

```
# List of packages required for this analysis
pkg <- c("dplyr", "ggplot2", "knitr", "devtools")
# Check if packages are not installed and assign the
# names of the packages not installed to the variable new.pkg
new.pkg <- pkg[!(pkg %in% installed.packages())]
# If there are any packages in the list that aren't installed,
# install them
if (length(new.pkg))
  install.packages(new.pkg, repos = "http://cran.rstudio.com")
# Load packages
library(dplyr)
library(ggplot2)
library(knitr)
```

EvoMining

Introduction

Enzyme promiscuity on metabolic families, can be looked on enzymes that are over a divergent process.

Gen families expansions on genomes

Pangenomes

Expansions are located on pangenome, Tools to analyse pangenome BPgA

EvoMining

EvoMining looks expansions on prokariotic pangenome.

Biological idea.

EvoMining was available as a consult website with 230 members of the Actinobacteria phylum as genomic data base, 226 unclassified nBGCs, and not interchangeable central database 339 queries for nine pathways, including amino acid biosynthesis, glycolysis, pentose phosphate pathway, and tricarboxylic acids cycle. [cruz-morales_phylogenomic_2016] EvoMining was proved on Actinobacteria Arseno-lipids

Pangenome

The sequenced genome of an individual in some species is just a partial print of the species genetic repertoire. Individuals can gain and loss genes.

[koonin_turbulent_2015] Pangenome is the total sequenced gene pool in a taxonomically related group. Supergenome all the possible extant genes. About 10 times genomes. there are open, closed pangenomes. Most genomes has a core a shell and a unique genes.

Gene history its a tree history

HGT doubles mutation rate on prokaryotes.

Maybe HGT is an selected feature, if is the case, so could be np production.

Some archaeas has open pangenome. [halachev_calculating_2011]

HGT doubles mutation rate on prokaryotes. [koonin_turbulent_2015] Maybe HGT is an selected feature, if is the case, so could be np production.

Some archaeas has open pangenome. [halachev_calculating_2011] Shell trees converge to core trees [narechania_random_2012]

EvoMining Implementation

EvoMining was expanded from a website (<http://evodivmet.langebio.cinvestav.mx/EvoMining/index.html>) with limited datasets to an easy to install distribution that allows flexibility on genomic, central and natural product databases. Evomining user distribution was developed on perl on Ubuntu-14.04 but wrapped on Docker. Docker is a software containerization platform that allows repetibility regardless of the environment. Docker engine is available for Linux, Cloud, macOS 10.10.3 Yosemite or newer and even 64bit Windows 10.

Dependencies that were packaged at EvoMining docker app are Apache2, muscle3.8.31, newick-utils-1.6, quicktree, blast-2.2.30, Gblocks_Linux64_0.91b perl and from cpan CGI, SVG and Statistics::Basic modules.

Github defines itself as an online project hosting using Git. Its free for open source-code hosting and facilitates team work. Includes source-code browser, in-line editing, and wikis.

Dockerhub is an apps project hosting.

Dockerhub nselem

EvoMining code is open source and it is available at a github repository [github/EvoMining](https://github.com/nselem/EvoMining)

Github and Dockerhub can be connected by the use of repositories automatically built. Among the advantages of automated builds are that the DockerHub repository is automatically kept up-to-date with code changes on GitHub and that its Dockerfile is available to anyone with access to the Docker Hub repository. EvoMining is stored on a DockerHub automated build repository linked to github EvoMining repository so that code is always actualized.

To download EvoMining image from docker Hub once Docker engine is installed its necessary to run the following command at a terminal:

```
docker pull nselem/newevomining
```

To run EvoMining container

```
docker run -i -t -v /home/nelly/docker-evomining:/var/www/html -p 80:80 evomining /bin/bash
```

To start evoMining app `perl startEvomining`

“ Detailed tutorial, EvoMining description, pipeline and user guide are available at a wiki on github at EvoMining wiki.

Other genomic apps were containerized to docker images during this work.

- *myRAST* docker- <https://github.com/nselem/myrast>

RAST is a bacterial and Archaeal genome annotator [aziz_rast_2008, overbeek_seed_2014, brettin_rasttk_2015] This app allows myRAST functionality to upload It allows EvoMining genome database annotation.

