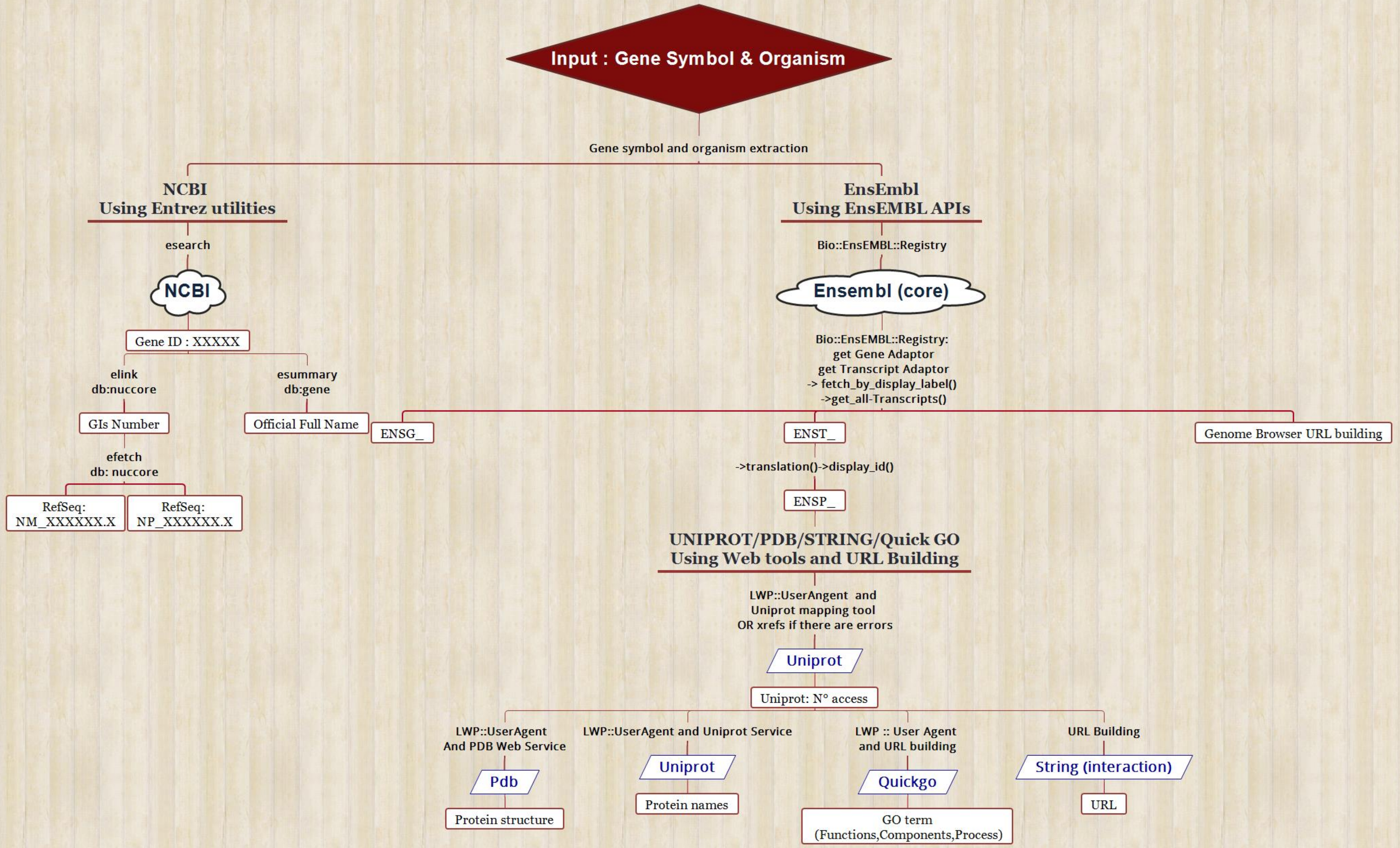
COSMOS 1.2:

SCRIPT FOR AGGREGATING ANNOTATIONS

Bioinformatics Analysis and Genome Annotation

Annthomy GILLES

Friday 4th March 2016


Input : Gene Symbol & Organism
Gene symbol and organism extraction
NCBI
Using Entrez utilities
esearch
NCBI
Gene ID : XXXXX
elink
db:nuccore
esummary
db:gene
GIs Number
Official Full Name
efetch
db: nuccore
RefSeq:
NM_XXXXXX.X
RefSeq:
NP_XXXXXX.X
EnsEmbl
Using EnsEMBL APIs
Bio::EnsEMBL::Registry
Ensembl (core)
Bio::EnsEMBL::Registry:
get Gene Adaptor
get Transcript Adaptor
-> fetch_by_display_label()
->get_all-Transcripts()
ENSG_
ENST_
Genome Browser URL building
->translation()->display_id()
ENSP_
UNIPROT/PDB/STRING/Quick GO
Using Web tools and URL Building
LWP::UserAngent and
Uniprot mapping tool
OR xrefs if there are errors
Uniprot
Uniprot: N° access
LWP::UserAgent
And PDB Web Service
LWP::UserAgent and Uniprot Service
LWP :: User Agent
and URL building
URL Building
Pdb
Uniprot
Quickgo
String (interaction)
Protein structure
Protein names
GO term
(Functions,Components,Process)
URL

