

Développement d'un utilitaire de sélection de particules observées au microscope électronique

Greffon Pick_EM pour ImageJ

FAUX - HÉRICÉ - PAYSAN-LAFOSSE - SANSEN

Master 1 Bioinformatique

Projet de programmation sous la direction de Jean-Christophe TAVEAU



25 Mai 2012

1 Introduction

- Contexte
- Objectifs

2 Analyse

- Besoins fonctionnels
- Besoins non fonctionnels

3 Conception - Réalisation

- Modularité
- Interface Graphique (GUI)
- Algorithmes
- Résultats
- Difficultés et améliorations

4 Conclusion

Introduction

CBMN

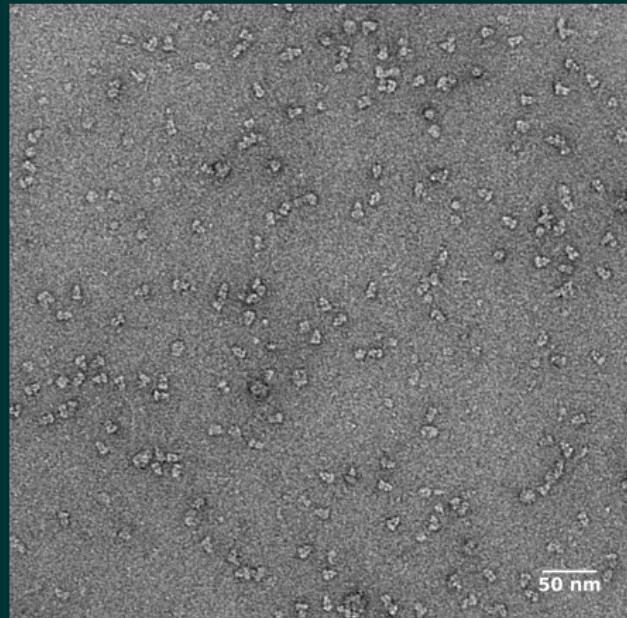
Laboratoire de Chimie et Biologie des Membranes et Nanoobjets

ACMPC

Équipe Architectures des Complexes Membranaires et Processus Cellulaires

Contexte

- Micrographies de complexes protéiques
- Utilisation du logiciel ImageJ
- Sélection manuelle fastidieuse
 - Chronophage
 - Accapare un membre de l'équipe
 - Répétitive



Exemple de micrographie

Objectifs

Création d'une interface

- Facile d'utilisation
- Claire et succincte
- Récupération des paramètres utilisateurs

Implémentation d'algorithmes

- Automatisation du traitement et de la sélection
- Récupération des coordonnées
- Images individuelles

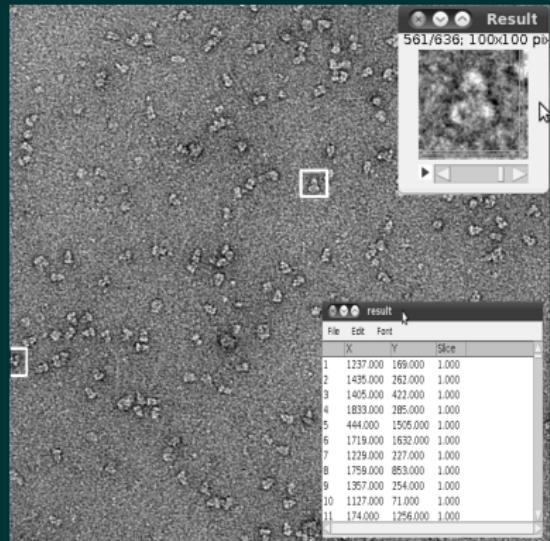
Besoins fonctionnels

Interface

- Choix entre plusieurs algorithmes
- Différente pour chaque algorithme

Algorithmes

- Sélection automatique
- Résultats : tableau de coordonnées (x , y , z) et pile d'images individuelles