- *Orthocores* docker-<https://github.com/nselem/orthocore>

Helps to obtain genomic core paralog free and construct genomic trees

- *CORASON* docker-<https://github.com/nselem/EvoDivMet/wiki>

- *PseudoCore* github- <>

Genomic Core with a reference genome has the advantage of more genomes, but it is not paralog free

- *RadiCal* docker image

To detect core differences on a set of genomes

- *BPGA* to analyze pangenome

EvoMining Dockerization was chosen to avoid future compatibility problems, for example dependencies unavailability, or incompatibility between future versions of its software components. As much as reproducible research was a concern while developing EvoMining app, reproducibility is also important on data analysis, for that reason this document was written using R-markdown and latex template from Reed College

[@chesterismay_updated_2016]. While R-markdown allows to write and run R code and interpolate text paragraph to explain scripts and analysis.

EvoMining Databases

Evomining containerized app is a user-interactive genomic tool dedicated to the study of protein function[].

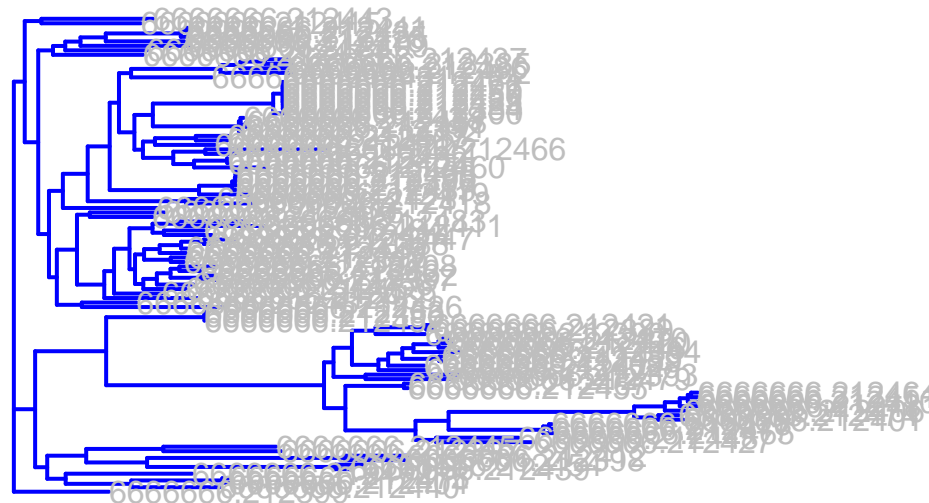
1. Genomes DB
2. Natural Products DB
3. Central Pathways DB

Archaea, *Actinobacteria*, *Cyanobacteria* were used as genome DB, MIBiG was used as Natural Product DB and different Central Pathways were used.

Genome DB

RAST annotation of genomes was done.

Phylogeny



To capture differences on genomes we sort them phylogenetically. Phylogenies can be constructed using different paradigms as Parsimony, Maximum Likelihood, and Bayesian inference. Short descriptions of the main phylogeny methods are included below.

Why is a tree useful {Book reference} why trees are useful for?

* Distance methods

* Parsimony * Maximum Likelihood * Mr bayes

General Trees

Actinobacteria Tree, ArchaeaTree, CyanobacteriaTree.

It's easy to create a list. It can be unordered like

To create a sublist, just indent the values a bit (at least four spaces or a tab). (Here's one case where indentation is key!)

1. Item 1
2. Item 2

3. Item 3

- Item 3a
- Item 3b

Central DB

We chose central pathways from [barona-gomez_what_2012]

* BBH Best Bidirectional Hits with studied enzymes from Central Actinobacterial pathways were selected.

