Metodi e tecniche alignment-free per l'analisi delle sequenze di Barcode

Antonino Fiannaca, PhD
Massimo La Rosa, PhD
ICAR-CNR, National Research Council of Italy

InterOmics Tutorial Day
Area della Ricerca CNR, Napoli 14-11-2013

Sections

- 1) **Traditional and Compression-Based Approaches**
- 2) Alignment-free Classification techniques

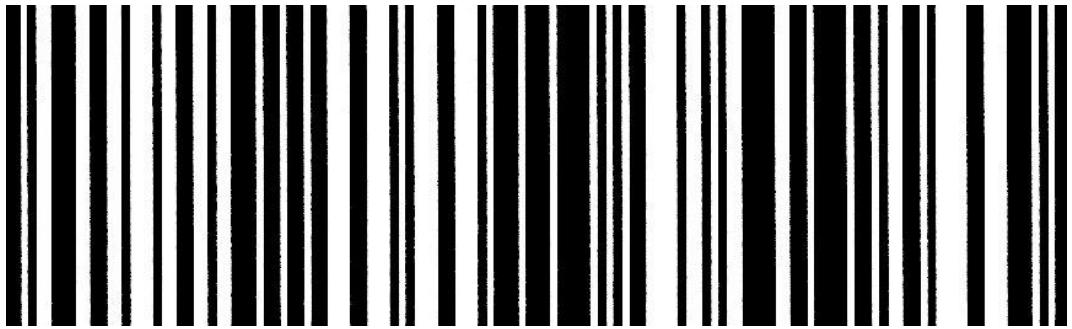
Outline

- Introduction to DNA Barcoding
- Traditional Bioinformatics Approach
- Universal Similarity Metric
  - Proposed Compression-based technique
- Results
- Practice

DNA Barcode

- Very short nucleotide sequence, acting as a unique element used for identification and taxonomic purposes.

- A single gene that works as a true “barcode” providing unique identification

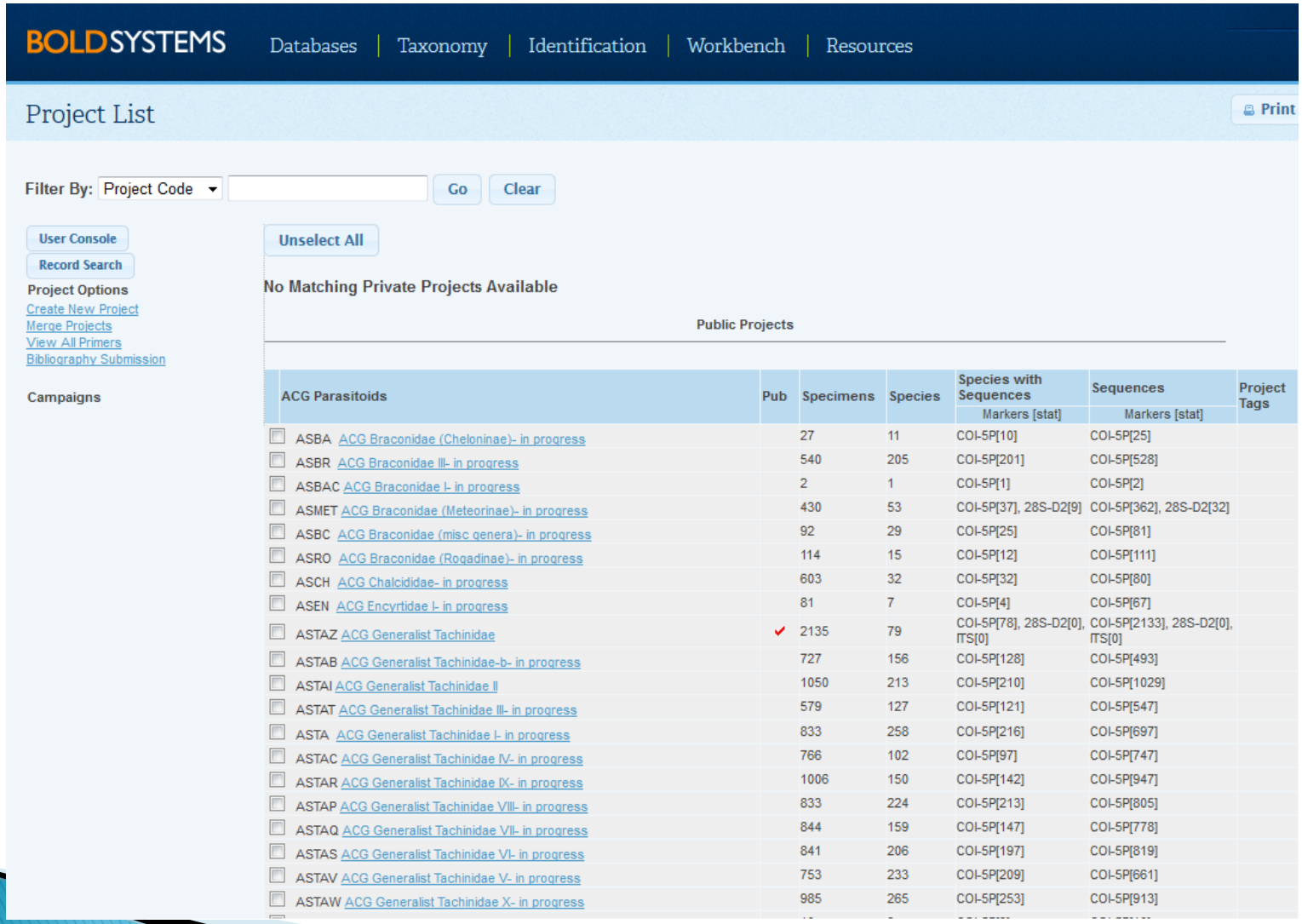


DNA Barcode

- In the animal kingdom, *mitochondrial gene cytochrome c oxidase subunit 1* (COI), about 650 bp long, has proven to be the best barcode sequence.

- DNA barcoding has been used for the study of the biodiversity of several species, such as fishes, birds and some bugs.