Images individuelles et tableau de coordonnées

Besoins non fonctionnels

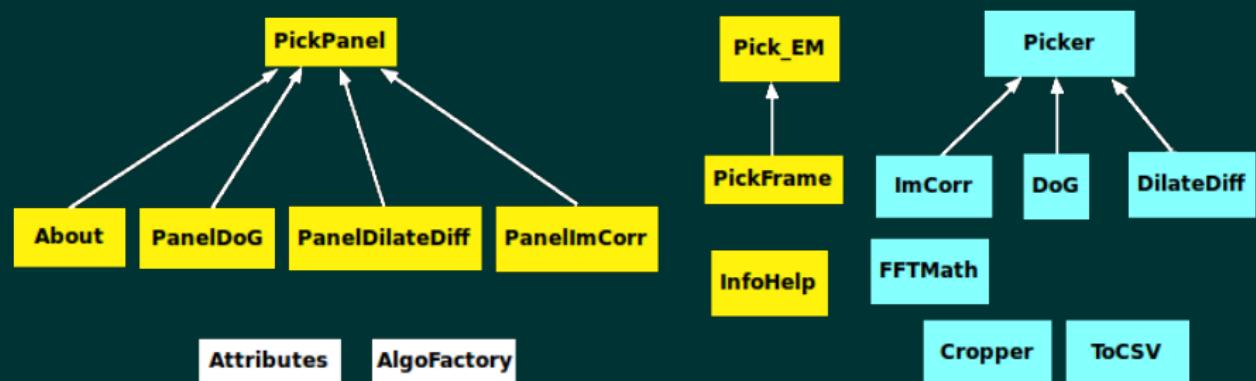
Interface

- Implémentation en Java
(AWT ou **Swing**)
- Modularité

Algorithmes

- Implémentation en Java
- Traitement rapide (grands jeux de données)
- Minimiser les étapes intermédiaires

Modularité - Diagramme des classes



Modularité - Patrons de conception

La classe Attributes

HashTable : couple clé/valeur des paramètres utilisateurs

Singleton

Restreint l'instanciation d'une classe à un seul objet

Algorithme 1

arg 1
arg 2

Algorithme 2

arg 1
arg 2
param 1

Algorithme 3

param 1
param 2
param 3
param 4

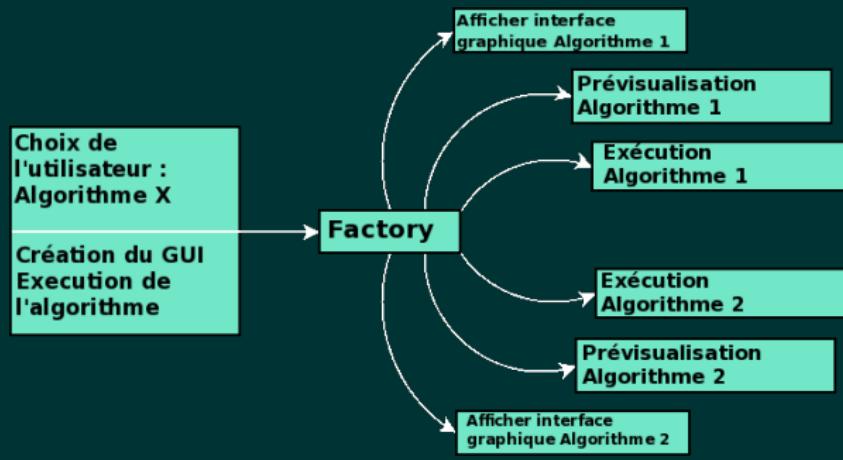
Modularité - Patrons de conception

La classe AlgoFactory

- actualisation de l'interface
- appel aux méthodes de sélection

Fabrique de création (*Factory*)

- Classe qui a pour rôle de construire des objets
- Seule responsable de la création / distribution de l'objet