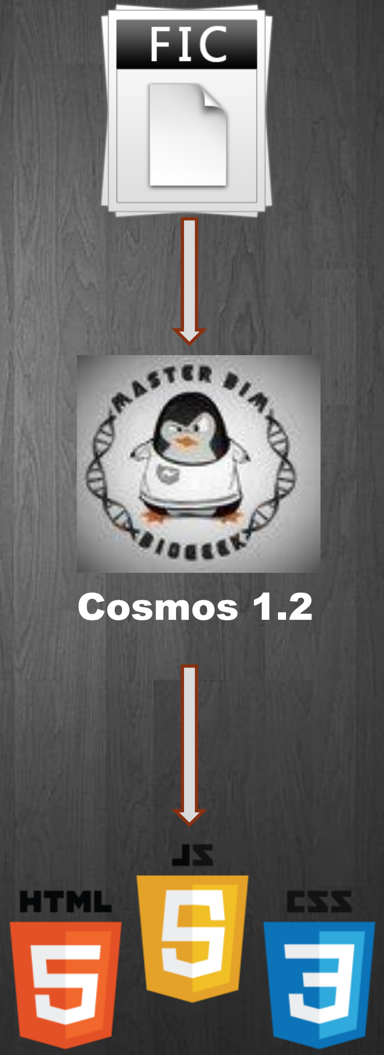
How Cosmos 1.2 works?

1
- Input:
- Text file

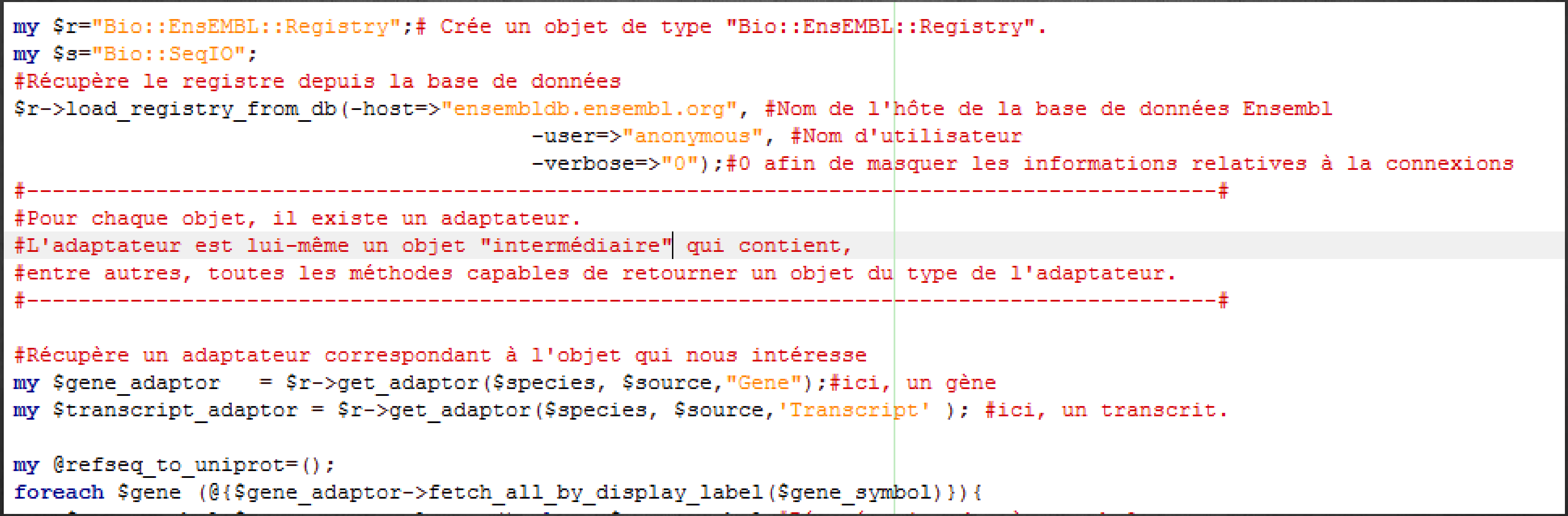
2
- PERL - Cosmos 1.2:
  Gene symbol & Organism extraction

