

Projet long

Projet G.L.A.S.S

Gene

List

Analysis using

Shiny

Server

Etudiante :

Coralie CAPRON

Master 2 Bioinformatique

Tuteur de projet :

Costas BOUYIOUKOS

Maître de Conférence

2019/2020

Table des matières

1. INTRODUCTION.....	3
1. MATERIELS ET METHODES	4
1.1. SHINY	4
1.1.1. Principe de Shiny.....	4
1.1.2. Shiny Dashboard.....	4
1.2. CLUSTERPROFILER	4
1.3. L'APPLICATION	5
1.3.1. DONNEES DE L'UTILISATEUR	5
1.3.2. PARAMETRES	5
1.3.3. CLASSIFICATION DES GENES	5
1.3.4. ENRICHISSEMENT DE TERMES GENEONTOLOGY	6
1.3.5. ENRICHISSEMENT DE VOIES KEGG	6
1.3.6. GENE SET ENRICHMENT ANALYSIS	6
1.3.7. ENRICHISSEMENT DE VOIES REACTOME	7
1.3.8. FONCTIONNEMENT GENERAL DE L'APPLICATION	8
3. RESULTATS.....	9
2.1. FONCTIONNEMENT DE L'APPLICATION	9
2.1.1. INPUT DATA	9
2.1.2. PARAMETRES	10
2.1.3. ENRICHISSEMENT DE TERMES GENEONTOLOGY	11
2.1.4. GENE SET ENRICHMENT ANALYSIS	12
2.1.5. ENRICHISSEMENT DE VOIES REACTOME	13
2.1.6. ENRICHISSEMENT DE VOIES KEGG	14
4. CONCLUSION ET PERSPECTIVES	15
5. ANNEXE.....	16
6. REFERENCES.....	18

1. Introduction

Avec l'essor des technologies de séquençage de nouvelle génération (NGS), les séquençages de génomes à haut-débit sont désormais plus accessibles (les tailles des génomes séquencés ont augmenté de l'ordre de 100-fold) (Mardis, ER, 2011). Ces séquençages sont effectués plus rapidement, et à moindre coût (de l'ordre du facteur 20 000) pour le même génome qu'avec de plus anciennes techniques de séquençage (Metzker, ML, 2010). Le volume de données génomiques, et *omiques plus globalement, ne cesse donc d'augmenter avec ce besoin en aval de les analyser.

L'utilisation du séquençage en protéomique, mais aussi des techniques comme les puces microarray ou de ChIP-chip par exemple, résulte en la production d'une liste de gènes d'intérêt (Huang, D., Sherman, B. & Lempicki, 2009). Bien que des pipelines bioinformatiques soient mis à disposition des chercheurs pour filtrer leurs données brutes issues de ce séquençage, cette liste de gènes, qui peut parfois atteindre plusieurs milliers de gènes d'intérêt à analyser, reste compliquée à étudier. Il semble impossible pour les scientifiques d'étudier chacun de ces gènes et d'identifier leur intérêt biologique et leurs interactions manuellement.

L'analyse de ces listes de gènes est simplifiée depuis la création de plusieurs outils destinés à l'extraction de l'annotation fonctionnelle de chacun de ces gènes. On note ainsi plusieurs outils qui font consensus dans l'étude et la classification de ces gènes tels que DAVID (Huang da *et al.* 2009), GSEA (Subramanian *et al.* 2005) et plus récemment clusterProfiler (Guangchuang *et al.* 2012). Ces outils permettent de réaliser différents types d'analyses telles que la visualisation de cette liste de gènes, leurs interactions, les enrichissements et la classification de ces gènes, etc.

Cependant, l'utilisation de ces outils est restreinte pour l'ensemble de la communauté scientifique. En effet, il est nécessaire de posséder des compétences de programmation pour les manipuler, il faut également être en mesure de pouvoir installer et comprendre les librairies nécessaires au bon fonctionnement de ces outils.

Ce projet consiste donc en la mise en place d'une application web permettant l'utilisation de l'un de ces outils, clusterProfiler fourni par le projet bioinformatique Bioconductor, et de ses options au travers d'un environnement implémenté sous R appelé Shiny. Cette application va permettre à l'utilisateur de réaliser, pour une liste de gènes donnée et un organisme référence sélectionné, des enrichissements de gènes ou de voies métaboliques et d'analyser ces résultats grâce à différents outils tels que des barplots ou dotplots par exemple.

1. Matériels et Méthodes

1.1. Shiny

1.1.1. Principe de Shiny

Shiny est un outil qui va permettre de créer une application web interactive de type framework, c'est-à-dire une infrastructure logicielle, implémentée et proposée par Rstudio (Chang *et al.*, 2019). L'intérêt de Shiny est de fournir à l'utilisateur une application manipulable grâce à une interface graphique sans avoir à connaître les langages nécessaires à l'utilisation des outils implémentés dans cette application. Une application Shiny est donc une page web HTML implémentée en R, qu'il est possible de mettre en ligne comme toute page web.

Shiny se compose de deux parties, la première est la partie serveur, celle-ci consiste en l'ajout de tous les processus ou calculs à réaliser lorsque l'utilisateur envoie des informations sur la page web. Ainsi, si l'utilisateur soumet un fichier dans l'application, c'est le bloc serveur qui va se charger de récupérer dynamiquement (c'est-à-dire en temps réel) l'objet fichier, de le lire, de le stocker dans une variable et de le réutiliser ensuite ailleurs dans le bloc.

La deuxième partie est la partie UI (User Interface), c'est donc la partie qui contient les éléments (ou widgets) qui servent au design de l'application mais aussi à la création des éléments interactifs pour l'utilisateur (les boutons par exemple) et enfin à l'affichage des résultats (un graphique par exemple).

L'ensemble serveur et UI forme ainsi l'objet ShinyApp qui est reconnu par R.

1.1.2. Shiny Dashboard

Le package Shiny Dashboard est un package qui permet de structurer graphiquement l'application Shiny. En effet, sans ce package, l'application n'est qu'une page blanche qu'il faut structurer manuellement et dont l'affichage est austère. Shiny Dashboard permet de définir 3 zones dans cette page : Une barre latérale interactive, une zone de titre en haut et une zone d'affichage centrale qu'il est possible de structurer facilement grâce à des widgets (window gadgets).