BOLD: Barcode of Life Data Systems



BOLD SYSTEMS | Databases | Taxonomy | Identification | Workbench | Resources

Project List

Print

Filter By: Project Code | Go | Clear

User Console

Record Search

Project Options

Create New Project
Merge Projects
View All Primers
Bibliography Submission

Campaigns

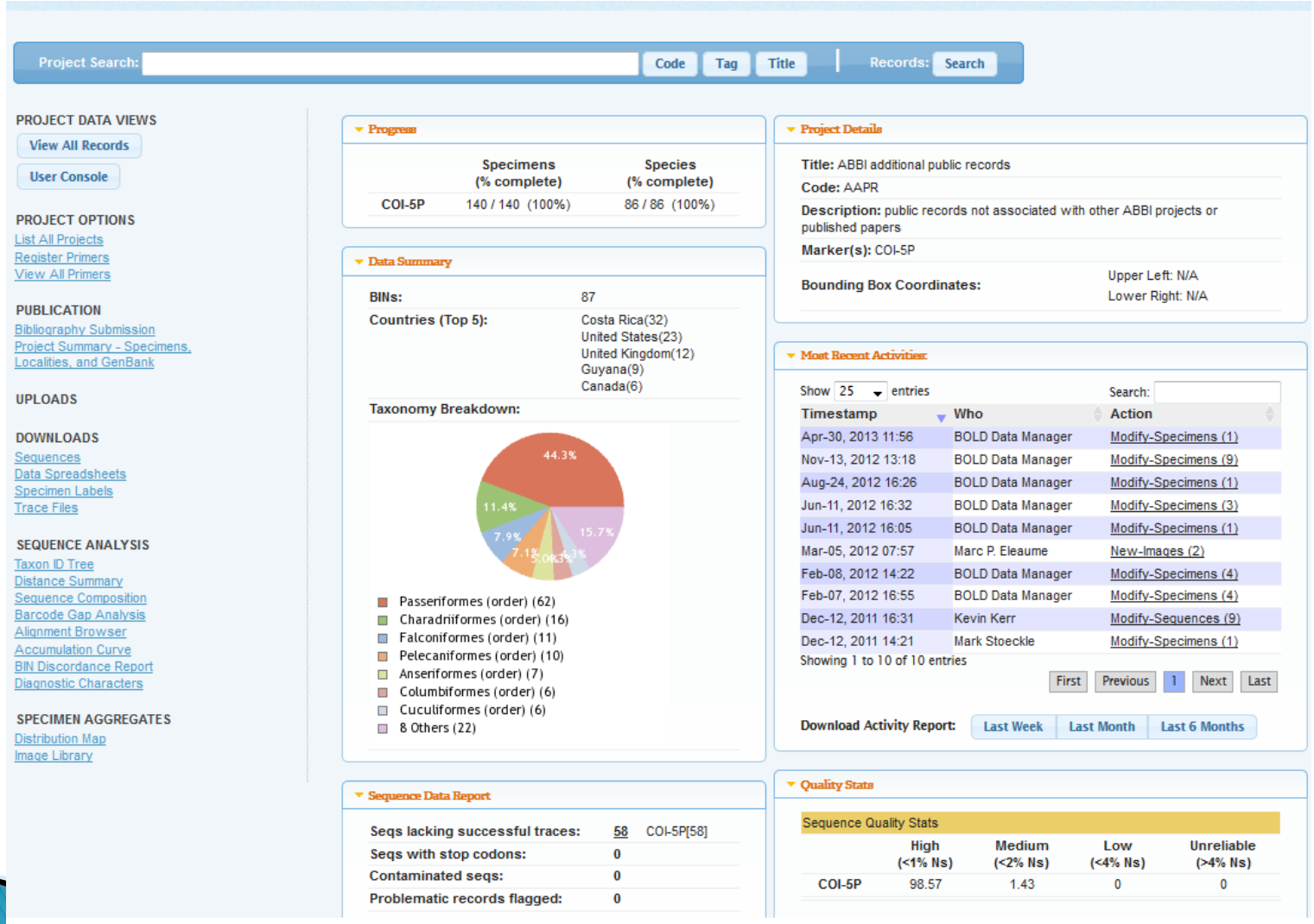
Unselect All

No Matching Private Projects Available

Public Projects

ACG Parasitoids	Pub	Specimens	Species	Species with Sequences Markers [stat]	Sequences Markers [stat]	Project Tags
ASBA ACG Braconidae (Cheloninae)- in progress		27	11	COI-5P[10]	COI-5P[25]	
ASBR ACG Braconidae III- in progress		540	205	COI-5P[201]	COI-5P[528]	
ASBAC ACG Braconidae I- in progress		2	1	COI-5P[1]	COI-5P[2]	
ASMET ACG Braconidae (Meteorinae)- in progress		430	53	COI-5P[37], 28S-D2[9]	COI-5P[362], 28S-D2[32]	
ASBC ACG Braconidae (misc genera)- in progress		92	29	COI-5P[25]	COI-5P[81]	
ASRO ACG Braconidae (Rogadinae)- in progress		114	15	COI-5P[12]	COI-5P[111]	
ASCH ACG Chalcididae- in progress		603	32	COI-5P[32]	COI-5P[80]	
ASEN ACG Encyrtidae I- in progress		81	7	COI-5P[4]	COI-5P[67]	
ASTAZ ACG Generalist Tachinidae	✔	2135	79	COI-5P[78], 28S-D2[0], ITS[0]	COI-5P[2133], 28S-D2[0], ITS[0]	
ASTAB ACG Generalist Tachinidae-b- in progress		727	156	COI-5P[128]	COI-5P[493]	
ASTAI ACG Generalist Tachinidae II		1050	213	COI-5P[210]	COI-5P[1029]	
ASTAT ACG Generalist Tachinidae III- in progress		579	127	COI-5P[121]	COI-5P[547]	
ASTA ACG Generalist Tachinidae I- in progress		833	258	COI-5P[216]	COI-5P[697]	
ASTAC ACG Generalist Tachinidae IV- in progress		766	102	COI-5P[97]	COI-5P[747]	
ASTAR ACG Generalist Tachinidae IX- in progress		1006	150	COI-5P[142]	COI-5P[947]	
ASTAP ACG Generalist Tachinidae VIII- in progress		833	224	COI-5P[213]	COI-5P[805]	
ASTAQ ACG Generalist Tachinidae VII- in progress		844	159	COI-5P[147]	COI-5P[778]	
ASTAS ACG Generalist Tachinidae VI- in progress		841	206	COI-5P[197]	COI-5P[819]	
ASTAV ACG Generalist Tachinidae V- in progress		753	233	COI-5P[209]	COI-5P[661]	
ASTAW ACG Generalist Tachinidae X- in progress		985	265	COI-5P[253]	COI-5P[913]	