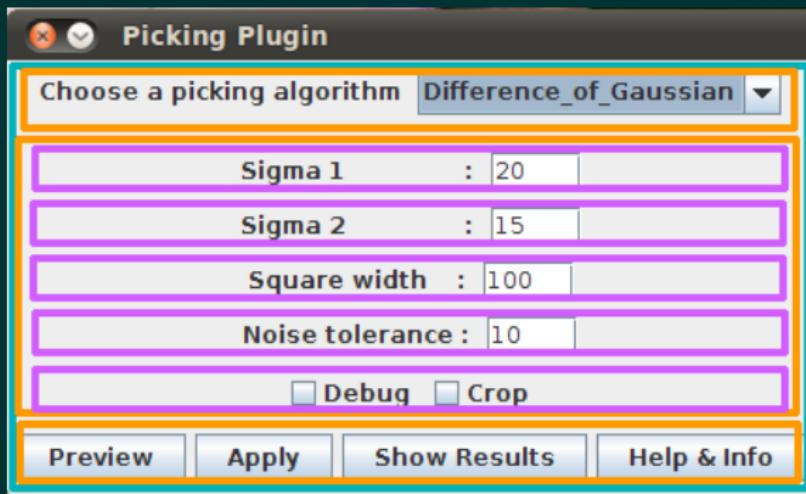


Modularité - Objectif réussi ?

Ajout d'un algorithme en 4 étapes :

- Modification du menu déroulant
- Ajout de l'interface et de la méthode de récupération des paramètres
- Création de la classe de sélection en suivant les modèles implémentés
- Modification de la classe AlgoFactory pour prendre en compte la nouvelle méthode de sélection

Interface Graphique (GUI)



Organisation générale de l'interface

GUI - Récupération des paramètres utilisateurs

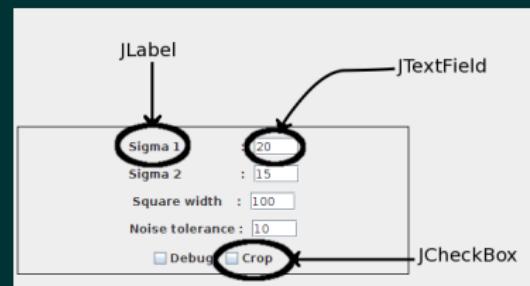
Choix de l'algorithme

- JComboBox



Paramètres propres aux algorithmes

- JTextField
- JCheckBox



Les algorithmes implémentés

3 algorithmes implémentés :

- Extraction de contours par morphologie mathématique
- Corrélation croisée d'images
- Différence de Gaussiennes (DoG)

Schéma général du traitement des images

- Pré-traitement (filtrage)
- Traitement
- Post-traitement (tri des résultats)