1.2. ClusterProfiler

ClusterProfiler est un outil permettant d'automatiser la classification de termes biologiques et l'enrichissement de clusters de gènes, c'est-à-dire plus de deux gènes dérivant d'un ancêtre commun après duplications successives (Guangchuang Yu *et al.*, 2012). clusterProfiler est implémenté sous R et a été mis à disposition de la communauté scientifique au travers de la plateforme Bioconductor. Grâce à ce projet Bioconductor, clusterProfiler peut utiliser des données issues de l'annotation par KEGG ou GeneOntology et permet également d'utiliser différents types de génomes de référence pour réaliser ses analyses (Huber *et al.* 2015). Différentes méthodes et fonctions sont ainsi disponibles telles que *groupGO* (classification des gènes), *enrichGO* (enrichissement se basant sur les termes de

GeneOntology) ou *enrichKEGG* (sur les voies métaboliques de la base de données KEGG), *GSEA* (Enrichissement de gènes ayant été préalablement ordonnés grâce à des mesures d'expression telle que le log fold change par exemple).

1.3.L'application

1.3.1. Données de l'utilisateur

La première fenêtre de l'application sur laquelle se situe l'utilisateur est la fenêtre de choix de données à importer. Ce dernier dispose au choix de deux options pour transmettre ses données, soit il souhaite transmettre un fichier contenant une liste de gènes, auquel cas il dispose d'un bouton d'exploration et chargement de fichier, soit il pourra écrire manuellement dans un champ de texte dédié les gènes qu'il souhaite étudier.

A noter que deux types de listes de gènes peuvent être attendus, les listes non ordonnées et les listes ordonnées. Les listes non ordonnées ne possèdent pas de log2 fold change (métrique de calcul de l'expression génique) et ne peuvent donc pas être triées. Ces listes ont la caractéristique de ne posséder qu'une seule colonne lorsqu'elles sont transmises dans un fichier ou manuellement. A l'inverse, les listes ordonnées possèdent une mesure associée à chaque gène qui va permettre le tri en fonction de cette mesure.

1.3.2. Paramètres

Après avoir soumis ses données à analyser, l'utilisateur est invité *via* un message pop-up à sélectionner dans l'onglet « paramètres », les paramètres qu'il souhaite utiliser pour réaliser les différentes analyses. Ce dernier doit ainsi choisir parmi une liste de génomes de références utilisables par clusterProfiler celui auquel appartient sa liste de gènes, par défaut ce dernier étant le génome humain. De plus, il doit préciser le type d'identifiants de gènes utilisé dans sa liste de gènes parmi un choix tel que les identifiants de gènes ENSEMBL ou ENTREZ. En fonction de l'identifiant précisé, une conversion sera réalisée vers les identifiants nécessaires pour les fonctions appelées par la suite.

Les choix disponibles par la suite concernent les enrichissements des termes Gene Ontology (GO), l'utilisateur peut choisir de réaliser un seul enrichissement (*i.e* : « Biological Process ») ou de tous les réaliser en les cochant tous.

Enfin, la possibilité de télécharger son propre « Univers » pour le GSEA (Gene Set Enrichment Analysis) est proposé si l'utilisateur souhaite utiliser un autre univers que celui par défaut. L'univers consiste en l'ensemble des gènes qui seront utilisés pour mener à bien cette analyse.

1.3.3. Classification des gènes

La classification des gènes consiste en la caractérisation des fonctionnalités biologiques associées pour chaque gène de la liste et dans un domaine de la base de données Gene Ontology (Ashburner, *et*

al.,2011). La fonction de clusterProfiler qui permet de rechercher ces processus biologiques sous-jacent est *groupGO*, cette fonction prend en arguments un vecteur d'identifiants de gènes ainsi que le type d'identifiant utilisé, le génome de référence contre lequel sera effectuée la comparaison, mais aussi le type de domaine de GeneOntology dans lequel faire la classification, qui sont au donnés par l'utilisateur dans l'onglet paramètres de l'application.

1.3.4. Enrichissement de termes GeneOntology

Un enrichissement de gènes par termes GeneOntology consiste à voir dans une liste de gènes, ceux qui sont sur-représentés. Cet enrichissement consiste donc à sélectionner une liste de gènes d'intérêt, puis à rechercher les annotations associées pour chaque domaine et enfin de déterminer avec un test statistique les gènes les plus enrichis (les plus représentés). La fonction de clusterProfiler qui permet de réaliser l'enrichissement de gènes par termes GO est *enrichGO*, cette fonction nécessite des arguments similaires à la fonction *groupGO* et utilise un test hypergéométrique. Cependant, il faut préciser également en arguments le test statistique utilisé ainsi que la valeur seuil de p-value à retenir ou la taille minimale et maximale des gènes annotés par GeneOntology et l'univers dans lequel réaliser l'analyse. L'univers correspondant à l'ensemble des gènes du génome étudié.

1.3.5. Enrichissement de voies KEGG

Un enrichissement de gènes par les voies métaboliques comprises dans la base de données KEGG se base sur le même principe qu'un enrichissement des termes GO. La différence étant que la base de données KEGG répertorie les voies métaboliques afin de connaître la fonction des gènes (Kanehisaa & Goto, 2000). La fonction de clusterProfiler permettant de réaliser l'enrichissement de gènes par les voies de KEGG est *enrichKEGG* et nécessite les mêmes arguments que *enrichGO*. Toutefois il est important de préciser que l'organisme choisi par l'utilisateur doit correspondre à l'un des organismes supportés par la fonction *enrichKEGG* (Cf Annexe 1 pour observer un échantillon des organismes pris en charge).

