

MÉMOIRE
PRÉSENTÉ EN VUE D'OBTENIR
LE MASTÈRE SPÉCIALISÉ EN BIO-INFORMATIQUE
CONSERVATOIRE NATIONAL DES ARTS ET MÉTIERS



DARIC VLADIMIR

GESTIONNAIRE GRAPHIQUE DE PROCESSUS
BIOINFORMATIQUES D'ANALYSE ET DE MANIPULATION DE
SÉQUENCES ET DE STRUCTURES DE PROTÉINES ET SON
APPLICATION POUR LA DÉCOUVERTE DES RELATIONS
STRUCTURE/FONCTION

DIRECTEUR DE STAGE : PEDRO ALZARI



UNITÉ DE BIOCHIMIE STRUCTURALE

TUTEURS : THIERRY ROSE, UNITÉ D'IMMUNOGÉNÉTIQUE CELLULAIRE,
INSTITUT PASTEUR
RACHID GHERBI, LIMSI-CNRS, UNIVERSITÉ PARIS-SUD

INFORMATIONS PRATIQUES :

Adresse postale du laboratoire:

Pedro Alzari

Unité de Biochimie Structurale

Institut Pasteur

25, rue du Docteur Roux

75724 Paris cedex 15, France

Tél. +33 (0)1 45 68 86 07

Fax. +33 (0)1 45 68 86 04

Adresses électroniques :

Pedro Alzari : alzari@pasteur.fr

Thierry Rose : rose@pasteur.fr

Gherbi Rachid : gherbi@limsi.fr

Daric Vladimir : daric@pasteur.fr / daric@iie.cnam.fr

Résumé :

Le module PASTEUR de PtolemyII que nous avons développé est une véritable « paillasse électronique ». Nous y avons introduit un certain nombre d'outils pour l'exploration et l'analyse des propriétés des protéines ainsi qu'un ensemble d'outils complets pour la modélisation comparative.

Ces outils peuvent aider à la prédiction de la fonction des protéines inconnues ou peuvent permettre l'exploration des mécanismes d'interactions des protéines.

Appliquée avec succès pour explorer le mécanisme d'action de l'IL-2, cette approche sera généralisée pour d'autres interleukines.

Remerciements :

Je tiens particulièrement à remercier **Thierry Rose** et **Pedro Alzari** pour m'avoir accueilli dans leurs équipes de recherche, pour leur disponibilité et pour leur enthousiasme à transmettre leur savoir.

Je souhaite également remercier **Michael Nilges** de l'Unité de Bio-Informatique Structurale pour m'avoir permis d'assister au cours « EMBO course on biomolecular simulation, July 2004 », **Warren DeLano**, pour son aide avec le logiciel PyMOL, **Marc Baudoin** du Groupe Systèmes et Réseau et **Huynh Quang Tru** de l'Unité de Bio-Informatique Structurale pour leur aide technique.

Merci à Perrine et Franck pour ces six mois de collaboration, à Floyd pour les conseils en administration système, à Martin et Cédric.

Et un merci massif à toute l'équipe de Biochimie structurale, de la Plate-Forme 6 en particulier ,dans le désordre: Anne-Marie, Gwenaelle, Marco, Alejandro, Pablo, Jocelyne, Marcelo, Frederic, Andrea, Fabrive, Nathalie, Ahmed.

Merci aussi à Virginie, Stephanie et Simon de l'IGC.

" Rien n'a de sens en biologie si ce n'est à la lumière de l'évolution."

Théodosius Dobzhansky

Gestionnaire graphique de processus bioinformatiques d'analyse et de manipulation de séquences et de structures de protéines et son application pour la découverte des relations structure/fonction

Table des matières

I. Introduction.....	7
La diffraction des rayons X.....	7
La structure, oui mais pourquoi faire ?.....	7
De la structure vers la fonction.....	7
Notre stratégie.....	8
II. Le gestionnaire graphique de workflows.....	8
III. Les outils d'analyse des structures.....	12
Diviser pour régner	12
Voir les molécules en 3D.....	12
PyMOL.....	13
PIG-Maker : Protein Image Gallery Maker.....	14
COSA – Clustal Output Structural Analysis.....	15
KISS & Chisel.....	16
APBS – Adaptive Poisson-Boltzmann Solver (Software for evaluating the electrostatic properties of nanoscale biomolecular systems).....	16
IV. Modélisation.....	17
Modeller – Modélisation comparative.....	17
Évaluation et tri des modèles.....	18
V. La mise en oeuvre.....	22
Le parc informatique.....	22
Partage de données.....	22
Gestion du projet.....	22
Résultats de l'expérience.....	24
Traitement de grandes quantités de données.....	24
Perspectives.....	25
VI. Application à la reconnaissance de l'IL-2.....	26
Introduction : L'IL-2 et ses récepteurs.....	26
Importance médicale de l'IL-2, et du développement d'agonistes et antagonistes de l'IL-2R.....	27
Approche bioinformatique.....	30
VII. Conclusions.....	43

Questionnaire graphique de processus bioinformatiques d'analyse et de manipulation de séquences et de structures de protéines et son application pour la découverte des relations structure/fonction

I. Introduction

La diffraction des rayons X

L'Unité de Biochimie Structurale, au sein de laquelle j'ai effectué mon stage, est un laboratoire de cristallographie.

La radio cristallographie, ou l'analyse de la diffraction des rayons X, permet de déterminer la structure des molécules cristallisées.

La technique consiste à soumettre un monocristal à un faisceau monochromatique de rayons X. La diffraction du rayonnement par les atomes, observée pour différentes rotations du cristal, permet de déduire leur disposition dans l'espace et de reconstruire ainsi, la structure moléculaire.

La structure, oui mais pourquoi faire ?

Les récentes avancées des techniques du séquençage, permettent de disposer des génomes complets d'un nombre croissant d'organismes. Il n'en est pas de même pour les données de structure. Celles-ci sont, en effet, encore difficiles à obtenir, elles sont donc moins abondantes¹. Néanmoins plusieurs projets de génomique structurale nationaux et internationaux ont pour but d'accélérer les transferts de technologies, de rationaliser et d'automatiser les méthodes.

L'information sur la structure est extrêmement utile lorsqu'on s'intéresse aux mécanismes moléculaires ou lorsqu'on essaie de comprendre les interactions entre les protéines. Elle est primordiale lorsqu'on essaie de spéculer sur la fonction d'une protéine inconnue. Mon travail dans le laboratoire était de mettre en oeuvre des méthodes pouvant aider à résoudre ce problème biologique dans le contexte d'un projet de génomique structurale appliqué à l'agent de la tuberculose, le *Mycobacterium Tuberculosis*.

De la structure vers la fonction

Une protéine est un enchaînement linéaire d'acides aminés. La synthèse des protéines se fait par un multicomplexe protéines/acide nucléique, situé dans le cytoplasme de la cellule. Pendant, ou peu après sa synthèse, un processus complexe de repliement aboutit à la conformation finale de la protéine. Quelques fois, ce processus est assisté par des protéines

1 Swiss-Prot Release 44.5, 13-Sep-2004: 158316 entries & Protein Data Bank : 27204 entries

chaperonnes. A l'issue de ce processus les zones initialement très éloignées sur la chaîne d'acides aminés peuvent se retrouver proches une fois la protéine repliée. Certaines protéines sont modifiées à ce stade (glycosylation, acylation, phosphorylation) et peuvent être coupées en fonction de leur localisation subcellulaire.

La chaîne protéique peut se replier en acquérant une organisation périodique à courte distance pour former une hélice ou s'associer à plusieurs segments de chaîne pour s'unir en un feuillet plissé : ces motifs périodiques sont des structures secondaires.

L'organisation dans l'espace de ces motifs de structure secondaire reliés par des boucles forme les structures tertiaires. Celles-ci sont des charpentes de la protéine; elles disposent en surface les acides aminés qui forment la « texture » chimique de la molécule et encodent l'interaction avec les autres molécules et définissent leur fonction.

Le gène encode la protéine, qui encode sa structure, qui encode la texture, qui définit la fonction. Chaque étape est dégénérée : un grand nombre de gènes peuvent encoder la même protéine ... un grand nombre de textures peuvent encoder une fonction. Chaque palier est un niveau de régulation.

Nous avons développé un certain nombre d'outils ou de procédures permettant l'analyse des structures. Nous avons voulu utiliser au maximum des outils existants et nous nous sommes efforcés d'utiliser, lorsque cela a été possible, des outils sous la licence GPL² ou du moins, ceux dont la licence est gratuite pour une utilisation en milieu académique.

Notre stratégie

Mon travail s'inscrit dans un projet plus large initié par Thierry Rose de l'Unité d'Immunogénétique Cellulaire à l'Institut Pasteur. Ce projet vise à mettre en place une plate-forme de gestion de workflows de logiciels bioinformatiques. Ce projet est mené par trois élèves du mastère : Perrine Barjou, Franck Valentin et moi-même. Cette plate-forme logicielle couvre un champ très vaste de la bioinformatique de manière à permettre de faire interagir les programmes d'origines et d'expertises diverses, manipulant des séquences, des arbres, des structures et des réseaux. C'est justement cette inter-opérabilité qui devait conférer à la plate-forme tout son intérêt. Franck Valentin a été en charge du développement du gestionnaire de workflows et de l'intégration des logiciels de traitement de la séquence. Perrine Barjou s'est occupée de la partie interaction protéine/protéine. Pour ma part, j'ai été en charge d'un prototype de gestionnaire de workflows sur une base de primitives JGraph puis de l'intégration des outils de visualisation, de prédiction et d'analyse de structures.

Dans la suite du document vous trouverez la description des outils que nous avons mis en place pour la partie « structure ». Je vais dans un premier temps décrire, succinctement le gestionnaire de workflows que nous avons sélectionné et décrire mon apport dans cette tâche. Par la suite, je détaillerai la stratégie que nous avons mis en oeuvre pour l'analyse des structures protéiques.

II. Le gestionnaire graphique de workflows

Voici un bref rappel du cahier des charges que nous avons défini dans la première phase du projet. Celui-ci avait déjà été décrit dans le rapport préliminaire rendu au mois d'avril.

Le package de programmes qui constituera le gestionnaire de pipelines de processus

2 GPL – GNU General Public License (<http://www.gnu.org/copyleft/gpl.html>)

bioinformatiques, devra satisfaire les contraintes suivantes:

Portabilité : Exécution indépendante de la nature du matériel, du système d'exploitation et du navigateur	<input checked="" type="checkbox"/>
Les processus créés par le gestionnaire devront lui survivre lors de sa fermeture (fonctionnement en arriere plan).	<input type="checkbox"/>
La réouverture d'un pipeline en cours d'exécution (identifié par userID/sessionID/pipeline ID) devra permettre de synchroniser les indicateurs d'états, les fichiers d'entrées et sorties avec les processus (process ID) appelés par le pipeline	<input type="checkbox"/>
Le pipeline doit pouvoir être stoppé à partir d'un gestionnaire actif et synchronisé, pas nécessairement depuis la même console qui a été utilisée pour le lancer.	<input type="checkbox"/>
Le gestionnaire doit permettre d'exécuter, de suivre l'exécution et d'accéder aux résultats de plusieurs pipelines indépendants	<input type="checkbox"/>
Un pipeline doit pouvoir autoriser plusieurs points d'entrées (départ) et de sorties (arrêt), plusieurs processus concurrents, les bifurcations logiques, les boucles conditionnelles	<input checked="" type="checkbox"/>
Une exportation des pipelines sous forme de scripts unix/linux exécutables ou de documents XML.	<input checked="" type="checkbox"/>
Autoriser une implantation aisée, voire assistée, d'un nouveau programme si possible via une description XML par exemple.	<input checked="" type="checkbox"/>

En entrée l'utilisateur fournira ou sélectionnera le choix simple ou multiple parmi:

Une séquence de protéine, son identificateur ou une liste de séquences de protéines.	<input checked="" type="checkbox"/>
Une structure de protéine, son identificateur ou une liste de structures de protéines.	<input checked="" type="checkbox"/>
Une ou des listes de paires de séquences de protéines formant un réseau continu ou disjoint.	<input checked="" type="checkbox"/>
Un arbre phylogénétique ou une liste d'arbres.	<input checked="" type="checkbox"/>
Le choix d'opérateurs avec leurs intermédiaires et wrappers s'ils ne sont pas ajoutés automatiquement, boucles et bifurcations logiques.	<input checked="" type="checkbox"/>
Le choix du niveau d'exécution : virtuel, test, script, exécution pas à pas, exécution globale.	<input type="checkbox"/>
Le choix des lieux et de l'instant d'exécution.	<input type="checkbox"/>

En sortie l'utilisateur obtiendra soit:

L'état d'exécution de chaque étape	<input checked="" type="checkbox"/>
Le résultat des exécutions de chaque étape dans des fenêtres indexées à onglets appelées par double-clics	<input checked="" type="checkbox"/>

J'ai été chargé de programmer plusieurs maquettes pour envisager des solutions concrètes et pour évaluer la quantité du travail à effectuer au cas où nous décidions de le développer nous-mêmes depuis des primitives de très bas niveau.

Nous avons choisi de réaliser la maquette en Java pour satisfaire la condition de portabilité mais aussi parce que ce langage se prête particulièrement au développement des interfaces graphiques.

Nous avons opté pour une architecture « client/serveur ».

Comme cela est montré sur la figure 1 le logiciel se compose de deux parties. Une partie « client » qui n'est en réalité qu'un éditeur de diagrammes capable de les exporter sous format XML et de passer ce fichier au serveur pour traitement.

L'idée initiale était d'utiliser JGraphPAD, un éditeur de diagrammes développé avec la librairie Jgraph³ (cf fig 2) pour le transformer en « workflow designer ». Les workflows, simples diagrammes, décrits dans des fichiers XML, seraient alors transmis au serveur. JGraphPAD étant très bien conçu, le code étant clair et bien commenté, le « détournement » envisagé ne semblait pas être une tâche trop ardue.

Pour l'application « coté serveur », nous avons envisagé de parser les fichiers XML pour en fabriquer des scripts. L'exécution de tâches aurait pu être gérée par un logiciel de type X-Flow⁴ qui permet d'encapsuler toute la communication entre le client, le serveur et les services dans des objets (Java) ou un échange en format XML (SOAP⁵). La figure 3 montre un schéma d'exemple de fonctionnement de X-flow.

L'utilisation d'un logiciel de ce type nous aurait permis d'économiser le développement de la couche bas niveau de communication entre le client et le serveur et de ne pas avoir à nous occuper de la distribution des tâches.

Il restait toutefois à développer le "parser" et la gestion des sessions utilisateurs.

