



L'interface du projet NARCISSE



<http://narcisse.toulouse.inra.fr/Beta>

Licence = CeCiLL (pas encore en téléchargement)

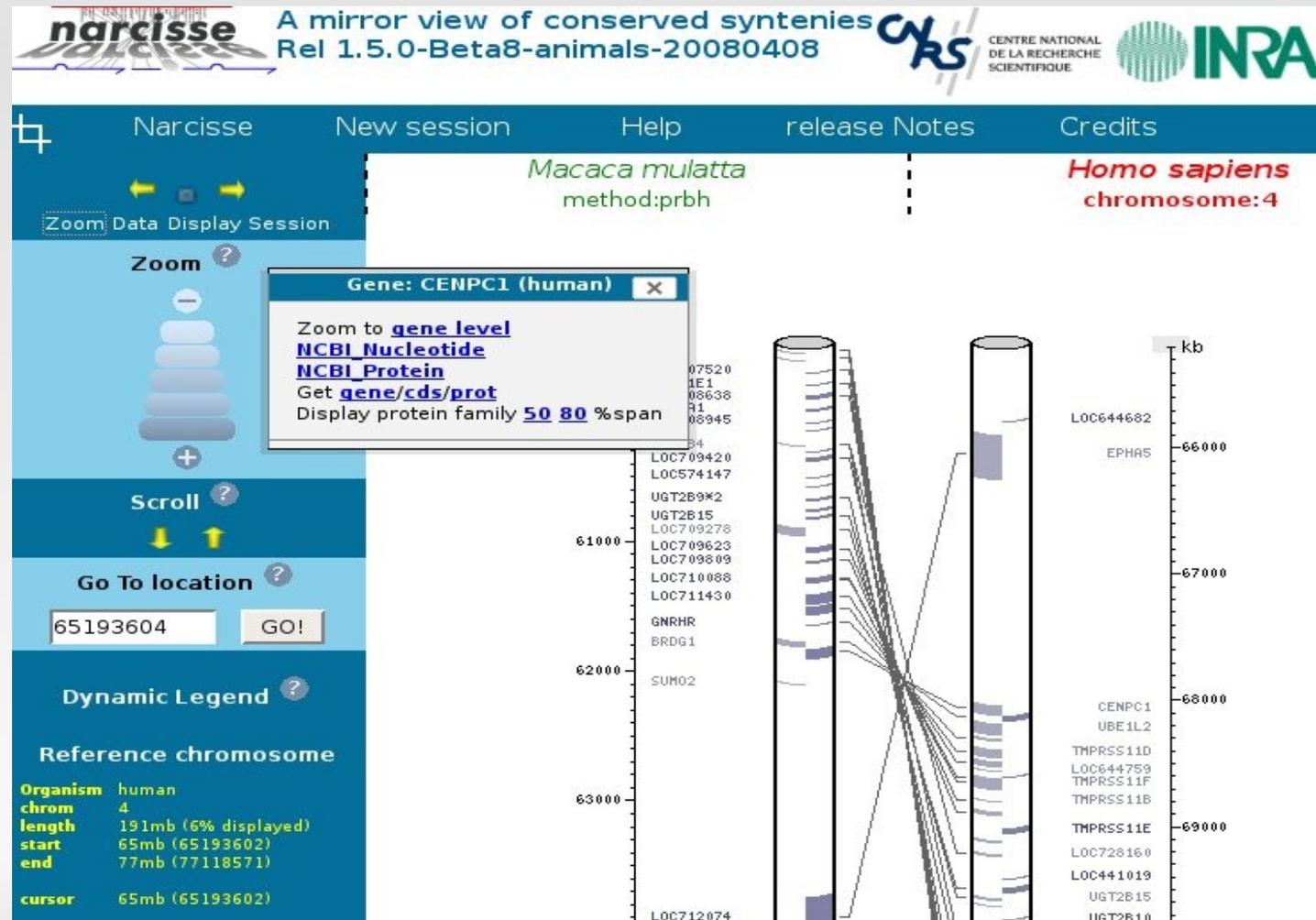
Emmanuel Courcelle LIPM (INRA/CNRS)

Le contexte



Génomique comparative

Comparer
tous les génomes
à
toutes les échelles



Interface web...