1.3.6. Gene Set Enrichment Analysis

La technique GSEA (Gene Set Enrichment Analysis) permet, à partir d'une liste de gènes d'intérêt ordonnée, de les regrouper selon différents facteurs tels que la présence d'une même voie métabolique associée à différents gènes puis de calculer un score permettant d'évaluer la quantité de gènes associés à ce facteur et leur niveau d'expression correspondant. Elle permet ainsi d'évaluer plus finement la moindre variation d'expression des gènes qu'une méthode avec une valeur seuil (Subramanian *et al.* 2005). La fonction clusterProfiler qui permet de réaliser le GSEA est la fonction *GSEA*, elle nécessite en argument une liste d'identifiants de gènes au type SYMBOL, une taille minimale des sets de gènes générés à analyser ainsi que les termes à utiliser pour créer le GSEA. Pour ce faire, la fonction *msigdb* est employée pour générer, en fonction de l'organisme sélectionné par l'utilisateur, l'ensemble des

termes (nom des gènes sous différents formats, base de données dans lequel il est décrit, etc.) appartenant à cette espèce. La base de données Msigdb contenant l'ensemble des gènes annotés nécessaires au bon fonctionnement de GSEA.

A noter que la technique GSEA est spécifique aux listes de gènes ayant été ordonnées grâce à des métriques d'expression des gènes préalablement calculées. Une autre technique d'enrichissement de gènes non-ordonnés (appelé enrichissement universel) est disponible et a été réalisée. Cette fonction se nomme *enricher*, fonctionne comme *GSEA* mais ne nécessite pas de liste de gènes ordonnés en argument. Elle utilise également la fonction *msigdb* pour générer l'ensemble des termes à analyser pour l'espèce choisie, ainsi que la taille minimale et maximale des gènes annotés à tester. Cette taille pourra être modifiée en temps réel après affichage des résultats grâce à une échelle graduée dynamique.

1.3.7. Enrichissement de voies Reactome

La base de données Reactome est dédiée aux voies biologiques de plusieurs espèces avec pour espèce la plus documentée, l'Homme (Croft *et al.*, 2011). Cette base de données permet ainsi la visualisation, l'interprétation et l'analyse de ces différentes voies biologiques. Réaliser un enrichissement spécifiquement sur ces voies biologiques va permettre, comme pour l'enrichissement GSEA de créer des « sets » de gènes en fonction de leurs voies biologiques. La fonction qui permet cet enrichissement est *enrichPathway*, elle prend en argument une liste d'identifiants de gènes au format ENTREZ.

1.3.8. Fonctionnement général de l'application

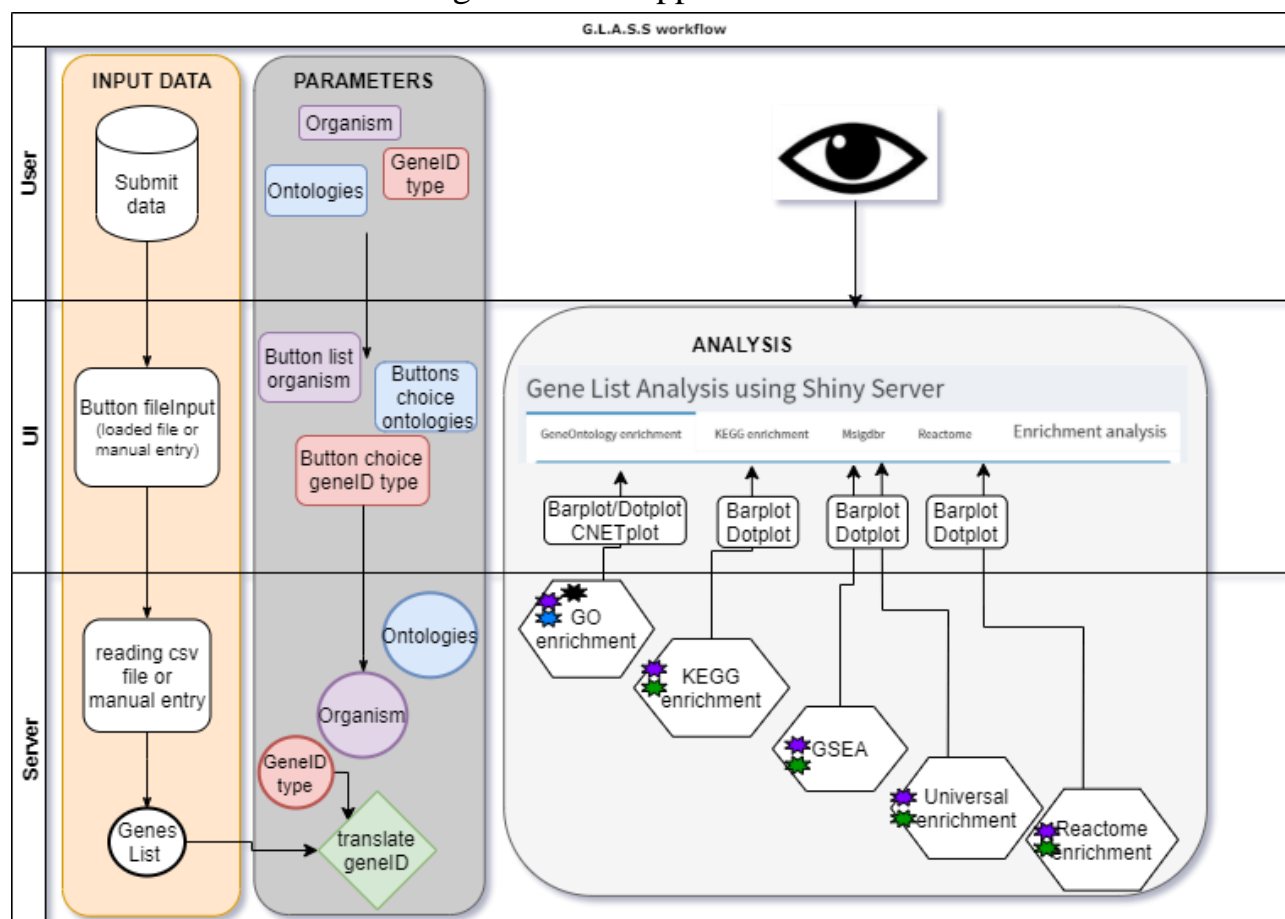


Figure 1: Workflow de l'application Shiny, les étoiles colorées dans les hexagones sont les différents paramètres passés en argument. i.e : étoile noire = liste de gènes, bleue = terme(s) de GO choisi(s)

Deux fichiers tests de listes de gènes ont été générés pour tester l'application, ces fichiers ne possèdent que très peu de gènes afin d'alléger les calculs réalisés par l'application. Le premier contient une liste non ordonnée de gènes (le gène BRCA2 et le gène IQSEC1), le deuxième contient une liste ordonnée de gènes (BRCA2, IQSEC1, CASC17, CT69, CASC22).