3 <http://www.jgraph.com/>
 4 <http://xflow.sourceforge.net/>
 5 <http://www.w3.org/TR/soap/>

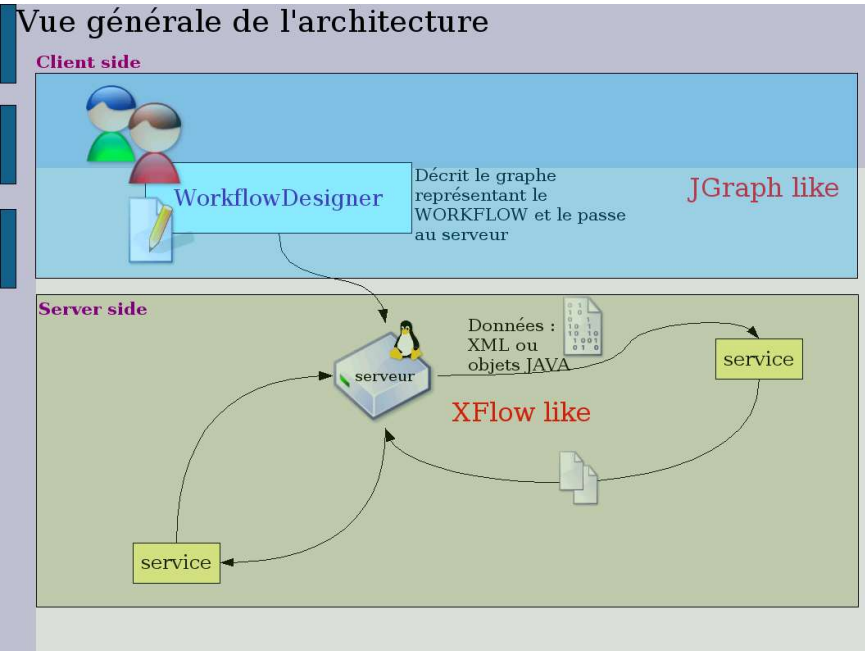


Fig 1: Un schéma de l'architecture logicielle imaginée pour répondre au cahier des charges

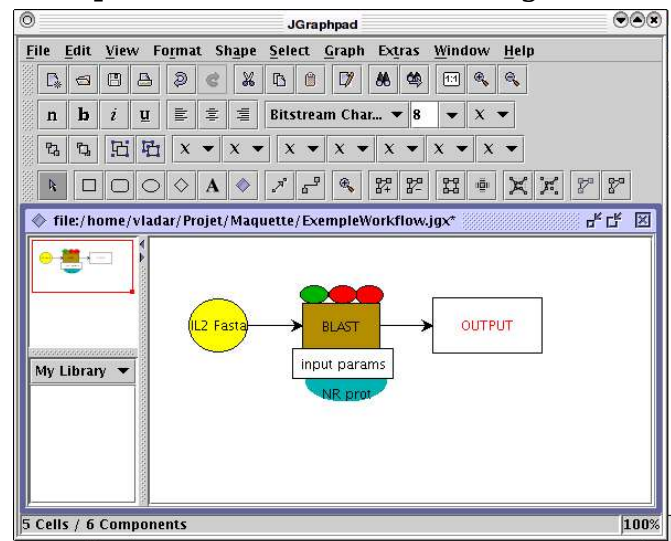
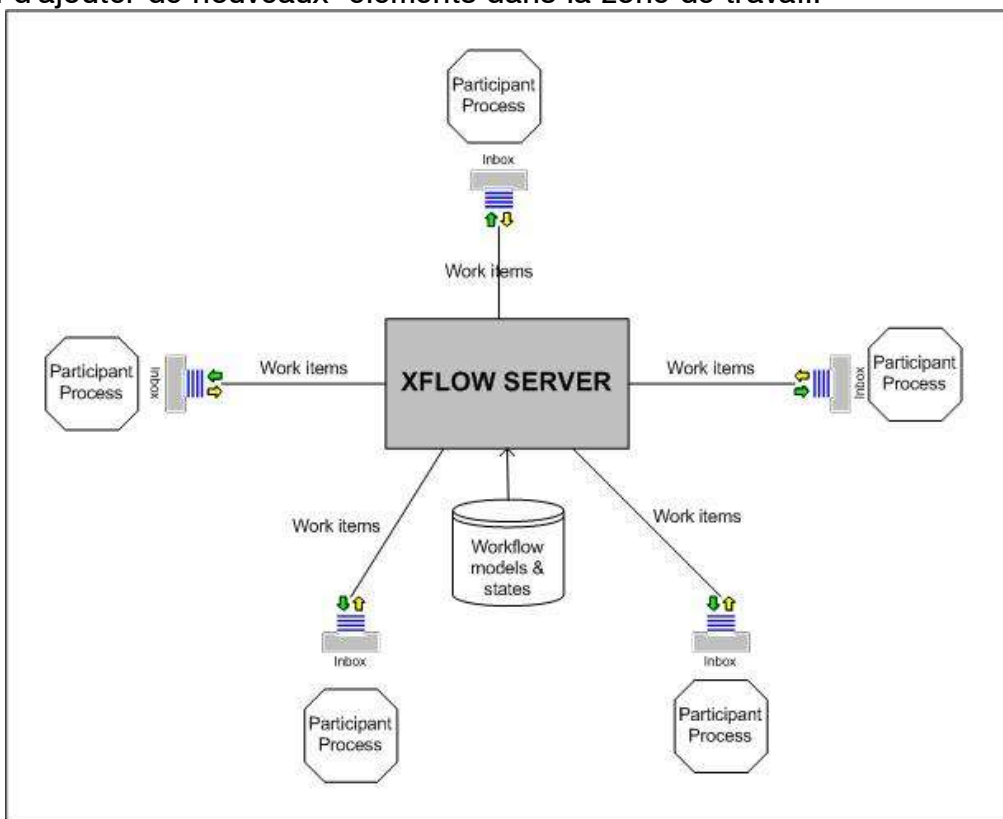


Fig 2: Capture d'écran de JGraphPAD montrant une représentation d'un pipeline simple avec les symbolisations graphiques que nous utilisons à ce stade du projet.

La maquette, dont la capture d'écran est présentée dans la figure 4, permettait de déclencher l'exécution locale d'un programme indépendant (ici Blast). Les deux boutons ont pour l'action d'ajouter de nouveaux éléments dans la zone de travail.



XFLOW Process Management System

Fig 3: Schéma montrant le fonctionnement de X-flow.

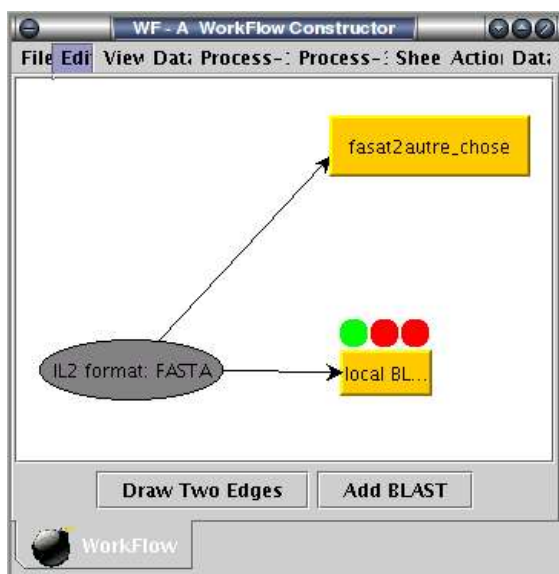


Fig 4: Capture d'écran de la maquette de test. Cette maquette exécute un BLAST représenté par une « brique », la requête, séquence d'IL2 sous format fasta est représentée sous forme d'ellipse. L'ellipse ne faisant pas partie des primitives graphiques de JGraph, pour la dessiner il a fallu modifier une classe de JGraph. Celle-ci utilise les composants Java Swing, il est très simple de remplacer une primitive par une autre. La brique BLAST est composée de plusieurs primitives graphiques simples.

Lors d'une réunion de travail de tous les intervenants du projet, nous avons eu à choisir entre cette solution et plusieurs autres logiciels de gestion de workflows déjà existants.

La solution maquettée avait pour le principal avantage de répondre très précisément au cahier de charges. Le travail de développement nécessaire a été jugé trop audacieux pour le temps et les ressources humaines dont nous disposions.

De plus, une des solutions existantes s'est avérée être très complète et même si elle ne répondait qu'en partie au cahier des charges initial, elle a été choisie. Nous avons, donc, décidé de développer le module *Pasteur* du workflow manager développé très activement à l'Université de Berkeley, PtolemyII⁶ (PTII dans la suite du document) et d'utiliser son interface graphique Vergil.

PTII a déjà été choisi pour le développement de workflows dans plusieurs disciplines, relevant plutôt de la physique ou des mathématiques et très récemment a séduit des groupes de géographes et quelques bioinformaticiens à titre expérimental (Kepler⁷, Spa⁸).

III. Les outils d'analyse des structures

Nous verrons plus loin dans le document la manière dont les outils d'analyse de structures de protéines ont été intégrés dans le module Pasteur de PTII.

Nous nous intéresserons, tout d'abord aux outils que nous avons mis en place.

Diviser pour régner

Tous les acides aminés de la protéine n'ont pas le même rôle. Certains sont enfouis au coeur de la structure et sont complètement inaccessibles aux solvants ou aux ligands. En revanche les acides aminés en surface de la protéine peuvent interagir avec des partenaires potentiels. Nous avons développé un ensemble de scripts permettant de générer automatiquement des documents montrant la protéine sous différents angles et en projetant ses différentes propriétés sur sa surface offrant une véritable cartographie de l'objet étudié. Ces « cartes » avec des vues multiples, sont habituellement fastidieuses à obtenir, non pas que le travail soit difficile, mais parce qu'il est long et très répétitif. Ces documents sont utiles pour analyser, prédire ou comparer des propriétés d'une molécule à l'autre. Ce type d'analyse est une étape clé dans l'étude des relations structure-fonction.

Pour accomplir cette tâche, nous avons tout d'abord dû choisir un logiciel de visualisation de molécules en 3D.

Voir les molécules en 3D

Le principal standard de format de fichier pour définir les structures des macromolécules est le format pdb⁹. Ces fichiers contiennent des coordonnées cartésiennes de tous les atomes qui constituent la molécule et peuvent être visualisés à l'aide de très nombreux logiciels.

Nous avons dans un premier temps envisagé d'utiliser O¹⁰, le logiciel de visualisation des

6 <http://ptolemy.eecs.berkeley.edu/ptolemyII/>

7 <http://kepler.ecoinformatics.org/>

8 <https://www-casc.llnl.gov/sdm/>

9 Description du format pdb : http://www.rcsb.org/pdb/docs/format/pdbguide2.2/guide2.2_frame.html

10 http://xray.bmc.uu.se/~alwyn/Distribution/distrib_frameset.html

molécules de prédilection des cristallographes dont le développement et la distribution ne sont plus assurés. Nous en avons catalogué beaucoup, testé quelques uns et notre choix s'est rapidement tourné vers PyMOL. Ce logiciel est beaucoup plus intuitif, écrit en python, il permet d'écrire avec peu d'efforts des scripts puissants. Il est robuste, fiable et présente toutes les fonctionnalités d'un visualisateur moderne. Les options d'optimisation et de recherche conformationnelle sont en cours de développement ou d'adaptation.

PyMOL

PyMOL est un logiciel de visualisation de molécules en 3D. Voici quelques unes de ses caractéristiques :

- PyMOL est portable. Il peut être exécuté sur un grand nombre des systèmes Unix, Windows et sur Macintosh(OSX). De plus il fait partie des packages officiels Debian, distribution que nous utilisons. Il est particulièrement aisé de l'installer avec cette distribution sur toutes les plate-formes supportées par Debian¹¹.
- Il est écrit en python, et permet d'exécuter des scripts, soit en python, soit en commandes pymol.
- Il est relativement intuitif, l'interaction avec le logiciel pouvant se faire soit à l'aide d'une interface graphique, soit à l'aide de commandes textuelles.
- PyMOL peut être exécuté avec ou sans interface graphique, il est ainsi possible d'effectuer de nombreuses fonctions complexes sur des molécules dans un shell.
- PyMOL utilise la librairie OpenGL, il utilise donc l'accélération graphique. La visualisation « temps réel » est ainsi rendue particulièrement fluide et agréable.
- De nombreux modes de visualisation sont disponibles. Le rendu de la représentation « cartoon » est de qualité comparable à celle de MolSCRIPT et celui des surfaces est aussi bonne que celui obtenu avec Grasp. Construire des scènes complexes avec PyMOL est d'une simplicité quasi enfantine.
- PyMOL permet l'affichage des molécules en mode stéréoscopique, « cross-eye » ou adapté pour une visualisation avec des lunettes 3D.
- PyMOL dispose d'un « ray tracer » incorporé qui permet d'améliorer le rendu 3D des scènes de qualité comparable à Pov-Ray ou Raster 3D, les standards de ce domaine d'application.
- Il est possible d'exporter des images au format png et l'enchaînement des vues pour obtenir des animations est extrêmement simple.
- Il est gratuit et *open source*, toutefois un soutien financier est encouragé. Il est donc possible de rajouter de nouvelles fonctionnalités.

Malheureusement l'équipe de développement étant très petite, la documentation est faite avec du retard. Ce point est en partie comblé par la communauté des utilisateurs, très active et dont la liste de diffusion très réactive. Tous nos problèmes ont ainsi trouvé une réponse rapide. J'ai en plus eu l'occasion de rencontrer Warren DeLano, le développeur principal de PyMOL, lors du cours « EMBO course on biomolecular simulation¹² » qui avait eu lieu à l'Institut Pasteur pendant mon stage et auquel j'ai eu la chance d'assister.

11 <http://www.debian.org>

12 <http://www.pasteur.fr/recherche/unites/Binfs/EMBO2004/>

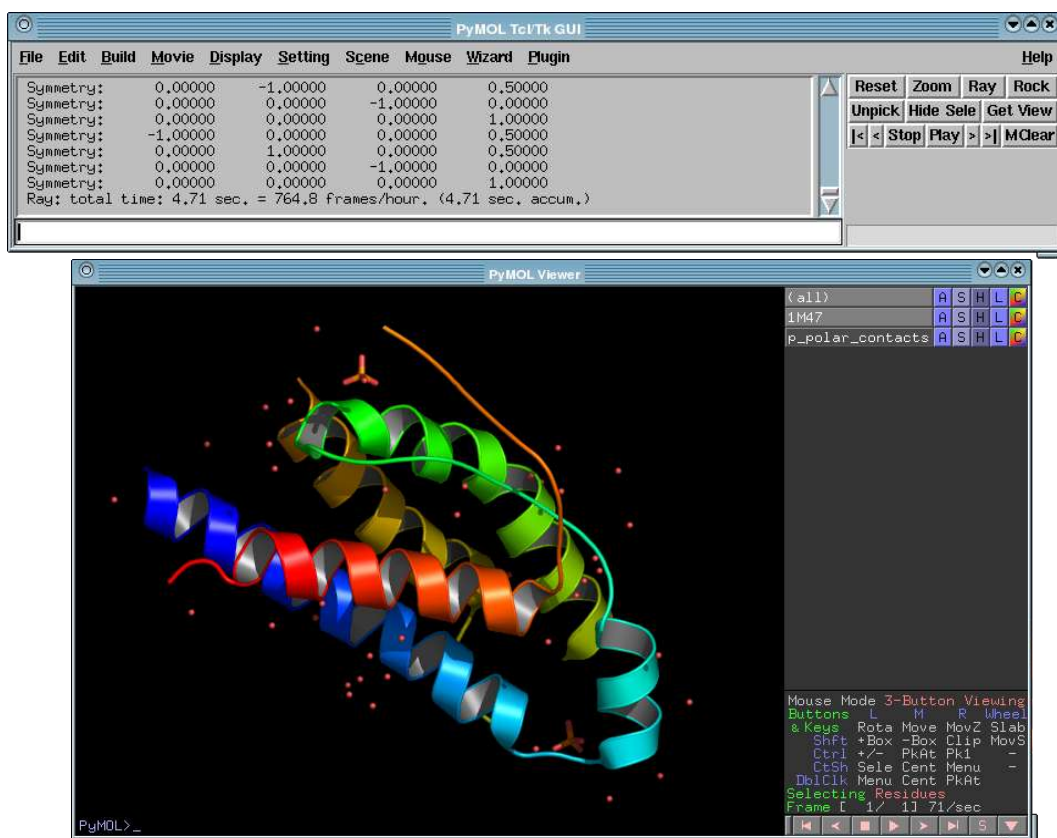


Fig 5. Capture d'écran de l'application PyMol. Montrant en haut la fenêtre de l'interface graphique externe et en bas la fenêtre de visualisation avec l'interface utilisateur interne.

L'image dans la fenêtre de visualisation montre une représentation « cartoon » d'IL2 entourée de quelques molécules d'eau et par deux ions sulfates.

PyMOL est utilisé pour visualiser les molécules mais aussi comme cela a été dit précédemment pour générer des documents avec des vues multiples de protéines.

PIG-Maker : Protein Image Gallery Maker

Le programme PIG-Maker est un script pour PyMOL. Je l'ai écrit en python et il permet de générer des représentations multiples sous des angles différents sous forme de document html.

Ce script utilise PyMOL pour générer des vues différentes. Pour chaque vue deux images png sont créées. Une première, avec la résolution 800x600 pixels servant à visualiser la protéine en vue plein écran ou pour les impressions. Une deuxième en 160x120 de manière à ce que le document montrant toutes les vues ne soit pas trop volumineux dans l'éventualité où il doit être montré sur le web. Le clic sur une des images de la galerie ouvre l'image détaillée montrant la même vue.

Toutes les images sont fabriquées avec ombrages et reflets (ray trace, le point fort de PyMOL), ce qui ralentit quelque peu l'exécution, mais contribue grandement au rendu esthétique du document et lui confère l'impression de volume, souvent nécessaire pour comprendre et interpréter l'image.

Chaque ligne du tableau (voir figure 6) est fabriquée par une fonction, le programme est conçu pour être personnalisable à souhait.