Algorithmes - Extraction de contours

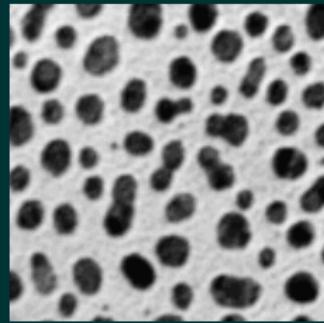


Image avant
traitement

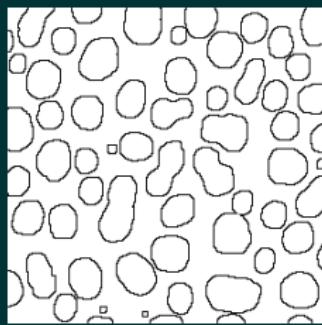


Image résultante

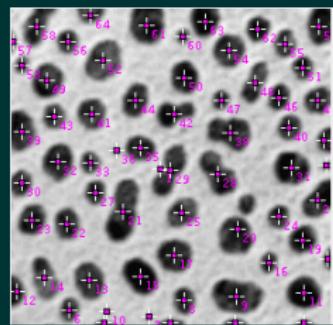


Image avec les
sélections

Algorithmes - Corrélation croisée d'images

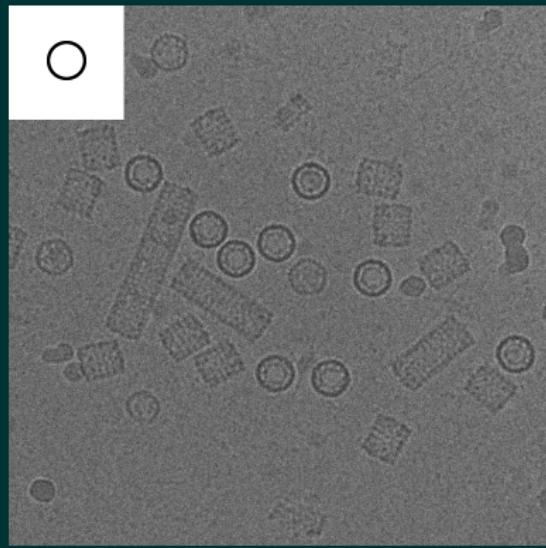


Image pour la corrélation croisée avec la référence

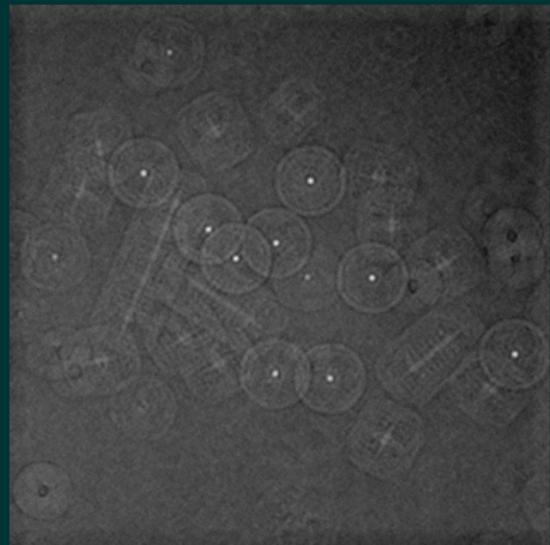
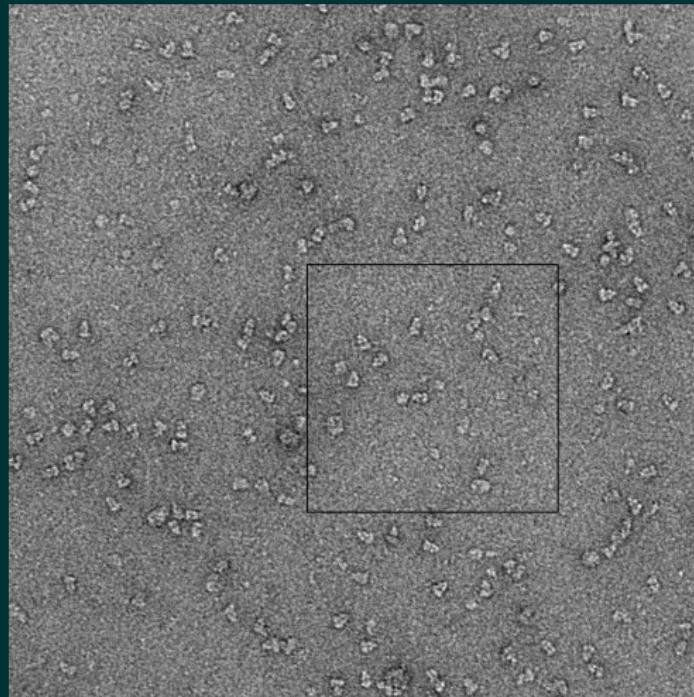


