SiLiX:
Ultra-fast sequence clustering from similarity networks

Vincent Miele,Simon Penel,Laurent Duret

Laboratoire Biométrie et Biologie Evolutive (BBE)

September 8, 2010

Menu

Comparative approaches at the whole genome scale in many taxa :

Several steps :

1. Compare all proteins of many genomes (BLAST)

Comparative approaches at the whole genome scale in many taxa :

Several steps :

1. Compare all proteins of many genomes (BLAST)
2. **Cluster homologous sequences into families**

Comparative approaches at the whole genome scale in many taxa :

Several steps :

1. Compare all proteins of many genomes (BLAST)
2. **Cluster homologous sequences into families**
3. Compute multiple sequence alignment for each cluster

Comparative approaches at the whole genome scale in many taxa :

Several steps :

1. Compare all proteins of many genomes (BLAST)
2. **Cluster homologous sequences into families**
3. Compute multiple sequence alignment for each cluster
4. Compute phylogenetic tree for each alignment

Example : HOGENOM (phylogenomic database of gene families from fully sequenced organism)

- 820 bacteria,62 archaea, 64 eukarya,

Example : HOGENOM (phylogenomic database of gene families from fully sequenced organism)

- 820 bacteria,62 archaea, 64 eukarya,
- 3,666,568 protein sequences (76% bacteria,3% archae and 20% eukarya),

Example : HOGENOM (phylogenomic database of gene families from fully sequenced organism)

- 820 bacteria,62 archaea, 64 eukarya,
- 3,666,568 protein sequences (76% bacteria,3% archae and 20% eukarya),
- distributed calculations (grid computing, cluster computing) used to generate the BLAST output

Example : HOGENOM (phylogenomic database of gene families from fully sequenced organism)

- 820 bacteria,62 archaea, 64 eukarya,
- 3,666,568 protein sequences (76% bacteria,3% archae and 20% eukarya),
- distributed calculations (grid computing, cluster computing) used to generate the BLAST output
  - 4 years cpu

Example : HOGENOM (phylogenomic database of gene families from fully sequenced organism)

- 820 bacteria,62 archaea, 64 eukarya,
- 3,666,568 protein sequences (76% bacteria,3% archae and 20% eukarya),
- distributed calculations (grid computing, cluster computing) used to generate the BLAST output
  - 4 years cpu
  - 110 Gb output (tabulated format)

Example : HOGENOM (phylogenomic database of gene families from fully sequenced organism)

- 820 bacteria,62 archaea, 64 eukarya,
- 3,666,568 protein sequences (76% bacteria,3% archae and 20% eukarya),
- distributed calculations (grid computing, cluster computing) used to generate the BLAST output
  - 4 years cpu
  - 110 Gb output (tabulated format)
  - 1,905,335,339 pairwise alignments (i.e. lines) (threshold 10^{-4},all local alignments)

Example : HOGENOM (phylogenomic database of gene families from fully sequenced organism)

- 820 bacteria,62 archaea, 64 eukarya,
- 3,666,568 protein sequences (76% bacteria,3% archae and 20% eukarya),
- distributed calculations (grid computing, cluster computing) used to generate the BLAST output
  - 4 years cpu
  - 110 Gb output (tabulated format)
  - 1,905,335,339 pairwise alignments (i.e. lines) (threshold 10^{-4},all local alignments)
- **Clustering step on a huge dataset**

Test of available clustering softwares on HOGENOM dataset

Method	CPU(min)	Mem (Gb)
Force	-	Out of Memory
hcluster_sg	5,604 *(4 days)*	99
MCL	>14,711 *(10 days, stopped)*	42
MC-UPGMA	2,711 *(2 days,not converged)*	0.5

Available softwares are not able to deal with this huge amount of data.

Need for a new software !

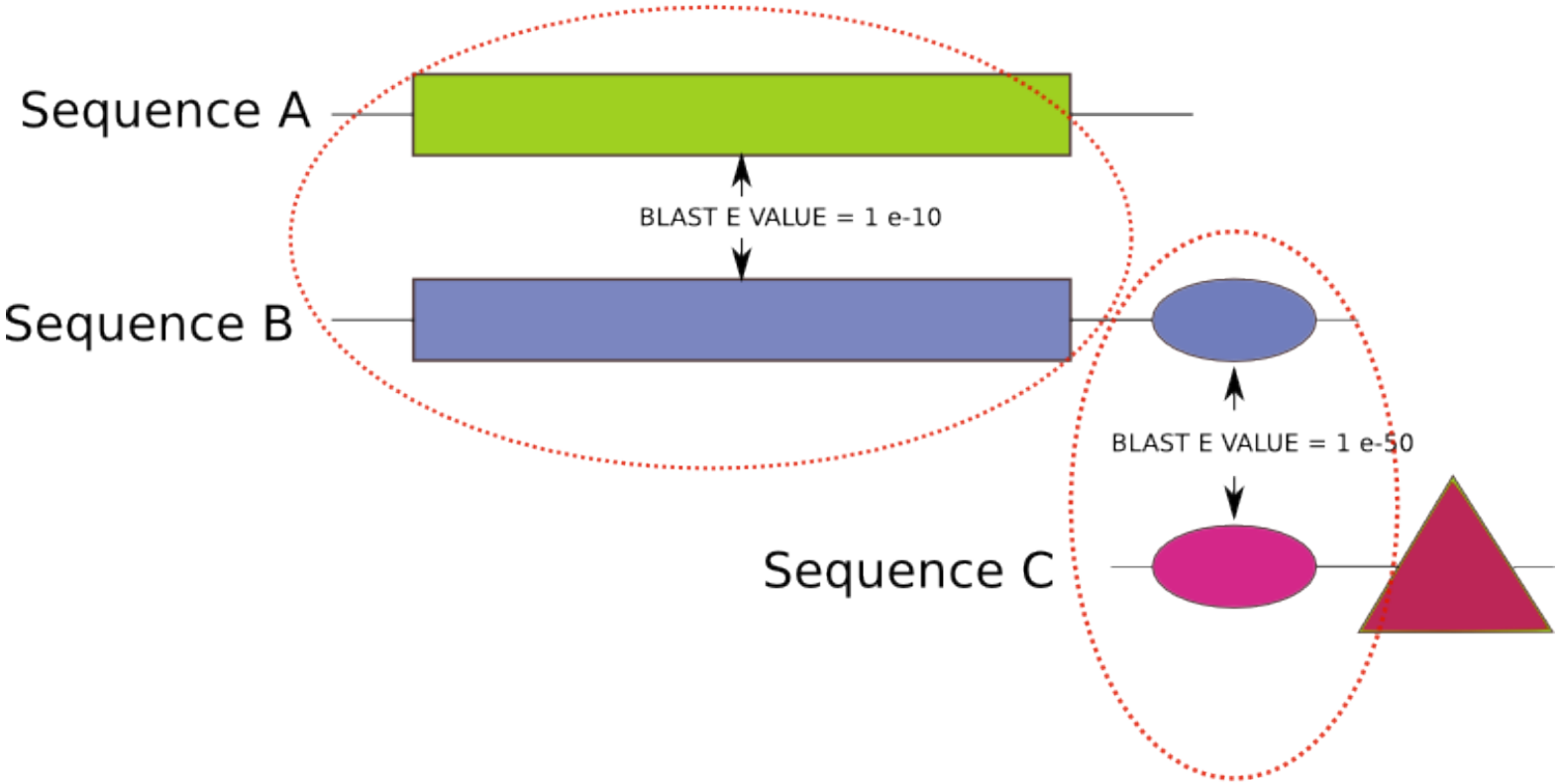
SiLix stands for*SIngle LInkage Clustering of Sequences*
It associates :