BOLD: Barcode of Life Data Systems



Traditional Analysis

- Well consolidated bioinformatics techniques:
  - Sequence alignment
  - Computation of evolutionary distances
  - Inference of Phylogenetic trees.

Sequence Alignment

- Much of bioinformatics involves sequences
  - DNA sequences
  - RNA sequences
  - Protein sequences
- We can think of these sequences as strings of letters
  - DNA & RNA: |alphabet|=4
  - Protein: |alphabet|=20

Sequence Alignment

- Main purposes:
  - Comparison among sequences
  - Finding similarities
  - Building phylogenetic trees
- Three types of alignment:
  - Local Alignment (Smith & Waterman, BLAST)
  - Global Alignment (Needleman & Wunsch)
  - Multiple Alignment (ClustalW)

Sequence Alignment

- Global vs Local Alignment

```
--T--CC-C-AGT--TATGT-CAGGGGACACG--A-GCATGCAGA-GAC
  |  || |  ||  | | | |||    || |  | |  | ||||   |
AATTGCCGCC-GTCGT-T-TTCAG----CA-GTTATG--T-CAGAT--C
```

```
                    tccCAGTTATGTCAGgggacacgagcatgcagagac
                       ||||||||||||
aattgccgccgtcgttttcagCAGTTATGTCAGatc
```

- Multiple Alignment

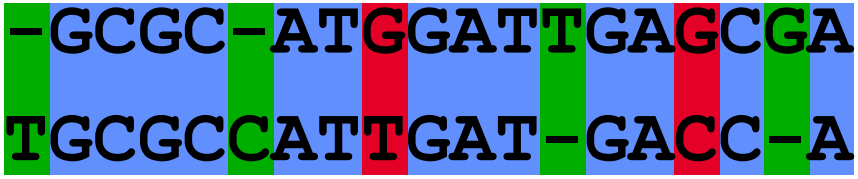
```
AGCCTTGTCATCCGTATC-TTTCAA----
AGCCTTGTCATCCGTATC-TTTCA-----
-GCCTTGTCATCCGTATC-TTTCAACG--
--CCTTGTCATCCGTATC-TTTCAACGTG
--CCTTGTCATCCGTATC-TTTCAAC---
---CTTGTCATCCGTATCTTTTCAAC---
---CTTGTCATCCGTATC-T---------
----TTGTCATCCGTATC-TT--------
------GTCATCCGTATC-TTTCAACGTG
------GTCATCCGTATC-TTTCAACGTG
--------CATCCGTATC-TTTCAA----
--------CATCCGTATC-TTTCA-----
---------ATCCGTATC-TTTCAACGTG
```

Sequence Alignment: Example

- **Input**: two sequences over the same alphabet
  - GCGCATGGATTGAGCGA and GCGCCATTGATGACCA
- **Output**: an alignment of the two sequences

```
-GCGC-ATGGATTGAGCGA
TGCGCCATTGAT-GACC-A
```

Sequence Alignment: Example



Three elements:

- Perfect matches
- Mismatches
- Insertions & deletions (indel)

Sequence Alignment: Example

- Score each position independently (Substitution Matrix):
  - Match: +1
  - Mismatch: -1
  - Indel: -2

- Score of an alignment is sum of position scores

```
-GCGC-ATGGATTGAGCGA
TGCGCCATTGAT-GACC-A
```

Score: (+1x13) + (-1x2) + (-2x4) = 3

```
------GCGCATGGATTGAGCGA
TGCGCC----ATTGATGACCA--
```

Score: (+1x5) + (-1x6) + (-2x11) = -23

Evolutionary Distances

- Compute (dis-)similarity among aligned sequences:
  - Number of substituions per site:

$$p = \frac{\text{Number of different nucleotides}}{\text{Total number of compared nucleotides}}$$

- It understimates the real number of substitutions because of biological phenomena like multiple hits

Evolutionary Distances

- Several stochastic models based on a set of *a priori* assumptions:
  - All sites evolve in an independent way
  - All sites can change with the same probability
  - All kinds of substitutions are equally probable
  - …
- The more complex the model, the less number of assumptions