3
- PERL – Cosmos 1.2:
  Iterative Data Processing

4
- PERL:
  Output .html

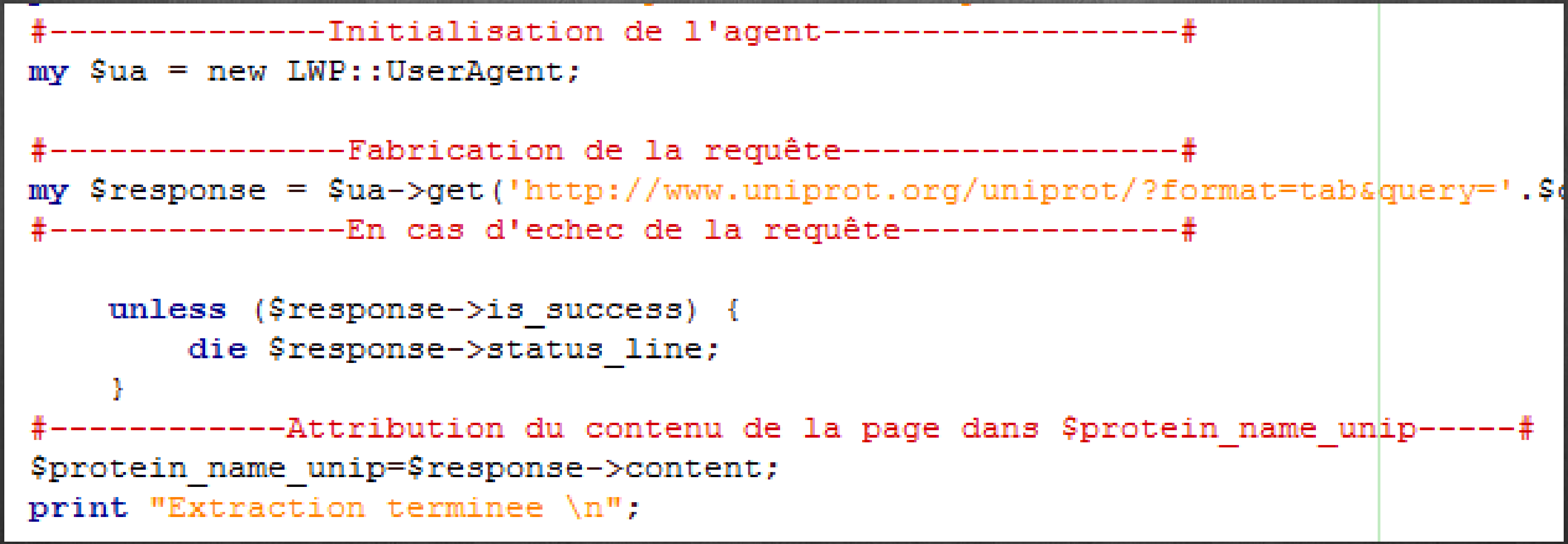


Ensembl API



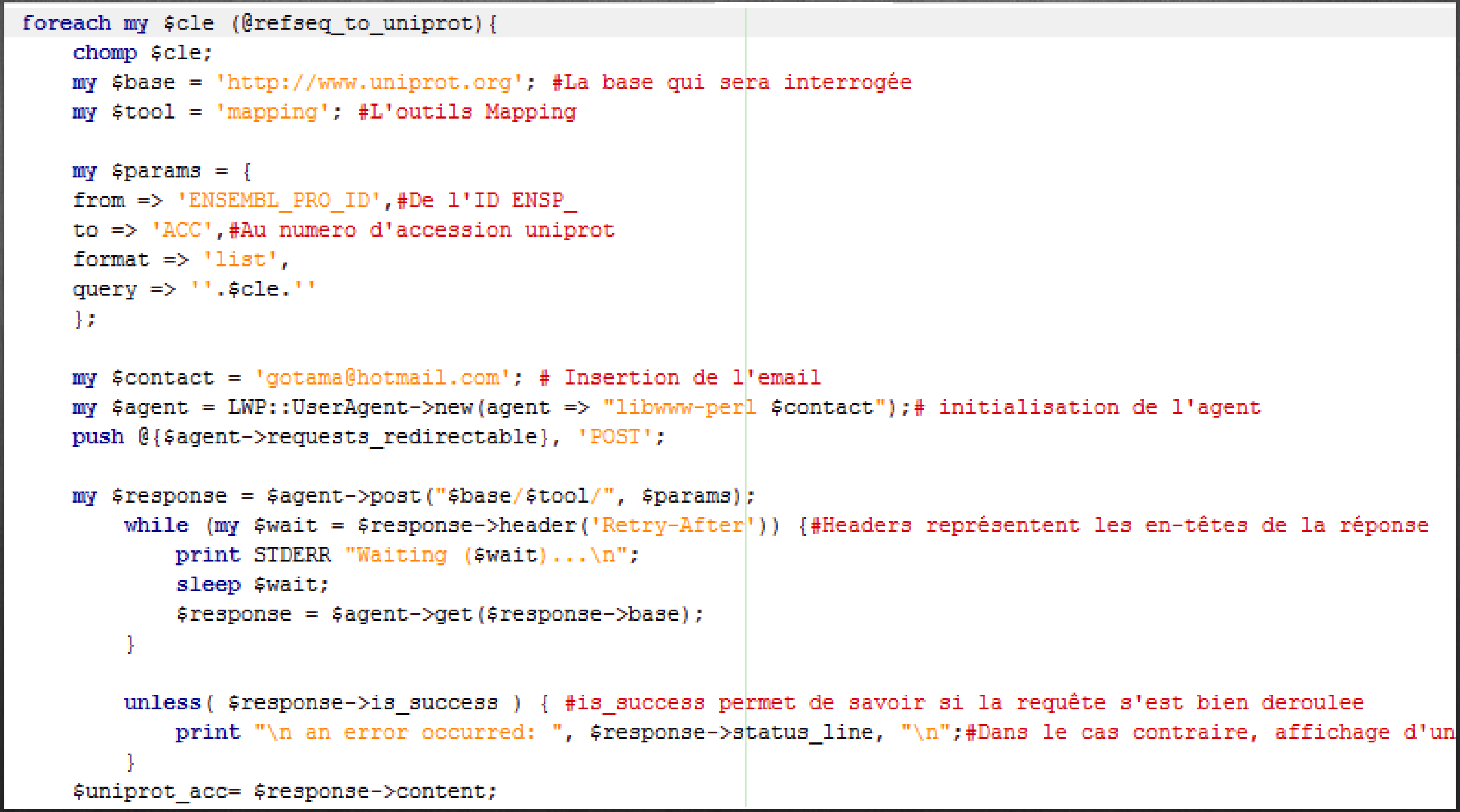
```perl
my $r="Bio::EnsEMBL::Registry";# Crée un objet de type "Bio::EnsEMBL::Registry".
my $s="Bio::SeqIO";
#Récupère le registre depuis la base de données
$r->load_registry_from_db(-host=>"ensembldb.ensembl.org", #Nom de l'hôte de la base de données Ensembl
                                        -user=>"anonymous", #Nom d'utilisateur
                                        -verbose=>"0");#0 afin de masquer les informations relatives à la connexions
#----------------------------------------------------------------------------------------------#
#Pour chaque objet, il existe un adaptateur.
#L'adaptateur est lui-même un objet "intermédiaire" qui contient,
#entre autres, toutes les méthodes capables de retourner un objet du type de l'adaptateur.
#----------------------------------------------------------------------------------------------#

#Récupère un adaptateur correspondant à l'objet qui nous intéresse
my $gene_adaptor    = $r->get_adaptor($species, $source,"Gene");#ici, un gène
my $transcript_adaptor = $r->get_adaptor($species, $source,'Transcript' ); #ici, un transcrit.

my @refseq_to_uniprot=();
foreach $gene (@{$gene_adaptor->fetch_all_by_display_label($gene_symbol)}){
```

