptibles d'être redondant entre eux. A charge alors pour les biologistes de vérifier les équivalence entre les séquences des 30 protéines correspondant aux accessions NON AGI. Ce travail effectué à l'aide de l'interface BLAST permettrait de créer de nouveaux groupes non redondants de séquences (clusters) et ainsi de réduire voir d'éliminer la redondance dans la base ProteomIs.

7.3.2 Comparaison de séquences

Le logiciel BLAST est désormais accessible à travers une interface (présentée **annexe 26**. Cette interface comporte simplement une zone de texte dans laquelle ils peuvent saisir la séquence à comparer avec les séquences en local d'une banque de données. Après validation, les résultats du BLAST apparaissent par ordre décroissant de similarité des séquences comparées avec la séquence requête. Ces résultats sont accompagnés d'une valeur de score. Pour l'instant, la comparaison a été testée avec les séquences d'Arabidopis provenant de la banque TAIR. Il reste comme prévu à pouvoir effectuer la comparaison d'une séquence avec celles contenues dans la base de donnée de ProteomIs. Cette fonctionnalité sera disponible lorsque sera terminé le développement du programme permettant d'importer en local toutes les séquences contenues dans ProteomIs.

7.3.3 Recherche de motifs

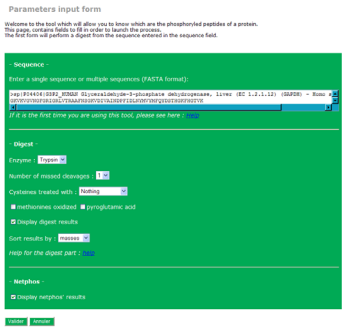
Sur le **document 37** est présentée une synthèse des fonctionnalités attendues dans la version orientée utilisateur l'application de recherche des motifs de phosphorylation. Dans l'état actuel des développements réalisés les résultats de MSDigest et NetPhos sont disponibles mais apparaissent dissociés. Il reste à implémenter la fonction qui permettra de fusionner les résultats des deux programmes avec les données expérimentales.

<u>Document 37 :</u> Schéma récapitulatif des fonctionnalités de la version utilisateur de l'application de recherche des motifs

ENTREE

Séquence ou liste de séquence
>MKSSTYERV…

Liste de masses peptidique



Formulaire HTML

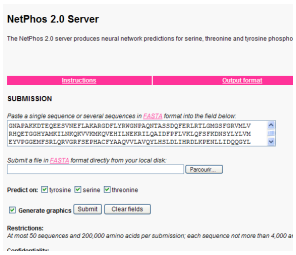
Limites du programme (en Perl)

Module Perl Digestion

Module Perl Netphos

Interrogation

résultats



Site http NetPhos :
http://www.cbs.dtu.dk/services/NetPhos/

Résultats

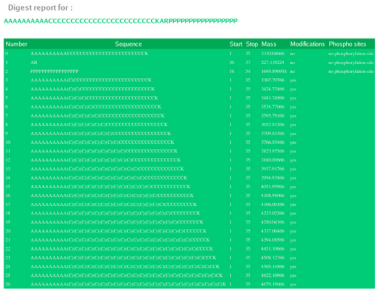
Liste de peptides digérés.

Résultats

Sites de phosphorylation.

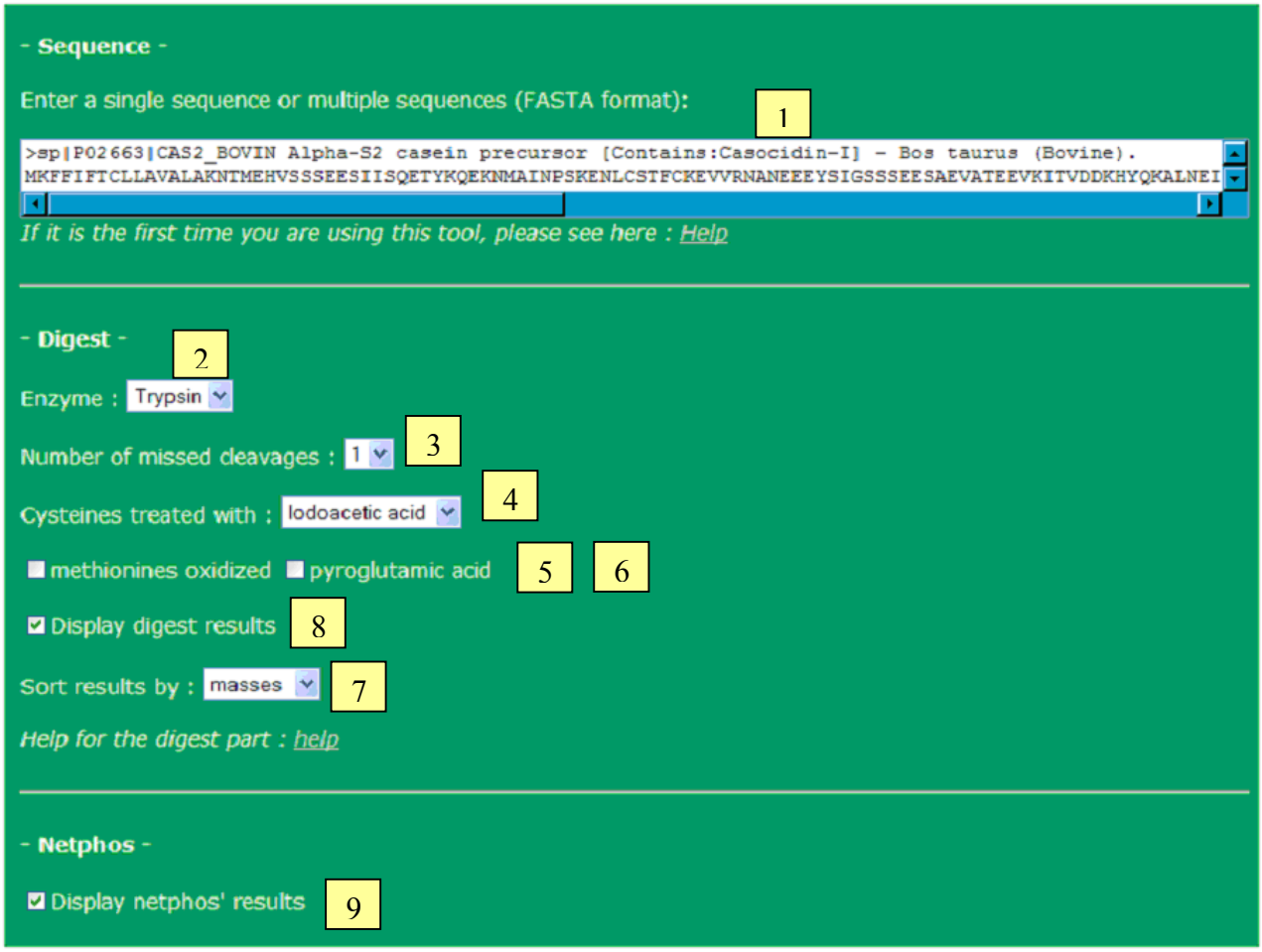
Combinaison des résultats Ms-Digest Netphos.

Combinaison des résultats Ms-Digest Netphos méthode expérimentale.



SORTIE HTML

Le **document 38** présente une copie d'écran du formulaire de saisie des paramètres requis pour l'utilisation du module de digestion et de NetPhos en automatique. Il est possible de voir qu'il existe différentes sections (appelées sequence, digest, et netphos) afin que l'utilisateur comprenne quelles options il est en train de modifier. Des liens vers des pages d'aide permettent de guider les premières utilisations. L'ajout de l'option 8 *display digest results*, permet à l'utilisateur de n'afficher que les résultats de la digestion. L'option 9 *display netphos results*, donne à l'utilisateur la possibilité de voir les résultats renvoyés uniquement par NetPhos.



Document 38 : Formulaire d'interrogation permettant de faire une analyse de la séquence avec MSDigest et NetPhos

1 : Zone de saisie des séquences, l'utilisateur peut en insérer une ou plusieurs.

2 : Choix de l'enzyme de digestion. L'application ne connaît que la trypsine. La liste de choix permettra d'ajouter des enzymes dans le futur.

3 : Nombre de découpes ratées par l'enzyme de digestion (il est possible avec le programme de simuler des erreurs de coupure dans la séquence comme le fait l'enzyme dans la nature)

4 : Pour savoir si l'utilisateur souhaite effectuer une digestion avec des Cystéines (C) modifiées.

5 : Pour savoir si l'utilisateur souhaite effectuer une digestion avec des Méthionines (M) modifiées.

6 : Pour savoir si l'utilisateur souhaite que la Glutamine (Q), en début de protéine, soit modifiée.

7 : Pour trier les résultats affichés. Il est possible de trier par masses, ou par peptide.

Le tri par masses va afficher les peptides par ordre de masse croissante. Le tri par peptide affiche les peptides dans l'ordre de leur traitement (ie : d'abord les peptides non modifiés sans erreur de découpe, puis ceux avec erreur de découpe, et enfin ceux avec modifications).

Voici maintenant sur le **document 39** le résultat final d'une digestion, avec les paramètres suivant :

- Séquence :
>sp|P02663|CAS2_BOVIN Alpha-S2 casein precursor [Contains:Casocidin-I] - Bos taurus (Bovine).
MKFFIFTCLLAVALAKNTMEHVSSSEESIISQETYKQEKNMAINPSKENLCSTFCKEVVR
NANEEEYSIGSSSEESAEVATEEVKITVDDKHYQKALNEINQFYQKFPQYLQYLYQGPIV
LNPWDQVKRNAVPITPTLNREQLSTSEENSKKTVDMESTEVFTKKTKLTEEEKNRLNFLK
KISQRYQKFALPQYLKTVYQHQKAMKPWIQPKTKVIPYVRYL
- Nombre de coupures ratées égal à 1 et cystéines modifiées.

Number	Sequence	Start	Stop	Mass	phosphorylation sites
0	NT(M)EHVSSSEESIISQETYK	17	36	2296.020989	
1	NTMEHV(S)SSEESIISQETYK	17	36	2360.001489	7
2	NTMEHV(S)(S)SEESIISQETYK	17	36	2439.981389	7 8
3	NTMEHV(S)(S)(S)EESIISQETYK	17	36	2519.961289	7 8 9
4	NTMEHV(S)(S)(S)EE(S)IISQETYK	17	36	2599.941189	7 8 9 12
5	NTMEHV(S)(S)(S)EE(S)II(S)QETYK	17	36	2679.921089	7 8 9 12 15

un acide aminé Sérine (S) phosphorylé

Position de l'acide aminé phosphorylé dans la séquence

Document 39 : Résultats obtenus à l'aide du module de digestion

Une amélioration devra être apportée concernant le choix des couleurs, car en l'état actue,l il est difficile de distinguer les peptides écrits en blancs sur un fond vert. De plus, l'acide aminé phosphorylé pourrait être mis en évidence par une couleur différente de celle des autres acides aminés.
Sur le **document 40** est présenté le résultat avec NetPhos.