3. Résultats

2.1. Fonctionnement de l'application

2.1.1. Input data

The screenshot displays the GLASS application interface. On the left, a dark sidebar contains navigation links: 'Import data' (selected), 'Enrichments parameters', 'Analysis', and 'About'. The main content area is titled 'Genes List Analysis using Shiny Server'. It includes a 'Choose a file' section with a 'Browse...' button and a file named 'test_gene_ensembl.csv' with an 'Upload complete' status. Below this is a text input area titled '...Or write your list of genes' containing 'geneID1' and 'geneID2'. A note below the text area says 'Write an ordered or unordered list of genes lines by lines with the same separator'. A blue notification box at the bottom right says 'Now choose your enrichment in the sidebar'.

Figure 2: Application GLASS, onglet d'importation de données

La première page de l'application sur laquelle est dirigé en premier lieu l'utilisateur est la page d'importation des données (Onglet de la barre latérale « import data »)(Cf figure2). Ce dernier est invité à soumettre soit une liste de gènes au format csv soit à écrire manuellement cette liste dans la zone de texte correspondante. Une fois que l'un de ces choix est réalisé, une petite fenêtre temporaire s'affiche afin de l'inviter à changer d'onglet dans la barre latérale pour définir ensuite les paramètres des enrichissements à réaliser.

A noter que si l'utilisateur transmet une liste ordonnée ou non ordonnée, le serveur de l'application détectera si la liste de gènes fait une colonne (liste non ordonnée) ou 2 colonnes (liste ordonnée) et affichera en conséquence des fenêtres de dialogue lors des analyses si des enrichissements ne peuvent être réalisés (*i.e* : GSEA avec une liste non ordonnée)

2.1.2. Paramètres

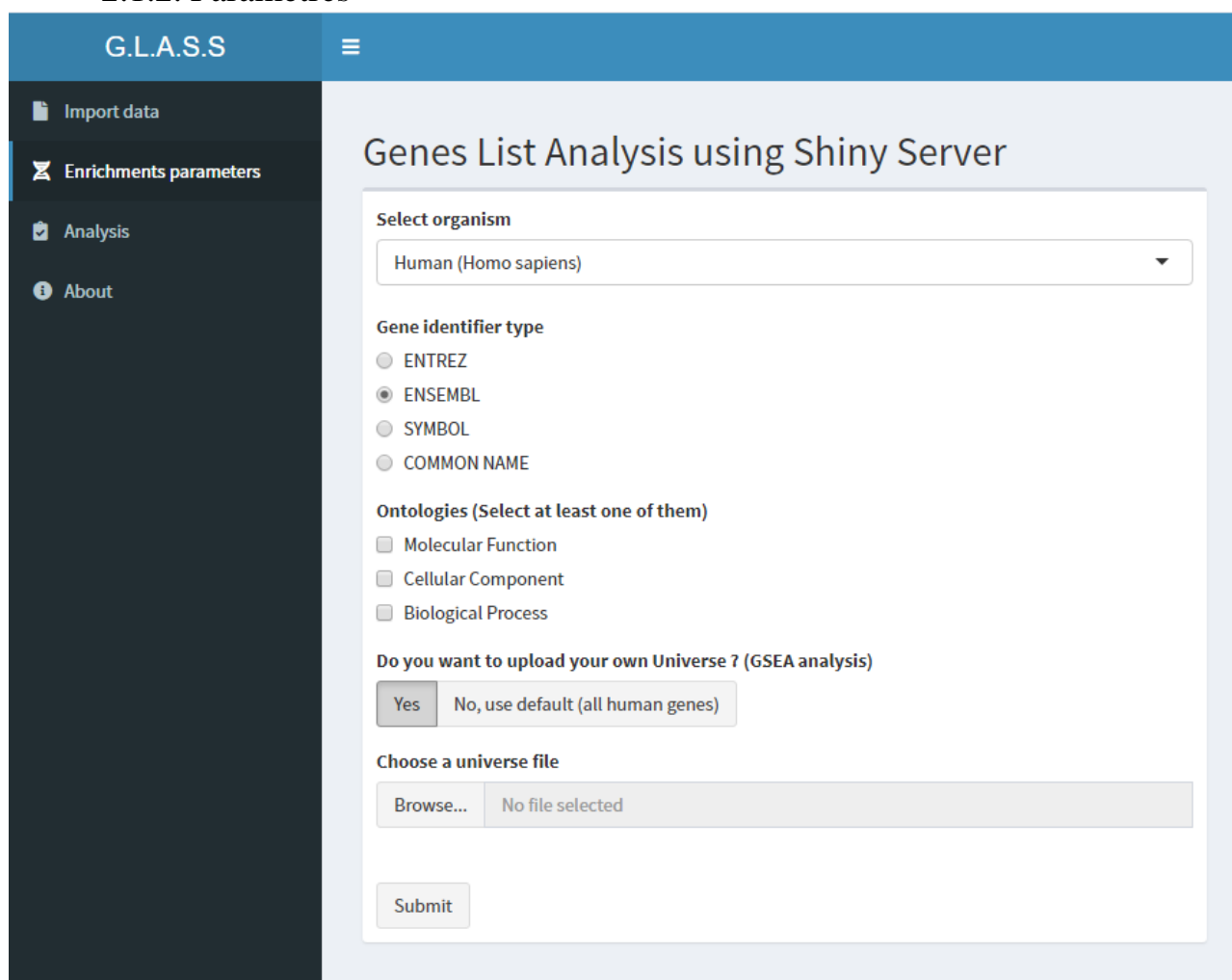


Figure 3: Application GLASS : onglet de paramètres

Le deuxième onglet vers lequel est amené l'utilisateur est celui du choix des paramètres pour les enrichissements (Cf figure 3). Le premier choix à réaliser est celui de l'organisme modèle sur lequel se baser. Parmi les organismes proposés sont ainsi disponibles des vertébrés (par défaut l'Humain), mais aussi des Eucaryotes (*i.e* : la drosophile), un procaryote (*Escherichia coli*) ou une plante (*Arabidopsis thaliana*) (Cf figure 4)