- By abundance
- By expansions on genomes

[largefiles,https://help.github.com/articles/installing-git-large-file-storage/]

Data Bases

Central pathways

Central database were chosen by BBH from

```
table <- read.csv("chapter1/WC_Central/BBH_Organisms.txt", row.names = 1,sep="\t")
kable(table, caption = "BBH_Organisms \\label{tab:BBH_Organisms}",caption.short = "BBH_Organisms ")
```

Table 1: BBH_Organisms

	RastId	Database	Taxa1	Taxa2
Corynebacterium glutamicum	6666666.112876	Actinobacteria		
Streptomyces coelicolor A3(2) NC_003888.3		Actinobacteria		
Mycobacterium tuberculosis H37Rv NC_000962.3	6666666.146923	Actinobacteria		
Methanosarcina acetivorans C2A AE010299.1	6666666.211599	Archaea	Euryarchaeota	Methanomicrob
Nanoarchaeum equitans Kin4-M - AE017199.1	6666666.211718	Archaea	DPANN group	Nanoarchaeota
Natronomonas pharaonis DSM 2160	CR936257.1	6666666.211909	Archaea	Euryarchaeota
Halobacteria				
Sulfolobus solfataricus P2 AE006641.1	6666666.211567	Archaea	TACK group	Crenarchaeota
Cyanothece sp. ATCC 51142 CP000806.1	6666666.212444	Cyanobacteria	Oscillatoriales	
Synechococcus sp. PCC 7002 CP000951.1	6666666.212477	Cyanobacteria	Synechococcales	
Arthrospira platensis C1	6666666.189647	Cyanobacteria	Cyanobacteria	

Genome Dynamics

Among BBH central databases, genomic dynamics was included.

Whats change site:WC Data

groups were formed with 100Cyanos, 100Archaea , 118 Actinos Closed, 43StreptosClosed

Selected orginims were

```
table <- read.csv("chapter1/WC_Central/WC_Organisms.txt", row.names = 1,sep="\t")
kable(table, caption = "WC_Organisms \\label{tab:WC_Organisms}",caption.short = "WC_Organisms ")
```

Table 2: WC_Organisms

	Rast.Id	Database
Arthrospira platensis NIES-39 AP011615.1	6666666.21	Cyanos
Synechococcus sp. PCC 7002	6666666.21	Cyanos
Cyanothece sp. ATCC 51142	6666666.21	Cyanos
Methanosarcina acetivorans	6666666.21	Archaea
Nanoarchaeum equitans Kin4-M	6666666.21	Archaea
Natronomonas pharaonis DSM 2160	6666666.21	Archaea
Sulfolobus solfataricus P2	6666666.21	Archaea
Mycobacterium tuberculosis H37Rv	83332.23	Actinos
Corynebacterium glutamicum ATCC 13032	196627.31	Actinos
Streptomyces coelicolor A3(2) NC_003888.3	6666666.11	Actinos and Streptomyces
Streptomyces sp. Mg1 NZ_CP011664.1	6666666.15	Streptomyces

Those families present on at least as much as genomes on the group

Cyanos 100 647

Abundant.Families.100Cyanos

Actinos 118 132

Abundant.Families.43Strepto

Archaea 100 35

Abundant.Families.Actinos

Streptomyces 43 1263

Abundant.Families.Archaeas

Those families expanded on at least two groups

```
cat *Abun* | cut -f3 | sort | uniq -c | sort >Abundance.all
```

Those Families expanded on Archaea and not expanded on Actino

```
comm -23 f3Archaeas f3Actinos >ArchaeasNoActinos
```

Those Families expanded on Actino and not on Archaea

```
comm -13 f3Archaeas f3Actinos >ActinosNoArchaea
```

Those families expanded on Streptomyces but not in ActinoBacteria

```
comm -13 f343Strepto f3Actinos >ActinosNoStrepto
```

Those Families expanded on Actinobacteria and not in Streptomyces

```
comm -23 f343Strepto f3Actinos >StreptoNoActinos
```

Those Families expanded on Cyano and not in Actino

```
comm -23 f3Cyanos f3Actinos >CyanosNoActinos
```

Natural Products DB

Natural products was improved from previous version

AntisMASH optional DB

AntiSMASH is [@weber_antismash_2015,@medema_antismash:_2011]

Archaeas Results Archaea is a kingdom of recent discovery were not many natural products has been known. On Actinobacteria, evoMining has proved its value to find new kinds of natural products. The clue to this discovery was that Actinobacteria has genomic expansions. Now Archaea has genomic expansions, even more has central pathways genomic expansions. Are this expansions derived from a genomic duplication?