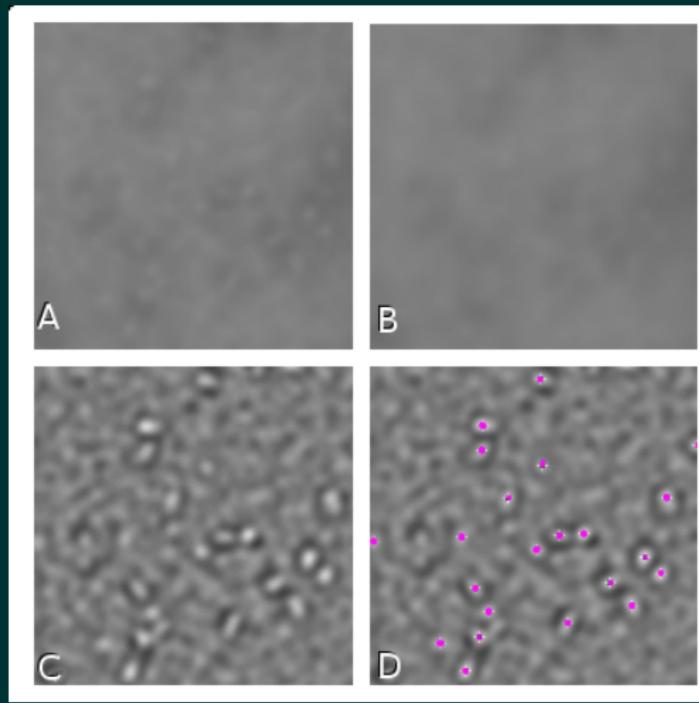
Image résultante

Démonstration avec DoG



Micrographie de protéines transmembranaires

Algorithmes - Les filtres



- A : Filtre Gaussien ($\sigma = 15$)
- B : Filtre Gaussien ($\sigma = 20$)
- C : Résultat de la soustraction
- D : Maxima locaux

Résultats

Statistiques préliminaires de sélection

Images	Variables	DoG	Extraction	Corrélation
Blobs	Vrais Positifs	63	61	X
	Vrais Négatifs	3	0	X
	Faux Positifs	9	2	X
	Faux Négatifs	0	1	X
	Sensibilité	1	0.98	X
	Spécificité	0.25	0	X
Protéines Mb	Vrais Positifs	167	X	X
	Vrais Négatifs	8	X	X
	Faux Positifs	16	X	X
	Faux Négatifs	8	X	X
	Sensibilité	0.95	X	X
	Spécificité	0.5	X	X
Virus	Vrais Positifs	35	X	34
	Vrais Négatifs	7	X	13
	Faux Positifs	43	X	36
	Faux Négatifs	2	X	3
	Sensibilité	0.95	X	0.92
	Spécificité	0.14	X	0.27

Difficultés rencontrées

API ImageJ

Distinction :

- ImagePlus
- ImageProcessor
- ImageStack

GUI

- Gestion des panneaux
- Organisation et tailles des fenêtres

Améliorations 1/2

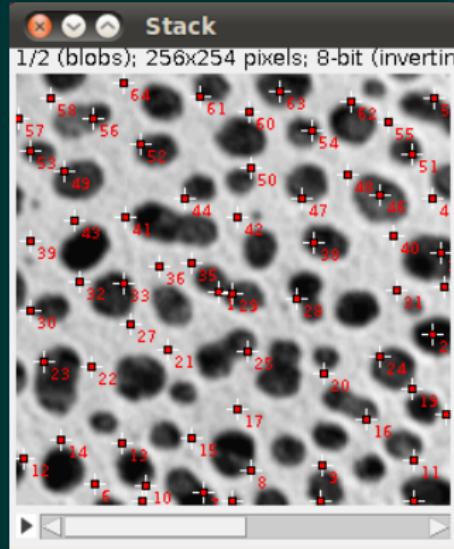
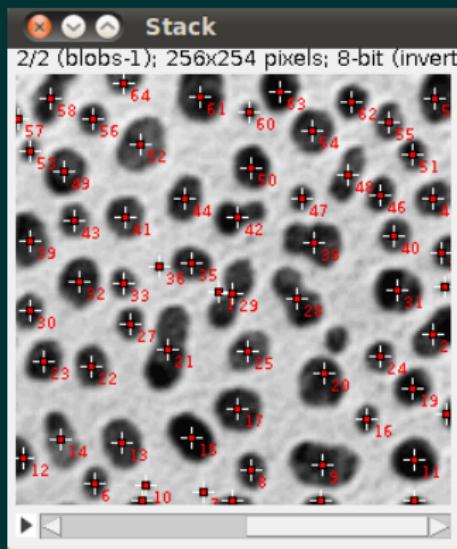
Améliorer