Letter	Position
S	23
S	24
S	25
S	28
S	31
T	34
Y	35

Un acide aminé Tréonine (T) phosphorylé dans la séquence

Position de l'acide aminé phosphorylé dans la séquence

Document 40 : Résultats obtenus à l'aide du module NetPhos

Il reste donc à mettre en correspondance le résultat d'analyse de ces deux logiciels et à ajouter la liste des masses expérimentales. Nous pourrons ainsi obtenir facilement un tableau du type de celui présenté ci-dessous qui permet de comparer les résultats expérimentaux et bioinformatiques. Il s'agit du tableau que les biologistes construisaient jusqu'alors manuellement et que nous avons déjà présenté dans la partie **5.2.3**.

Résultats théoriques de MSDigest			Résultats théoriques de NetPhos	Résultats expérimentaux
Combinaison :Peptide	Séquence du peptide	Masses théoriques	Validation par NetPhos des résultats de MSDigest	Validation expérimentale des résultats de MSDigest
Combinaison 1 : Peptide 2	AQYSQR	735	No	No
Combinaison 2 : Peptide 2	AQYSQR	815	No	No
Combinaison 3 : Peptide 2	AQYSQR	815	Yes	Yes
Combinaison 4 : Peptide 2	AQYSQR	895	Yes	No

Tableau 4 : Tableau de comparaison des résultats expérimentaux et bioinformatiques

7.5 Test et amélioration de la montée en charge de l'application

Deux versions de l'application ont en fait été testées et livrées à Génoplante Info (voir partie **6.2.4**). Je décris en première partie les tests effectués sur la version 1 de ProteomIs/GnpProt livrée en avril 2004 à Génoplante Info. Les résultats de ces tests montrant une limite au niveau de la montée en charge, je présente ensuite les améliorations techniques qui ont suivi pour conduire à une version 2 de ProteomIs/GnpProt satisfaisante.

➢ Procédure de test de la version 1 d'avril 2004 de ProteomIs

Avant d'être installée sur les postes des différents laboratoire partenaires de Génoplante et sur le serveur de Génoplante Info, l'application web devait être préalablement testée avec l'ensemble des données. Une procédure de test a donc été élaborée pour cela.

Le logiciel devait être testé à la fois sous les systèmes d'exploitation Windows et Linux pour vérifier sa portabilité. J'ai tout d'abord testée l'application ProteomIs sur ma machine de développement fonctionnant sous Windows à Montpellier. Sur cette machine la base de donnée ProteomIs fonctionnant sous Postgres contenait les données des laboratoires de protéomique de Grenoble, Toulouse, Montpellier. J'ai ensuite laissé le soin aux biologistes de l'unité de Montpellier de tester l'application installée avec la base de donnée Postgres cette fois sur un serveur Linux (redhat 9.0). Les caractéristiques de ce serveur sont les suivantes : 1 processeur (i686 à 1,8 Ghz), 500 Mo de RAM, 26 Go de disques. L'application pouvait donc être testée dans un environnement multi-utilisateurs. En fait les biologistes testaient chacune des sous-versions du logiciel que je déposais régulièrement sur ce serveur d'application. Ainsi le jour de la livraison aux clients la version 1 officielle de l'application ProteomIs/GnpProt était validée à Montpellier au niveau de sa portabilité et d'un point de vue fonctionnel et technique.

Il restait ensuite à tester la montée en charge de l'application ("combien l'application peut-elle absorber d'utilisateurs simultanés et quel est le temps d'attente sur les pages"). Une procédure de test devait alors être mise en place pour simuler le trafic auquel devrait faire face l'application une fois installée chez le client. Ce trafic pouvait être assez élevé puisque que l'application, une fois installée sur le serveur de Génoplante Info, devait à terme être rendu accessible à tout l'Internet. Les caractéristiques de ce serveur sont les suivantes : système d'exploitation Solaris 7, machine Sun Enterprise 5500, 12 processeurs (Ultra Sparc II à 400 MHz), 6Go RAM, 1 To de disques. Cependant je n'ai pas eu la possibilité de faire des tests de montée en charge de l'application sur le serveur de Génoplante Info. Les tests ont donc été effectués sur le serveur prodb Linux d'application de l'unité de Montpellier.

Sur l'ensemble des tests, deux requêtes (les plus gourmandes en terme de ressources) se sont montrées limitantes dans la montée en charge de l'application :

- **Requête 1 :** Quel est la liste totale des spots dans la base et pour chaque spot la liste des protéines associées avec leurs accessions ?
- **Requête 2 :** Quel est la liste totale des protéines dans la base et pour chacune de ces protéines quelle est la liste des numéros d'accessions et la localisation subcellulaire de ces protéines ?

Dans cette première version de l'application qui a été livré en avril 2004 à Génoplante j'avais averti les informaticiens de Génoplante Info que les requêtes 1 et 2 précédentes constituaient le facteur limitant au niveau des performances de l'application en terme de temps de réponse et de tenue en charge. Pour établir ce constat j'avais effectué plusieurs séries de tests jusqu'à obtenir des valeurs seuil limitante.

Les tests de montée en charge ont été réalisés avec le logiciel JMeter **[i71]**. Ce logiciel open-source, fourni par la fondation Apache permet à travers une interface simple d'envoyer des requêtes http de type GET à serveur web. L'application prend en charge plusieurs paramètres : une url pour interroger l'application web et un nombre de connexions dans une fréquence choisie

Les tests sont ainsi, d'une certaine manière, automatisés et ainsi il n'est plus nécessaire de faire appel à plusieurs personnes pour tester l'application dans un environnement multi-utilisateurs. Dans le **tableau 5** est présenté un ensemble de tests effectués sur la version 1 de l'application au niveau des requêtes 1 et 2. Le logiciel JMeter fut d'abord paramétré pour mesurer le temps de réponse moyen de ces deux requêtes (test1) puis pour simuler 3 utilisateurs effectuant simultanément la requête 1 puis la requête 2 en moins de 3 secondes (test 2). Le nombre moyen d'échec au niveau de la requête 1 étant de 1/3 et le nombre moyen d'échec étant de 1/5, il a été décidé, en accord avec les informaticiens de Génoplante, qu'un effort devait être fournit pour améliorer la montée en charge de l'application avant sont installation sur le serveur de Génoplante Info.

La version 1 non améliorée de l'application fut cependant installée en avril 2004 sur les serveurs des unités de recherche en protéomique de l'INRA de Montpellier et de l'unité de l'INRA de Nante. Le nombre d'utilisateurs au sein de ces unités étant assez réduit, il n'y a pas eu constatation de problèmes liés à la montée en charge de l'application.

	Version de l'application	type de requête	Nbre d'éléments renvoyés	Nbre de requêtes effectuées (en 3 secondes)	temps de réponse moyen (sur 5 essaies)	Nbre d'échec moyen/essaie
Test 1	1	requête 1	1766	1	10s	0
	1	requête 2	533	1	6	0
Test 2	1	requête 1	1766	3	15 s	1/3
	1	requête 2	533	3	9	1/5

Tableau 5 : Première série de tests : évaluation de la montée en charge de la version 1 de l'application

➢ **Optimisations effectuées sur la version 1 de ProteomIs pour l'amélioration de la montée en charge en vu de son installation sur le serveur de Génoplante Info**

Un certain nombre d'optimisations techniques ont été envisagées pour améliorer les temps de réponse mais aussi la charge mémoire sur le serveur. En effet l'objectif était aussi que l'application consomme un minimum de ressources pour économiser celles du serveur de Génoplante-Info. Certaines de ces optimisations comme la mise en place d'un cache objet ont déjà été présentées et discutées dans ce mémoire au niveau de leur conception. Cette partie **7.4** permet de les replacer dans leur contexte.

- Optimisation 1 : Création de tables temporaires :

Le premier travail devait être de soulager les traitements au niveau de la couche java d'accès aux données. Dans la version 1 de l'application la totalité des requêtes SQL associés aux requêtes 1 et 2 précédentes sur les spots et les protéines étaient effectuées à partir la couche d'accès aux données, le code SQL étant imbriqué dans le code applicatif java. Le résultat de chacune des requêtes 1 et 2 était obtenu à partir de l'assemblage au sein du code java des résultats d'une combinaison de plusieurs requêtes SQL ; chacune de ces requêtes étant composée elle-même d'une combinaison de jointures et de SELECT imbriqués assez importantes. Ces traitements étaient donc importants pour l'application java, et afin d'alléger la charge sur celle-ci une alternative était de déporter ces traitements sur le SGBD lui-même. Pour cela plusieurs solutions furent envisagées.

Une première solution était d'utiliser les procédures stockées. Une procédure stockée est un programme mémorisé au sein du SGBD, qui peut être exécuté comme n'importe quel ordre SQL. PostgreSQL propose le langage *PL/pgSQL* pour l'implémentation de procédures stockées. *PL/pgSQL* est une extension de *SQL* mais disposant de structures de contrôle comme les boucles.

La deuxième solution qui fut envisagée pour soulager la charge cliente fut d'utiliser des vues pour stocker à l'avance le résultat des requêtes SQL. Cependant le PL/pgSQL et les vues sont des traitements consommateurs de ressources au niveau du SGBD puisque ce sont des programmes exécutés à chaque appel.

Une alternative aux procédures stockées et aux vues était de créer des tables dans la base qui contiendrait à l'avance le résultat des requêtes SQL. Cette solution est avantageuse en ce sens que les requêtes sont exécutées une fois pour toute à la création des tables. De plus les colonnes des tables peuvent être indexées ce qui accélère encore l'accès aux données par le SGBD. Il suffit ensuite de récupérer le contenu de ces tables à l'aide d'une requête SELECT. Cependant le problème est qu'il faut recréer les tables à chaque fois qu'il y'a des mises à jour dans la base contrairement aux vues qui sont toujours synchronisées avec le contenu de la base de donnée. Pour cette raison je considère ces tables comme des tables temporaires. Dans notre cas cela n'est pas gênant car les mises à jour dans la base ne sont pas fréquentes puisqu'elles se font à intervalle régulier par l'intermédiaire du format d'échange.