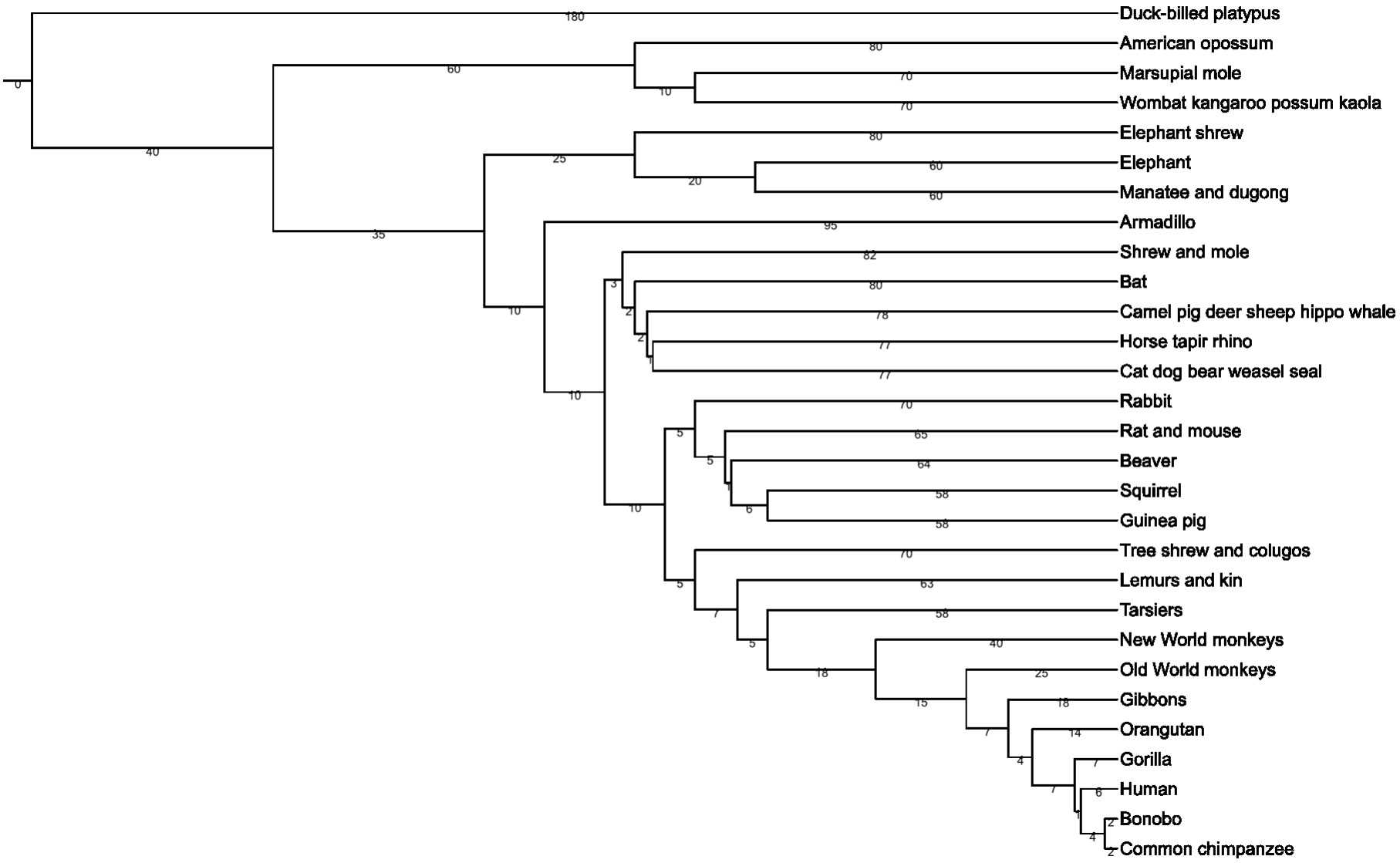
Evolutionary Distances

- Most common stochastic models (from simpler to more complex):
  - Jukes and Cantor (1969)
  - Kimura (1980)
  - Tamura (1992)
  - Tajima and Nei (1982)
- They DO NOT represent a metric!
  - i.e. NO triangle inequality

Phylogenetic Trees

- Exploit evolutionary relations among species
- Hierarchical structure made of nodes and branches:
  - Terminal nodes (leaves): taxa
  - Internal nodes: ancestor taxa
  - Branches: link two nodes. Their length proportional to the (evolutionary) distance between nodes

Phylogenetic Trees



Phylogenetic Trees

- Distance-Based algorithms:
  - Unweighted Pair Group Method with Aritemtic Mean (UPGMA)
  - Neighbor-Joining
- Sequence-Based algorithms:
  - Maximum Parsimony
  - Maximum Likelihood

Drawbacks of traditional analysis

- Sequence Alignment needs a lot of parameters and does not give a unique result
- High computation time for long sequences ($O(n^2)$)
- Different distance models according to a priori assumptions
- Evolutionary distances are stochastic models that do not define a distance metric

Alternative analysis

- An alignment-free methodological approach, based on compression-based distances derived from Universal Similarity Metric (USM):
  - Does not require a prior alignment of genomic sequences.
  - Parameterless
  - Strong theoretical assumptions
  - Definition of a distance metric

Universal Similarity Metric (USM)

- USM [Li et al.,'04] is a class of distance measures, defined in terms of the *Kolmogorov* complexity.