Les deux dernières lignes du document nous servent à montrer la coloration de la protéine en fonction du B-factor.

Le B-factor est une valeur dans la 11^{ème} colonne du fichier pdb. Cette valeur sert initialement pour stocker le facteur d'incertitude de la position de l'atome. Quasiment tous les logiciels de visualisation permettent de colorier la surface en fonction de cette valeur. Il est très courant de détourner l'utilisation des B-factors de son utilisation initiale. C'est de cette manière là que nous avons procédé.

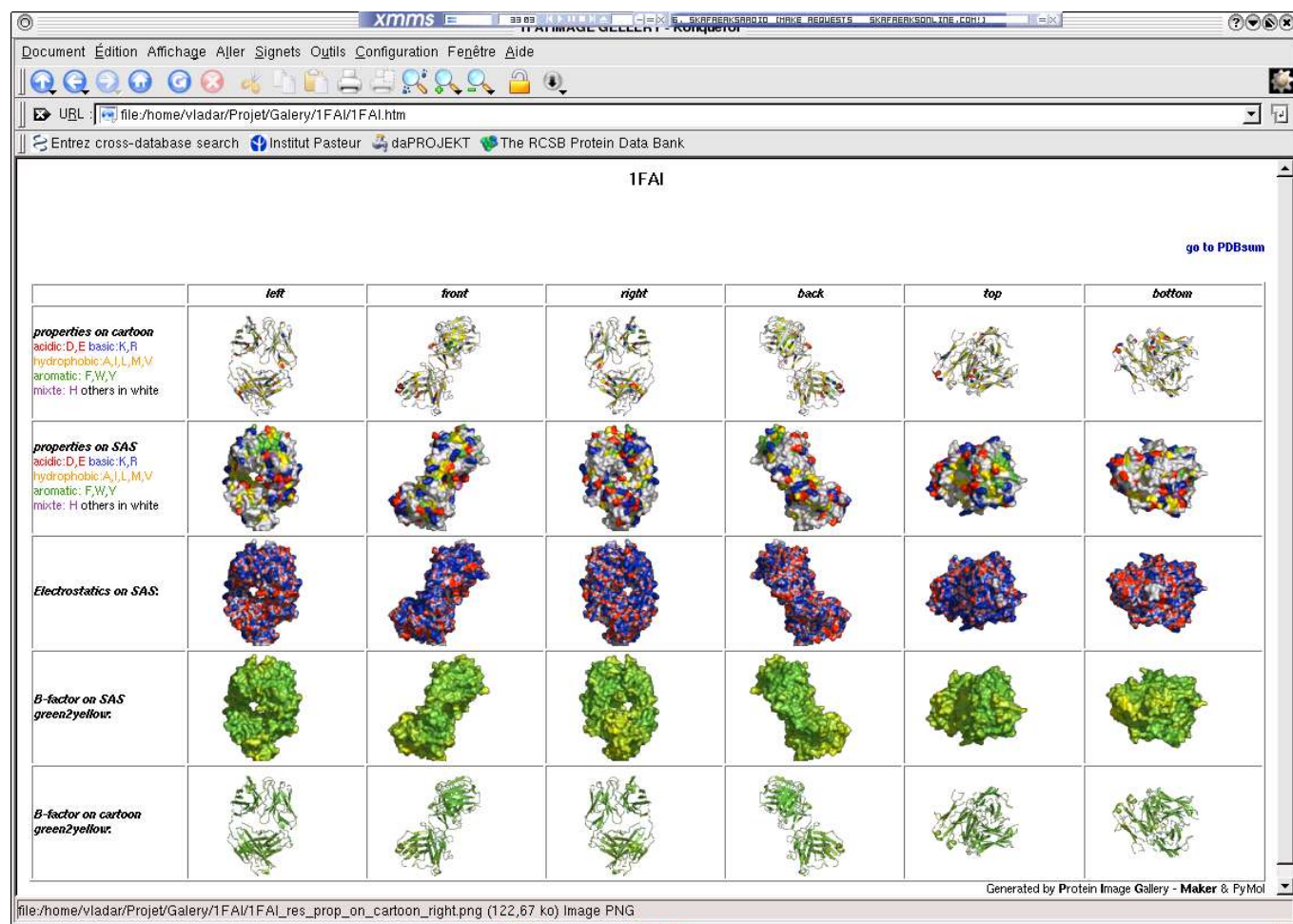


Fig. 6 : Capture d'écran de la page html générée par PIG-Maker.

Lignes 1 et 2 montrent la projection des propriétés chimiques des acides aminés sur une représentation schématique du squelette protéique et sur la surface accessible aux molécules d'eau (SAS – Solvent accessible surface).

Ligne 3 montre la projection du potentiel électrostatique sur la surface, celui-ci est calculé par le logiciel apbs (expliqué plus loin dans le document)

Lignes 4 et 5 montrent la projection du B-factor sur la surface et sur une représentation en « boucles/hélices ».

COSA – Clustal Output Structural Analysis

Nous utilisons le programme COSA (T. Rose, Institut Pasteur) pour mettre en évidence des

régions de la protéine qui ont été plutôt conservées au cours de l'évolution, ou au contraire celles qui sont très variables. La fréquence de l'acide aminé le plus représenté est calculée à partir d'un alignement de séquences multiples au format clustal. Il est possible aussi de figurer la diversité des résidus à chaque position, calculée à partir de la somme $\sum_{x=1}^{20} f_x \ln(f_x)$ ou f_x est la fréquence de l'acide aminé x. Le raisonnement qui est derrière

cette démarche présume que la nature ne conserve rien sans raison; seulement sous la contrainte; le coût de la conservation serait plus élevé que celui du changement. Habituellement les résidus enfouis sont plutôt conservés car ils assurent l'acquisition et le maintien du repliement. D'autres sont également conservés, alors qu'ils sont accessibles aux solvants, ou à toute autre molécule. Fréquemment ces résidus sont impliqués dans la fixation d'un ligand, d'une cible ou alors dans une fonction catalytique.

Le programme COSA donne de simples statistiques sur la conservation à partir d'un fichier d'alignement et les place dans la colonne des B-factors du fichier pdb. La plupart des logiciels de visualisation permettent de colorier la surface de la protéine en fonction de cette valeur.

KISS & Chisel

Lorsque nous avons une interaction entre macro-molécules, bien souvent nous souhaitons identifier les résidus formant l'interface.

Les programmes KISS et Chisel (T. Rose, Institut Pasteur) nous permettent de cartographier toute interface entre molécules décrites au format pdb.

Chisel est un petit programme qui permet d'extraire des sous-chaînes d'un fichier pdb lorsque celui-ci en contient plusieurs. Il est possible de choisir les chaînes que l'on souhaite isoler. Ce programme permet aussi de dissocier les fragments d'une même chaîne ou extraire un ligand de la protéine. A partir d'un fichier pdb, Chisel en produit donc au moins deux.

KISS prend en argument deux fichiers pdb et calcule la distance entre les paires d'atomes les plus proches de chaque fichier. Ces valeurs sont placées dans la colonne des B-factors.

Encore une fois nous utilisons les valeurs du B-facteur pour colorier la surface de la protéine pour y visualiser les zones potentiellement impliquées dans l'interaction. Il est important de noter que la proximité ou le contact entre deux résidus ne garantit en rien leur contribution positive dans l'association des deux molécules. Si ces contacts sont souvent « constructifs », ils peuvent aussi être « destructifs ».

APBS – Adaptive Poisson-Boltzmann Solver (Software for evaluating the electrostatic properties of nanoscale biomolecular systems)

Depuis cet été, avec la version 0.97 de Pymol et APBS, nous avons une alternative économique à DelPhi(Accelrys¹³) et Grasp¹⁴. APBS est développé par Nathan Baker (Washington University, St Louis) et est disponible gratuitement sous la licence GPL.

APBS calcule une solution numérique pour des équations de Poisson-Boltzmann qui permettent de modéliser les interactions électrostatiques entre les molécules dans un solvant

13 <http://www.accelrys.com>

14 <http://trantor.bioc.columbia.edu/grasp/>

(4) et de visualiser la valeur du potentiel électrostatique en tout point de l'espace (cf Fig 7).

APBS nous fournit une matrice 3D qui définit le potentiel électrostatique en chaque point. Pour le calculer APBS requiert un champ de force. Il s'agit d'un fichier de paramètres décrivant les interactions entre atomes liés et non liés. Il existe beaucoup de champs de force. Les plus connus sont ceux de « Amber », et « Charmm ».

Le fichier pdb, dont nous souhaitons étudier les propriétés électrostatiques doit être transformé en fichier pqr. Nous utilisons pour cela un script Pymol, que j'ai écrit, qui rajoute les hydrogènes manquants et vérifie l'état de protonation des chaînes latérales des histidines, des aspartates et des glutamates, lysines et arginines.

Ce fichier est ensuite passé à APBS. Pour cela j'ai écrit un script en python pour générer le fichier de lancement de APBS pour chaque molécule. Ce script fait appel au module psize.py livré avec APBS. Psize calcule un certain nombre d'informations sur la molécule représentée dans le fichier pdb. Nous l'utilisons pour calculer les dimensions de la molécule et adapter la résolution de la matrice 3D.

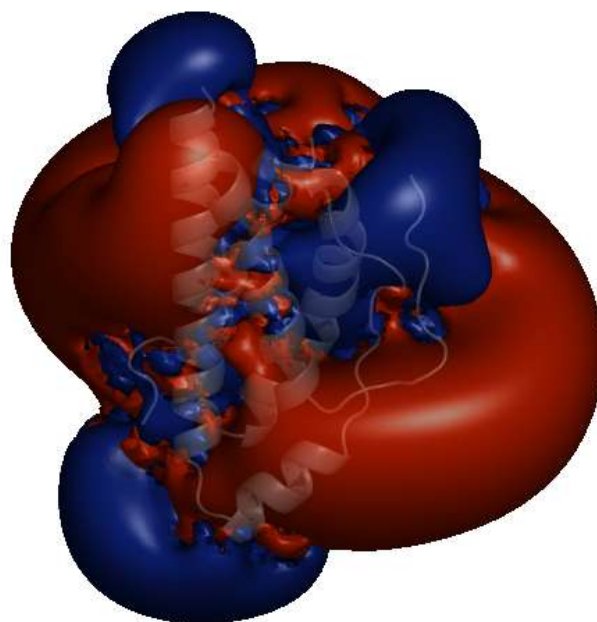


Fig 7 : Représentation des surfaces d'isopotential électrostatique au tour de la molécule d'IL2 (rouge pour les valeurs négatives et bleu pour les valeurs positives)

IV. Modélisation

Modeller – Modélisation comparative

La modélisation comparative permet de prédire la structure de protéines inconnues. Pour cela nous nous basons sur un alignement de la protéine requête, que l'on souhaite replier, avec une ou plusieurs protéines cibles dont la structure est connue (template).

Le processus se passe, donc, en quatre étapes :

- **Choix du template** : Cette étape est primordiale. Pour employer une image, il s'agit ici de choisir le moule. Ce choix doit être pertinent et souvent l'expertise humaine est un avantage. En augmentant le nombre de templates, et s'ils sont compatibles, on arrive à un meilleur résultat. Les résultats sont en général bons lorsque le niveau d'identité entre la requête et la séquence du template est supérieur à 30% et pour une longueur inférieure à 300 résidus. En dessous de 30% et au dessus de 300 résidus la construction du modèle est délicate.
- **Alignement de séquences de la cible et du/des template(s)** : Cette étape est également extrêmement importante et conditionne la qualité du résultat. Dans la réalité on revient souvent sur cette étape pour tenter d'améliorer la qualité du modèle.

- Construction des modèles : Cette étape est entièrement automatique et est effectuée par le logiciel « Modeller »(Sali & Blundel, 1993)¹⁵. Ce logiciel replie la séquence de la protéine requête, par une méthode de dynamique moléculaire, sous les contraintes géométriques du template, et produit une structure 3D ou un modèle moléculaire.
- Plus on augmente le nombre de modèles générés plus on explore l'espace conformationnel et on augmente ainsi les chances théoriques d'obtenir une bonne solution. Il faut garder en tête qu'une protéine est une structure dynamique et qu'il s'agit ici de la figer dans un état particulier pour des raisons pratiques évidentes.
- Évaluation des résultats : Cette étape sera expliquée en détail car de très nombreuses stratégies existent, mais il s'agit de choisir les modèles satisfaisants pour n'en garder qu'un à la fin.

Le logiciel académique Modeller est distribué sans le code source et son utilisation est gratuite pour le milieu académique. Il est également vendu par Accelrys, qui en a acquis la licence de distribution commerciale. La version commerciale, apporte en plus une interface graphique. Dans les travaux décrits dans ce mémoire, c'est la version 6v2 de Modeller qui a été utilisée.

Modeller se présente comme un logiciel s'exécutant en ligne de commande. L'exécution est contrôlée par un script contenant, entre autre, l'emplacement du fichier d'alignement et des fichiers pdb des templates.

Modeller replie bien les structures secondaires sur le modèle mais moins bien les boucles et les extrémités peu contraintes. Les chaînes latérales flexibles doivent être souvent revues. DrawBridge¹⁶ offre une recherche conformationnelle intéressante pour la « finition » des boucles et SCWRL¹⁷ est excellent pour ré attribuer une orientation acceptable des chaînes latérales. La version 7v7 de Modeller a été annoncée comme remédiant à ces difficultés (20 septembre 2004).

Le programme Robetta (David Baker, University of Washington, Seattle) sera aussi intégré pour replier les séquences sur des modèles dans les cas de faible niveau d'identité (<15%) entre les séquences des protéines requêtes et cibles. Ce programme basé sur la reconstruction du modèle par des fragments de peptides extraits de la PDB par un algorithme de Monte-Carlo est très performant mais gourmand en ressources.

Évaluation et tri des modèles

a) Procheck

Ce programme, développé par le groupe de J.Thornton (UCL, Londres) vérifie la stéréochimie de la molécule qu'on lui passe en paramètre sous forme de fichiers pdb. Bien qu'un grand nombre de paramètres soit vérifié par procheck nous n'utilisons pour l'instant que les angles ϕ et ψ qui décrivent le squelette peptidique carbonique.