- Single Linkage Clustering
- Clustering with Alignment Coverage Constraint (C.A.C.C.)

Single-linkage (transitive-link) clustering :

- if sequence A is considered homologous to sequence B,
- if sequence B is considered homologous to sequence C,
- whatever the level of similarity between A and C, A, B and C are grouped into the same family

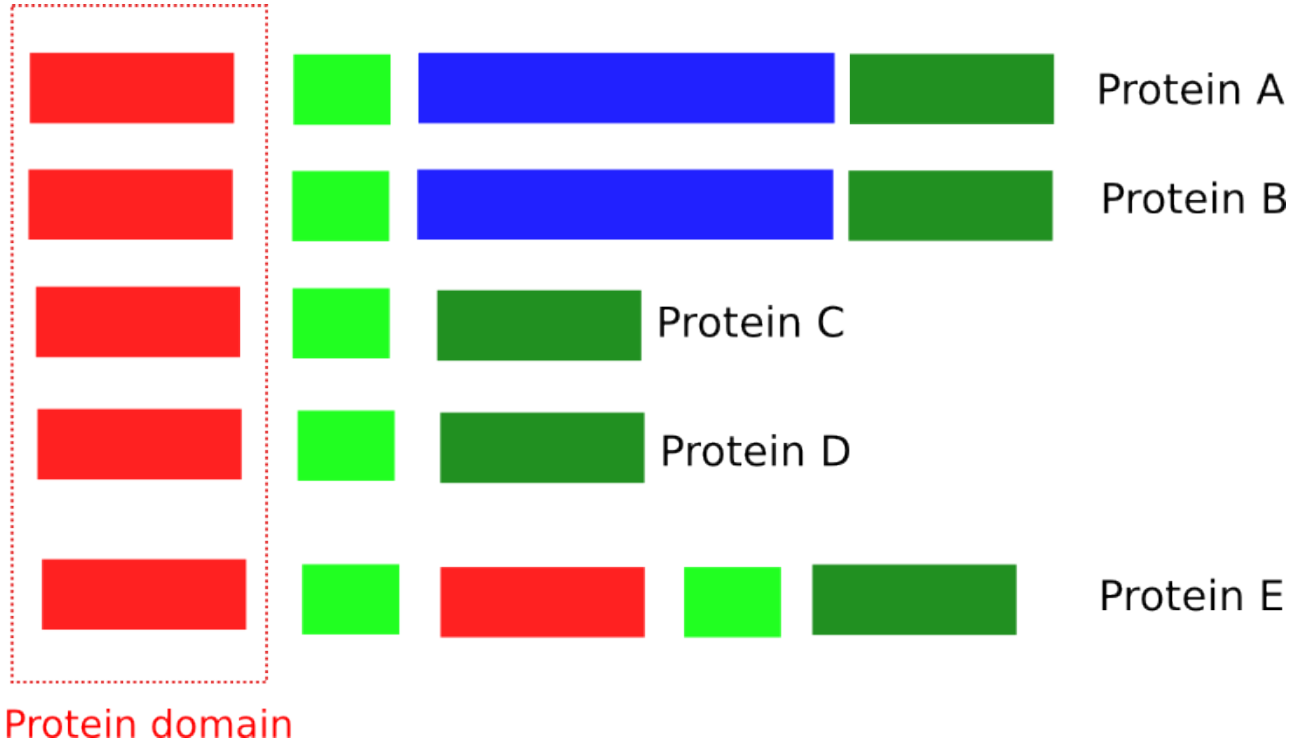
$$(A, B) and (B, C) \Rightarrow (A, B, C)$$



Single Linkage using domains : Accumulation of unrelated sequences.
Cluster (A,B,C) : A and C not homologous

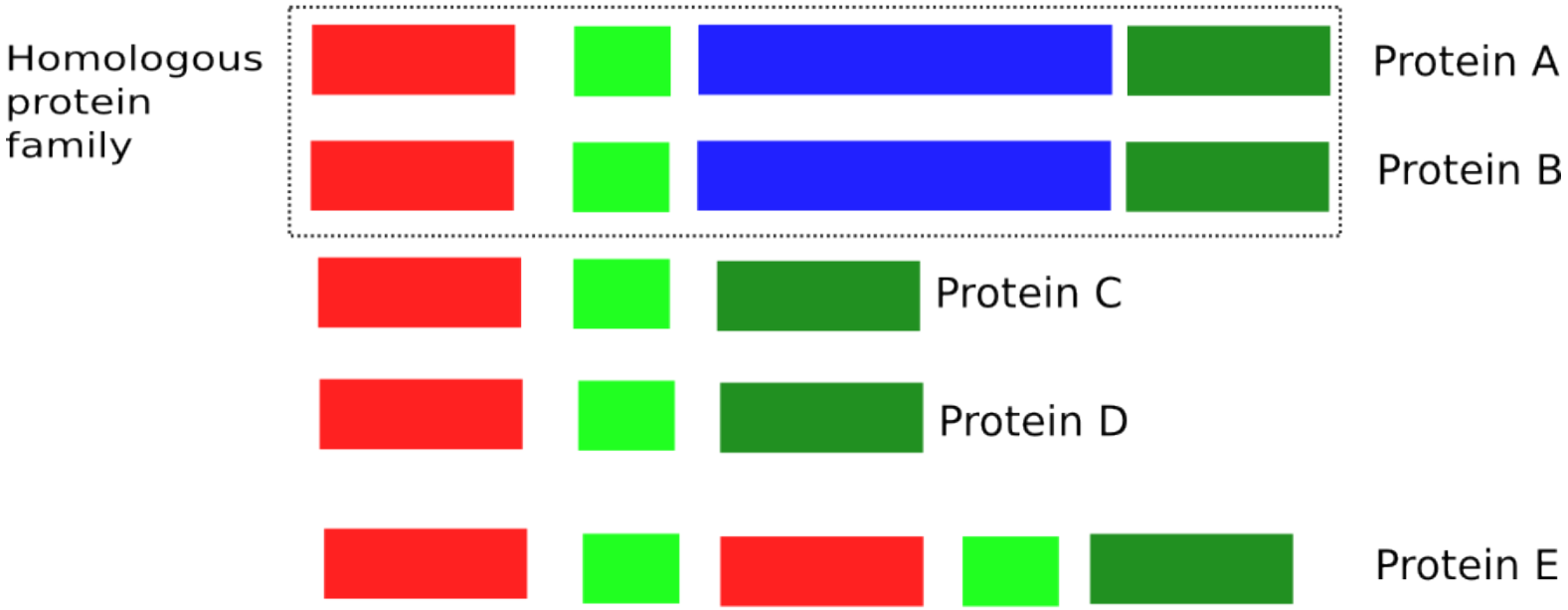
Clustering with Alignment Coverage Constraint (C.A.C.C.)