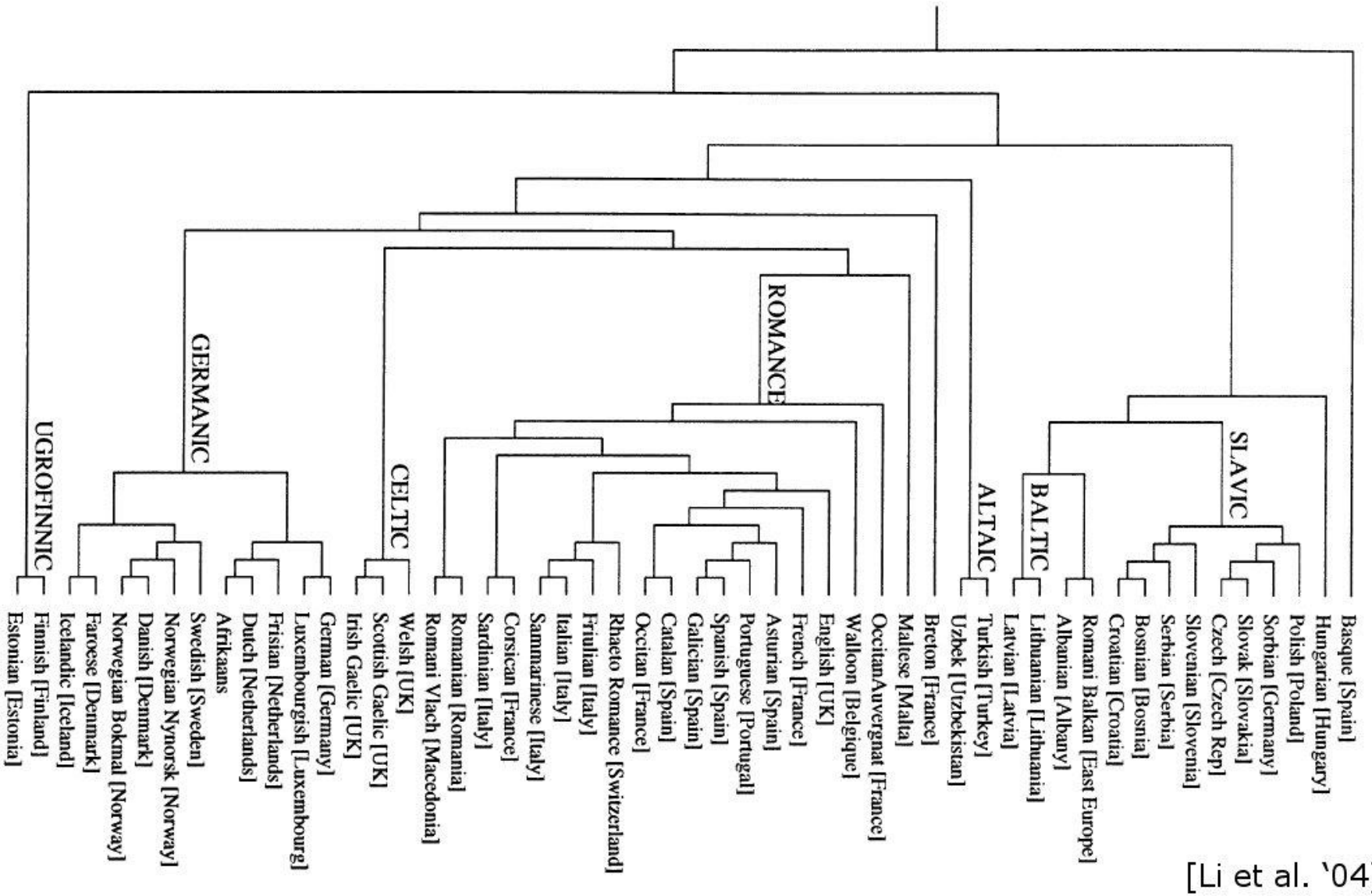
$$\mathrm{USM} = \frac{\mathrm{ID}(x,y)}{\max\{K(x),\ K(y)\}} = \frac{\max\{K(x|y),\ K(y|x)\}}{\max\{K(x),\ K(y)\}}$$

- USM is a metric, normalized (it ranges between 0 and 1), is "universal".

Universal Similarity Metric (USM)

- Universality:
  - Text files
  - Images
  - Music files
  - …
- Used for example to compute linguistic trees among different languages

Universal Similarity Metric (USM)



[Li et al. '04]

Kolmogorov Complexity

- Definitions:
  - "*The Kolmogorov complexity K(x) of a string x is the length of the shortest binary program x* to compute x on a universal Turing machine*"
  - *"K(x|y) is the conditional Kolmogorov complexity of two strings, x and y, defined as the length of the shortest binary program that produces x as output, given the input y"*
- It represents a theoretic concept $=>$
  it is NOT computable!, only approximated