15 <http://salilab.org/modeller/modeller.html>

16 <http://www.cmpharm.ucsf.edu/cohen/software/>

17 <http://128.220.22.46/Modeling/SCWRL.html>

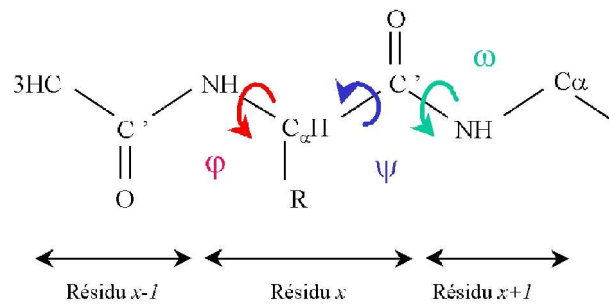


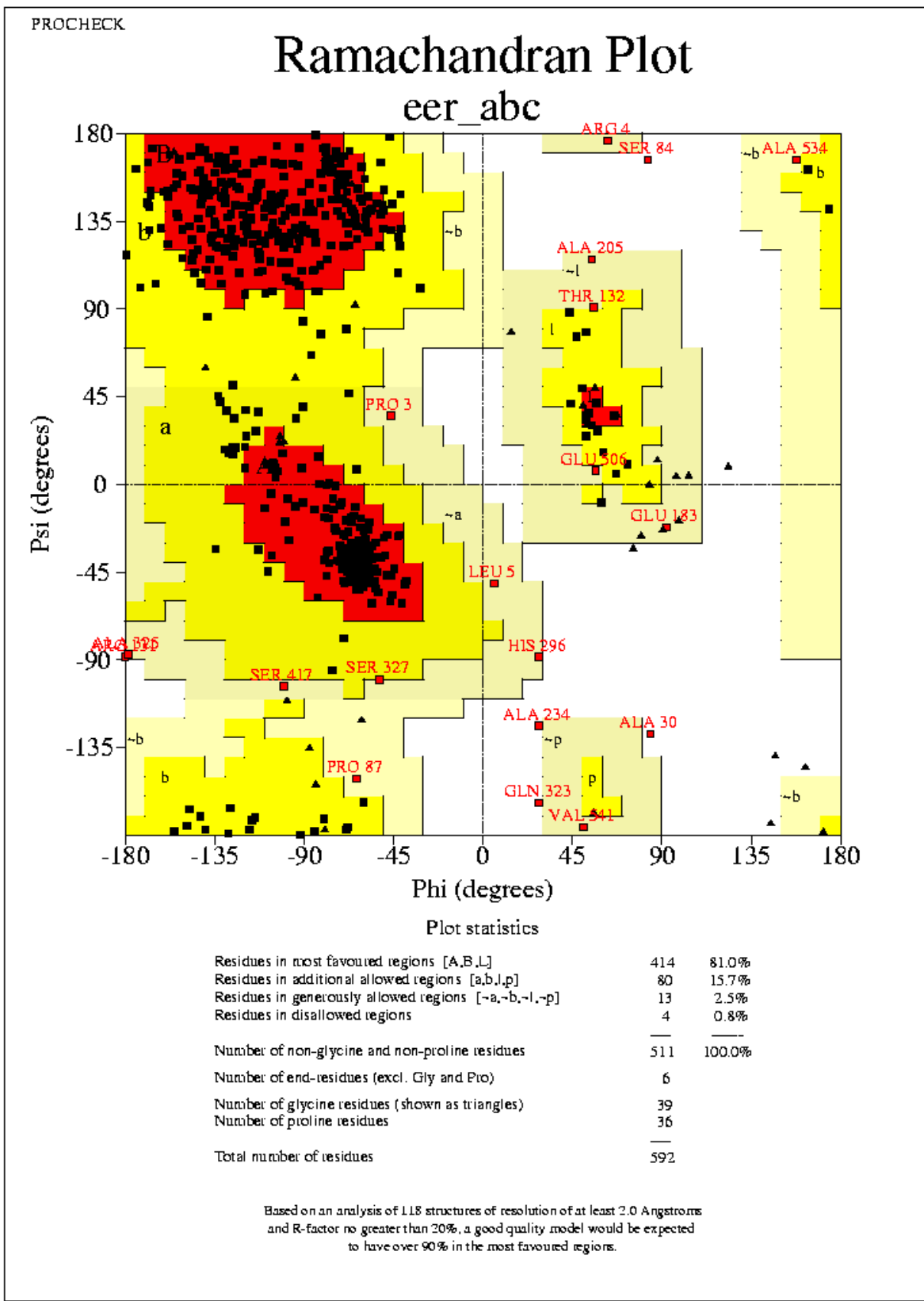
Fig. 8 : Schéma montrant la position des angles ϕ et ψ dans la chaîne polypeptidique. L'angle ω est indiqué également, mais celui-ci est très constant et avoisine les 180°

Les angles ϕ et ψ sont classiquement représentés l'un par rapport à l'autre dans le « diagramme de Ramachandran » (cf. Figure 9); Il fut utilisé pour la première fois en 1968 par G.N. Ramachandran qui se servait à l'époque de modèles énergétiques pour caractériser les zones préférentielles de ces angles de torsions (8). Les zones blanches de la figure 9 sont interdites à cause des percussions des atomes autour des dièdres ϕ et ψ .

Procheck classe tous les angles ϕ et ψ de la molécule dans quatre zones d'acceptation. Ces zones ont été définies par analyse statistique des structures présentes dans la PDB (Protein Data Base, RCSB, Rutgers University).

L'utilisation de Procheck, intégré pour des raisons « historiques », dans la séquence d'évaluation des modèles, est provisoire. Elle sera, à terme, remplacée par Whatcheck¹⁸ et par l'utilisation des profils de déviation stéréochimique et de violations de contraintes créés par Modeller.

¹⁸ <http://www.cmbi.kun.nl/gv/whatcheck/>



eer_abc_01.ps

Fig. 9 : Exemple d'une des sorties graphiques de procheck. Les résultats sont fournis sous forme de plusieurs fichiers PostScript, mais aussi, sous forme de fichiers texte.

b) DSSP (Define Secondary Structure of Proteins), Verify3D

DSSP est un programme, écrit par W. Kabsch et C. Sander. Nous l'utilisons pour définir les structures secondaires des protéines (3).

Ce logiciel, dont la première version avait été écrite en 1983 repose sur une définition géométrique des structures secondaires.

c) Verify3D

La théorie :

Comme cela a été mentionné précédemment, nous disposons de milliers de structures qui ont été définies expérimentalement. Ces structures peuvent être classées par familles de repliements identiques. Il est maintenant avéré que dans la même famille on peut trouver des protéines montrant très peu d'identité entre leurs séquences.

Le groupe de David Eisenberg (UCSF) a développé une méthode qui analyse l'adéquation d'une séquence par rapport à sa structure(2).

Cette méthode comporte trois étapes :

1. La structure tridimensionnelle est transformée en une chaîne unidimensionnelle définissant l'entourage de chaque résidu. Trois paramètres sont calculés dans la structure 3D pour chaque acide aminé :
 - la surface de la chaîne latérale qui est enfouie dans la protéine
 - la surface de la chaîne latérale qui est accessible aux atomes polaires
 - la structure secondaire dans laquelle le résidu est engagé
2. Calcul du profil 3D : ce profil est une matrice calculée à partir des trois paramètres qui décrivent l'environnement de chaque résidu, calculés à l'étape précédente. Ces données sont pondérées en fonction de la position du résidu dans la séquence et en fonction de la nature du résidu. En effet, le logiciel interroge une matrice précalculée à partir des structures connues, qui donne la probabilité pour chacun des 20 acides aminés d'être trouvé dans une classe d'environnement.
3. Comparaison de la séquence avec le profil3D de sa structure. Le score de cet alignement traduit la compatibilité de la séquence avec sa structure 3D.

Verify3D(1,2) produit le profil du score, une valeur moyenne et la valeur attendue déduite de la séquence. Ces valeurs donnent une idée de la qualité de la structure. Comme le profil calculé contient une valeur pour chaque résidu, il est possible d'estimer la qualité de repliement de chaque zone. Il est ainsi possible de former des protéines chimères en combinant les meilleurs fragments provenant de différents modèles.

Verify3D est écrit en Fortran et est disponible gratuitement pour une utilisation académique. Il fait, également partie du module « Homology » de la suite InsightII de Accelrys dans sa version commerciale.

Je n'ai pas réussi à le faire fonctionner sous Linux. Le programme ne compile pas avec g77, ni avec f2c. La compilation se passe sans erreurs avec le compilateur fortran de Intel,

mais les binaires obtenus provoquent un « segmentation fault » dont la cause n'a pas encore été identifiée.

Nous avons temporairement éludé ce problème en utilisant une station DEC alpha sur laquelle nous n'avons pas rencontré de problèmes de compilation.

V. La mise en oeuvre

Le parc informatique

Pour les besoins du projet nos dispositions de quatre PCs sous Linux, d'un Mac G5 sous mac OSX, d'une station Silicon Graphics Octane sous IRIX et d'une station DEC alpha sous Compaq Tru64 UNIX. Deux des quatre PCs avaient la possibilité de démarrer au choix sous Windows ou sous Linux. Le réseau et la DEC alpha sont gérés par le service informatique de l'Institut Pasteur.

Je me suis occupé de l'administration du parc et de l'installation des quatre PCs. J'ai choisi d'utiliser la distribution Debian pour sa facilité d'administration. La majorité des logiciels couramment utilisés en bioinformatique sont disponibles sous forme de packages Debian. Leur installation est par conséquent facilitée.

Partage de données

Un des quatre PCs a été dédié comme « serveur ». Il contenait les données qui devaient être partagées.

Pour la mise en commun de ces données un serveur ftp et des montages NFS ont été mis en place.

Cet ordinateur servait également comme serveur de backup. Le backup était assuré par un script lancé par la crontab tous les jours. Ce script utilise rsync pour éviter les transferts de données qui n'ont pas été modifiées et utilise une partition NFS lorsque cela est possible ou scp lorsque le montage NFS avait été rendu impossible par la configuration des routers¹⁹.

Gestion du projet

Pour gérer le projet, nous avons chacun une page web sur laquelle nous pouvions noter notre progression personnelle au jour le jour – le « notebook » - permettant aux tuteurs de nous suivre, en plus des réunions. Un agenda partagé avait également été mis en place. Pour cela il a fallu installer et configurer Apache²⁰, MySQL²¹ et WebCalendar²².

a) Calcul sur cluster

Au cour du mois d'août nous avons tenté une expérience de calcul sur cluster. Pour cette expérience nous avons utilisé 32 PCs (Dell Optiflex fx265) qui servent habituellement aux élèves des différents cours qui ont lieu à l'IP (Institut Pasteur). Ces ordinateurs sont régulièrement réinstallés selon les exigences des enseignants.

19 A cause de la politique de gestion du réseau de l'IP notre parc informatique était dissout sur quatre sous-réseaux différents.

20 <http://www.apache.org>

21 <http://www.mysql.org>

22 <http://www.k5n.us/webcalendar.php>

Nous avons, dans un premier temps, testé au laboratoire la technologie de clusterisation OpenMosix²³ sur un PC maître pour contrôler deux PC asservis (Dell Optiflex fx270). Pour cela nous avons utilisé la distribution LiveCD – ClusterKnoppix²⁴.

Avec l'aide de Michel Keller (Micro-informatique) et Marc Baudoin (Systèmes et Réseaux), nous avons testé cette même solution sur les 32 PC du Service des Enseignements au bâtiment de Biotechnologie.

L'ordinateur « maître » a été démarré à partir du CD Cluster Knoppix. Les 32 ordinateurs « esclaves » ont été initialisés à travers le réseau (boot on LAN). Cette option n'étant jamais sélectionnée dans le *bios* par défaut, ceci doit être fait manuellement.

La solution que nous avons utilisée permet de ne rien installer sur aucun des disques des PC asservis: ni le système, ni les programmes exécutés (importés en mémoire vive), ni même les résultats (exportés vers le disque du PC maître). Il s'est avéré indispensable que tous les noeuds soient derrière le même switch, pour éviter les latences dues au routage. Les performances de ce type de cluster se dégradent rapidement lorsque le réseau est lent (100Mb/s est le minimum pour obtenir de bonnes performances du cluster). En 7 heures, 32000 modèles d'IL-2 de 32 organismes différents ont été créés en utilisant le cluster. Nous n'avons pas rencontré de problèmes techniques particuliers. Ces tests sont un succès qui révèle la légèreté de la solution utilisée, la facilité de mise en place, l'innocuité pour les machines et l'absence d'opération de maintenance après utilisation puisque après extinction ou redémarrage les ordinateurs reviennent à leurs états initiaux.

OpenMosix se présente sous forme d'un patch à appliquer au kernel et d'un ensemble de logiciels permettant le contrôle du cluster et sa surveillance. Si ces services sont démarrés, il est possible de se connecter au cluster par n'importe lequel des protocoles habituellement

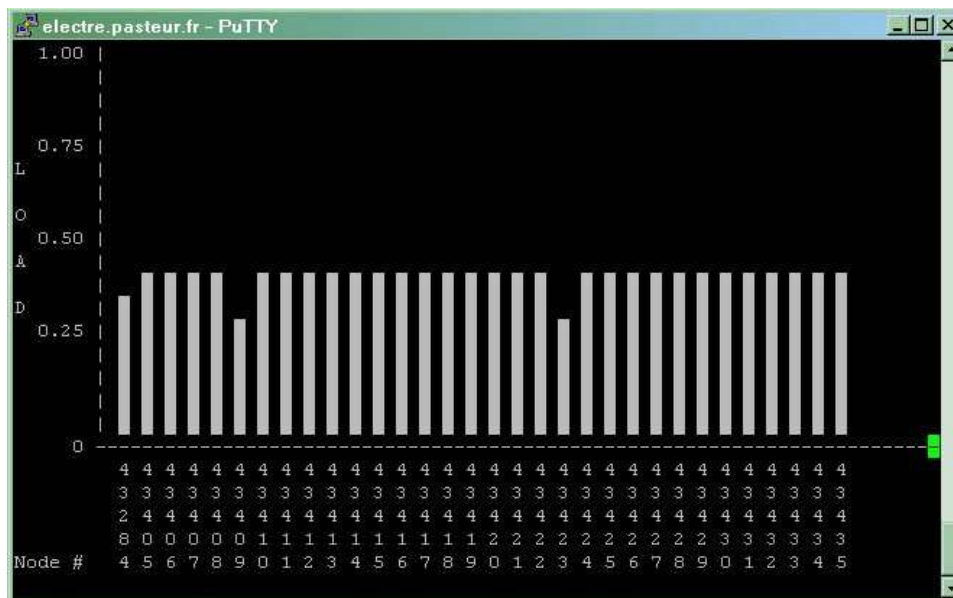


Fig. 10 : Capture d'écran de 'mosmon'. C'est un utilitaire permettant de surveiller la charge de tous les noeuds du cluster. Ici la surveillance se fait à distance, à partir d'une station sous Windows, connecté par ssh au cluster à l'aide du logiciel PuTTY.

23 <http://openmosix.sourceforge.net/>

24 <http://bofh.be/clusterknoppix/>

disponibles sous Linux (ssh, ftp ...). La figure 10 montre une capture d'écran de 'mosmon', un utilitaire de surveillance de l'activité du cluster.

La technologie OpenMosix, à défaut de permettre une véritable parallélisation des calculs, permet une distribution de processus sur les noeuds du cluster en fonction de leur disponibilité et de leur charge respective (load balancing). Le véritable avantage de cette technologie est qu'elle permet l'exécution de **tous** les programmes habituels, comme sur n'importe quelle autre station Linux. Les clusters de type Beowulf²⁵ ne sont capables de paralléliser que des exécutables écrits en utilisant des bibliothèques dédiées; un « portage » des logiciels est alors souvent nécessaire. Enfin, en raison du format même des 32000 exécutions d'un programme identique durant 1min 27sec, la technologie OpenMosix est tout à fait adaptée avec un ratio de temps d'exécution meilleur que 30/32. Une exécution parallélisée sur un véritable cluster Beowulf n'aurait pas significativement amélioré les performances sur la globalité du travail.

Pour lancer les instances de Modeller nous avons testé deux approches. Nous avons à calculer 1000 modèles pour chacune des 36 espèces animales. Une première approche consistait donc à lancer des tâches longues, de manière à ce que les 1000 modèles de chaque organismes ne soient calculés que sur un seul noeud à la fois. Dans la deuxième approche, nous générions n scripts d'initialisation de Modeller, nous lançons n instances de Modeller à la fois, chaque instance, qui s'exécute sur un processeur du noeud, ne produisait alors, que $\frac{1000}{n}$ modèles (n étant le nombre de processeurs dans le cluster). Ces tâches étaient plus courtes à exécuter dans le temps, les deux solutions étant théoriquement aussi performantes l'une que l'autre, c'est cette deuxième qui s'est avérée plus facile à gérer car la fin des tâches était plus facile à prévoir. Elle s'est donc avérée être plus performante dans la pratique, car elle permet d'exploiter au mieux les ressources en réduisant les périodes d'inactivité des noeuds individuels. En effet, OpenMosix, ne disposant pas d'utilitaire de « batch queuing » par défaut, nous étions obligés de lancer les calculs « à la main ».

Résultats de l'expérience

Compte tenu des excellents résultats obtenus, malgré les problèmes que nous avons rencontrés, chose inévitable lorsqu'on doit développer des outils en flux tendu; nous avons généré 125 405 modèles en deux semaines, nous envisageons d'entreprendre la construction des 200 000 complexes cytokines-récepteurs selon les mêmes protocoles et la même architecture.

Nos besoins en calcul sont ponctuels: beaucoup de modèles à construire ou de banques de molécules à cribler, de temps en temps. Un accès à cette grappe de PC de quelques jours par mois ou tous les deux mois serait adapté à nos besoins. La gestion du grand nombre de modèles et de leur analyse bénéficie grandement de la "paillasse électronique" que nous avons développée. Bien sûr, le gestionnaire de workflow PII est tout à fait compatible avec cette architecture distribuée.

Traitement de grandes quantités de données

Les 125405 modèles générés sur cluster représentent environ 34 Go de données qu'il a fallu traiter. Les modèles ont donc été évalués avec procheck, puis avec dssp et verify3D. Les résultats ont été classés puis dix meilleurs modèles de chaque set ont été choisis.

²⁵ Beowulf Project : <http://www.beowulf.org>

Pour cela j'ai écrit plusieurs scripts. Assez rapidement le besoin de pouvoir définir une liste de « jobs » est apparu afin de systématiser leur exécution.