- Très courant en bio: rien à installer en local
- Données sur le serveur... 400 Go
- Service public en ligne => visibilité



... Mais **Interface « riche »**

Langages et versions

perl – C – C++

xhtml strict 1.0

css 1.1

javascript 1.5

Séparer structure, présentation, comportement

Choix technologiques (2)

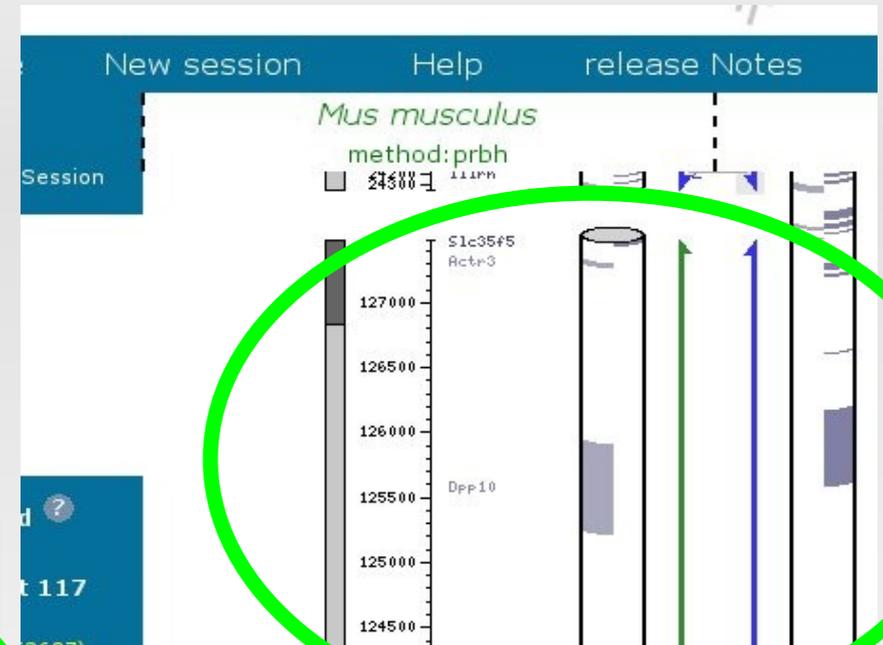
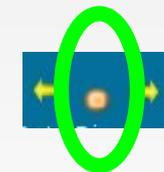


Asynchronous

Javascript

And

Xml

A screenshot of a web application interface for species comparison. It features two input fields for 'Reference and target species', both containing 'Homo Sapiens'. Below these is a section for 'Available organisms' and a 'Comparison method' section with radio buttons for 'glint (nucleic)' and 'prbh (proteic)'. A red error dialog box is overlaid on the interface, containing the text: 'DATA ERROR x', 'comparison Homo Sapiens vs Homo Sapiens not available'. A green oval highlights the error dialog box.

Choix technologiques (3)



Côté client: javascript portable

- **bas niveau:** prototype v 1.6
 - Permet de définir des classes avec héritage



haut niveau: Yahoo User Interface v2.5.2



wforms, objets maison

Firefox, plateforme de développement

- html validator
- firebug:
 - débogage du javascript et des requêtes ajax
 - accès interactif au source html, aux css



Outils pour MSIE

- firebug lite, debugbar



Une fonctionnalité = Un objet javascript

Faire le lien entre javascript et html

- Par les id html, en passant l'id au constructeur
- Les id sont utilisés le moins possible
- A partir d'un élément, on se déplace dans le DOM
- **Dépendance entre html et javascript**

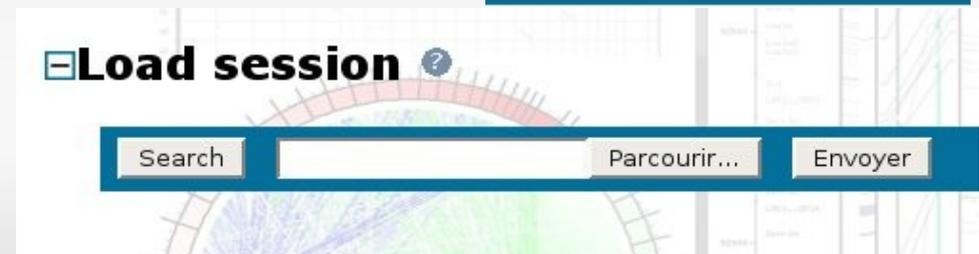
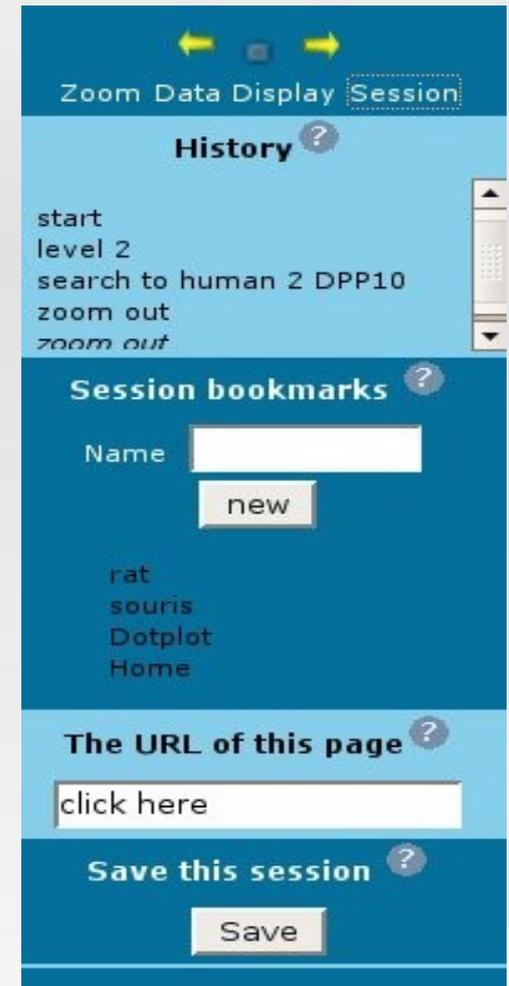