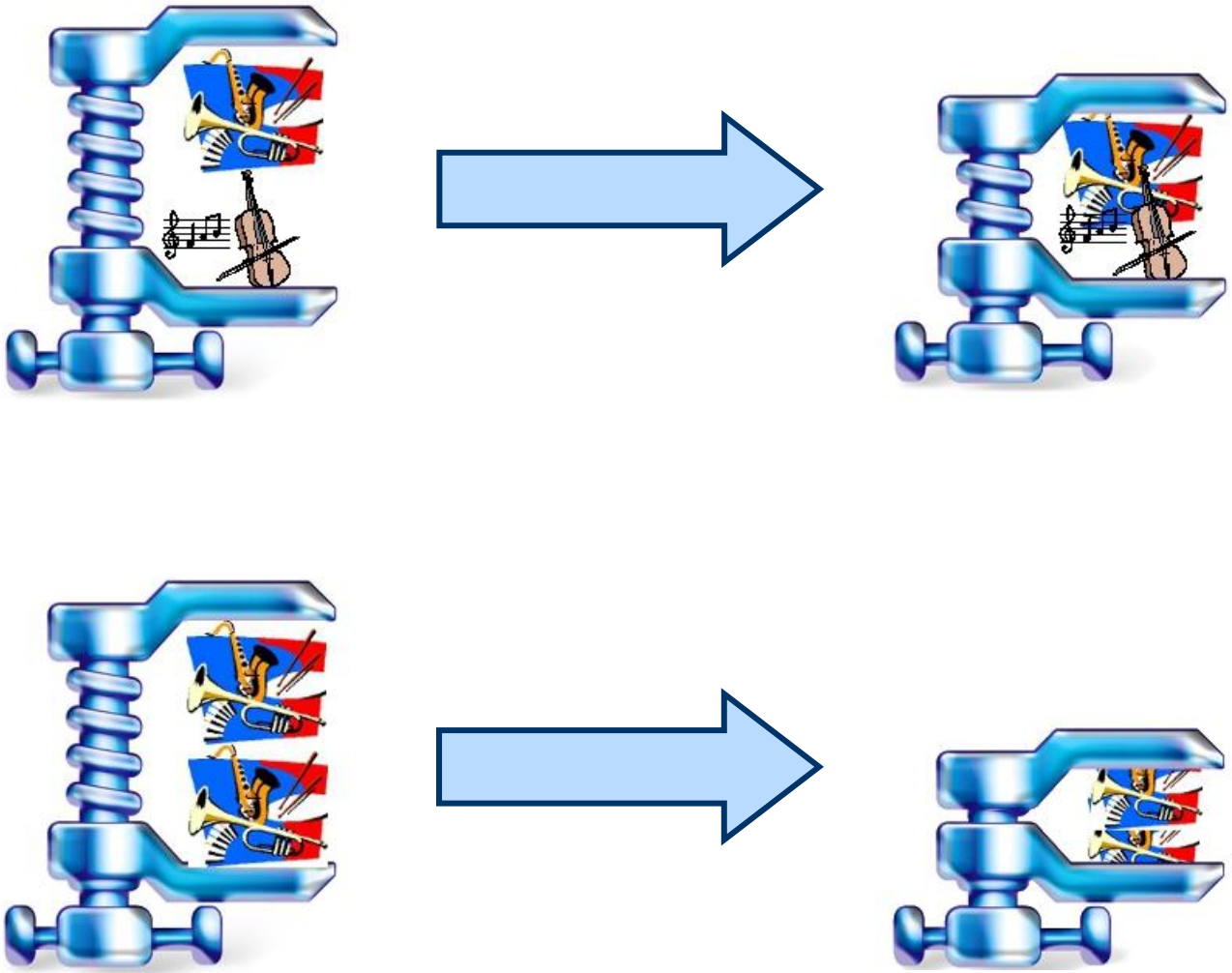
Compression-based Distances

- Normalized Compression Distance (NCD):

$$\mathrm{NCD}(x,y) = \frac{C(xy) - \min\{C(x), C(y)\}}{\max\{C(x), C(y)\}}$$

- C(*x*) is the size, in byte, of the compression version of string *x*
- C(*xy*) is the size of the compressed version of the concatenation of string *x* and *y*

NCD: How it works

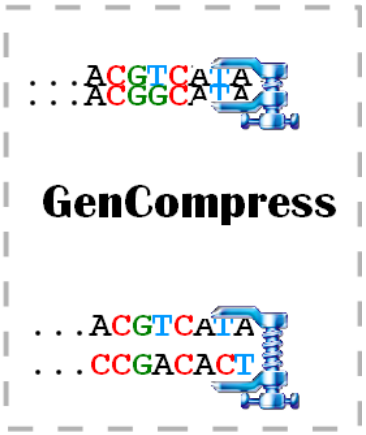


NCD: How it works

- Normalised Compression Distance

$$\mathrm{NCD}(x,y) = \frac{C(xy) - \min\{C(x), C(y)\}}{\max\{C(x), C(y)\}}$$

Seq1: . . . ACGTCATA . .
Seq2: . . . ACGGCATA . .



High compression → Low distance

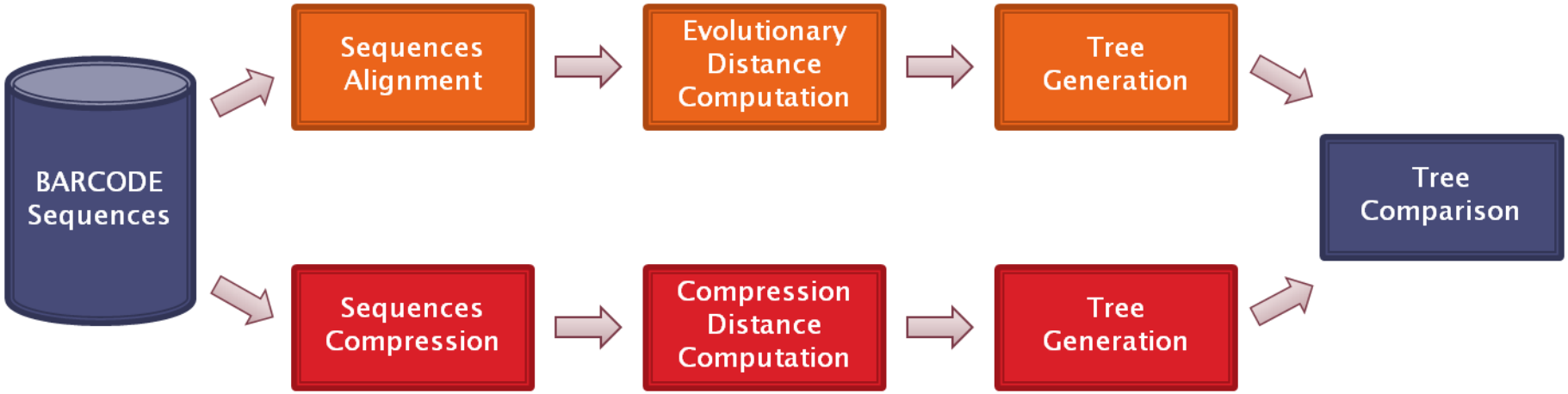
Seq1: . . . ACGTCATA . .
Seq3: . . . CCGACACT . .