De plus à cause des problèmes de compilation que nous avons eu avec Verify3D, nous avons dû exécuter une partie des traitements sur la station alpha, bien qu'elle soit mise à disposition de tout le personnel de l'IP et de ce fait très sollicitée.

Voici comment nous avons procédé :

- un premier script exécute procheck sur tous les modèles du même set, parse les résultats à l'aide d'un script 'awk' pour en extraire les informations qui nous intéressent et les mettre toutes dans un seul fichier comportant un modèle par ligne.
- pendant ce temps là un deuxième script, exécuté sur la station alpha récupère les données sur le serveur. Pour cela nous utilisons le protocole ssh.à l'issue. Un échange des clés de cryptage permet un transfert de données entièrement automatisé et sécurisé (les mots de passe ne « traînent pas » en « clair » dans les scripts). Encore une fois, issu nous permet de réduire la quantité de données échangées, en ne transférant les données que lorsque cela n'a pas déjà été fait. Les résultats de dssp et de Verify3D, parsés par un script 'awk', sont renvoyés sur le serveur, sous forme de deux fichiers comportant un modèle par ligne, en utilisant la même stratégie.
- un troisième script finalise les traitements en utilisant les résultats de procheck, de dssp, de Verify3D et la valeur d'une fonction d'énergie fournie par Modeller pour calculer un score final pour chaque modèle, puis, classe les modèles en fonction de ce score.

Les sets de données à traiter sont décrits dans un fichier texte. Celui-ci est lu par chacun des scripts. De cette manière chaque script « sait » où sont les données, comment les repérer, comment nommer les fichiers de résultats et où les renvoyer.

A terme une solution de « batch queueing » sera testée.

Perspectives

Notre expérience de calcul sur cluster a reçu un excellent écho dans le service d'Enseignement et dans le service d'Informatique Scientifique. En effet, ces deux services disposent de plusieurs salles de cours avec des ordinateurs qui ne servent que pour l'enseignement. Les besoins en « temps processeur » des chercheurs sont croissants, mais souvent ponctuels pour chaque laboratoire. Le partage des ressources des ordinateurs inutilisés semble rationnel et adapté aux besoins de calculs occasionnels.

Aussi, le service de Micro-informatique va s'inspirer de notre expérience pour proposer l'utilisation du cluster à la communauté.

Nous envisageons également de tester une solution « écran de veille » de type Seti@home²⁶ ou Folding@home²⁷. Il s'agit du projet Model@home²⁸ (6). En effet pour mettre en place un partage de ressources à grande échelle (il existe actuellement plusieurs milliers d'ordinateurs à l'IP) la solution OpenMosix dépend des performances du réseau. Une solution de type SETI a déjà démontrée son efficacité et ne souffre en rien de la lenteur du réseau et semble plus adaptée à l'hétérogénéité du parc et éteindrait les plages de disponibilité.

26 <http://setiathome.ssl.berkeley.edu/>

27 <http://folding.stanford.edu/>

28 <http://www.cmbi.kun.nl/models/>

Nous venons de parcourir les différentes applications mises en place. Regardons, sur un cas concret, comment elles peuvent être employées pour répondre à une question biologique précise.

VI. Application à la reconnaissance de l'IL-2

Le gestionnaire graphique de processus bioinformatiques d'analyse de séquences, de structures, d'arbres phylogénétiques et de réseaux d'interactions de protéines à été conçu autour d'une application commandée par l'utilisateur final, l'Unité d'Immunogénétique Cellulaire. L'application est focalisée sur la détection des protéines interagissant directement ou indirectement au récepteur des cytokines de la famille des hématopoïétines, la localisation des interfaces sur les séquences et sur les structures, et l'identification des résidus encodant la spécificité de la reconnaissance de l'interleukine-2 humaine par ses récepteurs au cours de l'évolution par comparaison aux autres systèmes cytokine-récepteur.

Introduction : L'IL-2 et ses récepteurs

L'interleukine-2 (IL-2) est une cytokine de 133 résidus sécrétée naturellement par les lymphocytes CD4 lors d'une stimulation par les cellules présentant les antigènes (monocytes, cellules dendritiques, lymphocytes B) au cours d'une infection ou en présence de cellules tumorales. L'IL-2 est reconnue par plusieurs types de récepteurs (IL-2R) à la surface des lymphocytes (Fig. 11) et stimule leur prolifération.

Les récepteurs de l'IL-2 comprennent jusqu'à trois chaînes membranaires α , β et γ dont le contrôle de l'expression et de la maturation varie d'un type de lymphocyte à l'autre. La proportion relative des chaînes exprimées dicte la composition oligomérique des récepteurs. Ils se distinguent entre eux par leurs affinités pour l'IL-2: $\alpha\beta\gamma$ (10pM), $\beta\gamma$ (1nM) (Fig. 12).

Aucune structure des chaînes du récepteur n'a été résolue, pas même celles de ses domaines intra ou extra-cytoplasmiques. Seules les structures cristallographiques et par RMN de l'IL-2 ont été mises à jour. Les bases moléculaires de la reconnaissance de l'IL-2 par ses récepteurs et le mécanisme de modification conformationnelle à l'origine de l'initiation transduction du signal ne sont pas documentées expérimentalement. En particulier les hypothèses de pré-

Fig. 11 : Quand et par quelles cellules est produite et sécrétée l'IL-2?

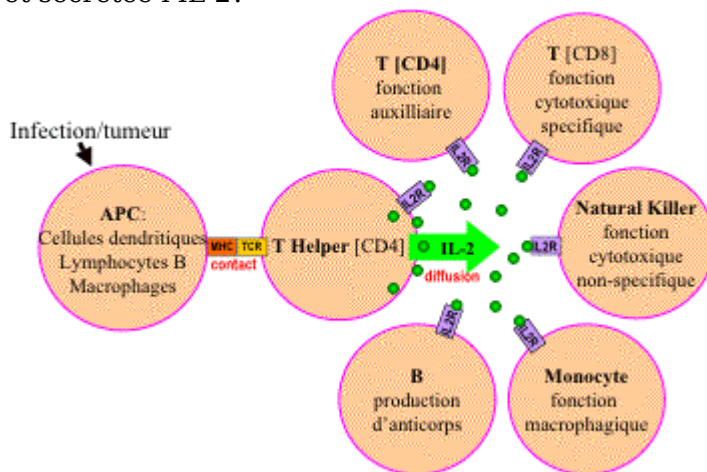
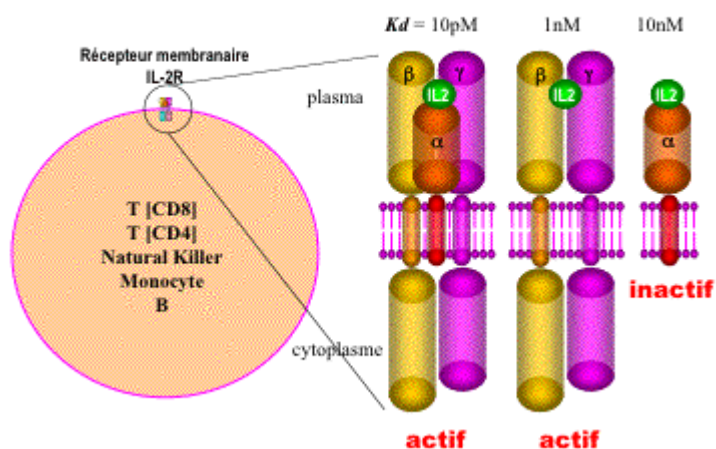


Fig. 12 : Quels sont les récepteurs actifs de l'IL-2?



assemblage du récepteur avant la fixation de la cytokine ou le rôle de l'assemblage induit par la cytokine sur la transduction du signal ne sont ni démontrés ni invalidés.

Comme le décrit la Fig. 12, le domaine cytoplasmique de la chaîne IL-2R β fixerait JaK1 (tyrosine kinase Janus 1) et IL-2R fixerait JaK3. Le rapprochement des domaines cytoplasmiques met en contact JaK1 et JaK3 qui forment un complexe actif qui catalyse la phosphorylation d'une tyrosine de l'extrémité C-terminale de la chaîne IL-2R β (5). Cette tyrosine modifiée est reconnue par le complexe de facteurs de transcription Stat5a-Stat5b qui s'active par auto-phosphorylation. Ces facteurs induisent l'expression de plusieurs gènes, celui de l'IL-2R α par exemple. L'activation du récepteur IL-2R induit aussi le blocage des mécanismes apoptotiques via Bcl2, la modification du cytosquelette par la voie Rho et la prolifération par l'activation de plusieurs gènes par la voie Ras/Raf/MapK (Fig. 13).

Importance médicale de l'IL-2, et du développement d'agonistes et antagonistes de l'IL-2R

L'injection d'IL-2 a été testée depuis 1980 chez des patients pour provoquer la prolifération de lymphocytes et réduire le développement de tumeurs (Fig. 15). La thérapie a été agréée depuis une dizaine d'années pour le traitement de carcinomes rénaux et de mélanomes. Plus récemment, l'IL-2 est utilisée pour reconstituer le taux de CD4 chez certains patients infectés par le virus VIH. Malheureusement, l'IL-2 induit aussi des effets secondaires indésirables. Le syndrome de fuite vasculaire (VLS) responsable entre autres d'œdèmes pulmonaires en est un exemple. La gravité des effets secondaires oblige à réduire les doses injectées ce qui diminue l'efficacité des traitements. Aucun agoniste du récepteur ni aucune IL-2 modifiée n'a permis jusqu'à présent d'améliorer significativement l'index thérapeutique de l'IL-2.

La recherche d'inhibiteurs du récepteur de l'IL-2 offrirait des espoirs de thérapies contre certaines maladies auto-immunes ou en complément de la cyclosporine prescrite dans le cas de greffes d'organes. En effet, la cyclosporine est toxique pour le rein et le seul moyen actuellement d'en réduire les doses, est l'injection d'anticorps anti-IL-2R α , inhibiteur du récepteur IL-2R.

Fig. 13 : Quels sont les rôles d'IL-2 ?

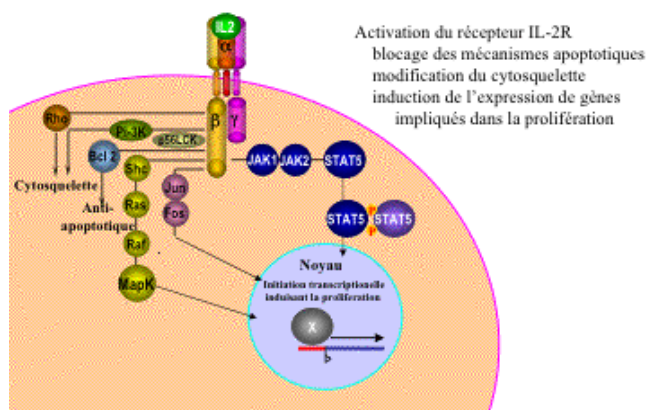
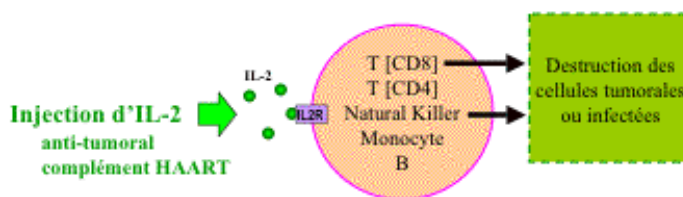


Fig. 14 : Quels usages thérapeutiques pour l'IL-2



Effet positif

- Réduction du nombre de cellules tumorales
 - adénocarcinome du rein
 - mélanome
- Stimulation du système immunitaire, remontée du taux de CD4
- Réduction de la charge virale ?

Effet négatif

- L'IL-2 est un facteur de croissance pléiotropique
- Effets secondaires majeurs associés au syndrome de fuite vasculaire (VLS)

a) Recherche en cours au laboratoire d'Immunogénétique Cellulaire (IGC)

L'Unité IGC est entre autres engagée dans deux projets conjoints l'un est l'identification du mécanisme de transduction induit par l'IL-2 sur les lymphocytes « natural killers » et l'autre de sélectionner ou concevoir des molécules susceptibles de stimuler cette transduction à la place de l'IL-2, toxique chez les patients en traitement.

Le groupe IGC a été le premier à démontrer la présence de récepteurs préassemblés de l'IL-2 spécifiques des NK (Eckenberg et al, 2000). Ceux-ci sont aussi stimulables spécifiquement par des peptides mimant la structure de l'IL-2 entière (9). En l'absence de données cristallographiques Thierry Rose a reconstruit par modélisation moléculaire comparative les structures tridimensionnelles du dimère de chaînes IL-2R $\beta\beta$ libres et associés au tétramère de peptide p1-30₄ (9) sur la base du complexe de l'érythropoïétine et de son récepteur résolue par cristallographie (EPO-EPOR, 1^{er} dans la PDB)(10). Ces simulations suggèrent un mécanisme de transduction de signal à travers la membrane par la modification des interactions des deux chaînes pré-assemblées $\beta\beta$ (Fig.15). Sous l'effet de la fixation du peptide, les extrémités C-terminales des deux domaines extra-cytoplasmiques distantes de plus de 80Å, se rapprochent à moins de 30 Å. Par extension de ces domaines, les hélices transmembranaires uniques sont elles aussi, rapprochées de 80Å à 30Å. Ce mouvement de plus de 50 Å serait responsable du rapprochement des domaines cytoplasmiques (Fig. 16). Chaque domaine cytoplasmique fixe une tyrosine kinase (JaK); le rapprochement des deux kinases permettrait de phosphoryler une tyrosine de l'IL-2R β reconnue comme site de fixation du complexe de facteur de transcription Stat5a-Stat5b.

La symétrie de l'homotétramère de peptides p1-30 (fragment N-terminal de l'IL-2 constitué uniquement de la première hélice) serait responsable de l'association préférentielle au récepteur symétrique $\beta\beta$ plutôt qu'au récepteur asymétrique $\beta\gamma$ (Fig.16). En effet, le peptide p1-30 a la même séquence que l'hélice A de l'IL-2 et l'interaction de A avec le récepteur IL-2R $\alpha\beta\gamma$, se ferait par l'intermédiaire de la chaîne β .

Ce mécanisme semble être généralisable aux IL-2 de tous les organismes (des vertébrés postérieurs

Fig. 15 : Modèles de la transition de conformation entre le récepteur libre et lié au tétramère de p1-30 ?

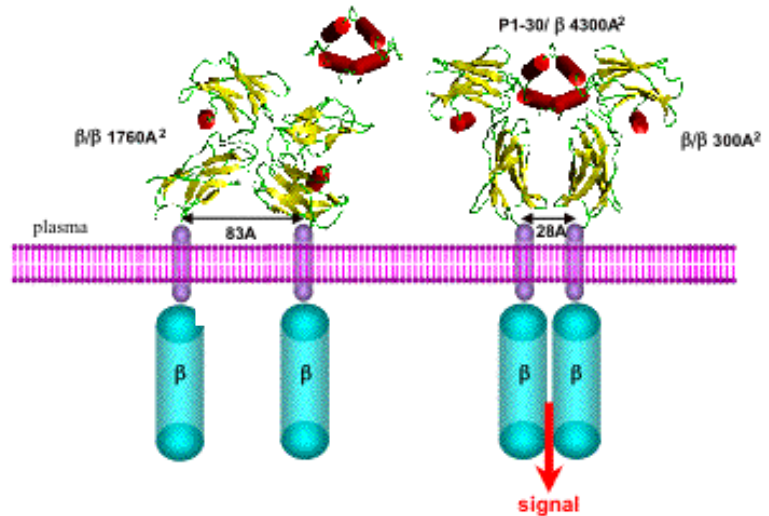
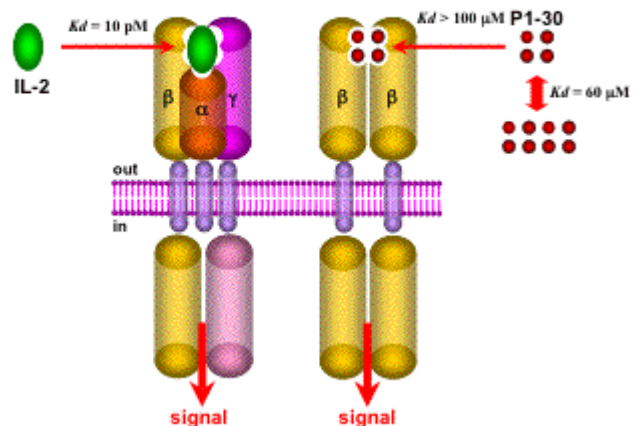


Fig. 16 : Spécificité de p1-30 par rapport à l'IL-2



aux poissons et osseux) sur leurs récepteurs naturels $\alpha\beta\gamma$, $\beta\gamma$ et $\beta\beta$. Nous supposons qu'il décrit aussi le mécanisme de tous les systèmes cytokine-récepteur de la famille des hématopoiétines. Le rapprochement des domaines internes juxtapose deux kinases Jak ; celles-ci peuvent différer en fonction des chaînes (Jak1, Jak2, Jak3, Tyk2) mais beaucoup utilisent les mêmes. Elles activeront à leur tour des facteurs de transcription STAT proches (1 à 6) voire identiques comme l'indique le tableau ci-dessous. Les chaînes de signalisation sont très intriquées et constituent de véritables réseaux. Pourtant les réponses physiologiques sont distinctes en raison des contextes d'activation et des régulations d'expression des différents protagonistes de la chaîne de signalisation.