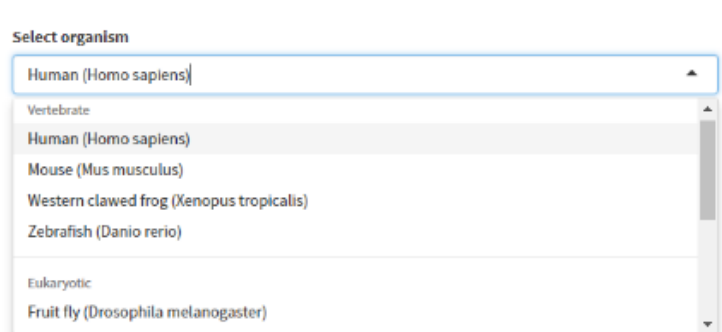


Figure 4: Organismes disponibles dans l'application

Le deuxième choix permet d'indiquer au serveur quel type d'identifiant de gènes est utilisé dans la liste. Le troisième choix consiste en les ontologies à choisir pour réaliser les enrichissements par

GeneOntology. Enfin, pour la technique GSEA, bien que par défaut l'univers soit l'ensemble des gènes humains, l'utilisateur peut définir son propre univers en important un fichier csv correspondant.

2.1.3. Enrichissement de termes GeneOntology



Figure 5: Application GLASS : Onglet analyse, visualisation de l'enrichissement GO

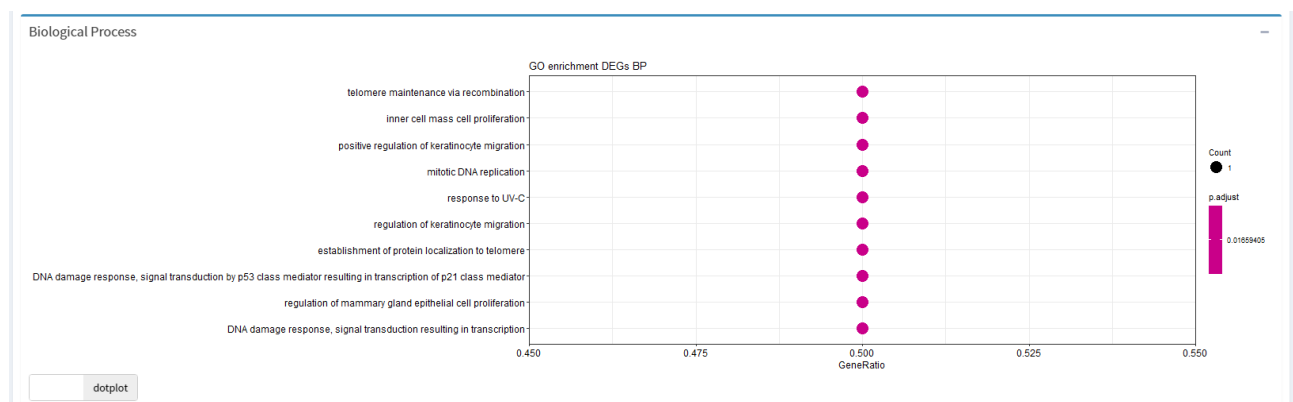


Figure 6: Application GLASS : Onglet analyse, visualisation de l'enrichissement GO, dotplot

Le premier onglet disponible est celui de l'enrichissement de gènes par termes GO. A noter que cette analyse se lance automatiquement lorsque l'utilisateur clique dans la barre latérale sur le bouton « Analysis ». La fenêtre se compose en 3 fenêtres (ou blocs) rétractables (Cf figure 5, encadré rouge) correspondants aux 3 termes de GeneOntology (Biological Process, Molecular Function, Cellular Component). Chaque bloc possède deux graphs, le premier graph par défaut, est le barplot de l'enrichissement, ici les gènes d'intérêt sont principalement impliqués dans des processus du développement, de la reproduction, etc. L'affichage de ce même enrichissement est également disponible en dotplot en appuyant sur le bouton dynamique « switch » (Cf figure 5 encadré bleu puis figure 6 pour le dotplot), cet affichage est pertinent lorsque la liste de gènes est conséquente.

Le deuxième graph affiché par défaut dans le bloc est un CNETplot, ce dernier permet d'informer l'utilisateur sur les relations entre les gènes et les termes de GeneOntology sous forme de réseau. Ici, par exemple le gène BRCA2 est lié en étoile à des termes tels que la réplication de l'ADN ou la réponse cellulaire après des dommages à l'ADN.

2.1.4. Gene Set Enrichment Analysis

Dans le cas du fichier test, l'enrichissement GSEA n'a pas fonctionné, ceci se produit notamment lorsque la liste de gènes est trop succincte et qu'aucun terme enrichi n'est observé. Dans ce cas, un nouveau message d'erreur s'affiche afin d'informer l'utilisateur. Cependant l'enrichissement universel (utilisant plusieurs bases de données) effectué par la fonction *enricher* est lui disponible (Cf *figure 7*). Les résultats sont peu lisibles mais 2 termes sont distincts en raison de leurs valeurs, ces deux résultats d'enrichissement sont obtenus sur la base de données GEO (Gene Expression Omnibus) car deux numéros d'accès sont lisibles : GSE8921 et GSE36392. De plus tous les autres clusters qui ont servi aux enrichissements sont au même niveau (Cf *figure 8, les points sont alignés*) ce qui signifie qu'ils sont très similaires entre eux.