Low compression → High distance

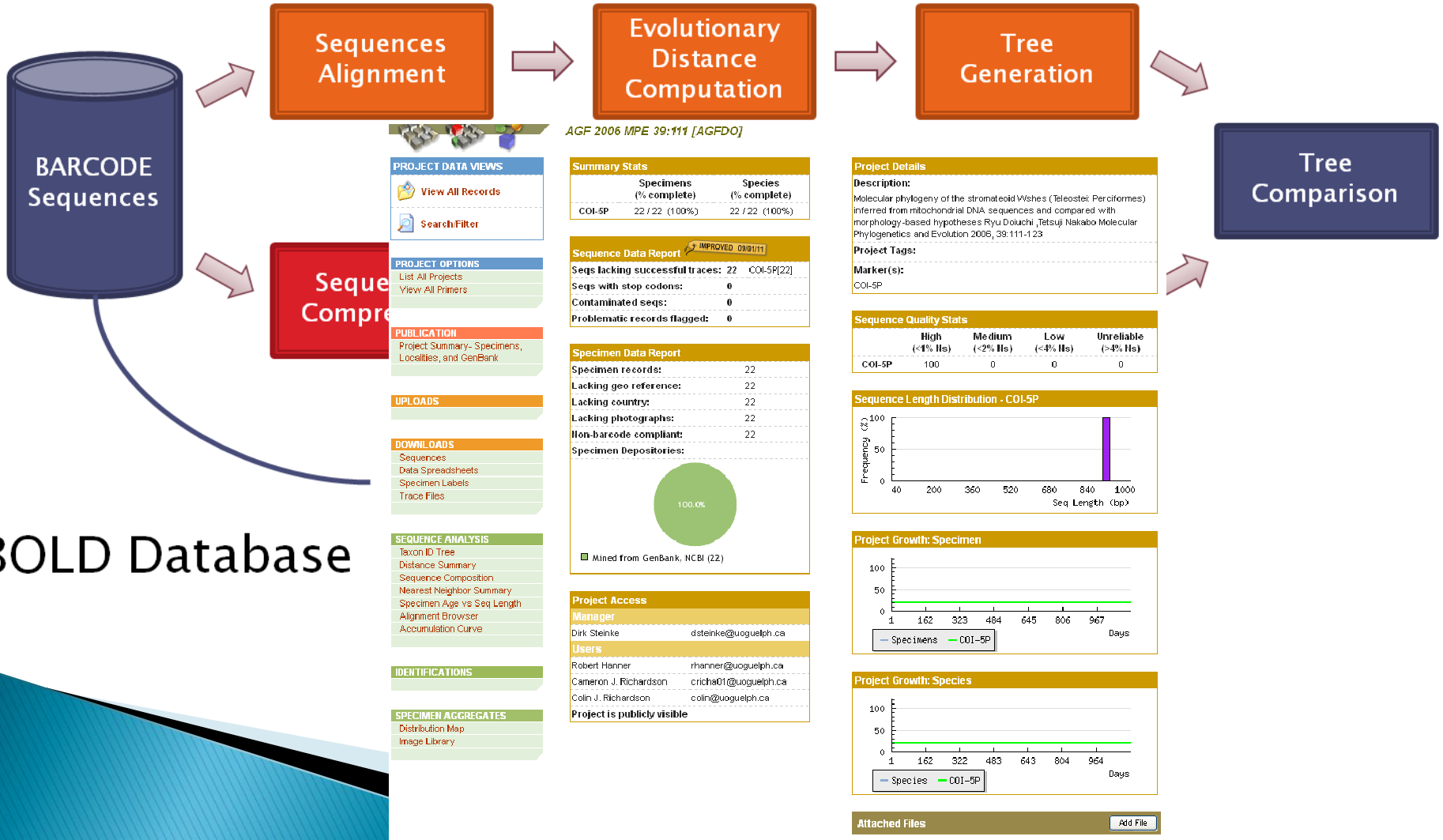
GenCompress

- String compression algorithms find portions of input string that are repeated and substitute them with a shorter reference
- The set of repeated string portions is indicated as “dictionary”.
- GenCompress dictionary based compressor optimized to work with DNA sequences, having only a 4 letter (A,C,G,T) alphabet