Phylogeny of a sequence : NOT interested in DOMAINS, but in FULLY HOMOLOGOUS sequences

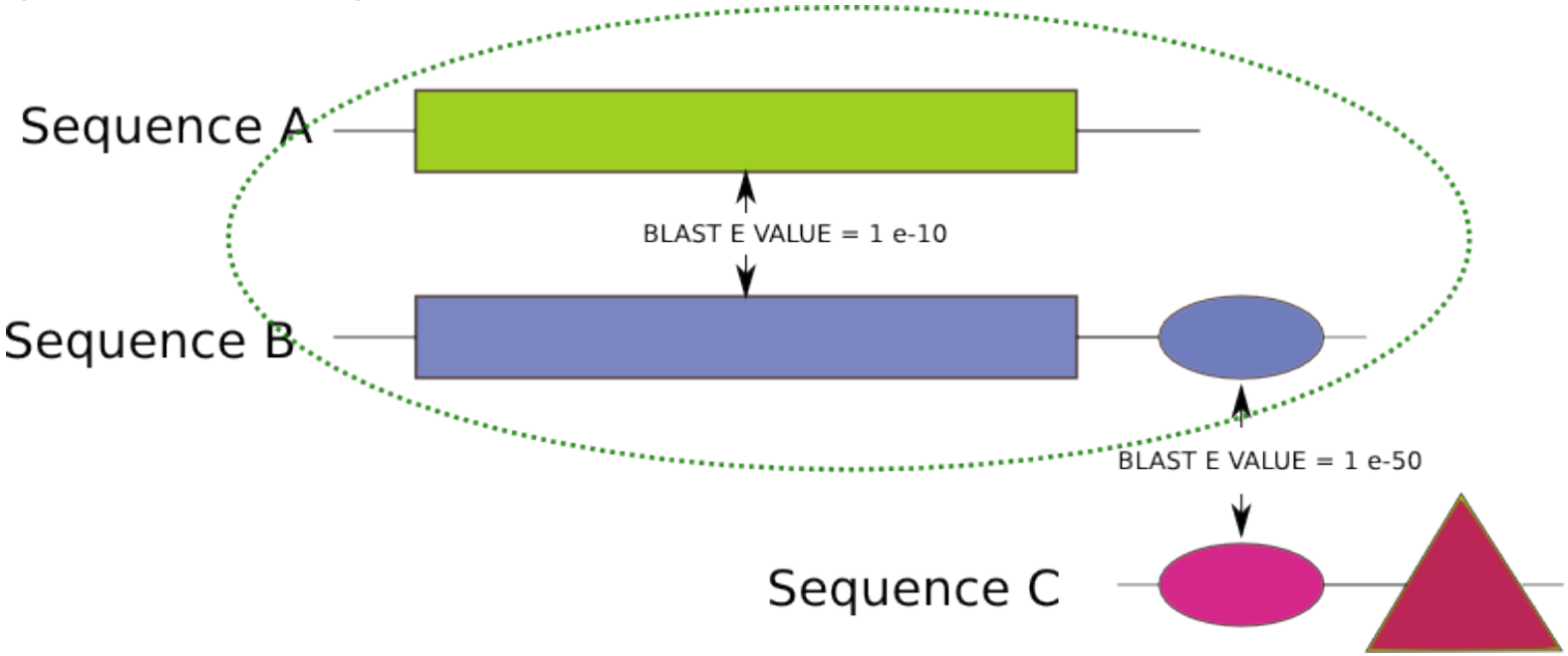


Clustering with Alignment Coverage Constraint (C.A.C.C.)

Phylogeny of a sequence : NOT interested in DOMAINS, but in FULLY HOMOLOGOUS SEQUENCES



Clustering with Alignment Coverage Constraint (C.A.C.C.) :
2 sequences are associated if present enough **similarity** on enough **length** (2 parameters).



Single Linkage with C.A.C.C.

No accumulation of unrelated sequences : Cluster (A,B)

Menu

We define an undirected graph $G = (V, E)$

- the set of vertices V representing sequences and
- the sets of edges E representing similarities between these sequences.
- $n = |V|$ and $m = |E|$.

Finding sequence families consists in computing the connected components of G ...

computing the connected components of G ...

We want to address the case of large n and m

- avoid storing the edges into a connectivity matrix
- examine the edges *online*

Menu

What we expect at the end ?

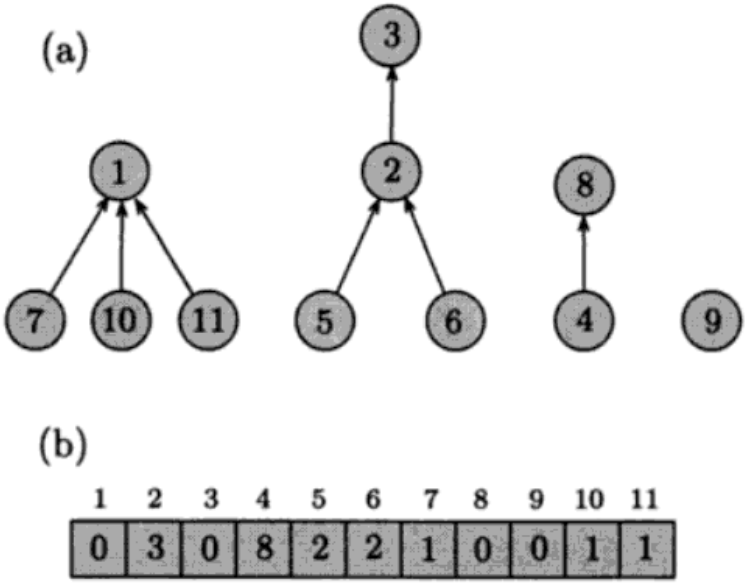
connected components of G
=
a partition of V into non-overlapping subsets
=
disjoint-sets