Les évènements

- Fonction `Event.observe` de prototype
 - Permet de gérer les évènements uniformément
- Les handlers d'évènements sont posés par les constructeur des objets
- javascript « discret » (*unobtrusive*)
 - **pas de liens javascript:fonction**

L'historique

- Remplacement des boutons 
- Ecriture d'un objet d'historique
 - On rejoue la requête en gardant les paramètres côté serveur
 - L'utilisateur peut définir des signets
- On peut sauvegarder la session

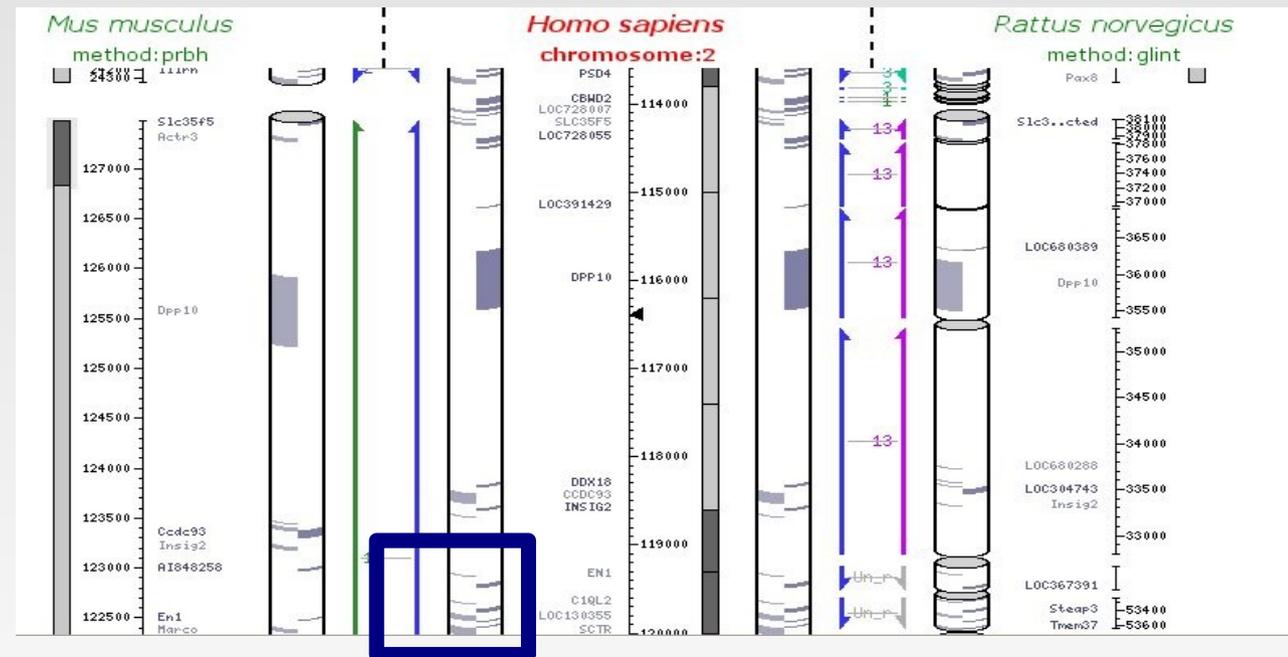


Communication

- client → serveur: requêtes **post**
 - Requetes ordinaires (nouvelle page)
 - Requetes ajax pour mettre à jour le graphique
 - Requetes ajax pour valider des formulaires
- serveur → client: **json**
 - Paramètres de configuration
 - Retour des requêtes de validation

Se rapprocher d'une application locale

- Fonctionnement « à la google map »



Sélection d'objets

Le bilan

- Interface plus agréable
- Base solide pour aller plus loin
- **mais...**
 - Développements lourds (« autre métier »)
 - Support de tous les navigateurs...
 - Demande une machine puissante côté client
 - **Accessibilité ?**