LWP



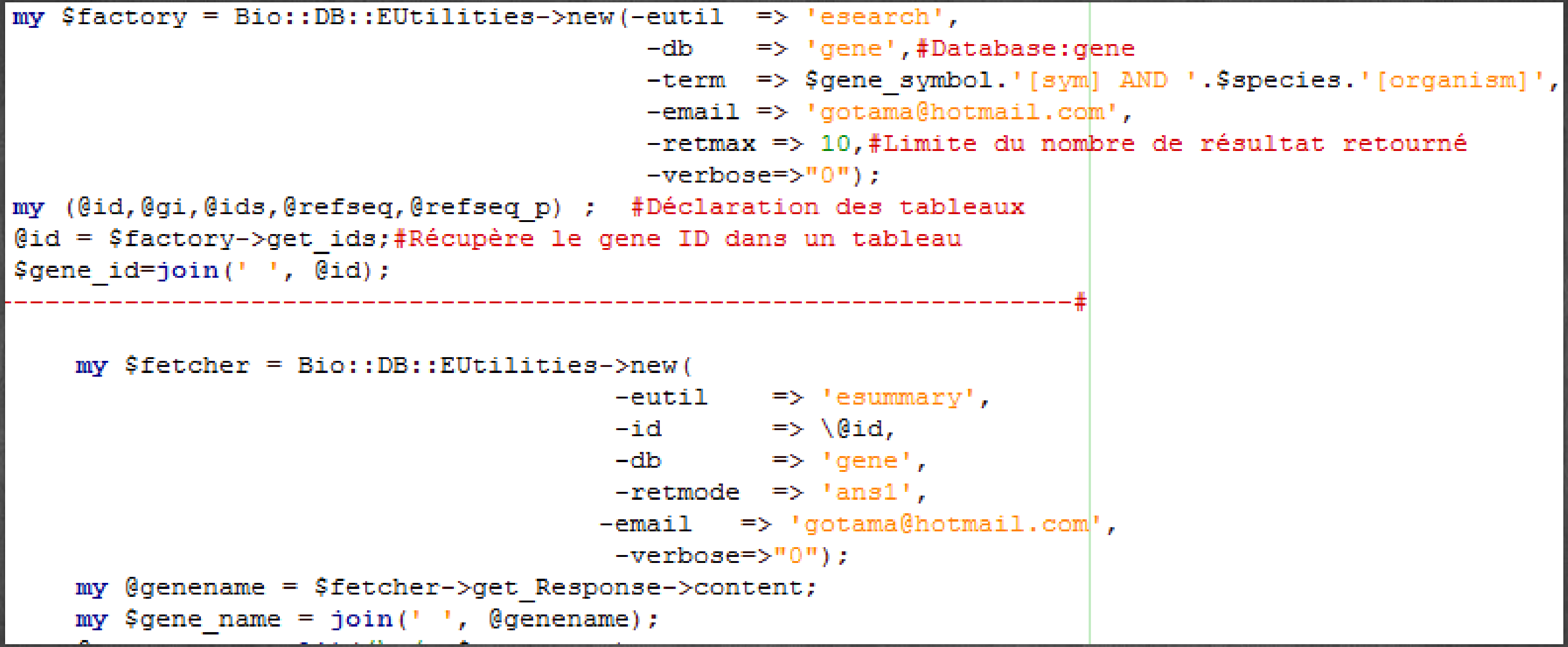
```
#---------------Initialisation de l'agent------------------#
my $ua = new LWP::UserAgent;

#----------------Fabrication de la requête-----------------#
my $response = $ua->get('http://www.uniprot.org/uniprot/?format=tab&query='.$
#----------------En cas d'echec de la requête--------------#

    unless ($response->is_success) {
        die $response->status_line;
    }
#------------Attribution du contenu de la page dans $protein_name_unip-----#
$protein_name_unip=$response->content;
print "Extraction terminee \n";
```

Web Services (Uniprot)

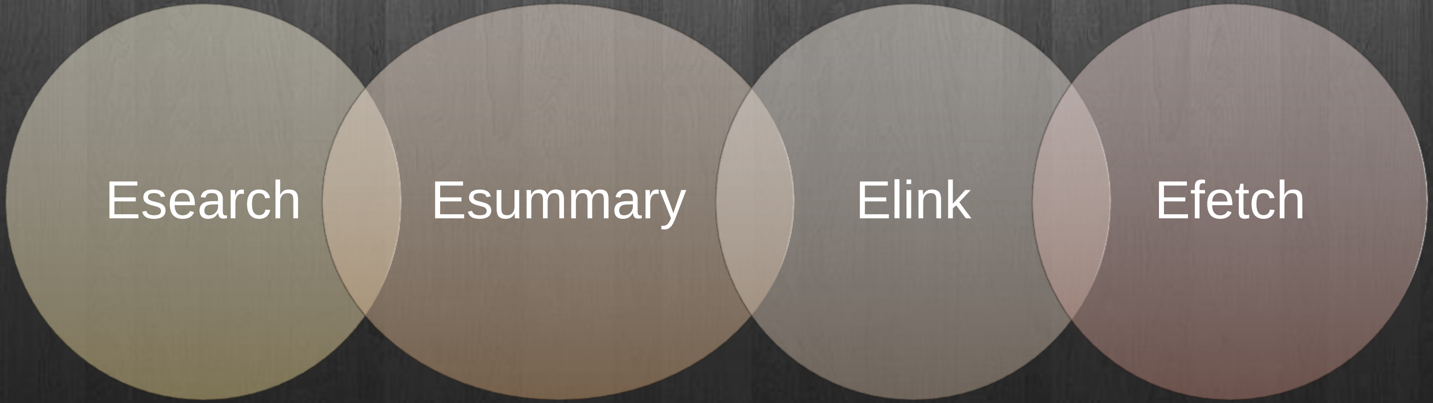


```
foreach my $cle (@refseq_to_uniprot){
    chomp $cle;
    my $base = 'http://www.uniprot.org'; #La base qui sera interrogée
    my $tool = 'mapping'; #L'outils Mapping

    my $params = {
    from => 'ENSEMBL_PRO_ID',#De l'ID ENSP_
    to => 'ACC',#Au numero d'accession uniprot
    format => 'list',
    query => ''.$cle.''
    };

    my $contact = 'gotama@hotmail.com'; # Insertion de l'email
    my $agent = LWP::UserAgent->new(agent => "libwww-perl $contact");# initialisation de l'agent
    push @{$agent->requests_redirectable}, 'POST';

    my $response = $agent->post("$base/$tool/", $params);
        while (my $wait = $response->header('Retry-After')) {#Headers représentent les en-têtes de la réponse
            print STDERR "Waiting ($wait)...\n";
            sleep $wait;
            $response = $agent->get($response->base);
        }

        unless( $response->is_success ) { #is_success permet de savoir si la requête s'est bien deroulee
            print "\n an error occurred: ", $response->status_line, "\n";#Dans le cas contraire, affichage d'un
        }
    $uniprot_acc= $response->content;
```