La solution des tables temporaires fut donc retenue et grâce à cette solution on soulageait la charge à la fois sur l'application cliente java et également au niveau de l'application serveur c'est à dire le SGBD. Deux tables temporaires furent crée chacune pour répondre aux requêtes 1 et 2. La première table temporaire est la table nommée *temp_spotband_list* qui permet de répondre à la requête 1 sur les spots. La deuxième table temporaire est la table nommée table *temp_protein_list* permet de répondre à la requête 2 sur les protéines. En annexe **27** est présent le code SQL de chacune de ces deux tables temporaires.

- Optimisation 2 : Création d'un système de navigation page par page et utilisation des curseurs en SQL pour limiter le nombre de lignes dans le navigateur :

Le deuxième travail fut de limiter le nombre de lignes renvoyées par le SGBD au navigateur. En effet 1766 lignes étaient affiché dans le navigateur pour la requête 1. Ensuite pour la requête 2 le nombre de lignes atteignait 533 lignes. Ce nombre de lignes était bien sûr susceptible d'augmenter au fur et à mesure des saisies dans la base.

Les conséquences sont :

- une charge mémoire importante dû à un nombre d'objet Java Beans important instanciés pour chacune des lignes renvoyées par les requêtes 1 et 2 (voir conception de la couche d'accès aux données partie **6.5.2**).

- un nombre important de données à faire transiter par le réseau et à afficher dans le navigateur

Dans le **document 33** de la **partie 7** on peut voir la procédure qui permet à l'utilisateur d'obtenir la réponse à la requête 2 précédente sur les protéines en utilisant l'interface d'interrogation de l'application. On peut voir que le nombre de lignes résultant total affiché dans le navigateur peut être sélectionné dans une liste déroulante (le nombre maximum sélectionnable étant 500). C'est l'utilisation d'un curseur SQL qui permet de sélectionner le nombre de lignes à renvoyer par le SGBD au navigateur. La déclaration d'un curseur en SQL permet d'effectuer une itération sur l'ensemble des tuples renvoyé par une requête **[i72]**. Ceci permet de sélectionner un intervalle de tuples (par exemple de 1er au 501ème tuple) à renvoyer par le SGBD et ainsi fixer une limite au nombre de tuples renvoyés par une requête. Le nombre de lignes renvoyées par le SGBD au navigateur pouvait ainsi être limité afin d'alléger la charge réseau et limiter le nombre de Java Beans instancié dans la couche applicative.

- Optimisation 3 : Mise en place d'un système de cache pour la gestion de la persistance :

Nous avons vu dans la partie **6.5.2** qu'un système de cache objet avait été mis en place pour assurer la persistance dans la couche d'accès aux données. L'avantage avec le cache objet était de pouvoir soulager la mémoire de la machine en ayant en mémoire une seule instance d'un objet Java Beans contenant le résultat d'une ligne de requête.

➢ **Test de performance de la version 2 de décembre 2004 de ProteomIs avec JMeter**

Une fois les différentes optimisations précédentes effectuées sur la version 1 de ProteomIs, une version 2 de ProteomIs pouvait être testée et livrée aux différents laboratoires partenaires et à Génoplante Info. J'ai alors utilisé plusieurs séries de test pour évaluer l'amélioration de la montée en charge de cette deuxième version de ProteomIs. Les test devaient permettre d'évaluer les améliorations apportées sur les valeurs seuil limitantes en terme de tenue de charge observée dans la version 1 précédente de ProteomIs.

Dans le **tableau 6** est présenté un ensemble de tests effectués sur la version 1 et 2 de l'application au niveau des requêtes 1 et 2. Le logiciel JMeter fut d'abord paramétré pour mesurer le temps de réponse moyen de ces deux requêtes (test1) puis pour simuler 10 utilisateurs effectuant simultanément la requête 1 puis la requête 2 en moins de 3 secondes (test 2) en limitant cette fois le nombre d'éléments renvoyés à 500. Le nombre moyen d'échec au niveau de la requête 1 est de 9/10 dans la version 1 de l'application tandis qu'avec la version 2 améliorée de ProteomIs le taux d'échec est de 0. Avec la requête 2, le nombre moyen d'échec est de 7/10 dans la version 1 tandis qu'avec la version 2 le taux d'échec est encore de 0.

Test	Version de l'application	type de requête	Nbre d'éléments renvoyés	Nbre de requêtes effectuées (en 3 secondes)	temps de réponse moyen (sur 5 essaies)	Nbre d'échec moyen/essaie
	1	requête 1	500	1	7 s	0
	1	requête 2	500	1	7 s	0
	2	requête 1	500	1	6 s	0
Test 3	2	requête 2	500	1	5 s	0
	1	requête 1	500	10	51 s	9/10
	1	requête 2	500	10	45 s	7/10
	2	requête 1	500	10	29 s	0
Test 4	2	requête 2	500	10	20 s	0

Tableau 6 : Deuxième série de tests : comparaison des performances entre la version 1 et 2 de l'application

Sur la base de l'ensemble des résultats de ces tests effectués à Montpellier et en accord avec les informaticiens de Génoplante Info l'application se révélait suffisamment robuste pour être installée sur le serveur de Génoplante Info. Aujourd'hui que l'application est installée sur le serveur de Génoplante Info je n'ai pas eu de retour sur d'éventuels problèmes de montée en charge avec ProteomIs/GnpProt. Il serait intéressant de poursuivre les tests sur le serveur de Génoplante Info à l'aide de JMeter. Ceci dit ce serveur étant beaucoup plus puissant que la machine de test il ne devrait normalement pas y avoir de problèmes de tenue de charge.

7.5 Livraison du logiciel et procédure d'installation

➢ **Format de livraison de l'application web**

La partie interfaces de ProteomIs/GnpProt est distribuée sous la forme d'un fichier war. Un fichier war est un fichier d'archive contenant une application web. Pour installer l'application il suffit de placer le fichier war dans le répertoire webapps du serveur web Tomcat et de redémarrer Tomcat. Le fichier war est alors décompressé et l'application installée avec son arborescence de répertoires contenant entre autres dans le cas de ProteomIs/GnpProt : les fichiers accompagnant les données (fichiers images de gel, pdf …), les fichier .class, les jsp, les fichiers xml de configuration … L'application est également livrée avec un fichier XML permettant de faciliter la configuration de l'application ainsi que sa maintenance à l'aide du programme Ant **[i76]**. Ant est une application java open source permettant d'automatiser un certain nombre de tâches à décrire dans un fichier XML appellé build.xml. Il suffit ensuite de se placer dans le répertoire contenant le fichier build.xml et de taper la commande `ant`. Le programme Ant exécutera ensuite chacune des tâches décrites dans le fichier. Parmi les tâches qui ont été automatisées dans le fichier build.xml livré avec ProteomIs/GnpProt il y'a :

- la compilation du code source Java de l'application et la production de fichiers .class (bytecode) à l'aide de la commande javac
- la mise à jour d'un certain nombre de fichiers de configuration de l'application (fichiers de configuration struts-config.xml pour l'accès à la base, fichier xml de configuration du cache …) à partir d'un fichier de propriétés build.properties
- la génération de la documentation html des classes java à l'aide de la commande javadoc
- la génération du fichier war contenant l'application

➢ **Inventaire des éléments distribués et sites d'implantation**

A la date où je soutiens ce mémoire les outils d'analyse bioinformatique des données n'ont pas pu être livrés à Génoplante puisque cette partie est cours de finition (voir partie **8.2 Perspectives**). L'utilisation des outils d'analyse est limitée pour l'instant au personnel de l'unité de recherche en protéomique de Montpellier où j'ai été affecté pour la réalisation de ce projet. Parmi ce qui a pu être livré à Génoplante est présent :

- l'application web en java (fichier war) permettant d'interroger et visualiser les données
- le script de création des tables de la base de donnée (implémenté sous Postgres)
- l'outils permettant la saisie des données (Format d'échange et script d'importation)

Une documentation très complète a également été fournie avec ces différents éléments :

- documentation d'installation de l'application web et documentation utilisateur pour utiliser les interfaces (accessible à partir d'un lien dans le menu de l'application web)
- documentation décrivant la saisie des données dans le format d'échange ainsi que la procédure d'importation de ces données dans la base
- le modèle conceptuel et physique de la base de donnée accompagnée du dictionnaire des données

Au niveau de Génoplante, l'application a également été livrée avec plusieurs formats d'échange remplis respectivement par les laboratoires partenaire suivant :

- UR 1199 de l'INRA de Montpellier
- UMR 5546 du pôle de Biotechnologie Végétale de Toulouse
- UMR 5168 du CEA de Grenoble

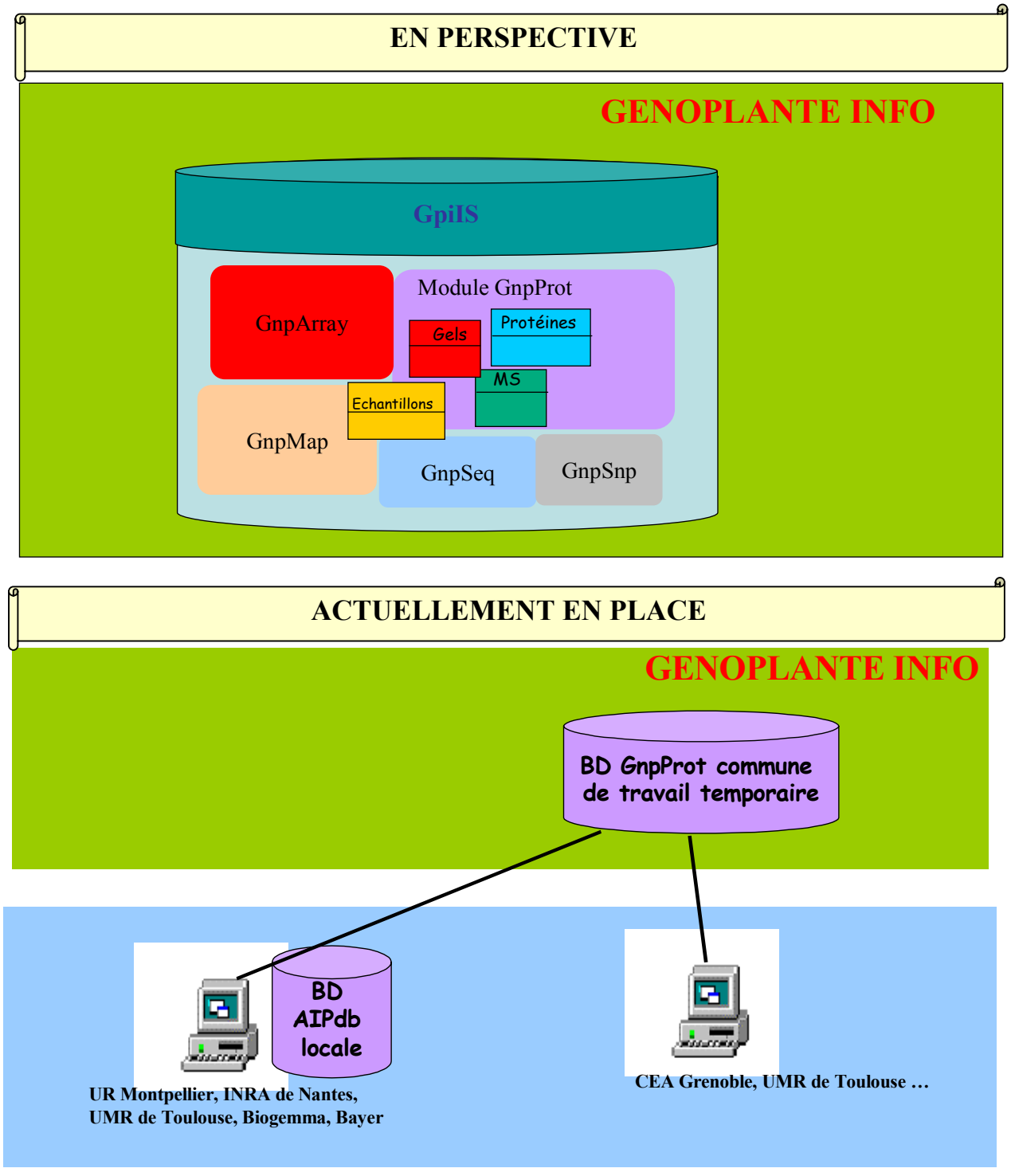
Toutes ces données ont ensuite été importées dans la base de donnée ProteomIs/GnpProt installée sur le serveur de Génoplante Info, à l'aide du programme d'importation correspondant (décrit **partie 7.1**).