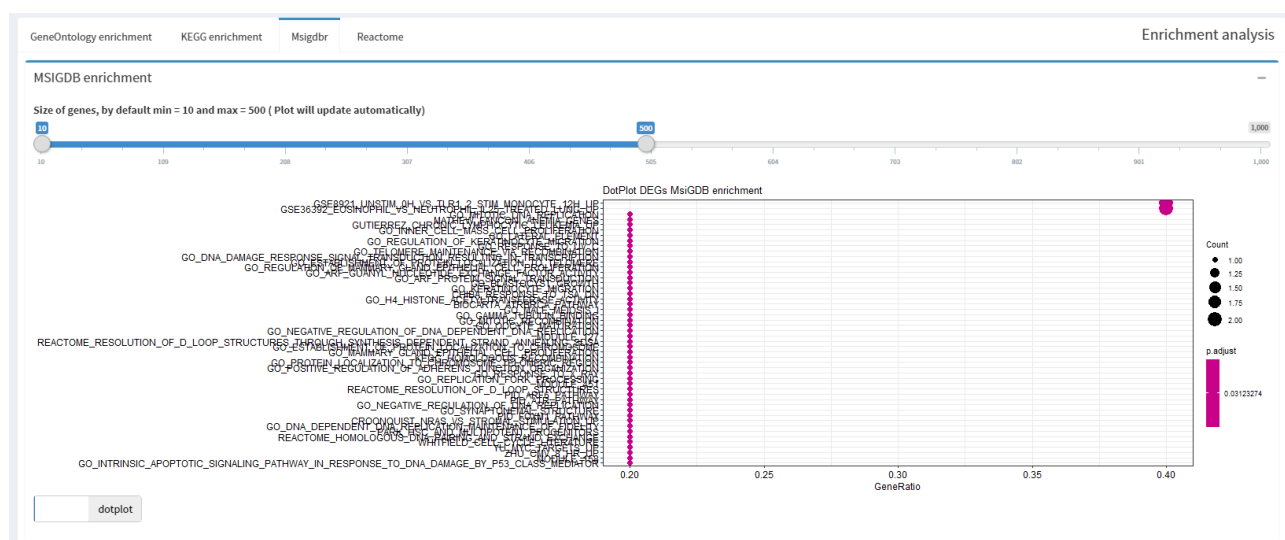


Figure 8: Application GLASS : Onglet analyse, visualisation de l'enrichissement universel, dotplot

2.1.5. Enrichissement de voies Reactome

Gene List Analysis using Shiny Server

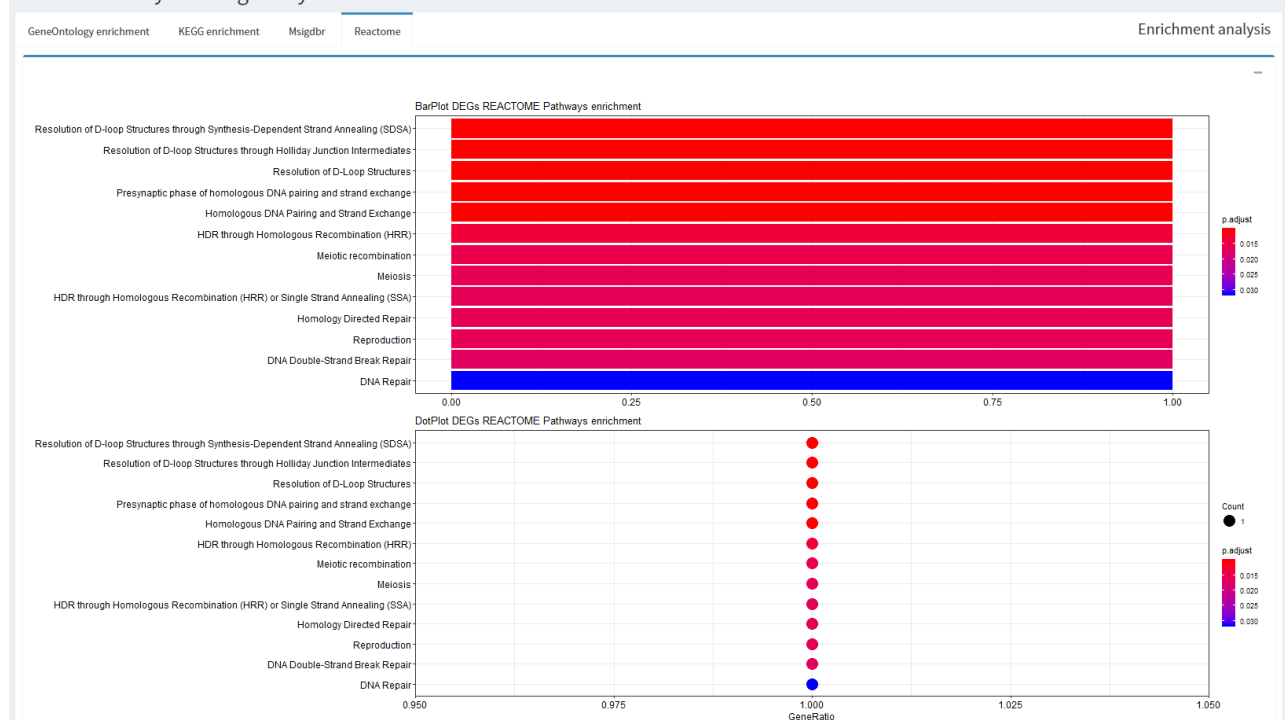


Figure 9: Application GLASS : Onglet analyse, visualisation de l'enrichissement REACTOME

Deux visualisations sont disponibles pour l'enrichissement réalisé sur la base de données Reactome. Le barplot et dotplot. L'enrichissement réalisé sur la base de données Reactome ne montre pas de set d'enrichissement sortant du lot des autres car tous les enrichissements ont une valeur de 1 (Cf figure 9).