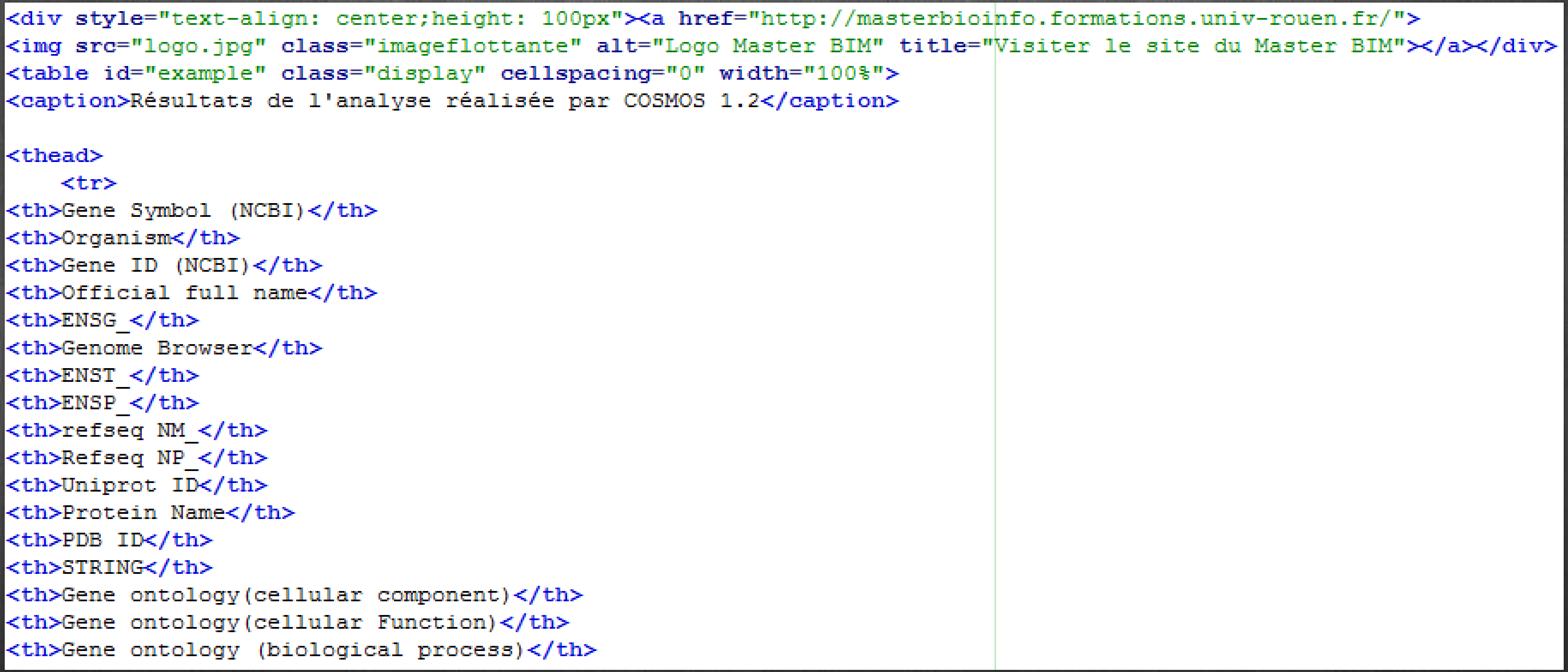
Entrez Utilities (NCBI)



```
my $factory = Bio::DB::EUtilities->new(-eutil  => 'esearch',
                                        -db    => 'gene',#Database:gene
                                        -term  => $gene_symbol.'[sym] AND '.$species.'[organism]',
                                        -email => 'gotama@hotmail.com',
                                        -retmax => 10,#Limite du nombre de résultat retourné
                                        -verbose=>"0");
my (@id,@gi,@ids,@refseq,@refseq_p) ;  #Déclaration des tableaux
@id = $factory->get_ids;#Récupère le gene ID dans un tableau
$gene_id=join(' ', @id);
--------------------------------------------------------------------#

    my $fetcher = Bio::DB::EUtilities->new(
                                      -eutil    => 'esummary',
                                      -id       => \@id,
                                      -db       => 'gene',
                                      -retmode  => 'ans1',
                                     -email    => 'gotama@hotmail.com',
                                      -verbose=>"0");
    my @genename = $fetcher->get_Response->content;
    my $gene_name = join(' ', @genename);
```