Tableau 1: Groupes des cytokines et de leurs récepteurs constitués d'une chaîne IL-2R γ .

Cytokines	Récepteurs
IL-2	IL-2R α , IL-2R β , IL-2R γ
IL4	IL4R α , IL-2R γ
IL7	IL7R α , IL-2R γ
IL9	IL9R α , IL-2R γ
IL13	IL13R α , IL4R α , IL-2R γ
IL15	IL15R α , IL-2R β , IL-2R γ
IL-21	IL-21R α , IL-2R γ

Tableau 2: Jak et STAT activés par les cytokines de la famille des hématopoiétines (extrait de Itoh et Arai, 1999, p173, The cytokine network and Immune functions, editor J. Thèze, Oxford Press)

Cytokines	Jak	STAT
IL-2, IL7, IL9, IL15, IL-21	Jak1, Jak3	STAT3, 5a,5b
IL4	Jak1, Jak3	STAT6
IL13	Jak1	STAT6
IL3,IL5	Jak1,Jak2	STAT5
IL6,IL11,LIF,CNTF,OSM	Jak1,Jak2/Tyk2	STAT3/1
IL12	Jak2, Tyk2	STAT3,4
G-CSF	Jak1, Jak2	STAT1,3
EPO	Jak2	STAT5
TPO	Jak2	STAT1,3,5
GH	Jak2	STAT5
PRL	Jak1,Jak2	STAT5
Leptin	?	STAT3,5,6
IFN α/β	Jak1,Tyk2	STAT1,2,3
IFN γ	Jak1,Jak2	STAT1

Cytokines	Jak	STAT
IL10	Jak1, Tyk2	STAT1,3,5
EGF	Jak1	STAT1,3
PDGF	Jak1, Jak2/Tyk2	STAT1,3
M-CSF	Jak1, Tyk2	STAT1,3,5

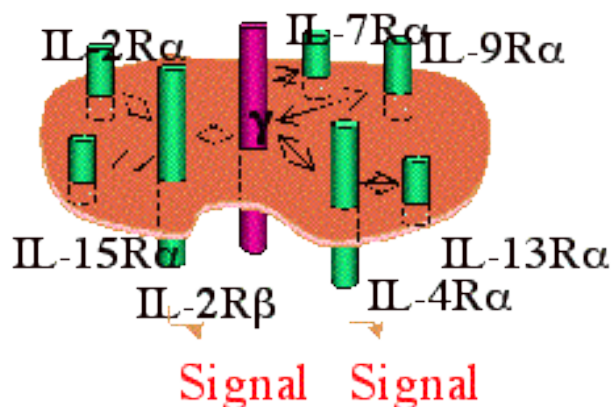


Fig 17 : Partage de chaînes entre les récepteurs du groupe de l'IL2

La structure d'aucun des domaines internes des récepteurs n'est connue, ni même celle d'aucune des Jak ou Tyk. Nous disposons actuellement de données structurales sur la partie externe du récepteur d'érythropoïétine (EPO-EPOR), de l'hormone de croissance (GH-GHR) et de l'IL4 (IL4-IL4R α), et sur un fragment d'un des facteurs de transcription STAT.

Les connaissances au niveau moléculaire sont encore plus confuses sur les autres voies de signalisation (Rho, MapK, Pi3K, BCl2) en raison du nombre d'intermédiaires, de la faible documentation de leurs fonctions biochimiques et de leurs structures. Une approche par modélisation est tout à fait adaptée pour rationaliser le choix des stratégies de modification des partenaires par mutagenèse dirigée afin de localiser leur interface et d'identifier les contacts cruciaux à la base de l'affinité ou des associations.

Approche bioinformatique

Les approches bioinformatiques que j'ai mises en application sur le projet cytokine-récepteur ont pour objet d'assembler les outils dont nous avons besoin dans des pipelines intégrés à la plate-forme PTII pour documenter les interactions entre l'IL-2 et son récepteur. Comme il n'y a pas de données cristallographiques pour le récepteur de l'IL-2, nous avons utilisé des méthodes prédictives. Nous avons ensuite procédé par comparaison entre les modèles des complexes IL2-IL2R chez l'homme et chez deux autres organismes, la souris et le rat. Nous avons étudié l'approche d'autres cytokines de la même famille des hématopoïétines dont les complexes partiels sont cristallisés : l'IL-4 qui reconnaît le récepteur IL4R α -IL2R mais pas IL2R β -IL2R γ et IL-6 dont aucune des chaînes du récepteur de l'IL-2 ne contribue à sa fixation. Pour identifier les différences nous avons utilisé les outils d'analyse et de cartographie des structures, articulés et intégrés à cette fin dans PTII et décrits dans le chapitre précédent.

La stratégie de cette phase d'application, peut être décrite ainsi :

- Rechercher les séquences orthologues à chaque cytokine et chaîne de récepteur,
- Produire des alignements multiples avec la ou les séquences des structures choisies comme référence.
- Produire les scripts d'exécution pour Modeller, choisir le nombre de modèles à produire, le poids et la nature des contraintes géométriques.
- Exécuter Modeller en local ou distribué en fonction du nombre de modèles.
- Analyser l'énergie potentielle, la stéréochimie des modèles et leurs adéquations avec la séquence.
- Classer les modèles.
- Retenir les meilleurs modèles lorsque leurs squelettes peptidiques différents 2 à 2 de plus de 2.5Å (rmsd).
- Analyser des violations stéréochimiques dans les rapports Procheck et Whatcheck.
- Éliminer des structures présentant des nœuds (1/250).
- Reprendre si nécessaire la recherche conformationnelle avec Modeller en ajoutant l'un des 10 meilleurs modèles comme template en plus de la structure cristalline et finir le cycle lorsqu'aucun meilleur modèle n'est obtenu à plus de 2.5Å des précédents.
- Optimiser la conformation des boucles et des terminaisons, soit par dynamique avec les extrémités fixées (Charmm²⁹, Harvard Medical School) soit en utilisant un algorithme génétique de (DrawBridge, Michael J. Bower, UCSF).
- Repositionner les chaînes latérales avec CWRL (Roland L. Dunbrack, Fox Chase Cancer Research, Philadelphie).
- Minimiser le modèle (Charmm, Harvard Medical School).
- Calculer le nombre de contacts entre les chaînes et leurs énergies libres de dissociation (Murphy et Freire 1993 Proteins, 15:113-120, Rose et al, 2003).
- Produire les cartes de conservation de résidus avec PyMOL/COSA et figurer les doubles mutations compensatoires aux interfaces avec PyMOL/evoluswap (IGC).
- Dissocier les systèmes avec Chisel (IGC) puis cartographier les interfaces avec PyMOL/Kiss (IGC).
- Afficher la distribution du potentiel électrostatique à la surface de la molécule et la surface d'isopotential électrostatique avec PyMOL/APBS.

a) Rechercher les séquences orthologues à chaque cytokine, chaîne de récepteur

Les paramètres de recherche des séquences (matrice de substitution, espérance e, création et extension de lacune) dans la database de séquences Nrprot (NCBI) par la méthode Blast (NCBI) ont été optimisés par un workflow de recherche itérative mis au point par Franck Valentin.

Le tableau ci-dessous résume notre recherche des orthologues des cytokines de la famille des hématopoïétines (homologue à l'EPO) du groupe de l'IL-2 (récepteur à chaîne IL-2R γ), des

²⁹ <http://www.charmm.org/>

chaînes des récepteurs, des protéines Janus kinases et des facteurs nucléaires de transcription STAT. Les différentes colonnes représentent de gauche à droite :

- le nom de la protéine
- la longueur de la séquence humaine avant maturation
- le nombre d'organismes où cette protéine a été identifiée
- la présence d'un signal de sécrétion détectée par la méthode SignalP
- le nombre de segments transmembranaires détecté par la méthode TMHMM
- les domaines d'interaction peptidiques reconnus (Pfam, Prosite, Smart)
- l'identification expérimentale d'une activité kinase
- nombre de tyrosines phosphorylées après activation
- nombre de ponts disulfures
- nombre de sites de glycosylation
- existence d'une structure cristallographique (Protein DataBase à RCSB)
- existence d'un modèle prédictif calculé par l'unité IGC.

Protéines	Long	Nombre	signal	TM	PRM	Kin	pY	S-S	Glyc	x3D	model
IL-2	153	34	1-20	non	non	non	non	1	1	oui	oui
IL4	153	26	1-24	non	non	non	non	3	1	oui	oui
IL7	177	7	1-25	non	non	non	non	3	3	non	oui
IL9	144	4	1-18	non	non	non	non	non	4	non	oui
IL13	132	11	1-20	non	non	non	non	2	4	non	oui
IL15	162	12	1-29	non	non	non	non	2	1	non	oui
IL-21	162	7	1-22	non	non	non	non	1	1	non	oui
IL-2R α	272	8	1-21	1	non	non	non	5	6	non	partiel
IL-2R β	551	3	1-25	1	non	non	1	2	4	non	partiel
IL-2R γ	379	6	1-22	1	non	non	non	2	5	non	partiel
IL4R α	825	10	1-25	1	non	non	4	2	6	non	partiel
IL7R α	459	3	1-20	1	non	non	2	non	6	non	partiel
IL9R α	522	4	1-40	1	non	non	1	non	2	non	partiel
IL13R α	427	8	1-36	1	non	non	1	non	oui	non	partiel
IL15R α	267	3	1-37	1	non	non	1	non	oui	non	partiel

b) Produire des alignements multiples avec la ou les séquences des structures choisie(s) comme référence

Les fichiers résultats de recherche de séquences par Blast ont été filtrés par le programme blast2list (IGC) qui permet de sortir, de filtrer et de classer les séquences par scores ou par organismes. Les listes ont été utilisées pour produire des fichiers où figurent les séquences complètes au format fasta avec le programme fastacmd du package Blast (NCBI). Ce fichier est alors utilisé en entrée par le programme ClustalW (Thompson et al, 1987) pour produire un fichier d'alignement multiple. Les paramètres d'alignement ont aussi été testés par Franck

Valentin (matrices de substitution, espérance e, création et extension de lacune) en vue d'optimiser le taux d'identité par paire et le taux de similitude sur la totalité du multialignement calculé par le programme clustal2. Comme décrit par Rose et al. (2003) la structure du complexe de l'érythropoïétine et de son récepteur (EPO-EPOR, réf. PDB: 1eer, Livhna et al, Science 1999) a été retenue en raison de sa qualité et de sa similitude avec le système IL2-IL2R.

c) Produire les scripts d'exécution pour Modeller, choisir le nombre de modèles à produire, le poids et la nature des contraintes géométriques

Nous avons écrit le programme en C, clustal2modeller qui produit à partir du fichier d'alignement multiple au format Clustal, un fichier d'alignement au format PIR spécifique pour Modeller et un script utilisé par Modeller pour fixer :

- le nom des fichiers de références structurales
- le nombre de modèles à construire et le numéro d'ordre du premier
- l'initiateur du générateur de nombre aléatoire
- l'indication de contraintes géométriques entre paires d'atomes, de création de ponts disulfures, d'extension et maintien de structure secondaire...
- l'optimisation des boucles

Voici un exemple simple de protocole standard :

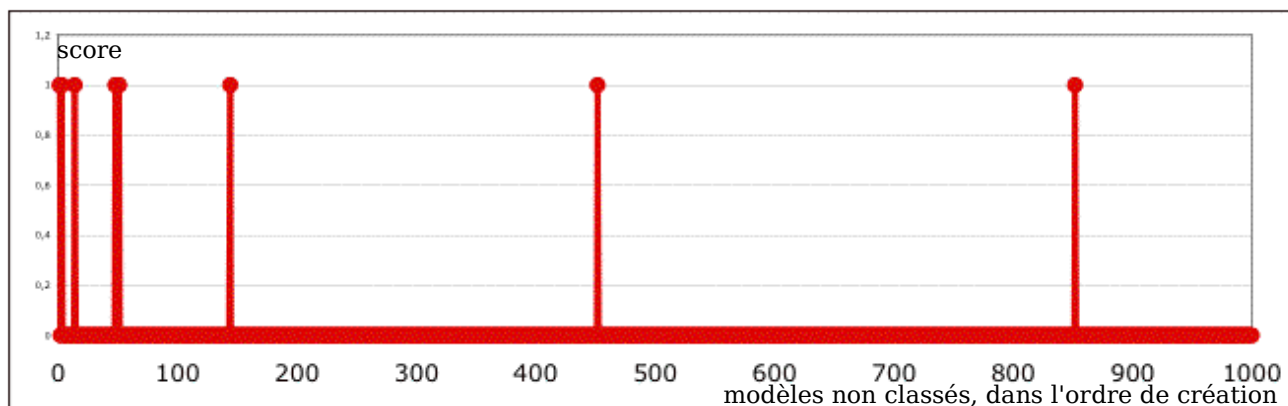
```
# Homology modeling with modeller 6.2
# Script written by the program clustal2modeller IL-2 human.top
# Target : sequence 12 in IL-2 modtest.aln
# 1 templates : IL-2 m47.pdb
#####
## Protocole execution
#$modeller4/IL-2 human.top
#####
## Protocole listing
INCLUDE
SET ALNFILE = 'IL-2 human.ali'
SET KNOWN = 'IL-2 m47.pdb'
SET SEQUENCE = 'IL-2_human'
SET ATOM FILES DIRECTORY = './IL-2/Modeling/'
SET OUTPUT DIRECTORY = './IL-2/Modeling/'
SET STARTING MODEL = 1
SET ENDING MODEL = 1000
SET RAND SEED = 666
CALL ROUTINE = 'model'
STOP
```

d) Exécuter Modeller en local ou distribué en fonction du nombre de modèles

Le nombre de modèles à exécuter dépend du niveau d'identité et du nombre d'insertions dans la séquence requête par rapport aux templates, en fonction de la taille des boucles. Si la structure des boucles ou des extrémités de la chaîne sont importantes et que l'alignement est bon dans ces régions, plusieurs modèles seront nécessaires pour en récupérer un

correct. Si par contre l'alignement est mauvais il n'y a aucune chance d'obtenir un bon modèle et il faudra préférer des méthodes comme Robetta (D.Baker, University of Washington, Seattle) (7) qui s'appuie sur une librairie de peptides dont la « fusion conformationnelle » par un algorithme d'optimisation de type Monte-Carlo donne de meilleurs résultats.

Ci-dessous le graphe montre l'acquisition d'une nouvelle structure meilleure que toutes les précédentes au cours de la construction de 1000 modèles. Sur 12 requêtes de complexes IL-2-récepteur, le point d'inflexion de la dérivée du « tirage » de meilleure solution est situé entre 100 et 200. Par conséquent, 200 modèles est le nombre optimal dans les conditions de niveau d'identité entre la requête (IL2-IL2R) et son template (EPO-EPOR). Ce type de figure nous sert à étalonner le nombre de modèles à construire et de choisir les paramètres de modeller. Il permet également, de tester si l'addition d'étapes d'optimisation, Drawbridge, CWRL ... est nécessaires.