How to treat one edge ?

find the name of the set containing each of the two vertices
&
union these sets by merging their vertices

Using conceptually-driven structure

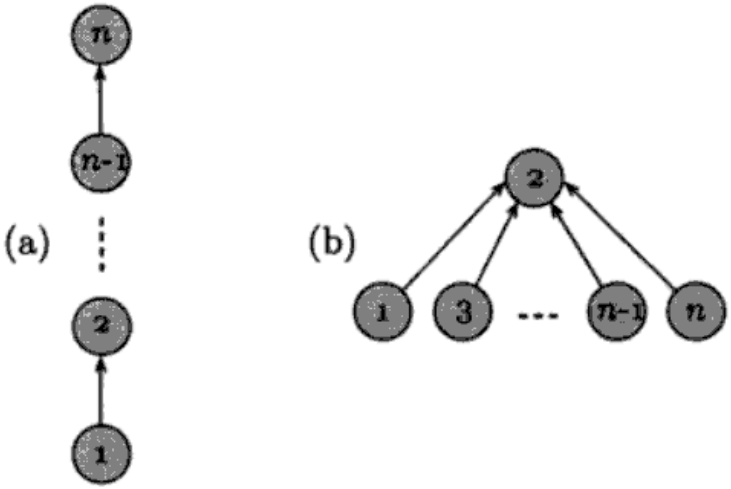
1. each set is a sorted list
2. *find(x)* : search x in all the lists = logarithmic complexity !
3. *union(x,y)* : merge two lists = memory management (most expensive operation on a computer) !



Requires only an array *parent*, such that $parent[r] = 0$ if $root(r) = r$

Using a memory-efficient structure

- *disjoint-sets data structure* (Tarjan, 1975)
- each set is represented as a tree
- its root is the *representative*
- *find(x)* : follows the path from x until the root of its tree, called $root(x)$
- *union(x,y)* : link $root(x)$ to $root(y)$

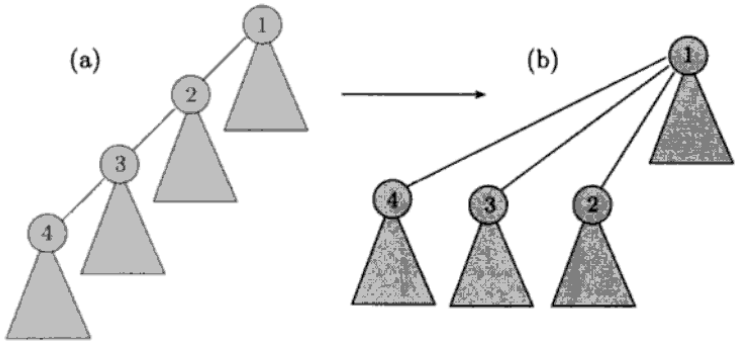


find could be in O(n)

(Exple : new edge (1,3))

Control the tree height

- each node has a rank, essentially its height.
- initially rank is 0
- in *union*, a root of lower rank is linked to a root of higher rank
- rank is incremented if $rank(x) == rank(y)$



(Exple : find(4))

Contract the trees

- *find(x)* : follows the path from x until $root(x)$, but link the nodes in this path to $root(x)$
- the amount of work increases for 1 find operation
- but shorter paths in average

Proved to be in almost O(m) time and in O(n) space !

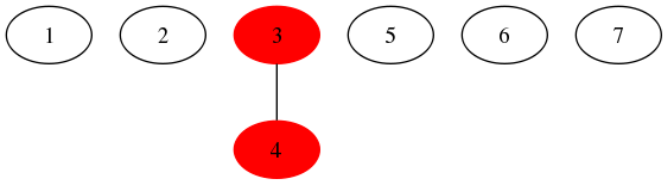
Vertice	1	2	3	4	5	6	7
Parent	0	0	0	0	0	0	0

(4, 3)