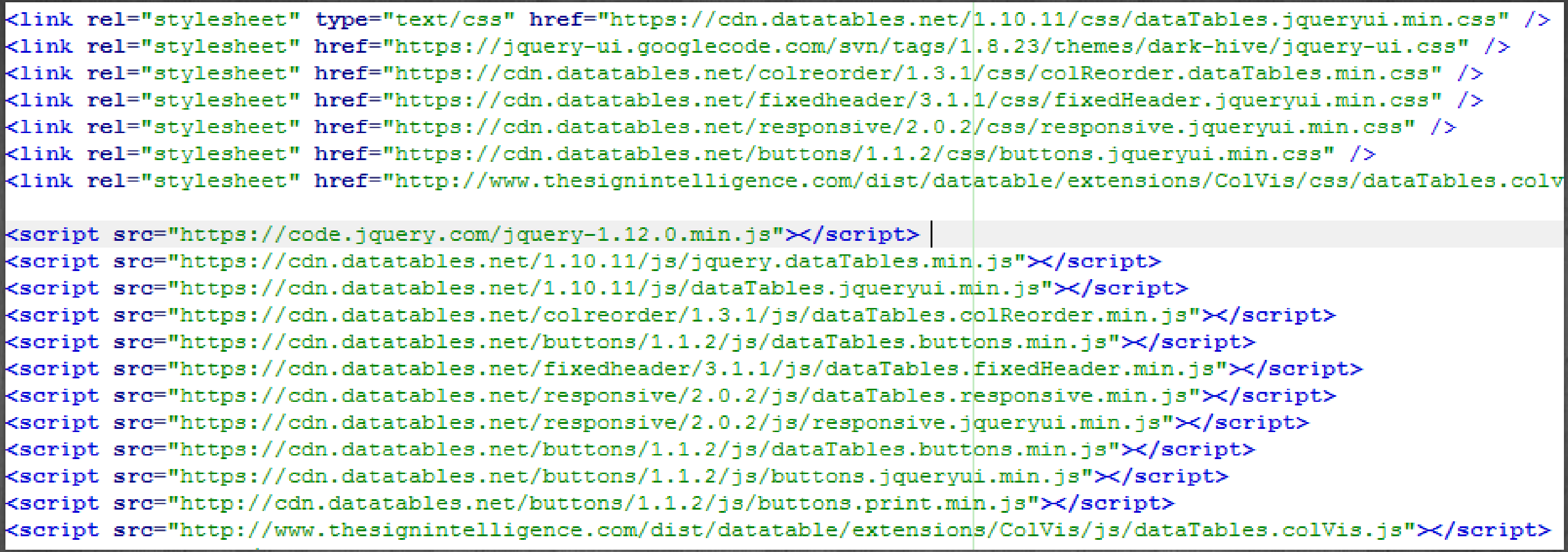


HTML

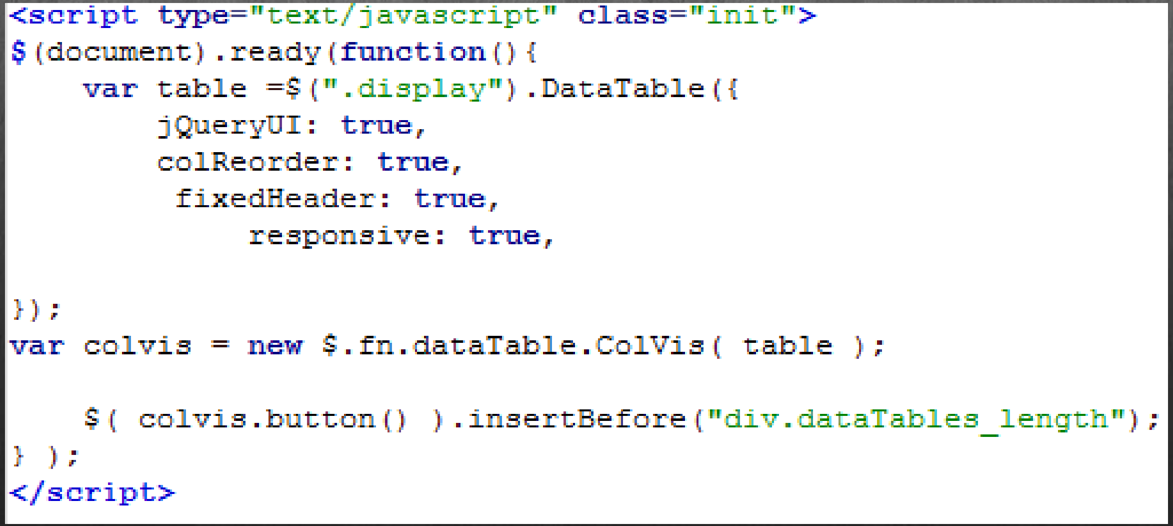


```
<div style="text-align: center;height: 100px"><a href="http://masterbioinfo.formations.univ-rouen.fr/">
<img src="logo.jpg" class="imageflottante" alt="Logo Master BIM" title="Visiter le site du Master BIM"></a></div>
<table id="example" class="display" cellspacing="0" width="100%">
<caption>Résultats de l'analyse réalisée par COSMOS 1.2</caption>

<thead>
    <tr>
<th>Gene Symbol (NCBI)</th>
<th>Organism</th>
<th>Gene ID (NCBI)</th>
<th>Official full name</th>
<th>ENSG_</th>
<th>Genome Browser</th>
<th>ENST_</th>
<th>ENSP_</th>
<th>refseq NM_</th>
<th>Refseq NP_</th>
<th>Uniprot ID</th>
<th>Protein Name</th>
<th>PDB ID</th>
<th>STRING</th>
<th>Gene ontology(cellular component)</th>
<th>Gene ontology(cellular Function)</th>
<th>Gene ontology (biological process)</th>
```

Javascript & Datatables



```html
<link rel="stylesheet" type="text/css" href="https://cdn.datatables.net/1.10.11/css/dataTables.jqueryui.min.css" />
<link rel="stylesheet" href="https://jquery-ui.googlecode.com/svn/tags/1.8.23/themes/dark-hive/jquery-ui.css" />
<link rel="stylesheet" href="https://cdn.datatables.net/colreorder/1.3.1/css/colReorder.dataTables.min.css" />
<link rel="stylesheet" href="https://cdn.datatables.net/fixedheader/3.1.1/css/fixedHeader.jqueryui.min.css" />
<link rel="stylesheet" href="https://cdn.datatables.net/responsive/2.0.2/css/responsive.jqueryui.min.css" />
<link rel="stylesheet" href="https://cdn.datatables.net/buttons/1.1.2/css/buttons.jqueryui.min.css" />
<link rel="stylesheet" href="http://www.thesignintelligence.com/dist/datatable/extensions/ColVis/css/dataTables.colv

<script src="https://code.jquery.com/jquery-1.12.0.min.js"></script>