La version 2 de ProteomIs/GnpProt a été rendue accessible (sous le nom de GnpProt) depuis le 10 février 2005 sur le portail du site privé de Génoplante Info à l'adresse : https://genoplante.infobiogen.fr. Pour pouvoir accéder à ce site il faut demander un login et un password à Genoplante Info à l'adresse gpsupport@infobiogen.fr

Sur le **document 41** est illustré de quelle manière on accède à GnpProt à partir de ce site. On peut constater que les autres modules de GpiIS sont également accessibles : GnpSeq, GnpMap, GnpArray. Cependant pour l'instant tout ces modules ne sont pas encore interconnectés (voir **partie 8.2**

Document 41 : L'accès à GnpProt sur le site privée de Génoplante Info

En juillet 2005 l'application GnpProt sera accessible sur le site public de Génoplante Info : http://genoplante-info.infobiogen.fr et disponible en open source. La règle à Génoplante est que les données transférées sur leur serveur doivent être accessibles pendant 6 mois sur le site privé avant de pouvoir être rendues publiques en consultation. Pour cette raison, pour l'instant, l'application ne peut être consultée que sur le site privé de Génoplante Info et ne peut être distribuée que dans les différents laboratoires appartenant au consortium Génoplante. Parmi les laboratoires publics de Génoplante où l'application a été installée on peut citer l'unité de recherche en protéomique de Montpellier et également l'unité de protéomique de l'INRA de Nantes. Des laboratoires privés de Génoplante ont également installé ProteomIs avec parmi eux : Biogemma et Bayer CropScience.Sur le **document 42** est illustrée l'implantation actuelle du projet ProteomIs/GnpProt avec une vue globale des perspectives. Ces perspectives seront détaillées dans la **partie 8.2**.



Document 42 : Implantation actuelle du projet AIPdb/GnpProt et perspectives

8 Conclusion et perspectives

8.1 Conclusion

La nécessité de gérer, analyser et corréler des données biologiques de nature diverse (génomiques, protéomiques, transcriptomiques), a conduit le consortium Génoplante à lancer une initiative nationale dont le but est la construction du système d'information GpiIS. Dans ce contexte, ma mission était de prendre en charge le développement d'un des modules du système d'information GpiIS. L'objectif de ce module appelé GnpProt était de pouvoir

- faciliter la gestion et l'exploration des données protéomiques
- analyser ces données d'un point de vue bioinformatique
- confronter ces données avec les autres données biologiques contenues dans GpiIS et également dans d'autres bases de données biologiques.

Parallèlement, ce module (cette fois sous l'appellation de ProteomIs) devait pouvoir fonctionner de manière autonome afin d'être installé localement au sein des différentes unités de recherche intéressées.

Aujourd'hui, la majeure partie de ces objectifs a été réalisée. L'application ProteomIs/GnpProt développée permet, comme prévu, de gérer et explorer les résultats d'analyses protéomiques au moyen d'une base de données PostgreSQL. La collecte de ces données se fait à l'aide d'un format d'échange constitué de feuilles Excel dont le contenu vient, à l'aide de scripts Perl/SQL, alimenter les tables de la base de données. L'exploration de ces données se fait à travers des interfaces conviviales développées en JSP, Servlets, JavaBeans à l'aide du framework Struts. Ces interfaces permettent l'interrogation croisée (par critères ou par mots clés) des données générées par chacune des étapes de la démarche expérimentale protéomique aboutissant à l'identification d'une protéine : extraction des protéines, séparation des protéines (gel d'électrophorèse, chromatographie), identification des protéines par spectrométrie de masse, etc … Pour les données issues de gels, l'application inclut aussi une applet Java pour la visualisation des spots, chacun de ces spots étant relié à une ou plusieurs protéines. L'étude des protéines étant le thème central du système, un effort particulier est consacré à la description de ces objets biologiques. Dans chacune des fiches protéines on réalise la synthèse des expériences des différents laboratoires qui ont participé à sa caractérisation ainsi que les annotations biologiques spécifiques sur ces protéines (fonction biologique, localisation subcellulaire, modifications post-traductionnelles, etc.). La base de données avec toutes ses interfaces, est actuellement accessible (sous l'appellation GnpProt) sur le site privé de Génoplante Info à travers un accès authentifié et réservé aux utilisateurs disposant d'un compte à Génoplante. Cependant le projet devrait être rendu public dans un délai maximum de six mois. Ensuite, l'application ProteomIs a pu être installée et utilisée dans différentes unités de recherche comme celle de l'INRA de Montpellier ou de l'INRA de Nantes et au sein des sociétés Biogemma et Bayer CropScience.

La deuxième partie du projet ProteomIs/GnpProt prévoyait de développer autour du système un ensemble d'outils bioinformatiques facilitant l'étude des protéines. Ces programmes présentés dans ce mémoire sont encore en cours de finition sur le site de Montpellier. La majeure partie de leurs fonctionnalités ayant été développée ces programmes sont utilisables par les biologistes de l'unité de recherche protéomique de Montpellier. Une des priorités a été de créer un programme de « clustering » capable d'éliminer la redondance des séquences protéiques dans la base de données et de normaliser la nomenclature des identifiants de ces protéines au format AGI. Grâce à ce programme, une version non redondante de ProteomIs a pu être proposée et devrait remplacer très prochainement la version redondante. Un ensemble de programmes basés sur l'utilisation de logiciels bioinformatiques et disponible au travers d'interfaces, a ensuite été développé pour l'analyse des séquences biologiques.

Ces analyses peuvent aller de la simple comparaison de ces séquences à l'aide de BLAST à la détection des sites de phosphorylation à l'aide de logiciels de prédiction comme NetPhos …

Le dernier volet du projet ProteomIs/GnpProt concerne l'interopérabilité des données protéomiques avec celle des autres modules de GpiIS mais aussi avec des bases de données biologiques publiques. En ce qui concerne les objectifs d'interopérabilité du système avec les autres bases de données biologiques, l'objectif est atteint. En effet, des liens dans les interfaces de visualisation des données sur les protéines de la base permettent d'accéder directement aux informations correspondantes au niveau de bases de données publiques (comme NCBI, Tair, Aramemnon, SwissProt). On peut également vérifier les mutants disponibles sur le gène correspondant et leur position sur les chromosomes grâce à un lien direct sur l'application FlagDB++ de Génoplante. Par contre il reste à poursuivre les travaux concernant l'intégration du module GnpProt au sein du système GpiIS.

En terme de valorisation, je présenterai ce projet à la conférence JOBIM (Journée Ouverte Biologie Informatique et Mathématique) à Lyon en juillet 2005. De plus, nous envisageons également de publier ces travaux dans une revue scientifique telle que Nucleic Acids Research ou Proteomics.

D'un point de vue plus personnel, le projet ProteomIs/GnpProt m'a offert l'opportunité de piloter les différentes étapes de la conduite d'un projet informatique. De plus, le projet ProteomIs/GnpProt m'a amené à travailler dans un contexte collaboratif avec des personnes de cultures scientifiques différentes (biologistes, informaticiens). Bien qu'ayant déjà été en interaction avec des biologistes et des informaticiens sur d'autres projets, le contexte du projet ProteomIs/GnpProt apportait une dimension nouvelle à ces interactions. En effet, j'étais cette fois responsable de la conduite du projet et de la définition du cahier des charges. Je devais donc être plus à l'écoute des attentes des futurs utilisateurs. La phase d'évaluation des besoins fut une des plus difficiles en raison de l'éloignement géographique des différents laboratoires et de la variabilité dans les protocoles expérimentaux utilisés par ces laboratoires. Cependant l'organisation de réunions avec les utilisateurs, l'envoi de questionnaires, la réalisation de maquettes et une analyse approfondie a permis de prendre en compte la majeure partie des besoins. L'application sera sans doute amenée à être modifiée en raison de nouveaux besoins et de l'évolution des technologies expérimentales. L'utilisation du modèle MVC pour la conception des interfaces permet de s'adapter facilement à ces évolutions et facilite la maintenance du projet ainsi que la réutilisation des différents composants.

En terme de langage informatique, le projet ProteomIs/GnpProt fut ma première expérience en développement java, et même si l'investissement personnel s'est avéré conséquent, je suis aujourd'hui satisfait du résultat. Ce langage objet de par la richesse de ces librairies et des projets open source qui lui sont associés permet d'utiliser une grande quantité de composants prêt à l'emploi. L'utilisation du framework java open source tels que Struts m'a permis notamment de faciliter l'implémentation du modèle MVC dans ProteomIs/GnpProt. Par ailleurs, l'utilisation d'un package de classes téléchargées sur le site d'un livre de chez O'Reilly m'a également beaucoup simplifié le travail. Une alternative aurait pu être l'utilisation du framework Hibernate qui m'aurait facilité entre autres la mise en place de certains aspects assez complexes tels que l'algorithme d'alimentation du cache, les fonctionnalités de pagination et pour l'avenir la migration sous Oracle. Cependant je n'ai remarqué l'existance de cet outil que trop récemment, la majeure partie de ma couche d'accès aux données ayant été implémentée.