Vertice	1	2	3	4	5	6	7
Parent	0	0	0	0	0	0	0

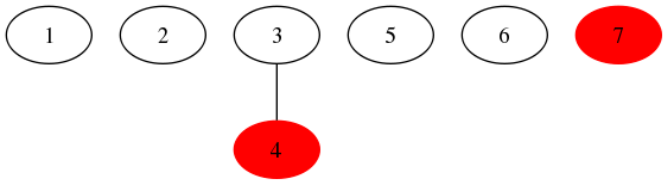
$(4, 3)$

Vertice	1	2	3	4	5	6	7
Parent	0	0	0	3	0	0	0



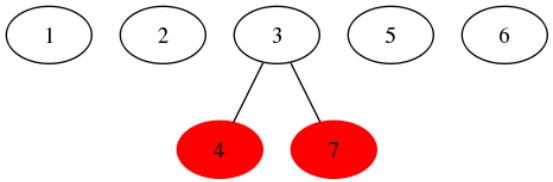
$(7, 4)$

Vertice	1	2	3	4	5	6	7
Parent	0	0	0	3	0	0	0



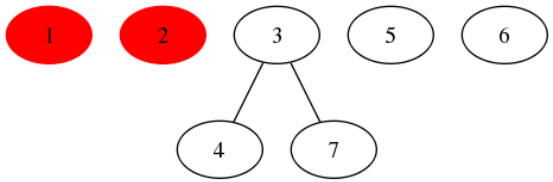
$(7, 4)$

Vertice	1	2	3	4	5	6	7
Parent	0	0	0	3	0	0	3



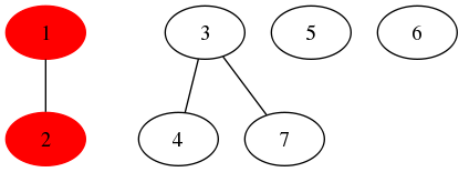
$(2, 1)$

Vertice	1	2	3	4	5	6	7
Parent	0	0	0	3	0	0	3



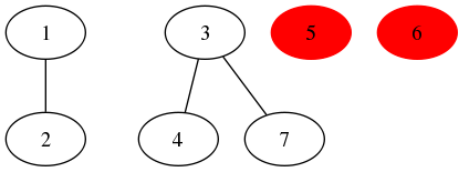
$(2, 1)$

Vertice	1	2	3	4	5	6	7
Parent	0	1	0	3	0	0	3



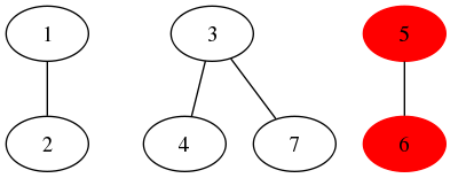
$(6, 5)$

Vertice	1	2	3	4	5	6	7
Parent	0	1	0	3	0	0	3



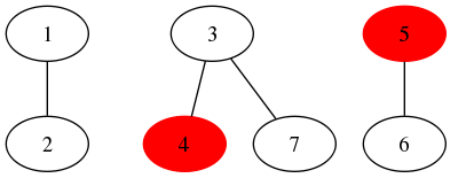
$(6, 5)$

Vertice	1	2	3	4	5	6	7
Parent	0	1	0	3	0	5	3



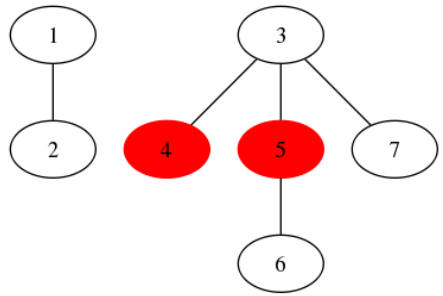
$(5, 4)$

Vertice	1	2	3	4	5	6	7
Parent	0	1	0	3	0	5	3



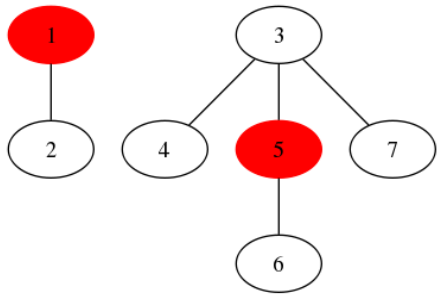
$(5, 4)$

Vertice	1	2	3	4	5	6	7
Parent	0	1	0	3	3	5	3



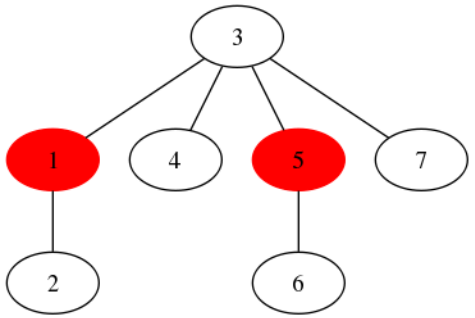
$(5, 1)$

Vertice	1	2	3	4	5	6	7
Parent	0	1	0	3	3	5	3



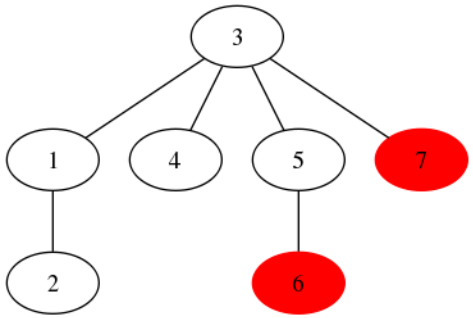
$(5, 1)$

Vertice	1	2	3	4	5	6	7
Parent	3	1	0	3	3	5	3



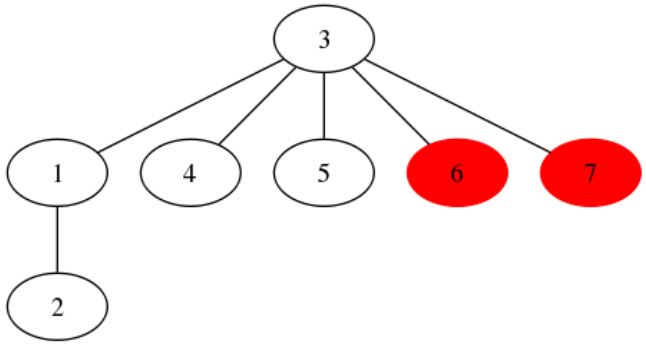
$(7, 6)$

Vertice	1	2	3	4	5	6	7
Parent	3	1	0	3	3	5	3



$(7, 6)$

Vertice	1	2	3	4	5	6	7
Parent	3	1	0	3	3	3	3



Comparison of available softwares and SiLiX on HOGENOM dataset

Method	CPU(min)	Mem (Gb)
SiLiX	**138**	**0.4**
Force	-	Out of Memory
hcluster_sg	5,604 *(4 days)*	99
MCL	>14,711 *(10 days, stopped)*	42
MC-UPGMA	2,711 *(2 days,not converged)*	0.5

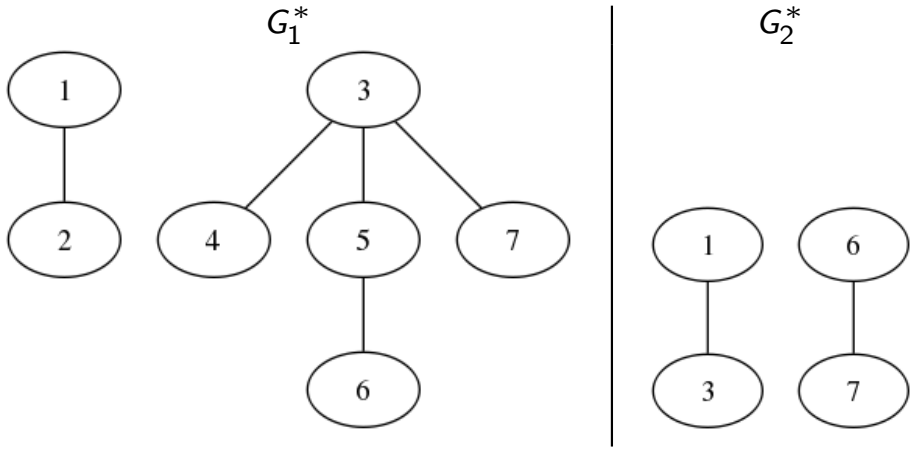
SiLiX is faster and more memory efficient than other methods

Menu

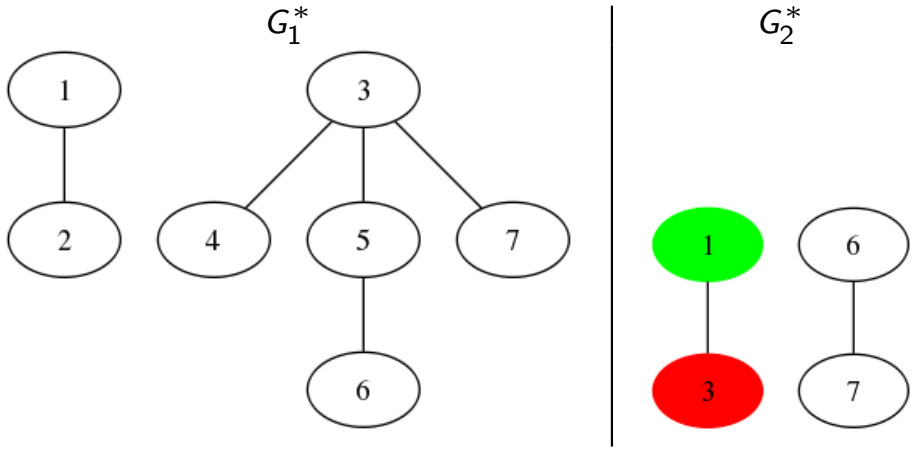
SiLiX is parallelisable !