<script src="https://cdn.datatables.net/1.10.11/js/jquery.dataTables.min.js"></script>
<script src="https://cdn.datatables.net/1.10.11/js/dataTables.jqueryui.min.js"></script>
<script src="https://cdn.datatables.net/colreorder/1.3.1/js/dataTables.colReorder.min.js"></script>
<script src="https://cdn.datatables.net/buttons/1.1.2/js/dataTables.buttons.min.js"></script>
<script src="https://cdn.datatables.net/fixedheader/3.1.1/js/dataTables.fixedHeader.min.js"></script>
<script src="https://cdn.datatables.net/responsive/2.0.2/js/dataTables.responsive.min.js"></script>
<script src="https://cdn.datatables.net/responsive/2.0.2/js/responsive.jqueryui.min.js"></script>
<script src="https://cdn.datatables.net/buttons/1.1.2/js/dataTables.buttons.min.js"></script>
<script src="https://cdn.datatables.net/buttons/1.1.2/js/buttons.jqueryui.min.js"></script>
<script src="http://cdn.datatables.net/buttons/1.1.2/js/buttons.print.min.js"></script>
<script src="http://www.thesignintelligence.com/dist/datatable/extensions/ColVis/js/dataTables.colVis.js"></script>
```



```javascript
<script type="text/javascript" class="init">
$(document).ready(function(){
    var table =$(".display").DataTable({
        jQueryUI: true,
        colReorder: true,
         fixedHeader: true,
            responsive: true,

});
var colvis = new $.fn.dataTable.ColVis( table );

    $( colvis.button() ).insertBefore("div.dataTables_length");
} );
</script>
```

Conclusion

- Difficulties and challenges:
  - Input
  - Output
  - <u>Documentation!!!</u>
  - Javascript & Datatable
  - Optimize Performances

- Positive Points:
  - Fast
  - Rendering
  - User experience
  - Javascript & Datatable
  - Brainstorming
  - skills improvement