Ainsi la réutilisation de briques logicielles ou de programmes préexistants est en mon sens une chose importante pour accélérer le développement d'un projet. De plus, les solutions développées et maintenues par toute une communauté open source sont souvent plus puissantes qu'un programme implémenté hâtivement par une seule personne dans des contraintes de délais forcés. Aussi j'accorde une très large part à l'activité prospective comme peut en témoigner l'importance du chapitre 4 Etat de l'art de ce mémoire. Cette démarche m'a permis par exemple de réutiliser en les adaptant la plupart des composants constituant l'applet de navigation dans les gels.

La veille technologique est d'autant plus importante en bioinformatique qu'un grand nombre de programmes sont disponibles en open source. En terme d'analyse bioinformatique, plusieurs logiciels ont pu être utilisés (BLAST, NetPhos) et la librairie BioPerl a permis d'accélérer les développements en Perl.

Pour conclure, la rédaction de ce mémoire m'a permis de porter un regard critique sur le travail effectué. Si l'ensemble des réalisations me semblent satisfaisantes, il n'en reste pas moins que certaines parties pourraient être améliorées. A la suite de cette synthèse, je présente les perspectives qui me paraissent intéressantes et complémentaires à ce projet.

8.2. Perspectives

Nous présentons ici les développements envisagés pour le projet ProteomIs/GnpProt. Les points prioritaires portent sur la finalisation des avancées déjà présentées dans le cadre de ce mémoire. Dans ce sens, il s'agit d'abord de finaliser le script permettant d'importer automatiquement les séquences dans la base ProteomIs à partir des banques de données publiques. Ensuite, il s'agira de modifier les interfaces de ProteomIs/GnpProt pour qu'elles intègrent les modifications apportées au modèle de données suite au travail sur la gestion de la redondance des séquences (« clustering ») (voir **annexe 25** la maquette du résultat attendu). Puis, il restera à terminer le travail concernant l'implémentation du script de recherche des motifs de phosphorylation afin qu'il fusionne les données bioinformatique et expérimentales. Par la suite, cette dernière application devra être étendue comme prévu (partie **6.2.3**) pour analyser automatiquement toutes les séquences contenues dans ProteomIs et réaliser une sauvegarde de ces résultats dans la base. Enfin, au niveau SGBD, il est prévu d'étendre la portabilité de l'application ProteomIs à Oracle en plus de Postgres.

8.2.1 Perspectives concernant la saisie des données

Nous avons vu que l'insertion des données dans ProteomIs/GnpProt se faisait par l'intermédiaire d'un format de soumission de données (partie **5.1.2**) et de programmes d'insertion correspondants. Un des aspects les plus contraignants de cette solution est qu'elle oblige le gestionnaire de données à corriger les erreurs de saisies afin de les rendre compatibles pour l'insertion. L'automatisation du processus de vérification est une étape obligatoire du processus, sans quoi celle-ci deviendra vite l'étape limitative de tout le système. Pour cela un programme de vérification des données est prévu pour être mis à la disposition des biologistes. Ce système ayant déjà été développé pour les autres modules de GpiIS, il reste à l'intégrer au processus de saisie de ProteomIs/GnpProt.

8.2.2 Perspectives concernant la recherche des motifs

➢Utilisation d'InterproScan dans la chaîne de traitement déjà existante

Nous avons présenté une application permettant la recherche des motifs de phosphorylation à partir d'un lot de séquences en provenance d'un utilisateur. Cette application doit à terme pouvoir analyser automatiquement toutes les séquences contenues dans la base ProteomIs. Cependant, comme évoqué dans la partie **5.2.3,** ce travail de prédiction sur un seul motif spécifique doit être étendu à tous les types de motifs mais aussi aux domaines protéiques. Dans la partie **4.2.2 Etat de l'art,** nous avons énuméré un certain nombre d'outils spécialisés dans la recherche de motif et domaines de toutes sortes. L'outil qui a été retenu pour cela est InterproScan pour les avantages déjà évoqués en **4.2.2**. L'objectif est d'intégrer cet outil à la chaîne de traitement déjà existante.

➢ Intégration des données d'annotations de SWISSPROT sur les motif

Nous avons vu en **4.1.1** que SWISSPROT avait pour avantage d'être une base de données dont les résultats biologiques sur les séquences protéiques étaient vérifiés par des experts.
L'objectif est de récupérer les données sur les motifs annotés de SWISSPROT (correspondant au champ Feature, voir **4.4.2**) pour les séquences contenues dans ProteomIs. Cependant toutes les séquences de ProteomIs ne sont pas contenues dans SWISSPROT. De ce fait, InterproScan, par son approche systématique, viendra compléter ces résultats.

➢ **Développement d'une interface de visualisation des motifs**

L'objectif de cette interface sera de réunir pour une séquence donnée au sein d'une même vue les motifs des pipelines de phosphorylation, InterproScan et de la base SWISSPROT. La représentation des données doit de plus offrir une vue graphique de la position des motifs sur la séquence *(et non plus une vue textuelle comme en 7.3.3)*. Sur le **document 43** est présentée une maquette de ce que pourrait être cette interface. Celle-ci est directement inspirée de l'applet de visualisation des motifs de SWISSPROT **[i78]** présentée dans le **document 8** de la partie **4.2.2** . En fait, le code source de cette interface est disponible en libre téléchargement. J'ai prévu de l'adapter pour qu'elle réponde aux objectifs du projet en intégrant en plus les motifs de phosphorylations et ceux d'Interpro.

➢ **Gestion de l'interopérabilité des données de phosphorylation, InterproScan et SWISSPROT dans ProteomIs**

L'objectif est de mettre en correspondance les données sur les motifs de phosphorylation InterproScan et SWISSPROT afin de permettre la consultation de ces données au sein de l'interface graphique précédente. Il se pose alors le problème de l'interopérabilité de ces données au sein du système ProteomIs. Dans la partie **4.3**, nous avons décrit deux approches permettant d'assurer l'interopérabilité entre différents systèmes : il s'agit de l'approche entrepôt de données/centralisée et de l'approche décentralisée/distribuée.

1- L'approche entrepôt de données/centralisée :

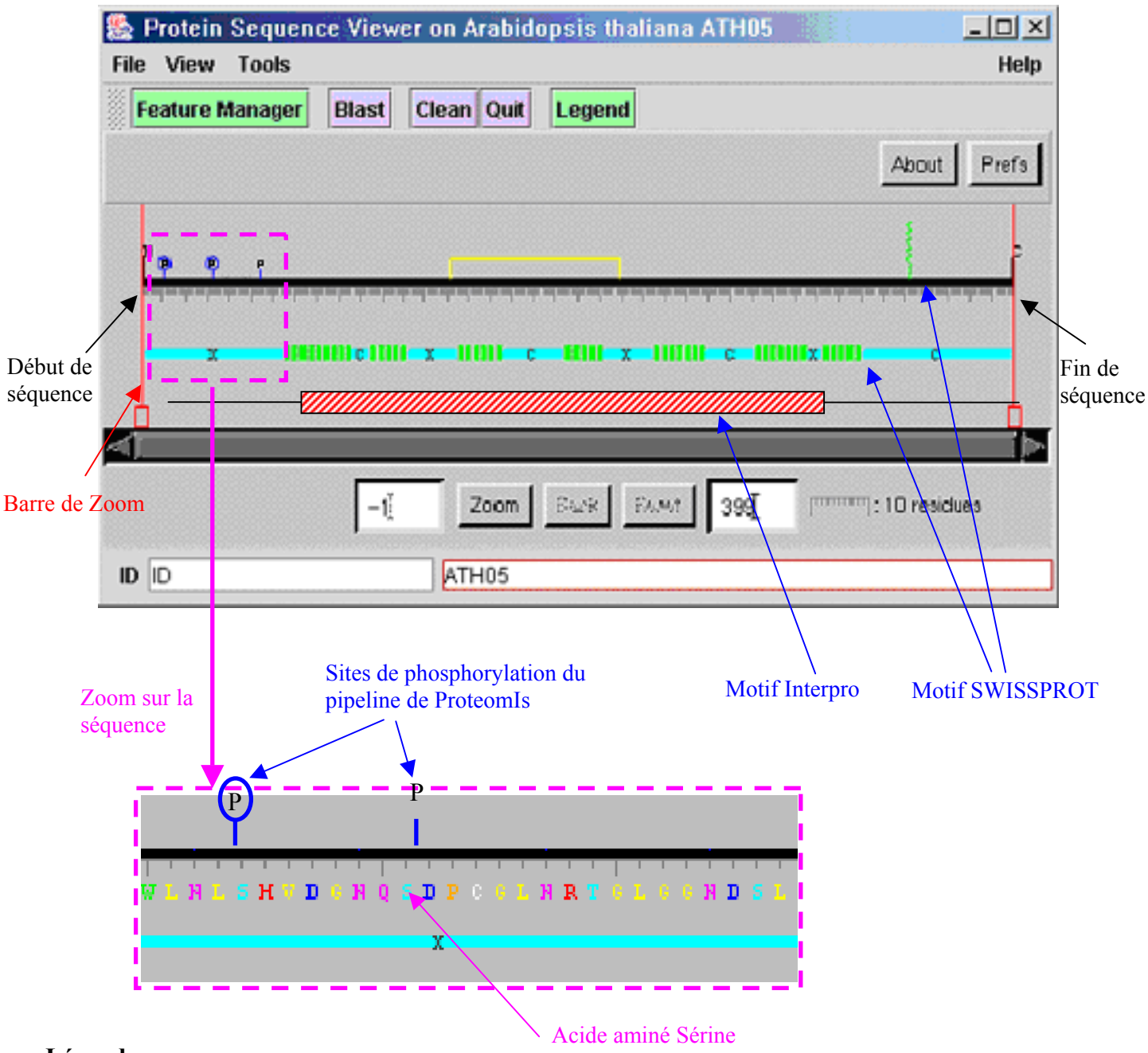
Cette solution consiste à gérer en local et au sein de ProteomIs les données sur les motifs de phosphorylation, InterproScan et SWISSPROT pour les rendre accessibles par l'interface de visualisation. Il se pose alors le problèmes de la gestion d'un nombre toujours croissant de données (mise à jours régulière, gestion de l'espace disque, maintenance et évolution de l'extension du schéma, compatibilité avec Postgres …). Une alternative à ce problème consiste à réfléchir à une approche non matérialisée/décentralisée (voir partie **4.3**) pour garantir l'accessibilité et l'interopérabilité de ces données.