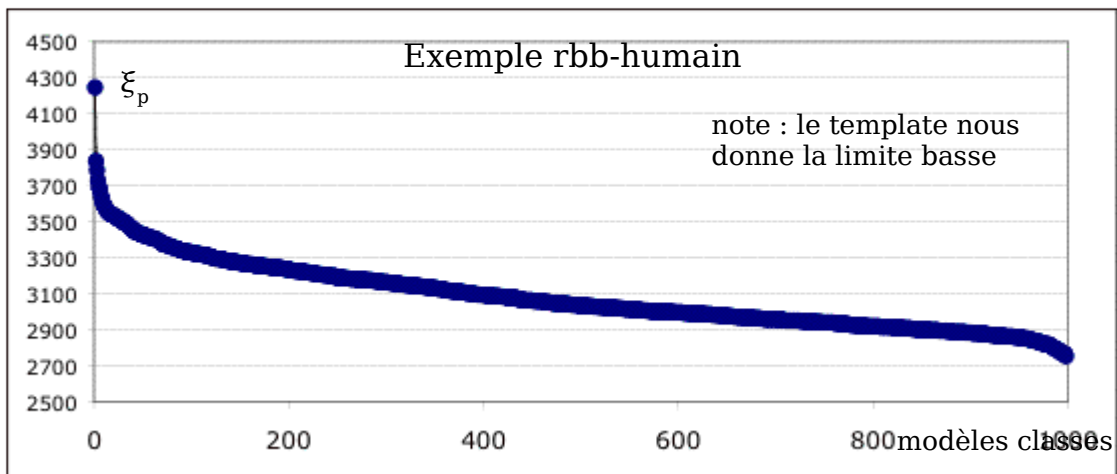


e) Analyser l'énergie potentielle, la stéréochimie des modèles et leurs adéquations avec la séquence

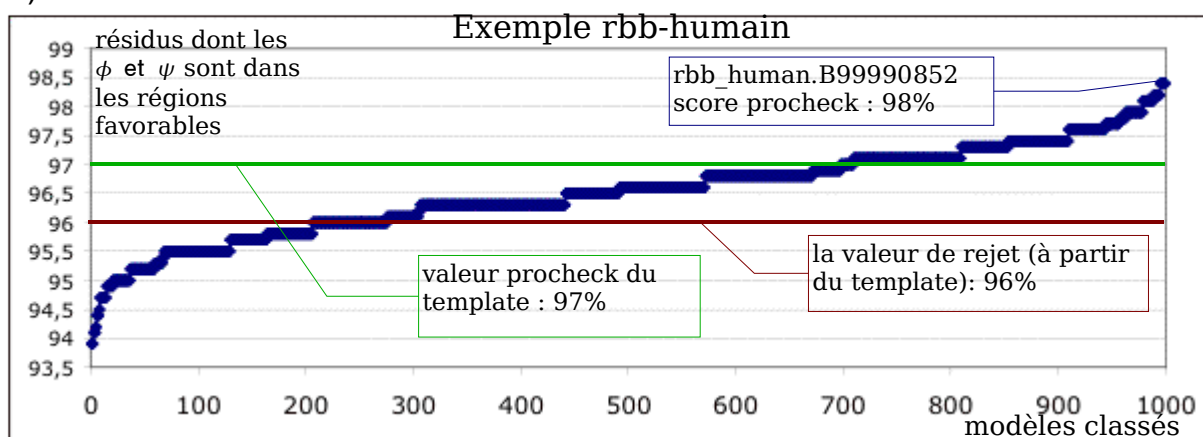
L'énergie potentielle calculée par Modeller est affichée dans le fichier décrivant le modèle :

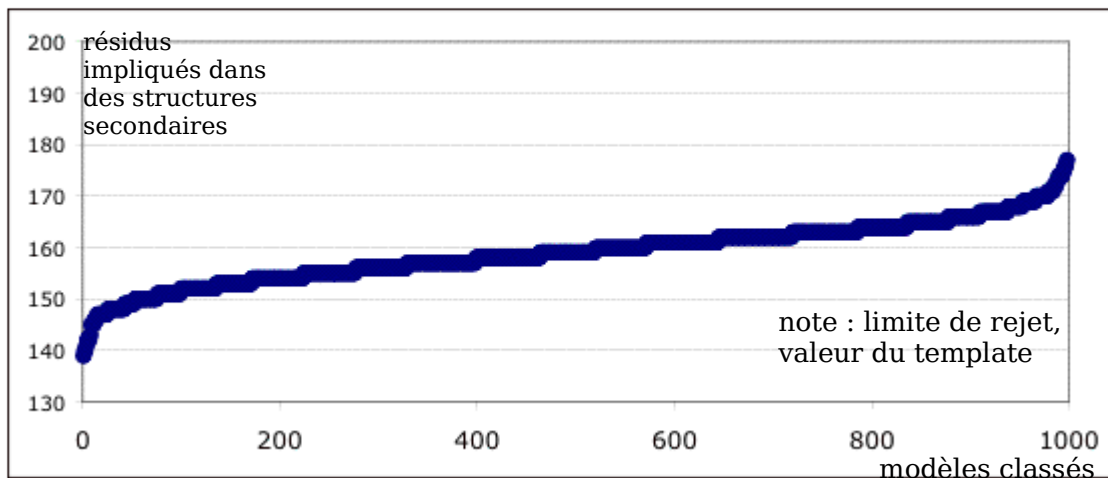
```
EXPDTA MODEL, MODELLER Version 6v2 2004/09/03 02:31:01.990
REMARK MODELLER OBJECTIVE FUNCTION: 3240.8820
ATOM 1 N ALA 1 0.943 -4.422 -24.054 1.00 25.00 1SG 2
```

La valeur 3240.880 kcal/mol de l'énergie potentielle du système, ci-dessus, rend compte à la fois du champ de force utilisé (Charmm22) et donc de la conformité stéréochimique (distance, angle, torsion, planéité des liens peptidiques et cycles...) mais aussi des éventuelles violations des contraintes harmoniques apportées par le template. Le graphe ci-dessous montre la distribution de l'énergie en kcal/mol sur un millier de modèles construits pour une requête. Les modèles sont classés par énergies décroissantes, les « bons » modèles sont à droite.



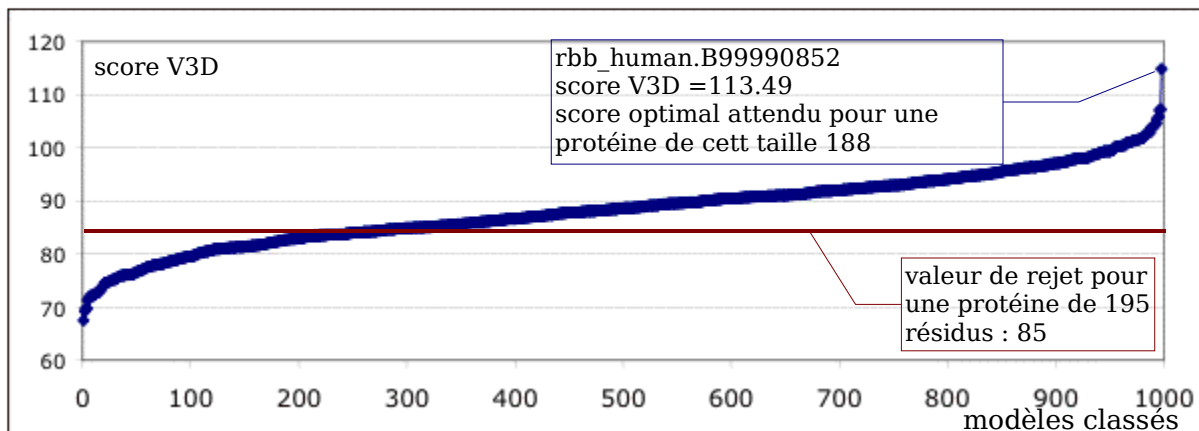
Le nombre de résidus dont les angles dièdres phi et psi sont dans les zones très favorables et favorables du diagramme de ramachadran calculé par procheck et en ordonnées ci-dessous est utilisé comme critère de sélection. Les autres critères stéréochimiques sont utilisés pour les dix meilleurs modèles fanaux, pour localiser d'éventuels problèmes (cause de rejet).





Le nombre de résidus engagés dans les structures secondaires en ordonnée dans le graphe ci-dessous (distribution classée) a été dans le cas présent retenu comme critère, du à la forte proportion de structure secondaires dans le template. Les « bons » modèles sont à droite.

L'adéquation du repliement et de la séquence est calculée avec verify3D dont le score est affiché en ordonnée ci-dessous dans la distribution classée. Les « bons » modèles sont à droite.



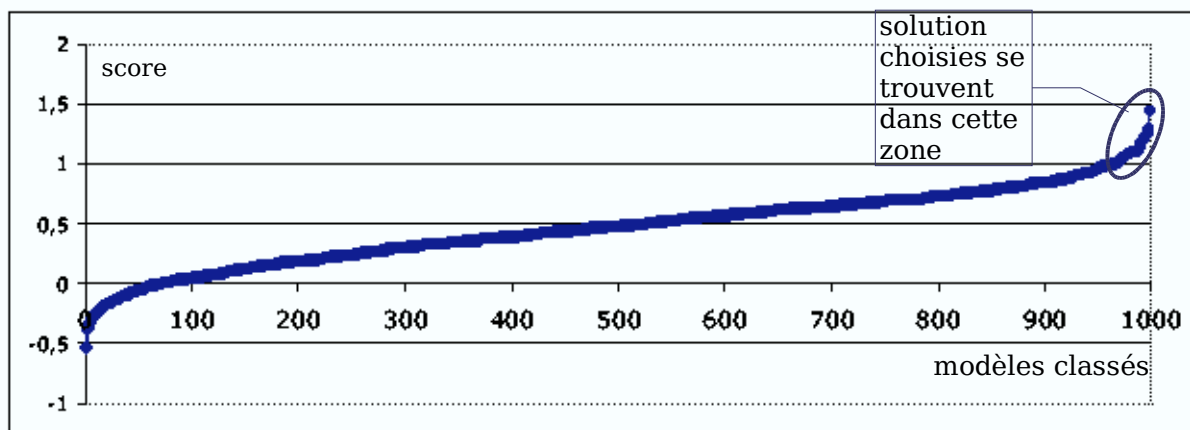
Pour le template
 profile3D score = 175.5
 score optimal attendu pour une
 protéine de cette taille 195 et une
 valeur de rejet de 88.

f) Classer les modèles

Les 4 critères : l'énergie (Ep), la population des aires de ramachandran (Ram), le taux de structure secondaire (2D), l'adéquation séquence-structure (verify3d) ont été normalisés puis sommés en une combinaison linéaire avec le programme score pour donner un score qui permet de classer les modèles entre eux. Un exemple des 5 meilleurs scores pour une requête de 1000 modèles :

Rank	Model	Ep	2D	Ram	Verify3D	score
995	250	3004	175	96.90	106.56	1.50
996	154	2943	173	97.40	104.41	1.52
997	544	2943	173	97.40	104.41	1.52
998	777	2943	173	97.40	104.41	1.52
999	954	2943	173	97.40	104.41	1.52
1000	234	2853	173	97.70	101.32	1.55

La distribution des scores en fonction de modèles classés est indiquée ci-dessous. Les bons modèles sont à droite.



g) Retenir les meilleurs modèles lorsque leurs squelettes peptidiques diffèrent 2 à 2 de plus de 2.5Å (rmsd)

J'ai composé un script python pour PyMOL pour calculer les rmsd entre 2 squelettes peptidiques, pour ne retenir que les meilleures solutions distinctes entre elles. Lorsque le meilleur modèle se détache bien dans le graphe des scores classés en fonction du modèle, comme c'est le cas ici (cf graphique ci-dessus), c'est en général, une bonne indication de la qualité du modèle.

h) Analyse des violations stéréochimiques dans les rapports Procheck et Whatcheck

Une analyse avec Procheck et Whatcheck permet d'éliminer les structures qui présentent des problèmes stéréochimiques ponctuels. Les modèles sont rejetés lorsqu'ils ne souscrivent pas grossièrement les critères de qualité des templates à une résolution fixe. Cela arrive lorsque les niveaux d'identité (local ou global) entre les séquences requête et cible est faible et engendre des alignements de qualité médiocre. Ces problèmes se présentent aussi, dans le

cas d'utilisation de plusieurs templates différents.

i) Élimination des structures présentant des nœuds (1/250)

L'examen de la recherche de nœuds est encore visuelle, car nous ne disposons pas encore de programmes accomplissant efficacement cette tâche. Cet événement arrive en particulier dans les boucles contraintes par plusieurs templates. Ici leurs fréquences sont inférieures à 1 pour 250.

j) Amélioration du modèle

Nous reprenons si nécessaire la recherche conformationnelle avec Modeller en ajoutant l'un des dix meilleurs modèles comme template en plus de la structure cristalline. Le cycle est terminé lorsqu'aucun modèle meilleur et à plus de 2.5Å des précédents n'est obtenu.

Les modèles de cette étude sont suffisants dans l'état actuel. Certains ajustements des alignements pourraient être opérés avant de pousser plus loin les optimisations

k) Optimiser la conformation des boucles et des terminaisons.

L'optimisation conformationnelle est réalisée soit par des méthodes de la dynamique moléculaire avec les extrémités fixées (Charmm, Harvard Medical School), soit en utilisant un algorithme génétique (DrawBridge, Michael J. Bower, UCSF)

Ce type d'optimisation a été réalisé par Rose et al. (2003) mais pas encore dans l'état actuel de travaux décrits dans ce mémoire.

l) Repositionnement des chaînes latérales avec CWRL (Roland L. Dunbrack ,Fox Chase Cancer Research, Phyladelphie)

Reste à réaliser.

m) Minimiser le modèle (Charmm, Harvard Medical School)

Reste à réaliser.

n) Calculer le nombre de contacts entre les chaînes et leurs énergies libres de dissociation (Murphy et Freire)

Reste à réaliser.

o) Afficher la distribution du potentiel électrostatique à la surface de la molécule et la surface d'isopotential électrostatique autour, avec PyMOL/APBS (cf fig. 7 et 17)

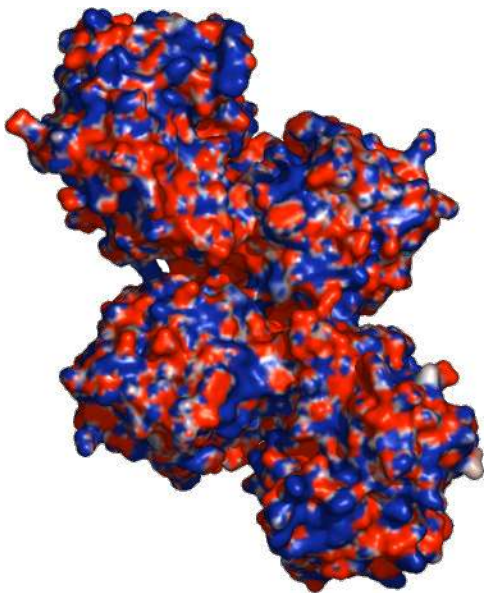
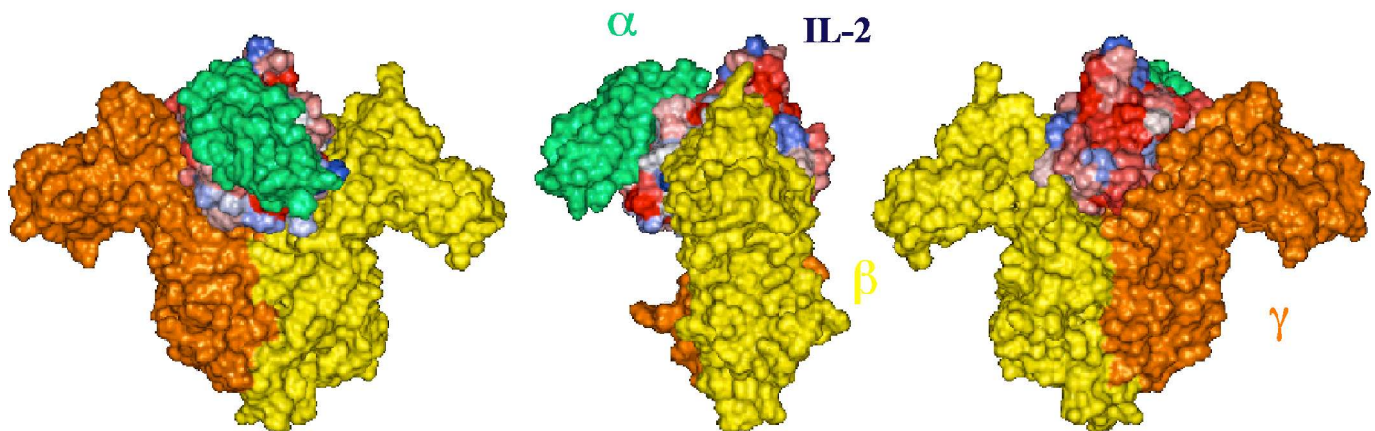


Fig. 17: Représentation montrant la projection du potentiel électrostatique sur la surface d'une protéine cristallographiée au laboratoire. Voir aussi la figure 7.

p) Produire les cartes de conservation de résidus avec PyMOL/COSA et dissocier les systèmes avec Chisel puis cartographier les interfaces avec PyMOL/Kiss

La figure suivante présente 3 orientations des 3 fragments extra-cytoplasmiques du récepteur IL-2R α , IL-2R β et IL-2R γ complexé avec IL-2. Les surfaces accessibles au solvant (l'enveloppe en contact avec l'eau) sont représentées en bleu-blanc-rouge pour l'IL-2, en vert pour l'IL-2R α , en jaune pour l'IL-2R β et en orange pour IL-2R γ .