2.1.6. Enrichissement de voies KEGG

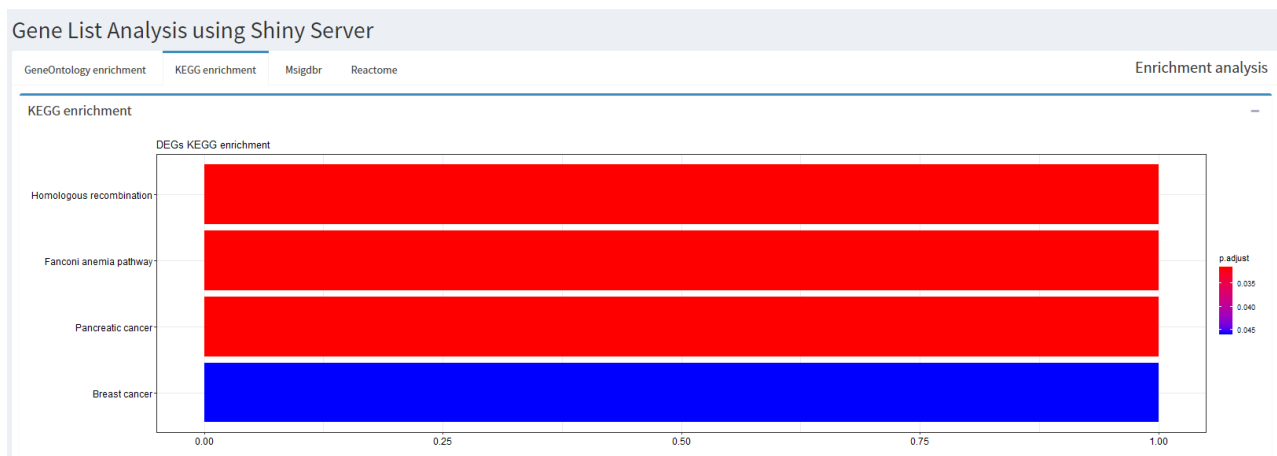


Figure 10: Application GLASS : Onglet analyse, visualisation de l'enrichissement KEGG

L'enrichissement des voies métaboliques de la base de données KEGG est visualisable grâce à un barplot (*Cf figure 10*). Dans le cas du fichier test, les voies biologiques ayant été enrichies de manière conséquente sont les voies de certains cancers (sein et du pancréas), de la recombinaison homologue et de l'anémie. Tous ces résultats sont cohérents car les gènes du fichier test contenant une liste ordonnée sont respectivement le gène du cancer du sein (BRCA2), du cancer pancréatique (CASC17, CT69 et CASC22 sont impliqués dans d'autres types de cancer notamment testiculaire pour CT69).

Un autre enrichissement est disponible en enrichissant *via* des modules de KEGG, cependant il est possible que la fonction ne trouve pas de module pour enrichir, ce qui est le cas ici. Un message d'erreur s'affiche donc (*Cf annexe 2*).

4. Conclusion et perspectives

L'application G.L.A.S.S développée sous l'environnement R Shiny permet à un utilisateur n'ayant pas de notions de programmation d'analyser une liste de gènes d'intérêt afin d'obtenir diverses informations biologiques. Cette application comprend des options laissant le choix à l'utilisateur pour paramétrer les enrichissements qu'il souhaite effectuer. L'affichage dynamique permet également de changer en temps réel de type d'affichage (barplot ou dotplot) pour avoir une meilleure précision pour son analyse des enrichissements réalisés. Sont ainsi implémenté, avec l'aide de l'équipe « Epigenetics And Cell Fate » de l'UMR UMR7216, une classification des gènes, un enrichissement par termes GeneOntology ou par les voies métaboliques KEGG. La possibilité d'analyser une liste de gènes ordonnée ou non permet également de réaliser des analyses tels que le GSEA ou simplement un enrichissement universel utilisant plusieurs bases de données. Enfin un enrichissement se basant sur la base de données Reactome est disponible.

L'ensemble de ces enrichissements permet à l'utilisateur d'obtenir des informations biologiques complètes et automatisées sur une liste de gènes d'intérêt sans avoir à maîtriser l'ensemble des notions de programmation nécessaire au bon fonctionnement de R et des librairies qui lui sont associées.

Cependant, l'application développée possède un défaut majeur lié à la librairie clusterProfiler et qui est son temps d'exécution. L'implémentation de l'application a été optimisée en se basant sur l'interaction avec l'utilisateur. Ainsi les calculs des différents enrichissements seront réalisés uniquement lorsque l'utilisateur sélectionne les onglets correspondants, mais le chargement initial des enrichissements (la première fois que l'utilisateur interagit avec les onglets des analyses) reste toutefois de l'ordre de plusieurs minutes. Des améliorations seraient également réalisables en ajoutant des visualisations telles que les valeurs des fold change dans les CNETplot, ou utiliser par exemple la fonction *enrichMap* de clusterProfiler, qui permet de visualiser un graph des clusters après enrichissement des voies métaboliques afin de voir les interactions entre eux.

Le code implémenté lors de ce projet sera disponible temporairement à l'adresse :