Has Archaea natural products detected by antismash, and if not, where are this NP's or may Archaea doesn't have NP's.

applying EvoMining to Archaea

Otras estrategias para los clusters Argon context Idea

Argonne

ssh nselem@login.mcs.anl.gov

phrase

ssh nselem@maple

password

cs close strain

wc whats chain

we source (edit bashrc)

link ln (create a link to ross directory)

run out of power:

screen

in Seqs (not mine)

cat

```
6666666.103569 6666666.112815 6666666.112823 6666666.112833 6666666.112841 6666666.112849
6666666.112857 > /home/nse/Concat_Full
```

to find paralogous sets

svr_representative_sequences -b -f Id_Clust -s 0.5 < Concat_Full > TempFull&

perl -p -i -e 's///' readable.tree to clean the tree

To find contexts o pegs of paralogous sets

Context midle point 5000 bp (using text tables)

scp 6666666.112839.txt nselem@maple:/homes/nselem/Strepto_01/.

fig|6666666.112839.peg.26

copy families.all file

on the file we have column1 family name column 5 peg id

cluster_objects < elements_to_cluster > ClusteFile

write a file with pegs

1 peg1 adjacent1, adjacent2

1 peg2

2

2

write a file similiar but with the family number

1 peg1 fn1, fn2

1 peg2

2

2

compare each peg on this file from the same family

Write the conexions file

peg1 peg2

peg1 peg3

peg2 peg3

cluster this file and score the cluster

Define

1. a "function set" is generated by the what's changed directory

as a "family"

2. a "paralog set" is a set of function sets in which paralogous

members span the sets

3. a PEG is in a paralog set if it is in one of the function sets

that make up the

4. a "context" of a PEG is the set of close pegs

4.1 First cluster operation would give us: context sets (CS)

5. a "context set" is a set of PEGs with "similar contexts"

5.1 second clustering operation would give us: cluster (Cl)

6. a "cluster" is a set of context sets (each context set is a different

compute:

Compute the context sets that are made from PEGs that occur in PS.

Compute the contexts of PEGs in PS.

cluster these context using the "similar contexts" relation

This gives a set of clusters, and the members of the clusters are context sets

That is, a cluster is a set of context sets

a. the number of contexts sets i

score the clusters

Take a paralog set PS.

Be the context sets: CS₁, CS₂, ..., CS_k members of the paralogous set

k the number of contexts sets on the paralogous set

n_i the cardinality of CS_i

PS={CS₁, CS₂, ..., CS_k}

Cl={ [CS₁, n₁] , [CS₂, n₂] , ..., [CS_k, n_k] }

let be M=max(n_i) i=1,2,...,k (Maximum cardinality of Context sets)

m=max(n_i) i=1,2,...,k, i!=M (second greatest cardinality of context sets)

(We are interested that a second copy is distributed)

We are interested on k, M, n to form a scoring function for the cluster set

S=f(k, m, M)=c₁*k+c₂*m+c₃*M

history

Para hacer un nuevo set de datos

```
591 cd Data/CS
592 mkdir Directorio
593 vi Directorio/rep.genomes
594 cd Directorio/
600 nohup svr_CS -d Directorio&
```

Contenido de rep.genomes

```
rast|390693 nselem35 q8Vf6ib
rast|390675 nselem35 q8Vf6ib
rast|388811 nselem35 q8Vf6ib
```

When you click the **Knit** button above a document will be generated that includes both content as well as the output of any embedded **R** code chunks within the document. You can embed an **R** code chunk like this (`cars` is a built-in **R** dataset):

```
summary(cars)
```

```
##      speed      dist
##  Min.   : 4.0    Min.   :  2.00
## 1st Qu.:12.0    1st Qu.: 26.00
##  Median:15.0    Median : 36.00
##   Mean :15.4    Mean   : 42.98
## 3rd Qu.:19.0    3rd Qu.: 56.00
##   Max. :25.0    Max.   :120.00
```

Inline code

If you'd like to put the results of your analysis directly into your discussion, add inline code like this:

The `cos` of 2π is 1.

Another example would be the direct calculation of the standard deviation:

The standard deviation of `speed` in `cars` is 5.2876444.

One last neat feature is the use of the `ifelse` conditional statement which can be used to output text depending on the result of an **R** calculation:

The standard deviation is less than 6.

Note the use of `>` here, which signifies a quotation environment that will be indented.