- Le mode Debug
- Continuer le travail pour l'utilisation en Macro
- Le post-traitement en fonction de la taille pour l'élimination des agrégats :
 - Vérification visuelle
 - Système d'apprentissage

Améliorations 2/2

Ajouter

- Afficher les sélections pour chaque image d'une pile
- Contrôle visuel par l'opérateur



Conclusion

Le projet

- Interface facile d'utilisation
- Sélection efficace
- Récupération fonctionnelle des images individuelles et des coordonnées
- Modularité pour l'ajout de nouveaux algorithmes
- Utilisation du logiciel *Eclipse* et du gestionnaire de versions *Git*

L'équipe

- Première expérience de travail en équipe sur un gros projet concluante
- Aperçu de notre futur métier

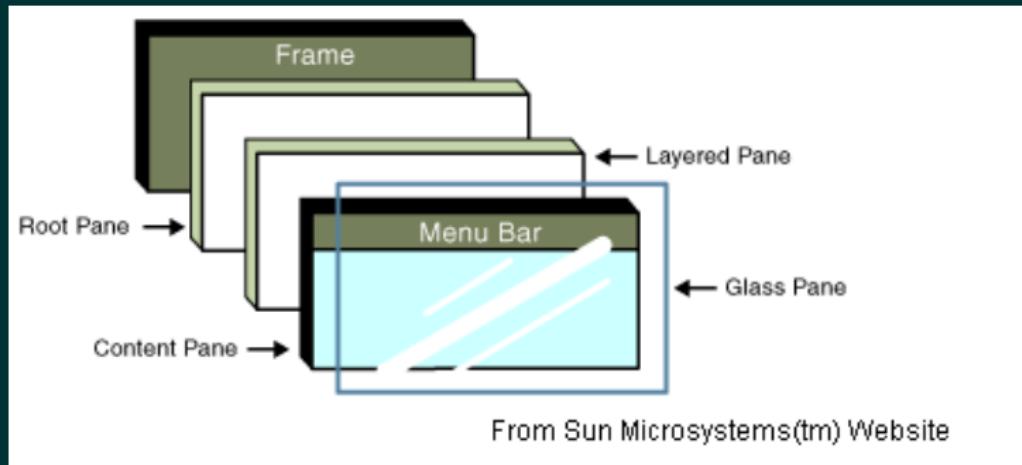
Merci beaucoup pour votre attention

Définitions

- **Vrais Positifs** = particules devant être sélectionnées et qui le sont par l'algorithme
- **Vrais Négatifs** = tout ce qui ne doit pas être sélectionné et qui ne l'est pas
- **Faux Positifs** = tout ce qui ne doit pas être sélectionné mais qui l'est
- **Faux Négatifs** = particules devant être sélectionnées mais qui ne le sont pas
- **Sensibilité** = probabilité qu'une particule devant être piquée le soit
- **Spécificité** = probabilité de ne pas sélectionner ce qui ne doit pas l'être

$$SE = \frac{VP}{VP+FN} \text{ et } SP = \frac{VN}{VN+FP}$$

Organisation d'une interface en Java



Exemple d'organisation de fenêtre