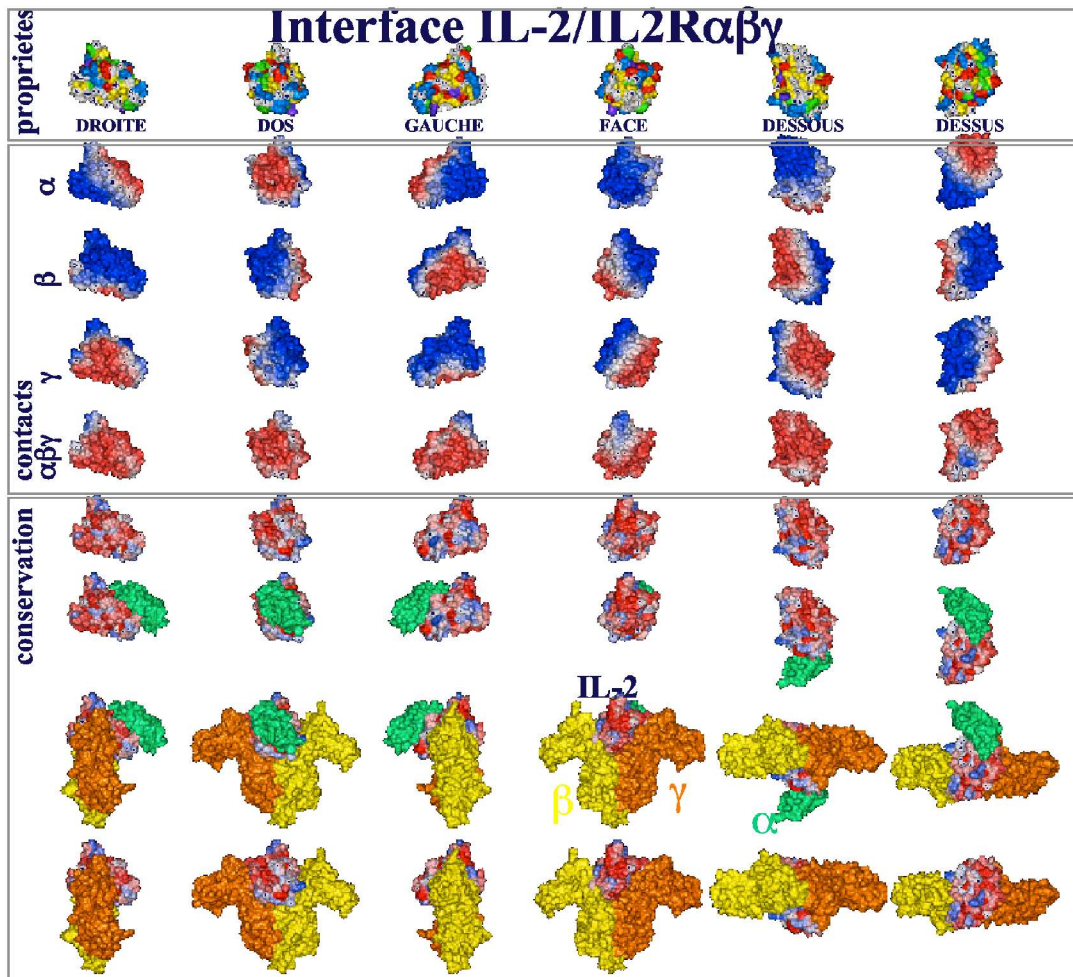
Interface IL-2/IL2R $\alpha\beta\gamma$



Nous avons intégré dans le pipeline géré par PTH plusieurs logiciels qui permettent d'afficher des propriétés à la surface d'une molécule en utilisant la colonne réservée usuellement aux B-factors. Cosa permet d'afficher le niveau de conservation des résidus (rouge = conservé, bleu = variable, blanc = intermédiaire) et Kiss l'empreinte des interfaces (rouge = contact, bleu >10Å).

La figure suivante présente, de gauche à droite, les six perspectives. Elles sont alignées de haut en bas pour permettre leur comparaison. Nous avons représenté, de haut en bas :

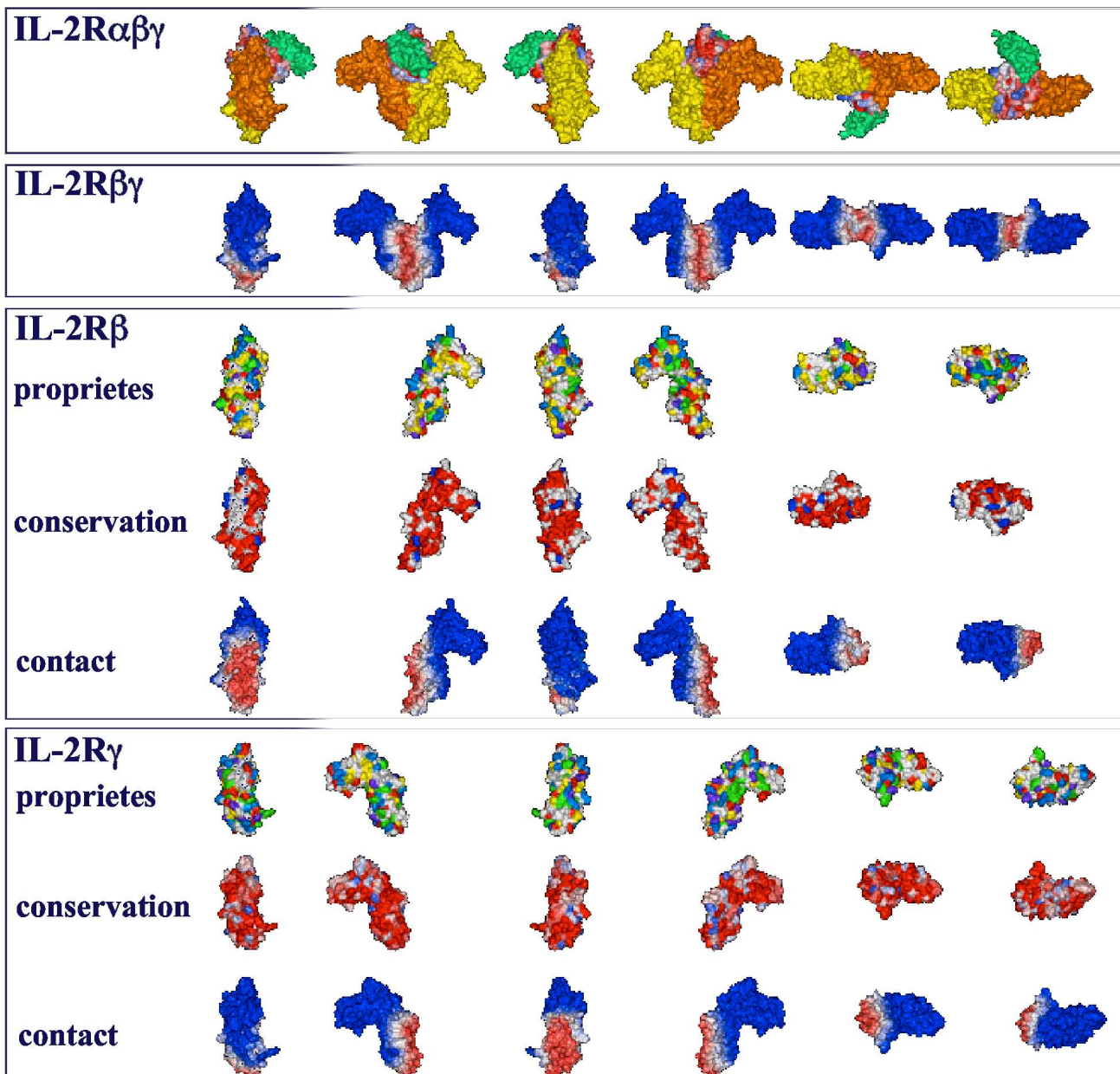
- la nature physico-chimique des chaînes latérales à la surface de l'IL2 (bleu : Arg, Lys; violet : His; rouge : Asp, Glu; jaune : Leu, Ile, Val, Met, Ala; Vert: Phe, Trp, Tyr)
- l'IL-2 aux couleurs Kiss des contacts intermoléculaires avec IL2R α
- l'IL-2 aux couleurs Kiss des contacts intermoléculaires avec IL2R β
- l'IL-2 aux couleurs Kiss des contacts intermoléculaires avec IL2R γ
- l'IL-2 aux couleurs Kiss des contacts intermoléculaires avec IL2R $\alpha\beta\gamma$
- l'IL-2 aux couleurs COSA représentant le niveau de conservation parmi 34 séquences d'IL-2 d'organismes différents
- représentation identique de l'IL-2 avec figuration de IL2R α
- représentation identique de l'IL-2 avec figuration de IL2R α , IL2R β et IL2R γ
- représentation identique de l'IL-2 avec figuration de IL-2R β ,IL-2R γ



Nous pouvons remarquer que les zones de l'IL-2 en contact avec IL-2R α et IL-2R γ sont conservées mais pas celle entre l'IL-2 et IL-2R β . Il y a une crête bleue dans la vue des contacts de l'IL-2 avec IL-2R $\alpha\beta\gamma$ qui ne contribue donc pas à l'interface cytokine récepteur alors que cette zone est particulièrement bien conservée. Il pourrait s'agir d'une éventuelle extension de la chaîne IL-2R α dont nous n'avons pas pu figurer la chaîne complète faute de template. Ainsi, soit l'IL2R α pourrait, vraisemblablement, recouvrir la totalité de la face supérieure de l'IL-2 soit, un autre partenaire reste à découvrir. Répondre par une approche expérimentale à cette incertitude soulevée par mon travail est maintenant une priorité du groupe IGC.

La figure suivante présente, de gauche à droite, les six perspectives pour, de haut en bas :

- le complexe IL2-IL2R $\alpha\beta\gamma$ aux couleurs identiques à la figure précédente
- le complexe IL2R β -IL2R γ , code couleur Kiss des contacts intermoléculaire
- IL-2R β , représentation des propriétés, conservation, contact avec IL-2R γ
- IL-2R γ , représentation des propriétés, conservation, contact avec IL-2R β



Le niveau de conservation est hétérogène à la surface de l'IL-2. L'interface avec les chaînes IL-2R α et IL-2R γ du récepteur est conservée (surfaces rouges) d'un organisme à l'autre mais pas l'interface avec IL-2R β . Ceci suggère que la chaîne IL2R β confère la spécificité pour la reconnaissance de l'IL-2 par son propre récepteur. Ceci est cohérent avec

le fait que l'IL-2 de souris ne se fixe pas sur le récepteur humain et qu'il faut produire une souris transgénique dotée du gène humain encodant pour l'IL2R pour observer un effet de l'IL-2 humain chez la souris. L'observation est identique pour l'IL4R, IL7R et IL9R qui encoderaient d'après notre approche bioinformatique, la spécificité pour les cytokines IL4, IL7 et IL9 respectivement. Bien qu'encore préliminaire, ce travail d'application dresse pour la première fois une liste des résidus qui encoderaient l'affinité à l'interface IL2-IL2R α , IL2-IL2R β et IL2-IL2R γ et la spécificité à l'interface IL2-IL2R β . Cela donne des lignes de guide au groupe IGC pour définir une stratégie d'étude de ces interactions par mutagenèse dirigée.

VII. Conclusions

Lors de ce stage ma tâche était essentiellement de trouver des outils adaptés à nos besoins de représentation, d'analyse et de prédiction de structures de protéines pour compléter nos pipelines. Lorsque les articulations, permettant de passer les données d'un programme à un autre n'existaient pas, j'ai dû les développer.

Nous avons également eu une attitude pragmatique, tournée vers le résultat. Il s'agissait au cours de ce stage de poser des jalons pour aller de la séquence jusqu'à l'analyse de sa structure. Des améliorations seront ainsi plus faciles à amener ultérieurement, une fois que la route a été tracée, tant dans la substitution de programmes par d'autres que dans la diversification des fonctionnalités. Le concept de « brique » dans le gestionnaire de programmes permettra au groupe d'IGC d'implémenter sur la même trame une recherche de fixation de ligands et de criblage virtuel de bibliothèques de molécules.

L'application sur l'IL-2 sera maintenant généralisée pour les autres interleukines dans le groupe IGC. Le développement du projet ne s'arrêtera pas après la fin de mon stage, Martin Grana qui commence sa thèse dans le laboratoire de Biochimie Structurale et qui a pour thématique la relation structure/fonction utilise déjà des outils qui ont été mis en place dans le cadre du projet sous PTII.

Le gestionnaire de workflow et en particulier les briques logicielles de représentation et d'analyse des structures servira de base au module bioinformatique de l'enseignement de maîtrise de Biochimie des protéines à l'Institut Pasteur.

Le gestionnaire et toutes les applications sont disponibles sur le site pasteur : www.pasteur.fr/panoramics.

Bibliographie

1. A method to identify protein sequences that fold into a known three-dimensional structure. , Bowie JU, Luthy R, Eisenberg D. *Science*. 1991 Jul 12;253(5016):164-70.
2. Assessment of protein models with three-dimensional profiles, Luthy R, Bowie JU, Eisenberg D. *Nature*. 1992 Mar 5;356(6364):83-5
3. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. , Kabsch W, Sander C. *Biopolymers*. 1983 Dec;22(12):2577-637.
4. Electrostatics of nanosystems: application to microtubules and the ribosome. , Baker NA, Sept D, Joseph S, Holst MJ, McCammon JA. *Proc. Natl. Acad. Sci. USA* 98, 10037-10041 2001
5. IL-2 receptor signaling through the Shb adapter protein in T and NK cells. , Lindholm CK. *Biochem Biophys Res Commun* . 2002 Aug 30;296(4):929-36.
6. Models@Home: distributed computing in bioinformatics using a screensaver based approach. , Krieger E, Vriend G. *Bioinformatics* . 2002 Feb;18(2):315-8.
7. Protein structure prediction and analysis using the Robetta server Kim DE, Chivian D, Baker D. *Nucleic Acids Res*. 2004 Jul 1;32(Web Server issue):W526-31.
8. Stereochemical quality of protein structure coordinates. , Morris AL, MacArthur MW, Hutchinson EG, Thornton JM. *Proteins* 1992 Apr;12(4):345-64.
9. Structural analysis and modeling of a synthetic interleukin-2 mimetic and its interleukin-2Rbeta2 receptor. , Rose T, Moreau JL, Eckenberg R, Theze J. , *J Biol Chem*. 2003 Jun 20;278(25):22868-76. Epub 2003 Apr 03.
10. The first alpha helix of interleukin (IL)-2 folds as a homotetramer, acts as an agonist of the IL-2 receptor beta chain, and induces lymphokine-activated killer cells. , Eckenberg R, Rose T, Moreau JL, Weil R, Gesbert F, Dubois S, Tello D, Bossus M, Gras H, Tartar A, Bertoglio J, Chouaib S, Goldberg M, Jacques Y, Alzari PM, Theze J. *J Exp Med*. 2000 Feb 7;191(3):529-40.
11. Crystallographic evidence for preformed dimers of erythropoietin receptor before ligand activation. , Livnah O, Stura EA, Middleton SA, Johnson DL, Jolliffe LK, Wilson IA. *Science*. 1999 Feb 12;283(5404):987-90.