Divide and conquer :
split a large set into q sets
+
consider a group of q processors =
use the previous sequential algorithm to obtain a collection of spanning (star) forests $G_1^* \ldots G_q^*$
+
merge $G_1^* \ldots G_q^*$
complexity in $O(m/q + n * q)$!

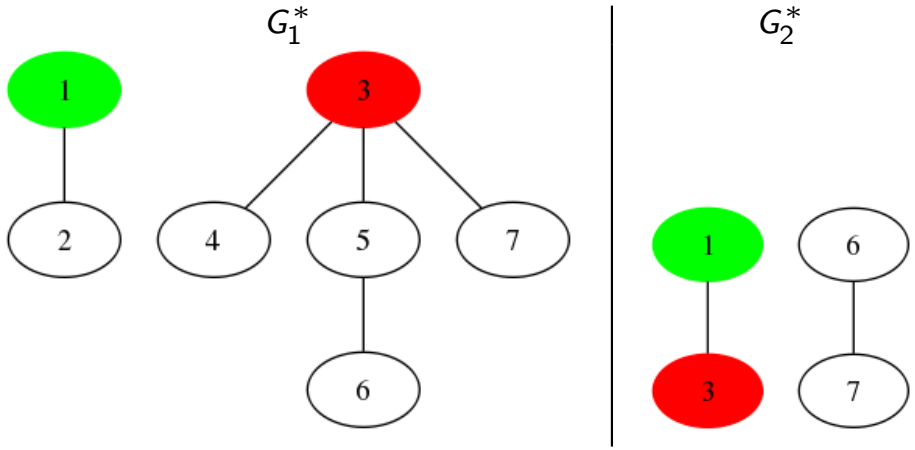
Merging G_2^* to G_1^*



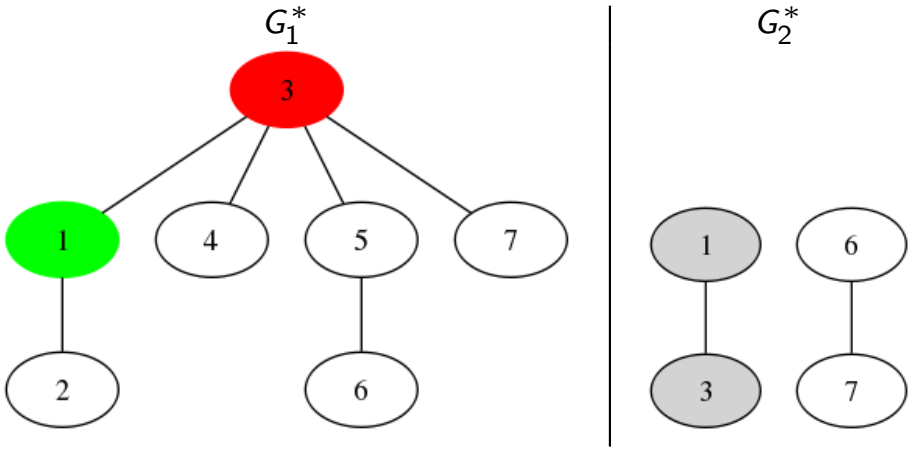
Merging G_2^* to G_1^*



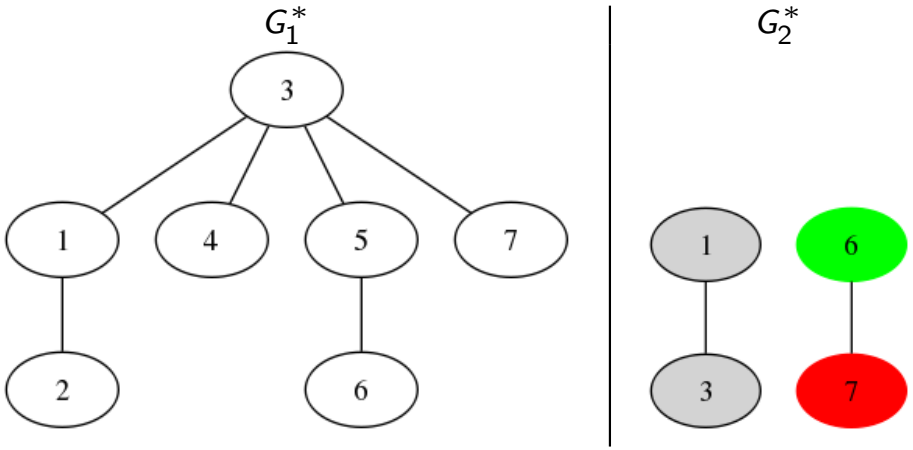
Merging G_2^* to G_1^*



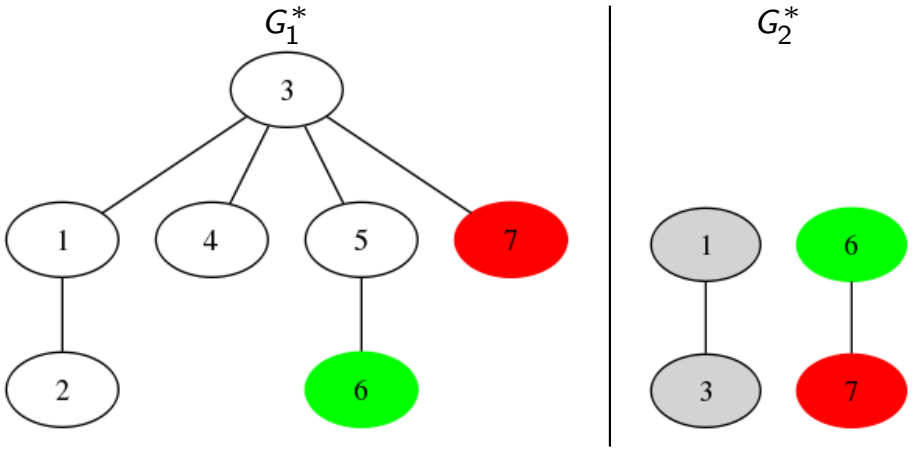
Merging G_2^* to G_1^*



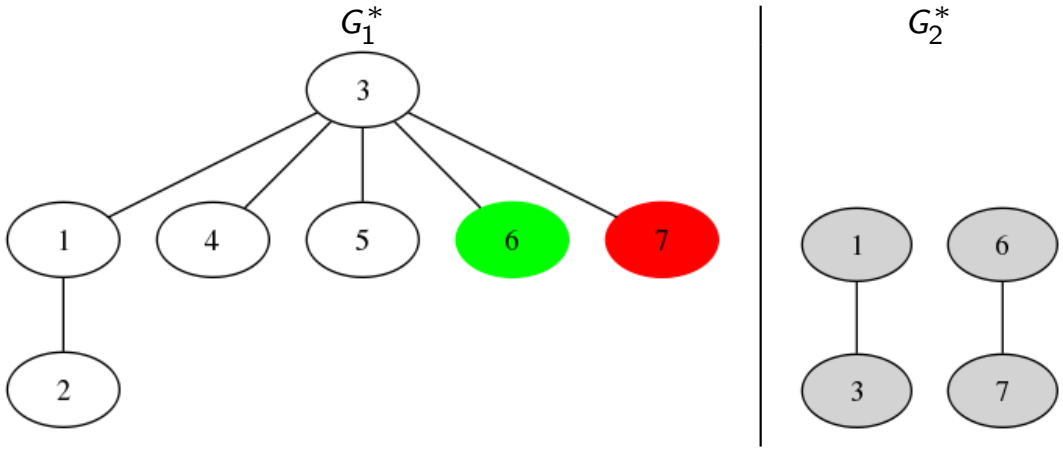
Merging G_2^* to G_1^*



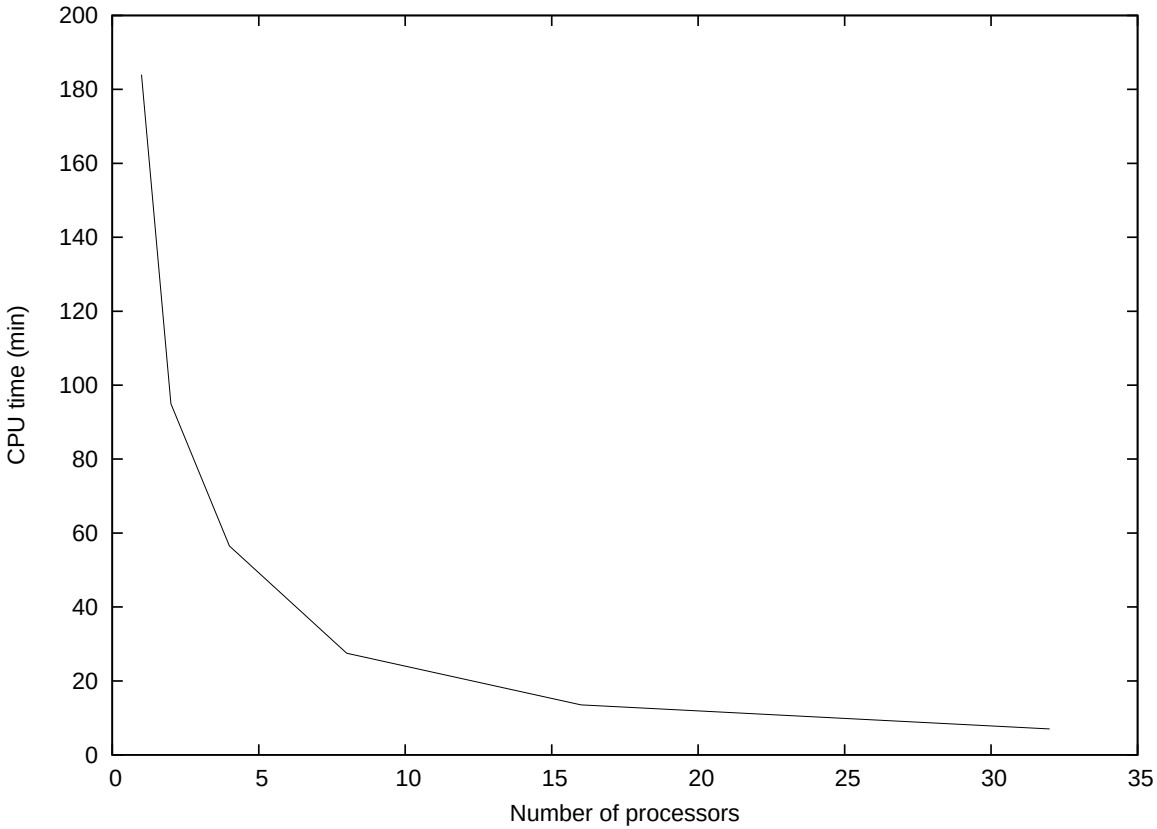
Merging G_2^* to G_1^*



Merging G_2^* to G_1^*



SiLiX scalable in practice



Allows to search for optimal clustering parameters
Allows to analyse many different datasets

About 2.10^9 BLAST pairwise alignments in 7 minutes with 32 cores

Menu

- What is a *good* clustering ?

- What is a *good* clustering ?
  - Question : which objective criterion to estimate a clustering ?

- What is a *good* clustering ?
  - Question : which objective criterion to estimate a clustering ?
  - Possible answer : estimation of its biological relevance.

- What is a *good* clustering ?
  - Question : which objective criterion to estimate a clustering ?
  - Possible answer : estimation of its biological relevance.
- Use of a animal mitochondrial sequence dataset

- What is a *good* clustering ?
  - Question : which objective criterion to estimate a clustering ?
  - Possible answer : estimation of its biological relevance.
- Use of a animal mitochondrial sequence dataset
  - classified into 13 well known gene families (13 genes per organism).

- What is a *good* clustering ?
  - Question : which objective criterion to estimate a clustering ?
  - Possible answer : estimation of its biological relevance.
- Use of a animal mitochondrial sequence dataset
  - classified into 13 well known gene families (13 genes per organism).
  - we can compare this classification with SiLiX clustering.

Build a mitochondrial and nuclear genome dataset from :

1. HOGENOM : complete nuclear genomes from bacteria, archaea and eukarya : 3,665,558 protein sequences

Build a mitochondrial and nuclear genome dataset from :

1. HOGENOM : complete nuclear genomes from bacteria, archaea and eukarya : 3,665,558 protein sequences
2. mitochondrion RefSeq 41 : all animal mitochondrial sequences :31,207 sequences - 2,174 species

Nuclear and mitochondrial sequences datasets are concatened into a single dataset of 3,686,765 sequences.