As you see with `2π` above, mathematics can be added by surrounding the mathematical text with dollar signs. More examples of this are in [Mathematics and Science] if you uncomment the code in [Math].

Recomendaciones de Luis

Para evoMining

Probar distintos métodos de filogenia y después hacer la coloración.

maximum likelihood, Protest phym

Atracción de ramas largas.

raxml

trim all vs Gblobs (Tony Galvador)

Comparar dos árboles

Para ver si la evolución de los genes concatenados ha sido simultánea

Robinson and foulds

Joe Felsestein

Phylip

2. dist tree

quarter descomposition

peter gogarten fendou Mao

Sets de experimentos.

Para el experimento de los streptomyces con ruta centrales el core, analizar el problema de dominios múltiples.

Dominios

Nan Song, Dannie durand

Después del blast

Para obtener

Pablo Vinuesa: Get Homologues

Burkholderias y su toxina (Preguntar a Beto)

Cianobacterias y la ruta de fijación de nitrógeno.

Servidor Viernes a las 12:00

CORASON: Other genome Mining tools context-based

CORe Analysis of Syntenic Orthologs to prioritize Natural Product-Biosynthetic Gene Cluster

Bacterial biosynthetic gene clusters (BGCs) known are always increasing, almost all bacterial genome sequenced contributes with new genes and gene clusters to the known Bacterial Pangenome. In consequence of gene diversity and sequence technology advances researchers often have a large set of genomes to analyze in search of a particular gene cluster variation. Answering BGCs analysis needs, CORASON allows users to find and visualice variations of a given gene cluster sorting them according to the conserved core cluster phylogeny.

The core genome on a taxonomical group is the set of coding sequences that are shared between all group members, this definition may be adapted to the cluster core by exploring a set of gene clusters instead of a set of genomes. The cluster core attempts to identify a set of functions conserved on a particular BGC variations. A report about gene function using RAST technology will be provided whenever a cluster core exists and core sequences will be concatenated to construct a phylogenetic tree and sort variation clusters accordingly.

To find cluster variations, given a query protein sequence that belongs to a reference cluster, CORASON will search on a Bacterial genome database all gene clusters that contains orthologues of the query-protein and at least another sequence from the reference cluster. Orthologues on variation clusters are coloured within a gradient according to its identity percentage with the reference cluster sequences.

Finally, in order to provide an easy to install distribution, CORASON was packaged on docker containerization platform. Software dependencies such as BLAST 2.2.30, muscle3.8.3, GBlocksLinux64_0.91b, quicktree, newick-utils-1.6, and CORASON code were wrapped together on CORASON docker container. Tutorial and software are available at [nselem/github](https://nselem.github.io).

CORASON inputs are a genomic database, a reference cluster and an enzyme inside this cluster, outputs are newick trees, core functional report and a cluster variation SVG file. SVG format among being high quality scalable graphics, also allow to display metadata such as gene function and genome coordinates just by mouse over figures on a browser facilitating genomic analysis.

In conclusion CORASON is an easy to install comparative genomic visual tool on a customizable genome database that allows users to visualice variations of a reference gene cluster identifying its core functions and finally sorting variations according to their evolutionary history helping to prioritize clusters that may be involved on chemical novelty.

Tree methods (from antiSMASH textual quotation)

Multiple methods exist to construct phylogenetic trees based on multiple sequence alignments. Depending on the desired output tree characteristics, the number of input sequences, and other constraints, the most appropriate method should be chosen. A popular algorithm among the distance-matrix based methods is the Neighbour-Joining algorithm that uses bottom-up clustering to create the tree. Neighbour-Joining is comparatively fast method, but the correctness of the tree depends on the accuracy and additivity of the underlying distance matrix. Maximum parsimony methods try to identify the tree that uses the smallest number of evolution events to explain the observed sequence data. While maximum parsimony algorithms build very accurate trees, their computation tends to be relatively slow compared to distance-matrix based methods. Maximum likelihood methods use probability distributions to assess the likelihood of a given 5 <http://mc.manuscriptcentral.com/bibManuscripts> submitted to Briefings in Bioinformatics phylogenetic tree according to a substitution model. This method unfortunately has a high complexity for computing the optimal tree. Many current tools use a combination of methods