2- L'approche non matérialisée/décentralisée :

L'approche non matérialisée/décentralisée est l'inverse de l'approche entrepôt de données/centralisée. Dans l'approche décentralisée, l'idée est d'interroger directement les producteurs de données (Interpro, SWISSPROT) sur leur site respectif plutôt que de gérer en local les données de ces bases. Dans le cadre de notre projet, la requête exécutée à partir de l'interface de visualisation pourrait être du type : « Quels sont, pour la séquence X, les motifs et domaines disponibles dans les bases de données Interpro et SWISSPROT». L'utilisation de web services devrait permettre d'accéder aux données distantes.

Un des inconvénient majeur de ce genre d'approche restera le temps d'attente pour l'obtention des résultats d'une requête. De plus, on reste dépendant du traffic assez élevé sur les bases de données Interpro et SWISSPROT. Ensuite, il faut mettre en correspondance les différentes données avant de les afficher dans une interface. Dans le cadre de la solution intégrée précédente, on évite ce genre de traitement puisque la correspondance entre les données est en grande partie enregistrée dans la base.

Les temps de réponse devraient être plus rapides avec l'approche intégrée. Le travail de maintenance des données en local sera par contre plus élevé. Ainsi, chacune des approches a ses avantages et ses inconvénients. Il reste maintenant à poursuivre l'évaluation de ces solutions afin d'effectuer un choix qui dépendra aussi des contraintes d'intégration avec le système GpiIS.



Légende :

P — Site de phosphorylation validé expérimentalement

P — Site de phosphorylation suggéré par Netphos

<u>Document 43 :</u> Maquette de l'application de visualisation des motifs protéiques de ProteomIs (Protein Sequence Viewer)

8.2.3 Perspectives concernant l'interopérabilité

➢ Création de nouvelles tables de liaisons inter-modules dans GpiIS

Nous avons vu (partie **6.3.2**) que pour assurer l'interopérabilité entre GnProt et les autres bases de GpiIS (GnpSeq, GnpMap …), nous avions établi des tables de liaisons décrivant des réalités biologiques communes. Cet effort doit être poursuivi afin que d'autres tables permettent d'établir des liens. La table séquence par exemple pourrait être adaptée afin de constituer le pivot d'une relation entre les objets protéines et gènes des autres bases ; l'utilisation des identifiants AGI venant par ailleurs renforcer ce lien.

➢ Développement des liens Web et interfaces permettant de faire des requêtes croisées

Au niveau des interfaces d'accès aux données, des liens Web de plus en plus nombreux apparaîtront sur chaque interface spécifique de GnpProt pour avoir des informations provenant des autres modules. Ces liens permettront donc de "sauter" d'une thématique à l'autre de façon transparente pour l'utilisateur. Un exemple déjà implémenté (voir **annexe 12**) est le lien Web qui est établit entre GnpProt (thématique protéomique) et FlagDB++ (thématique Génomique et Mutant d'insertion **[G1]**). D'une manière générale, aussi bien au niveau des bases de données que des interfaces de consultation, tous ces développements devront être prévus de manière à ce que la consultation des données se fasse de la façon la plus transparente possible. L'étape suivante sera la mise en oeuvre d'interfaces d'interrogation ayant la capacité d'effectuer des requêtes sur plusieurs bases de données à la fois. C'est elle qui donnera toute sa puissance à l'intégration des données sur une même plate-forme. Cette étape utilisera pleinement les tables de liaisons déjà évoquées. Les interfaces d'interrogations se brancheront sur les différents modules de GpiIS, leur développement sera facilité par l'utilisation de technologies communes au-dessus de chacune des bases. On développera également des interfaces utilisant des requêtes prédéfinies (typiquement, requête récupérant toute l'information disponible dans toutes les bases sur un gène donné).

➢ Développement d'un système de gestion des projets

Un système d'authentification pour accéder aux différents modules de GpiIS est également nécessaire en vue de permettre aux différents projets de garder leur confidentialité. Une gestion des utilisateurs et de leur appartenance à un ou plusieurs groupes est donc à mettre en place dans GnpProt. Une première version de ce système a été testée par GPI sur les bases transcriptome, Séquence, et sur la base Carto de GpiIS et reste à valider pour le module SNP.

➢ Intégration des Ontologies dans ProteomIs

La conception d'un système d'information comme GpiIS intégrant dans un même entrepôt plusieurs types de données biologiques (séquences, génomes annotés, transcriptome, protéome …) entraîne une hétérogénéité sémantique nécessitant de standardiser le vocabulaire à l'aide des ontologies (définies dans la partie **4.3.2**). Cette hétérogénéité de sens est tout aussi présente au sein de ProteomIs. Par exemple, une protéine peut avoir été associée au niveau de sa description ou de son nom avec un terme biologique qu'un autre laboratoire n'utiliserait pas pour la même protéine. L'association de cette protéine à un terme issue d'un vocabulaire contrôlé dans une ontologie du domaine permettrait d'augmenter les chances de retrouver cette protéine lorsque le biologiste utilise une interrogation par mot clé. Afin d'implémenter cette fonctionnalité, il sera question, comme pour les motifs, de réfléchir à une solution d'intégration : il faudra en effet faire un choix entre une gestion locale de données provenant de ressources externes (ici ce sera les ontologies issues du projet Gene Ontology) et l'interrogation à distance de ces ressources sur leur site d'origine : http://www.geneontology.org/

BIBLIOGRAPHIE

Références citées dans ce mémoire :

Sites Internet :

[i1] Site officiel de Génoplante Info : http://genoplante-info.infobiogen.fr/

[i2] Site officiel de l'INRA : http://www.inra.fr/presentation-inra/

[i3] Site officiel du centre INRA de Montpellier : http://www.montpellier.inra.fr/internet/centre/presentation.html

[i4] http://www.matrixscience.com/

[i5] TAIR : http://www.arabidopsis.org/index.jsp

[i6] GpiIS : http://genoplante-info.infobiogen.fr/data/

[i7] La solution SQL*LIMS appliquée à la plate-forme protéomique du Génopôle Toulouse Midi-Pyrénées : http://www-helix.inrialpes.fr/article615.html#3

[i7] The Make2ddb package : http://us.expasy.org/ch2d/make2ddb.html

[i8] PROTICdb : http://moulon.inra.fr/~bioinfo/PROTICdb/

[i9] PARIS : http://www.inra.fr/bia/J/imaste/Projets/PARIS/

[i10] ARAMEMNON, a Novel Database for Arabidopsis Integral Membrane Proteins : http://aramemnon.botanik.uni-koeln.de/

[i11] MIPS Arabidopsis thaliana Database (MAtDB): an integrated biological knowledge resource based on the first complete plant genome.: http://mips.gsf.de/proj/thal/db/

[i12] Les sites constituant la plate-forme protéomique de service de Montpellier : http://www.genopole-montpellier-lr.org/PF/proteome/index.htm

[i13] Le site de Zope : http://www.zope.org/

[i14] Pedant: http://pedant.gsf.de/

[i15] Le réseau proteome vert : http://www.pierroton.inra.fr/genetics/2D/Proteomevert/

[i16] SWISS2DPAGE : http://www.expasy.org/ch2d/

[i17] World-2Dpage : http://au.expasy.org/ch2d/2d-index.html

[i18] BLAST au ncbi : http://www.ncbi.nlm.nih.gov/BLAST/

[i19] FASTA (GENET) : http://www.univ-tours.fr/genet/gen001400_fichiers/chap5/genach5ec22.htm

[i20] PROSITE : http://www.expasy.ch/prosite/

[i21] ScanProsite : http://www.expasy.org/tools/scanprosite/

[i22] Expasy : http://www.expasy.org

[i23] PRINTS : http://bioinf.man.ac.uk/dbbrowser/PRINTS/

[i24] Interpro : http://www.ebi.ac.uk/interpro/

[i25] Pfam : http://www.sanger.ac.uk/Software/Pfam/

[i26] InterproScan : http://www.ebi.ac.uk/InterProScan/

[i27] Phospho.ELM : http://phospho.elm.eu.org/

[i28] SEView: a Java applet for browsing molecular sequence dataThomas Junier and Philipp Bucher : http://www.bioinfo.de/isb/1998/01/0003/main.html

[i29] NetPhos : http://www.cbs.dtu.dk/services/NetPhos/

[i30] EMBOSS : http://emboss.sourceforge.net/

[i31] GCG : http://www.accelrys.com/products/gcg_wisconsin_package/index.html

[i32] BioPerl : http://bioperl.org/

[i33] MS-Digest : http://prospector.ucsf.edu/ucsfhtml4.0/msdigest.htm

[i34] LWP : http://search.cpan.org/~gaas/libwww-perl/lib/LWP.pm

[i35] Didier GIRARD (Mars 2001), *OpenSourceJava-Deuxième partie : servlet/JSP/MVC avec Tomcat*, : http://www. http://www.application-servers.com/articles/pdf/opensourcejava-tomcat.pdf

[i36] Les expressions régulières en Perl : http://www.april.org/groupes/doc/perl/perl-6.html

[i37] Biomoby : http://www.biomoby.org.

[i38] PSA : http://python.fyxm.net/psa/

[i39] Le module BioPerl pour NetPhos : http://doc.bioperl.org/bioperl-live/Bio/Tools/Analysis/Protein/NetPhos.html

[i40] ClustalW : http://www.infobiogen.fr/services/analyseq/cgi-bin/clustalw_in.pl

[i41] GMOD Modular Schema – Chado : http://www.gmod.org/schema/index.shtml

[i42] FlagDB++ : http://193.51.165.9/projects/FLAGdb++/HTML/index.shtml

[i43] PIR : http://pir.georgetown.edu/

[i44] Mapping objet-relationnel, Couches d'accès aux données et Frameworks de persistance *par Sébastien Ros :* http://www.dotnetguru.org/articles/Persistance/livreblanc/ormapping.htm

[i45] Choisir un outil de mapping objet-relationnel : http://www.dotnetguru.org/articles/articlets/choixmapping/mapping.htm

[i46] SYSRA : http://www.sysra.com/

[i47] Hibernate : http://www.hibernate.org/

[i48]Spring : http://www.springframework.org/

[i49] Struts : http://struts.apache.org

[i50] OJB : http://db.apache.org/ojb/

[i51] Persistance et mapping : http://www.objectiva.fr/dossiers.2002-10-17_persistance.php

[i52] Le site privée de Génoplante Info : https://genoplante.infobiogen.fr/

[i53] ehcache : http://ehcache.sourceforge.net/

[i54] ArgoUML : http://argouml.tigris.org/

[i55] Rational Rose : http://www-306.ibm.com/software/rational/

[i56] Mouse Genome Database (MGD) : http://www.informatics.jax.org/

[i57] GO, Gene Ontology : http://www.geneontology.org/

[i58] Infobiogen : http://www.infobiogen.fr/

[i59] EBI : http://www.ebi.ac.uk/

[i60] PlaNet http://www.eu-plant-genome.net/

[i61] Mygrid : http://www.mygrid.org.uk/

[i62] MGIS (*Musa* Germplasm Information System) : http://mgis.grinfo.net/Homepage.htm