<https://github.com/parisepigenetics/shiny-your-gene-list>

<https://parisepigenetics.github.io/shiny-your-gene-list/>

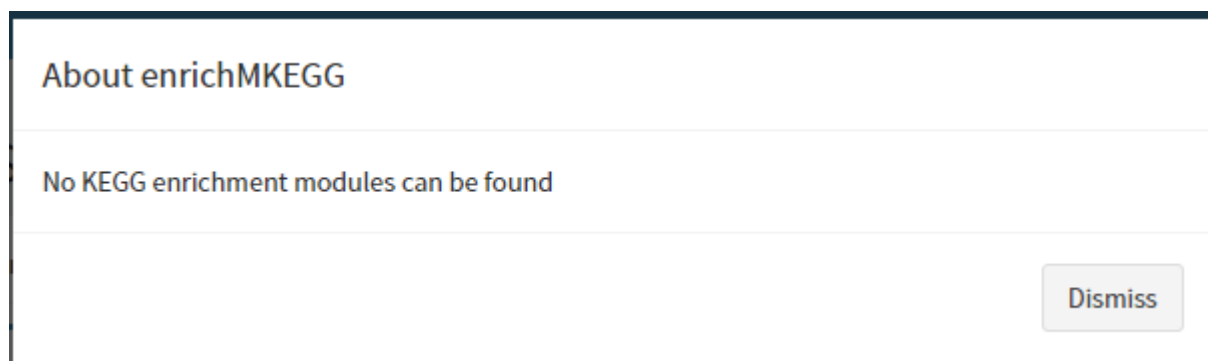
5. Annexe

Annexe 1 : Base de données des organismes modèles disponibles pour *enrichKEGG*

Eukaryotes

Category		Organisms	Source
Mammals	hsa	Homo sapiens (human)	RefSeq
	ptr	Pan troglodytes (chimpanzee)	RefSeq
	pps	Pan paniscus (bonobo)	RefSeq
	ggo	Gorilla gorilla gorilla (western lowland gorilla)	RefSeq
	pon	Pongo abelii (Sumatran orangutan)	RefSeq
	nle	Nomascus leucogenys (northern white-cheeked gibbon)	RefSeq
	mcc	Macaca mulatta (rhesus monkey)	RefSeq
	mcf	Macaca fascicularis (crab-eating macaque)	RefSeq
	csab	Chlorocebus sabaeus (green monkey)	RefSeq
	rro	Rhinopithecus roxellana (golden snub-nosed monkey)	RefSeq
	rbb	Rhinopithecus bieti (black snub-nosed monkey)	RefSeq
	cjc	Callithrix jacchus (white-tufted-ear marmoset)	RefSeq
	sbq	Saimiri boliviensis boliviensis (Bolivian squirrel monkey)	RefSeq
	mmu	Mus musculus (mouse)	RefSeq
	mcal	Mus caroli (Ryukyu mouse)	RefSeq
	mpah	Mus pahari (shrew mouse)	RefSeq
	rno	Rattus norvegicus (rat)	RefSeq
	mun	Meriones unguiculatus (Mongolian gerbil)	RefSeq
	cge	Cricetulus griseus (Chinese hamster)	RefSeq
	ngi	Nannospalax galili (Upper Galilee mountains blind mole rat)	RefSeq
	hgl	Heterocephalus glaber (naked mole rat)	RefSeq
	ccan	Castor canadensis (American beaver)	RefSeq
	ocu	Oryctolagus cuniculus (rabbit)	RefSeq
	tup	Tupaia chinensis (Chinese tree shrew)	RefSeq
	cfa	Canis familiaris (dog)	RefSeq
	vvp	Vulpes vulpes (red fox)	RefSeq
	aml	Ailuropoda melanoleuca (giant panda)	RefSeq
	umr	Ursus maritimus (polar bear)	RefSeq
	uah	Ursus arctos horribilis	RefSeq
	oro	Odobenus rosmarus divergens (Pacific walrus)	RefSeq
	elk	Enhydra lutris kenyoni (northern sea otter)	RefSeq
	fca	Felis catus (domestic cat)	RefSeq
	ptg	Panthera tigris altaica (Amur tiger)	RefSeq
	ppad	Panthera pardus (leopard)	RefSeq
	aju	Acinonyx jubatus (cheetah)	RefSeq
	bta	Bos taurus (cow)	RefSeq
	bom	Bos mutus (wild yak)	RefSeq
	biu	Bos indicus (zebu cattle)	RefSeq
	bbub	Bubalus bubalis (water buffalo)	RefSeq
	chx	Capra hircus (goat)	RefSeq
	oas	Ovis aries (sheep)	RefSeq
	ssc	Sus scrofa (pig)	RefSeq
	cfr	Camelus ferus (Wild Bactrian camel)	RefSeq
	cdk	Camelus dromedarius (Arabian camel)	RefSeq
	bacu	Balaenoptera acutorostrata scammoni (minke whale)	RefSeq
	lve	Lipotes vexillifer (Yangtze River dolphin)	RefSeq

Annexe 2 : Message d'erreur lorsque l'enrichissement ne peut aboutir



6. Références

- Ashburner, Michael, Catherine A. Ball, Judith A. Blake, David Botstein, Heather Butler, J. Michael Cherry, Allan P. Davis, et al. 2000. « Gene Ontology: tool for the unification of biology ». *Nature genetics* 25 (1): 25-29. <https://doi.org/10.1038/75556>.
- Huang, Da Wei, Brad T. Sherman, et Richard A. Lempicki. 2009. « Systematic and Integrative Analysis of Large Gene Lists Using DAVID Bioinformatics Resources ». *Nature Protocols* 4 (1): 44-57. <https://doi.org/10.1038/nprot.2008.211>.
- Huber, W., V.J. Carey, R. Gentleman. «Orchestrating high-throughput genomic analysis with Bioconductor». *Morgan Nature Methods*, 2015:12, 115.
- Kanehisa, Minoru, et Susumu Goto. 2000. « KEGG: Kyoto Encyclopedia of Genes and Genomes ». *Nucleic Acids Research* 28 (1): 27-30.
- Mardis, Elaine R. 2011. « A Decade's Perspective on DNA Sequencing Technology ». *Nature* 470 (7333): 198-203. <https://doi.org/10.1038/nature09796>.
- Metzker, Michael L. 2010. « Sequencing Technologies - the next Generation ». *Nature Reviews. Genetics* 11 (1): 31-46. <https://doi.org/10.1038/nrg2626>.
- Subramanian, Aravind, Pablo Tamayo, Vamsi K. Mootha, Sayan Mukherjee, Benjamin L. Ebert, Michael A. Gillette, Amanda Paulovich, et al. 2005. « Gene Set Enrichment Analysis: A Knowledge-Based Approach for Interpreting Genome-Wide Expression Profiles ». *Proceedings of the National Academy of Sciences of the United States of America* 102 (43): 15545-50. <https://doi.org/10.1073/pnas.0506580102>.
- « Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources. - PubMed - NCBI ». s. d. Consulté le 16 janvier 2020. <https://www.ncbi.nlm.nih.gov/pubmed/19131956?dopt=Abstract>.
- Wetterstrand, KA. s. d. « DNA Sequencing Costs: Data ». Genome.Gov. Consulté le 11 janvier 2020. <https://www.genome.gov/about-genomics/fact-sheets/DNA-Sequencing-Costs-Data>.
- Winston Chang, Joe Cheng, JJ Allaire, Yihui Xie and Jonathan McPherson (2019). shiny: Web Application Framework for R
- Yu, Guangchuang, Li-Gen Wang, Yanyan Han, et Qing-Yu He. 2012. « clusterProfiler: an R Package for Comparing Biological Themes Among Gene Clusters ». *OMICS: a Journal of Integrative Biology* 16 (5): 284-87. <https://doi.org/10.1089/omi.2011.0118>.