Is Single Linkage able to

- retrieve the classification of mitochondrial sequences into 13 families among all the sequences ?
- avoid spurious clustering of non homologous sequences ?

Parameters :

- identitiy $>$ 35 %
- coverage $>$ 80 %

Comparaison of the 13 SiLiX famillies containing the biggest number of RefSeq sequences and the 13 mitochondrial genes

gene name and length	nb of mitochondrial sequences	sensibility	specificity
ND1 315aa	1981	100%	100%
ND2 340aa	1982	89%	100%
COX1 515aa	2167	98%	99%
COX2 230aa	2121	99%	100%
ATP8 55aa	1988	49%	100%
ATP6 225aa	2112	95%	100%
COX3 260aa	2121	98%	100%
ND3 115aa	1985	99%	100%
ND4L 100aa	1981	92%	100%
ND4 450aa	1983	97%	100%
ND5 600aa	1983	98%	100%
ND6 170aa	1982	70%	100%
CYTB 380aa	2005	100%	100%

Parameters :

- identitiy $>$ 35 %
- coverage $>$ 80 %

Comparaison of the 13 SiLiX famillies containing the biggest number of RefSeq sequences and the 13 mitochondrial genes

gene name and length	nb of mitochondrial sequences	sensibility	specificity
ND1 315aa	1981	100%	100%
ND2 340aa	1982	89%	100%
COX1 515aa	2167	98%	99%
COX2 230aa	2121	99%	100%
ATP8 55aa	1988	49%	100%
ATP6 225aa	2112	95%	100%
COX3 260aa	2121	98%	100%
ND3 115aa	1985	99%	100%
ND4L 100aa	1981	92%	100%
ND4 450aa	1983	97%	100%
ND5 600aa	1983	98%	100%
ND6 170aa	1982	70%	100%
CYTB 380aa	2005	100%	100%

Specificity of SiLiX families is excellent.
Clusters dot not associate spurious non homologous sequences.
$\Rightarrow$ **When used with alignement coverage constraint criteria, single linkage works fine.**

Menu

New proposal :Explore gene families from a network point of view.

- Post-treat and curate the families by removing similarities that must be false positive
- Find community structure and interpret topology from an evolutionary perspective

- SiLiX higly faster and memory efficient than available programs
  $\Rightarrow$ allows to deals with huge datasets
- Single Linkage clustering combined with coverage constraint is reliable
  $\Rightarrow$ clustering quality is good
- SiLiX sofwtare is now available

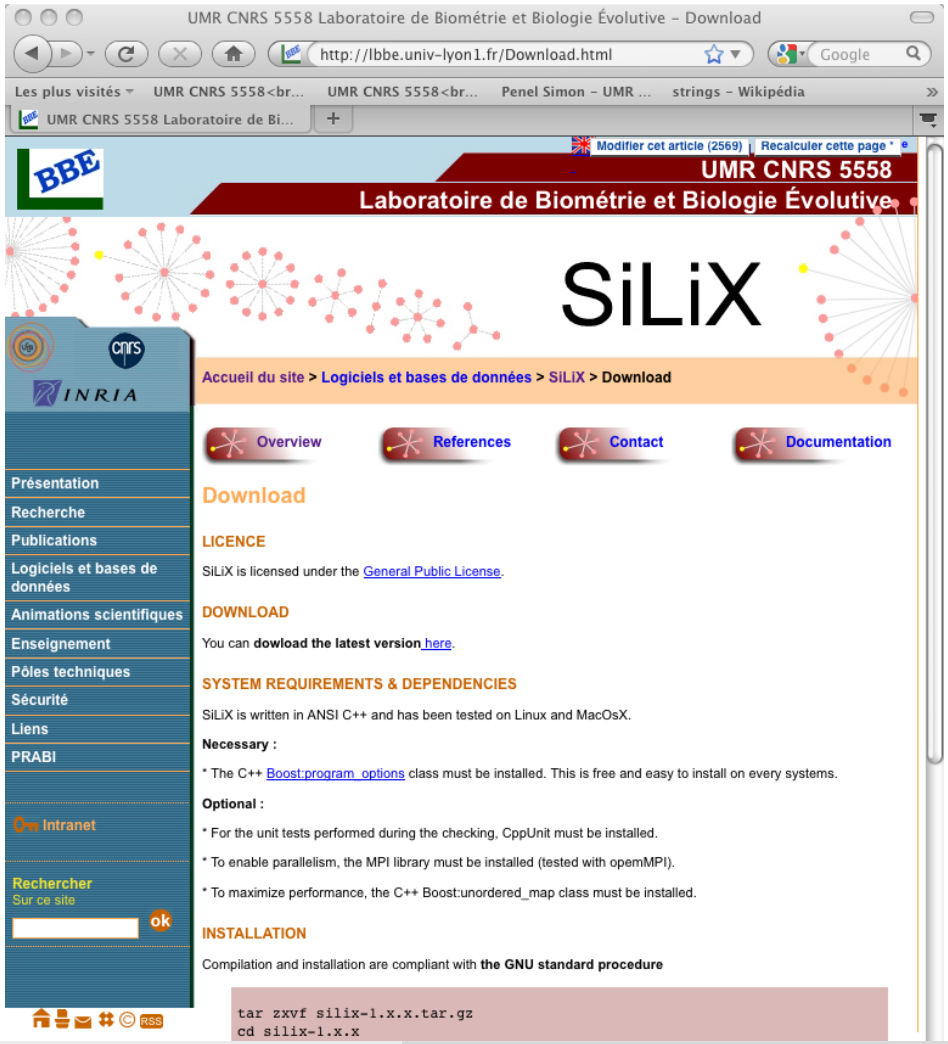
Menu

Availability

Web site

Software (single and parallelised versions) and documentation available at :

`http ://lbbe.univ-lyon1.fr/silix`



Acknowledgements

Pole Informatique LBBE

Bruno Spataro
Stephane Delmotte
Lionel Humblot

Calculation Center IN2P3

Pascal Calvat
Yonny Cardenas
Jean-Yves Nief

Quality comparison

Specificity and Sensibility according to InterPro famillies

(HOGENOM dataset)

method(% identity)	J	Spec.	Sens.
SiLiX (0.25)	0.69	0.73	0.95
SiLiX (0.35)	0.78	0.88	0.89
SiLiX (0.45)	0.74	0.94	0.78
SiLiX (0.25)+$hcluster\_sg^{(100)}$	0.77	0.85	092
hcluster_sg	0.76	0.84	091

$hcluster\_sg^{(100)}$:hcluster_sg on SiLiX families with more than 100 sequences.

Calculation time

hcluster_sg	5 days
SiLiX (0.25)+$hcluster\_sg^{(100)}$	5 hours