[i63] TropGeneDB : http://tropgenedb.cirad.fr/

[i64] Comparaison BioMoby et LeSelect Pierre Larmande : http://www.cines.fr/textes/pl.pdf

[i65] Le Web sémantique, une infrastructure d'intégration de sources de données Chantal Reynaud : http://www.lri.fr/~cr/Exposes/ParisV.ppt

[i66] ELOGE : http://tagc.univ-mrs.fr/bioinformatics/eloge/

[i67] Integration of Biological Sources: Current Systems and Challenges. Ahead Thomas Hernandez & Subbarao Kambhampati : http://www.public.asu.edu/~thomas98/papers/biosurvey_sigrec04.pdf

[i68] Htdig : http://www.htdig.org/

[i69] Webglimpse : http://webglimpse.net/

[i70] L'unité de protéomique de l'INRA de Montpellier : http://www.montpellier.inra.fr/internet/recherche/unites/unites/proteomique.pdf

[i71] Jmeter : http://jakarta.apache.org/jmeter/

[i72] Performance Tips for the Data Tier (JDBC) By John Goodson : http://www.theserverside.com/articles/article.tss?l=JDBCPerformance_PartII

[i73] MySQL : http://www.mysql.com/

[i74] PostgreSQL : http://www.postgresql.org/

[i75] Oracle : http://www.oracle.com/database/index.html

[i76] Ant : http://ant.apache.org/

[i77] ProDom : http://protein.toulouse.inra.fr/prodom/current/html/home.php

[i78] JavaServer Pages Hans Bergsten O' Reilly, 2001 : http://www.oreilly.fr/catalogue/javaserver-pages.html

[i79] GeneFarm : http://genoplante-info.infobiogen.fr/Genefarm/index.htpl

[i80] Predotar : http://genoplante-info.infobiogen.fr/predotar/

[i81] EMBL : http://www.ebi.ac.uk/embl/

[i82]Entrez : http://www.ncbi.nlm.nih.gov/entrez

[i83] DDBJ : http://www.ddbj.nig.ac.jp/

[i84] Introduction aux Java Server Pages : http://www.commentcamarche.net/jsp/jspintro.php3

[i85] SWISSPROT : http://www.expasy.org/sprot/

[i86] Uniprot : http://www.expasy.uniprot.org/

[i87] Java Web Start : http://java.sun.com/products/javawebstart/

[i88] J2EE : http://java.sun.com/j2ee/

[i89] Servlet : http://java.sun.com/products/servlet

[i90] Introduction aux servlets : http://www.commentcamarche.net/servlets/servintro.php3

[i91] CGI : http://www.w3.org/CGI

[i92] Introduction aux CGI : http://www.commentcamarche.net/cgi/cgiintro.php3

[i93] JSP : http://java.sun.com/products/jsp/

Articles :

[a1] GénoPlante-Info (GPI) : a collection of databases and bioinformatics resources for plant genomics *Nucleic Acids Research (2003), Vol. 31, No.1 179-182* Delphine Samson, Fabrice Legeai.

[a2] The Make 2D-DB II package: Conversion of federated two-dimensional gel electrophoresis databases into a relational format and interconnection of distributed databases. *PROTEOMICS, Volume 3, Issue 8, Pages 1385-1657 (August 2003)* Khaled Mostaguir [1*], Christine Hoogland [1], Pierre-Alain Binz [1], Ron D. Appel [1 2 3]

[a3] PPMdb: a plant plasma membrane database. *Journal of Biotechnology 78 (2000) 235–246*
Ilhem Sahnoun a, Patrice De´hais a, Marc Van Montagu a, Michel Rossignol b, Pierre Rouze´ c,*

[a4] TAIR: a resource for integrated Arabidopsisdata. *Funct Integr Genomics (2002) 2:239–253*
Margarita Garcia-Hernandez · Tanya Z Berardini

[a5] MIPS Arabidopsis thaliana Database (MAtDB): an integrated biological knowledge resource for plant genomics. Heiko Schoof1,2,*, Rebecca Ernst1, Vladimir Nazarov1, Lukas Pfeifer1,
Hans-Werner Mewes1,2 and Klaus F. X. Mayer1

[a6] ARAMEMNON, a Novel Database for Arabidopsis Integral Membrane Proteins
Plant Physiol. 2003 Jan;131(1):16-26. Rainer Schwacke,* Anja Schneider, Eric van der Graaff, Karsten Fischer

[a7] Functional and structural genomics using PEDANT. *Bioinformatics. 2001 Jan;17(1):44-57*
Frishman D, Albermann K, Hani J, Heumann K, Metanomski A, Zollner A, Mewes HW.

[a8] Removing redundancy in swiss-prot and trembl. *Bioinformatics*, *vol. 15, n-.25em.2ex 3, 1999, pp. 258--259.* O'Donovan (Claire), Martin (Maria Jesus), Glemet (Eric), Codani (Jean-Jacques) et Apweiler (Rolf).

[a9] LASSAP, a LArge Scale Sequence compArison Package *Bioinformatics, Vol 13, 137-143*
E Glemet and JJ Codani

[a10] SWISS-2DPAGE, ten years later. *Proteomics 2004, 4(8), 2352-2356.* Hoogland C., Mostaguir K., Sanchez J.-C., Hochstrasser D.F., Appel R.D.

[a11] Basic local alignment search tool. *J. Mol. Biol., 215, 403-410.* Altschul, S. F., Gish, W., Miller, W., Myers, E. W., & Lipman, D. J. (1990).

[a12] ProDom and ProDom-CG : tools for protein domain analysis and whole genome comparisons., *Nucleic Acids Res., 2000, 28, 267-269.* Corpet F, Servant F, Gouzy J, Kahn D.

[a13] Sequence- and structure-based prediction of eukaryotic protein phosphorylation sites. *Journal of Molecular Biology: 294(5): 1351-1362, 1999.* Blom, N., Gammeltoft, S., and Brunak, S.

[a14]The SWISS-PROT protein sequence data bank and its supplement TrEMBL. *Nucleic Acids Res. 25:31-36(1997).* Bairoch A. and Apweiler R.

[a15] K2/Kleisli and GUS : Experiments in Integrated Access to Genomic Data Sources. *IBM Systems Journal, Vol.40, Num. 2, pp. 512-531, 2000.* S. B. Davidson, J. Crabtree, B. Runk, J. Schug, V. Tannen, G.C. Overton, and C.J. Stoeckert.

[a16] The Gene Ontology Consortium. Gene Ontology : tool for the unification of biology. *Nature Genetics, Vol. 25 pp. 25-29, 2000.*

[a17] H.M.Wain, R.C. Lovering, E.A. Bruford, M.J. Lush, M.W.Wright, and S. Povey. Guidelines for Human Gene Nomenclature. *Genomics, Vol. 79, Num. 4, pp. 464-470, 2002.*

[a18] Federated 2-DE database: a simple means of publishing 2-DE data. *Electrophoresis 17, 1996, 540-546*, 1996. R.D. Appel, A. Bairoch, J.C. Sanchez, J.R. Vargas, O. Golaz, C. Pasquali, D.F. Hochstrasser.

[a19] SRS--an indexing and retrieval tool for flat file data libraries. *Bioinformatics, Vol 9, 49-57, 1993.* T Etzold and P Argos

[a20] BioMOBY : An open source biological web services proposal. *Briefings in Bioinformatics, 3, 331-341.* Wilkinson,M. and Links,M. (2002)

[a21] The PlaNet consortium : a network of European plant databases connecting plant genome data in integrated biological knowledge resource. *Comparative and Functionnal Genomics 2004; 5: 184-189.* H. School, R. Ernst and K. F. X. Mayer

[a22] WIEDERHOLD G., « Mediators in the Architecture of Future Information Systems »,
IEEE Computer, 1992, p. 38-49.

[a23] K2/Kleisli and GUS : Experiments in Integrated Access to Genomic Data Sources. IBM Systems *Journal, Vol.40, Num. 2, pp. 512-531, 2000.* S. B. Davidson, J. Crabtree, B. Runk, J. Schug, V. Tannen, G.C. Overton, and C.J. Stoeckert

[a24] Creating and Maintaining Curated View Databases. *Knowledge Discovery and Data Mining in biological Databases, 2001.* S. B. Davidson, H. Liefke, and L. Wong.

[a25] « TAMBIS : Transparent Access to Multiple Bioinformatics Information Sources », Bioinformtics, vol. 16, n. 2, 2000, p. 184-186. STEVENS R., BAKER P., BECHHOFER S., NG G., JACOBY A., PATON N., GOBLE, C., BRASS A.

[a26] A Strategy for Database Interoperation. *Journal of Computational Biology, 1996.* Peter D. Karp

[a27] Modélisation d'un entrepôt de données dédié à l'analyse du transcriptome hépatique. *Actes de JOBIM, pp. 319-324, 2002.* E. Guerin, F. Moussouni, B. Courselaud, and O. Loréal.

[a28] OIL in a Nutshell. Knowledge Acquisition, *Modeling and Management, pp. 1-16, 2000.* D. Fensel, I. Horrocks, F. van Harmelen, S. Decker, M. Erdmann, and M. C. A. Klein

[a29] DiscoveryLink : A system for integrated access to life sciences data sources. *IBM Syst. J., Vol.40, Num.2, pp.489-511, 2000.* L. Haas, P.Schwarz, P. Kodali, E. Koltar, J. Rice, and W. Swope.

[a30] PARIS: a proteomic analysis and resources indexation system. *Bioinformatics Vol. 20 no. 1 2004, pages 133-135* Juhui Wang [1,*], Christophe Caron [1], Michel-Yves Mistou [2], Christophe Gitton [2] and Alain Trubuil [1]

Livres :

[L1] JavaServer Pages. *O' Reilly, 2001.* Hans Bergsten

[L2] Jakarta Struts par la pratique. *Eyrolles, 2002.* James Goodwill

[L3] Struts. CampusPress. James Trurner & Kevin Bedell

[L4] Les Cahiers du Programmeur : UML Modéliser un site e-commerce. *Eyrolles.* Pascal Roques

[L5] Introduction à la bioinformatique. *O' Reilly, 1re édition, février 2002.* Cynthia Gibas & Per Jambeck

[L6] Introduction à Perl pour la bioinformatique. *O'Reilly, 2002.* James Tisdall

[L7] J2EE et les design pattern. *Campus Press*

Rapports :

[r1] Rapport de Stage de Maîtrise Génie Mathématique et Informatique, IUP d'Avignon. Titre : « Projet AFPDB ». Soutenu en septembre 2003 par M.Arnaud Nemrod. Stage du 15 avril 2003 au 30 août 2003 à l'UR Protéomique INRA de Montpellier et encadré par M. BOUTTES Cédric.

[r2] Rapport de Stage de DESS IAO (Informatique Appliquée aux Organisations » à l'Université Montpellier 2). Titre : « Développement d'applications bioinformatiques pour l'organisation des accessions au sein d'une base de données protéomique. ». Soutenu en septembre 2004 par M.Mohamed Ndiaye. Stage de juin 2004 à septembre 2004 à l'UR Protéomique INRA de Montpellier et encadré par M.BOUTTES Cédric et Mme Isabelle Mougenot

[r3] Rapport de Stage de deuxième année d'école d'ingénieur à l'EPSI Montpellier
Titre : « Outils d'analyse de protéines ». Soutenu en octobre 2004 par M.Cyril GENIN. Stage de juin à août 2004 à l'UR Protéomique INRA de Montpellier et encadré par M.BOUTTES

[r4] PARIS Proteomic Analysis and Resources Indexation System. Mémoire CNAM .Soutenu le 3 mai 2004 par Christophe Caron

GLOSSAIRE

Liens utiles :

Liste de DICTIONNAIRES spécialisés en ligne :
http://www.infobiogen.fr/deambulum/menu.php?page=dictionnaires
GLOSSAIRE en ligne spécialisé en biologie et bioinformatique :
http://www.infobiogen.fr/glossaire/glossaire_idx.php

[G1] Mutants d'insertion :
Introduction d'une mutation précise dans un fragment d'ADN cloné.

[G2] Les Javabeans :
Il s'agit de classes java possédant essentiellement deux particularités :
- Elle doit rendre accessible ces propriétés publiques grâce à des méthodes set (écrire dans la variable) et get (lecture de la variable).
- Elle doit aussi fournir un constructeur (fonction portant le même nom que la classe JavaBean) sans argument.

[G3] Les EJB (*Enterprise JavaBeans*):
Il s'agit de composants essentiels de la plate-forme J2EE de Sun. Les EJB (qui sont en fait des Java Beans en plus évolués) sont un modèle de composants distribués portables facilitant le développement de solutions métier, et ce notamment en gérant les aspects les plus complexes de l'informatique d'entreprise que sont la gestion des transactions, la sécurité, la persistance et la concurrence.

[G4] Frameworks
La traduction littérale en français est : cadre d'application. En informatique il s'agit d'une bibliothèque de classes fournissant une ossature générale pour le développement d'une application dans un domaine particulier. Les frameworks facilitent le travail du développement en fournissant un squelette d'application qu'il suffit de remplir pour l'adapter à ses besoins.

[G5] Parseur
C'est un analyseur syntaxique destiné à récupérer certaines informations dans un document. Cet outil distinguera les informations en fonction de leur contenu et de leur situation dans le document. Si c'est un document XML par exemple : balise de début, balise de fin, etc. Plus généralement, un parseur peut être assimilé à un outil d'analyse syntaxique. C'est d'ailleurs le sens premier du terme anglais parser.

[G6] CSV
CSV signifie "Coma Separated Value". Dans un fichier au format CSV les champs sont séparés par une virgule et les lignes séparées par un retour chariot. Ce format est directement importable dans Excel qui suivant le séparateur utilisé pourra présenter chacun des champs par colonne. On peut également facilement extraire les informations d'un tel format grâce à des parseur.

[G7] AGI
AGI (Arabidopsis Genome Initiative) est le terme utilisé pour désigner le consortium en charge du séquençage d'Arabidopsis (terminé en 2000). Ce consortium a décidé d'utiliser une nomenclature standard pour identifier les gènes d'Arabidopsis thaliana. C'est la nomenclature nommée AGI utilisée dans la banque de données de référence sur Arabidopsis thaliana : TAIR.
Un identifiant (ou numéro d'accession) AGI par exemple At4g22690 est composée de 9 caractères :
- At (Arabidopsis thaliana)
- le numéro du chromosome : 1,2,3,4,5 ou M pour le chromosome mitochondrial ou C pour le chromosome chloroplastique
- g (gène) puis 5 chiffres : 22690

[G8] SSL
SSL (**Secure Sockets Layers**, que l'on pourrait traduire par couche de sockets sécurisée) est un procédé de sécurisation des transactions effectuées via Internet, mis au point par *Netscape,* en collaboration avec Mastercard, Bank of America, MCI et Silicon Graphics. Il repose sur un procédé de cryptographie par clef publique afin de garantir la sécurité de la transmission de données sur Internet. Le système SSL est indépendant du protocole utilisé, ce qui signifie qu'il peut aussi bien sécuriser des transactions faites sur le Web par le protocole http que des liaisons via le protocole FTP ou Telnet protocole FTP ou Telnet.

[G9] Expression régulière
Une expression régulière est un modèle de texte constitué de caractères ordinaires (par exemple les lettres de a à z) et de caractères spéciaux, appelés *métacaractères*. Le modèle décrit une ou plusieurs chaînes à mettre en correspondance lors d'une recherche effectuée sur un texte. Le langage Perl utilise beaucoup les expressions régulières.

[G10] Eucaryote
Organisme vivant possédant un noyau isolé du cytoplasme par une membrane et qui contient de l'ADN, par opposition aux procaryotes (bactéries, cyanophycées) qui n'ont pas de noyau.

[G11] IMAC
IMAC signifie chromatographie de pseudo affinité sur ions métalliques immobilisés. Ce procédé expérimental consiste à enrichir les phosphopeptides avant leur analyse par spectrométrie de masse.Les ions métalliques les plus souvent utilisés sont le Fe 3+ et le Gallium. En effet, les phosphopeptides donnent un signal faible en spectrométrie de masse par le fait qu'ils s'ionisent plus difficilement que les autres peptides
Source : http://sfeap.free.fr/lettres/lettre31.pdf

[G12] Chromatographie en phase liquide
La chromatographie en phase liquide est une technique de séparation des molécules en chimie organique et en biochimie. Comme toutes les techniques chromatographiques, elle est basée sur une différence de vitesse de migration des molécules. Les molécules sont mises en solution dans un solvant, et ce solvant est « poussé » dans un tube fin appelé « capilaire ». Certaines molécules présentes des affinités avec le matériau composant la paroi du tube (notamment, les molécules polaires se fixent sur cette paroi) et avancent plus lentement que les molécules ne présentant pas d'affinité. La taille et la masse des molécules influent aussi sur leur vitesse de déplacement. Ces phénomènes provoquent un « étalement du peloton » (pour utiliser une métaphore cycliste), qui permet de séparer les molécules. La chromatographie en phase liquide se caractérise comme une technique à la fois qualitative de séparation, mais aussi quantitative car elle permet la récupération, selon leur temps de rétention dans la colonne, des analytes individualisés. La chromatographie en phase liquide à haute pression (CLHP, ou HPLC pour high pressure liquid chromatography) a permis la miniaturisation des colonnes, l'utilisation de fortes pressions augmentant les débits et l'efficacité de résolution de la colonne par diminution des quantités de produits à analyser.
Source : http://fr.wikipedia.org/wiki/Chromatographie_en_phase_liquide

[G13] Sérialisation
Java, permet de sauvegarder l'état d'un objet à un instant donné dans un flux d'octets. On dirigera généralement ce flux dans un fichier pour effectuer une sauvegarde. Pour créer une classe sérialisable il suffit simplement que la classe implémente l'interface java.io.Serializable".

[G14] Motif
Un motif est un élément structural que l'on retrouve dans tous les membres d'une famille de protéines. Il contient des acides aminés essentiels à une fonction conservée. Ces acides aminés ne sont pas nécessairement consécutifs (ils peuvent même être très éloignés dans la séquence), mais on s'attend à les trouver assez proches dans la structure 3D, car ils participent à la même fonction (par exemple formation d'un site actif).

[G15] L'électrophorèse monodimensionnelle (ou Gel 1D)
L'électrophorèse monodimensionnelle permet de séparer les protéines en 1 dimension selon 1 seule propriété physico-chimique qui est leur moléculaire (contrairement à l'électrophorèse bidimensionnelle décrite **annexe 3 étape I** ou la séparation des protéines se fait selon deux dimensions à l'aide de deux propriétés physico-chimiques différentes).

[G16] Voie métabolique
Suite de réactions enzymatiques ayant pour but de satisfaire une fonction biologique.

LISTE DES DOCUMENTS

LISTE DES TABLEAUX

LISTE DES DIAGRAMMES

RESUME

Génoplante est un programme fédérateur de génomique végétale qui associe au niveau national à la fois la recherche publique (INRA, CIRAD, IRD, CNRS) et les principales sociétés privées impliquées dans l'amélioration et la protection des cultures (Biogemma, Bayer Cropscience, Bioplante).

Tous ces organismes, au travers de leurs différents laboratoires génèrent en masse des données complexes et hétérogènes provenant de diverses disciplines biologiques que sont la génomique (qui traite de la structure des gènes et de leur position sur l'ADN), la transcriptomique (qui étudie leur niveau d'expression) et la protéomique (qui décrit les protéines découlant de ces mêmes gènes).

La nécessité de gérer, analyser et corréler ces données entre elles, a conduit le consortium Génoplante à lancer une initiative d'envergure dont l'objectif est la construction du système d'information intégré GpiIS *(Genoplante-info Information system)*.

Le travail présenté dans ce mémoire s'inscrit dans le cadre du développement d'un des modules du système d'information GpiIS. Ce module appelé GnpProt *(GeNoPlante Proteomic module)* a pour rôle de prendre en charge la gestion et l'analyse de données protéomiques ainsi que leur intégration avec les autres données biologiques contenues dans GpiIS. Cet outil présente également l'avantage de pouvoir fonctionner de manière autonome afin de pouvoir être installé localement au sein des différentes unités de recherche intéressées. Dans ce contexte le nom de ProteomIs *(Proteome Information system)* est donné à l'application.

Ce mémoire décrit l'ensemble de mon travail et la démarche qui m'a permis d'aboutir, avec l'aide de mes partenaires, à la version fonctionnelle de l'application ProteomIs/GnpProt aujourd'hui déployée sur le serveur de Génoplante-Info et installée dans différentes unités de recherche.

Mots clés :

Génoplante-Info, GpiIS, biologie moléculaire, bioinformatique, protéomique , ProteomIs/GnpProt, base de donnée intégrée, clustering, comparaison de séquences, motif, phosphorylation.

Keywords :

Génoplante-Info, GpiIS, molecular biology, bioinformatics, proteomics, ProteomIs/GnpProt, integrated database, clustering, comparison of sequences, motif, phosphorylation.