Proposed Methodology

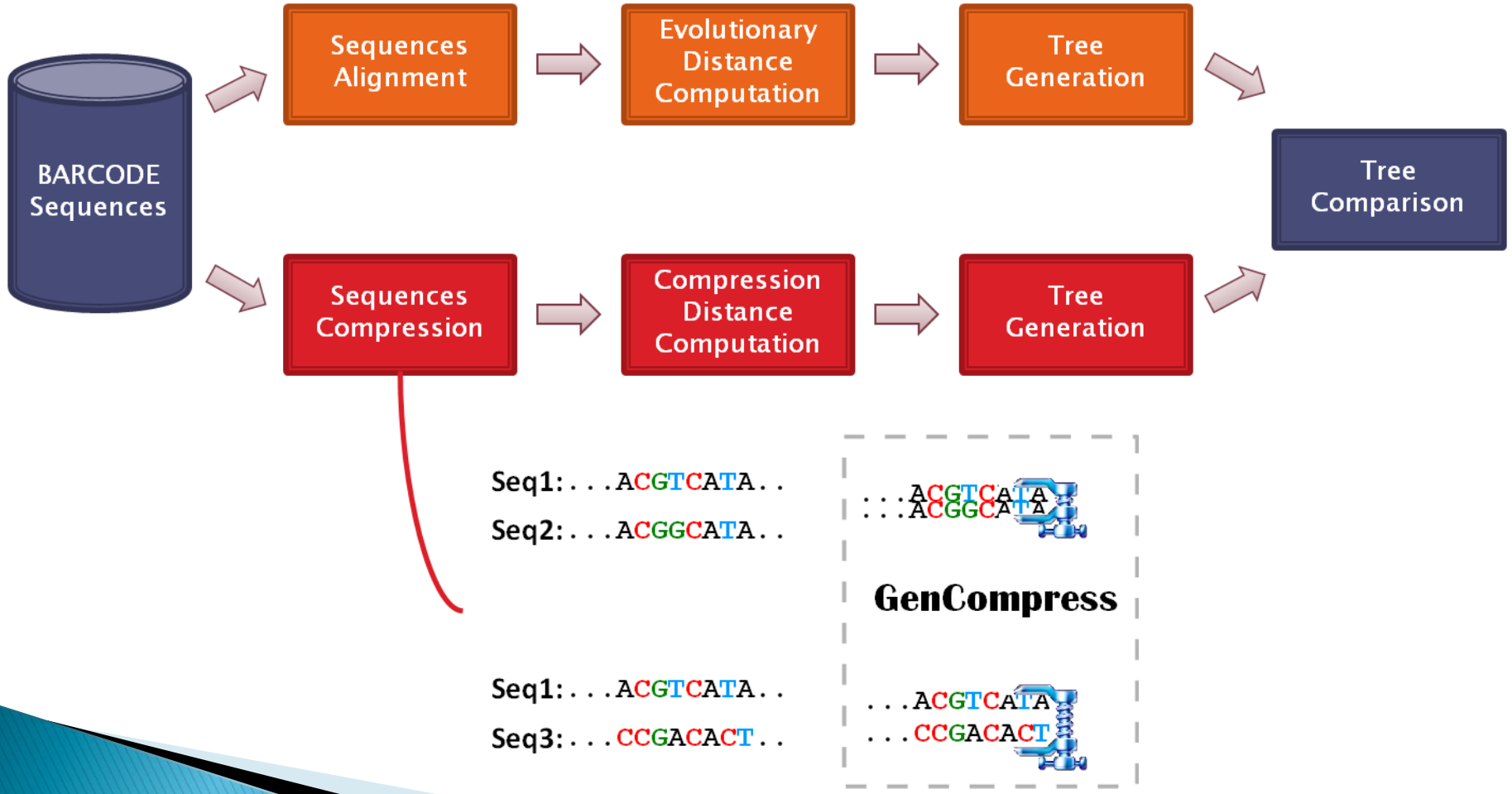


Proposed Methodology

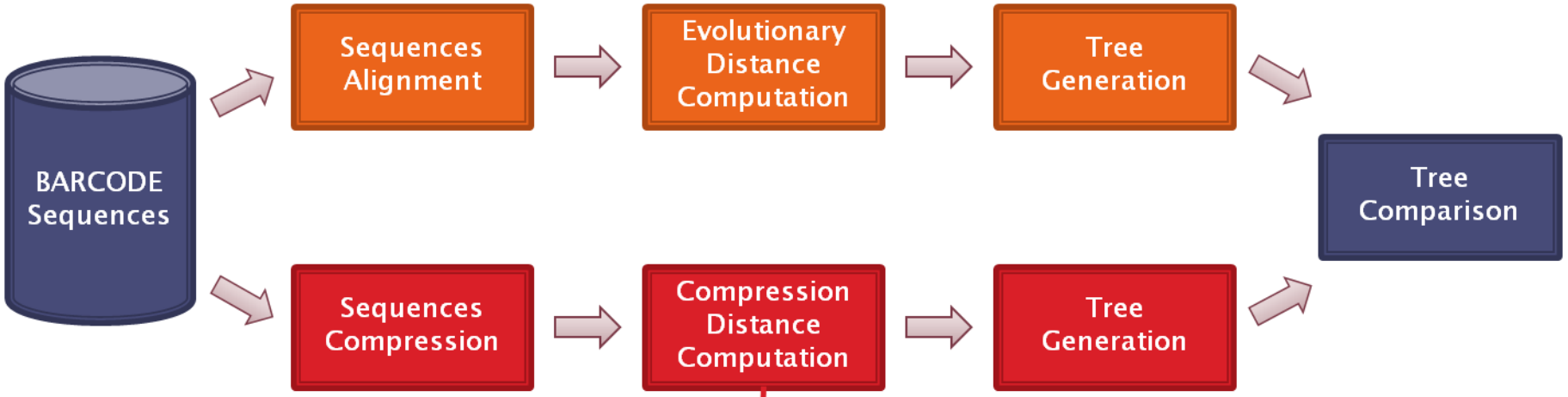


BOLD Database

Proposed Methodology

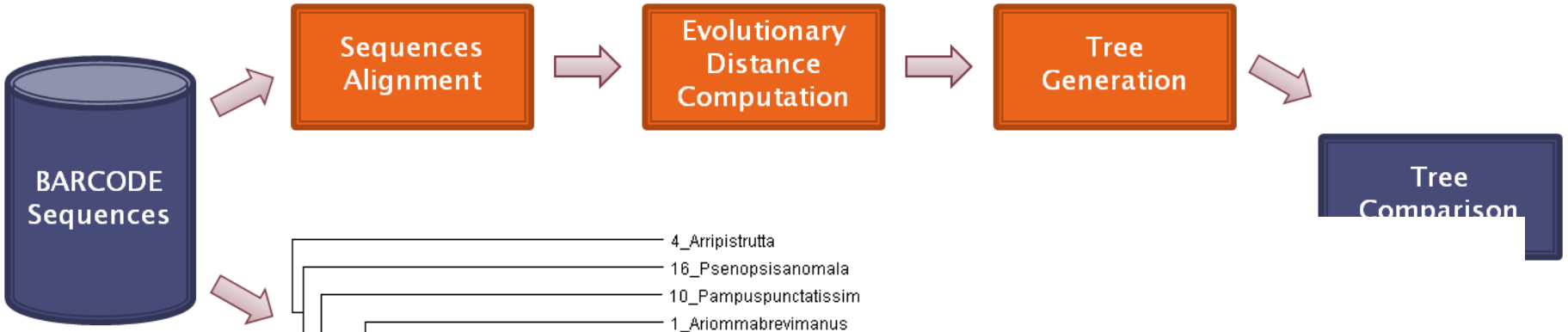


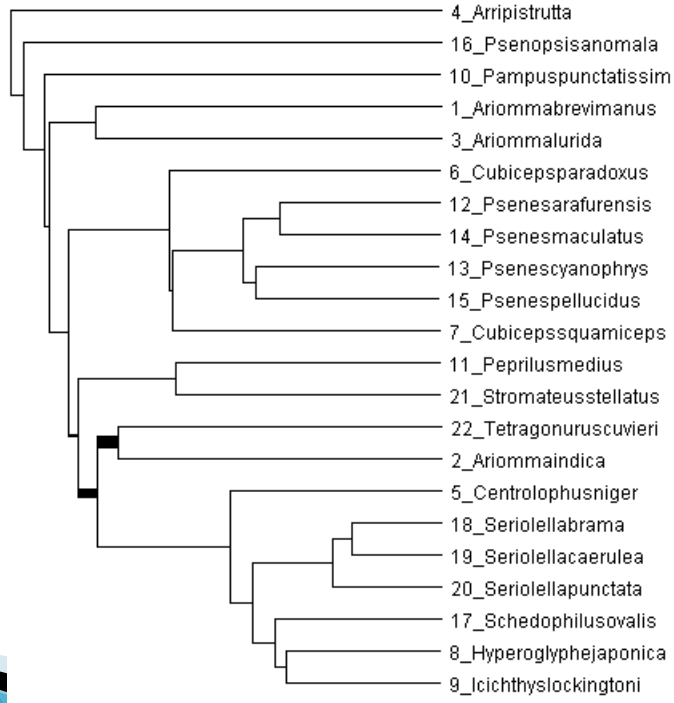
Proposed Methodology



$$\mathrm{NCD}(x,y) = \frac{C(xy) - \min\{C(x), C(y)\}}{\max\{C(x), C(y)\}}$$

Proposed Methodology

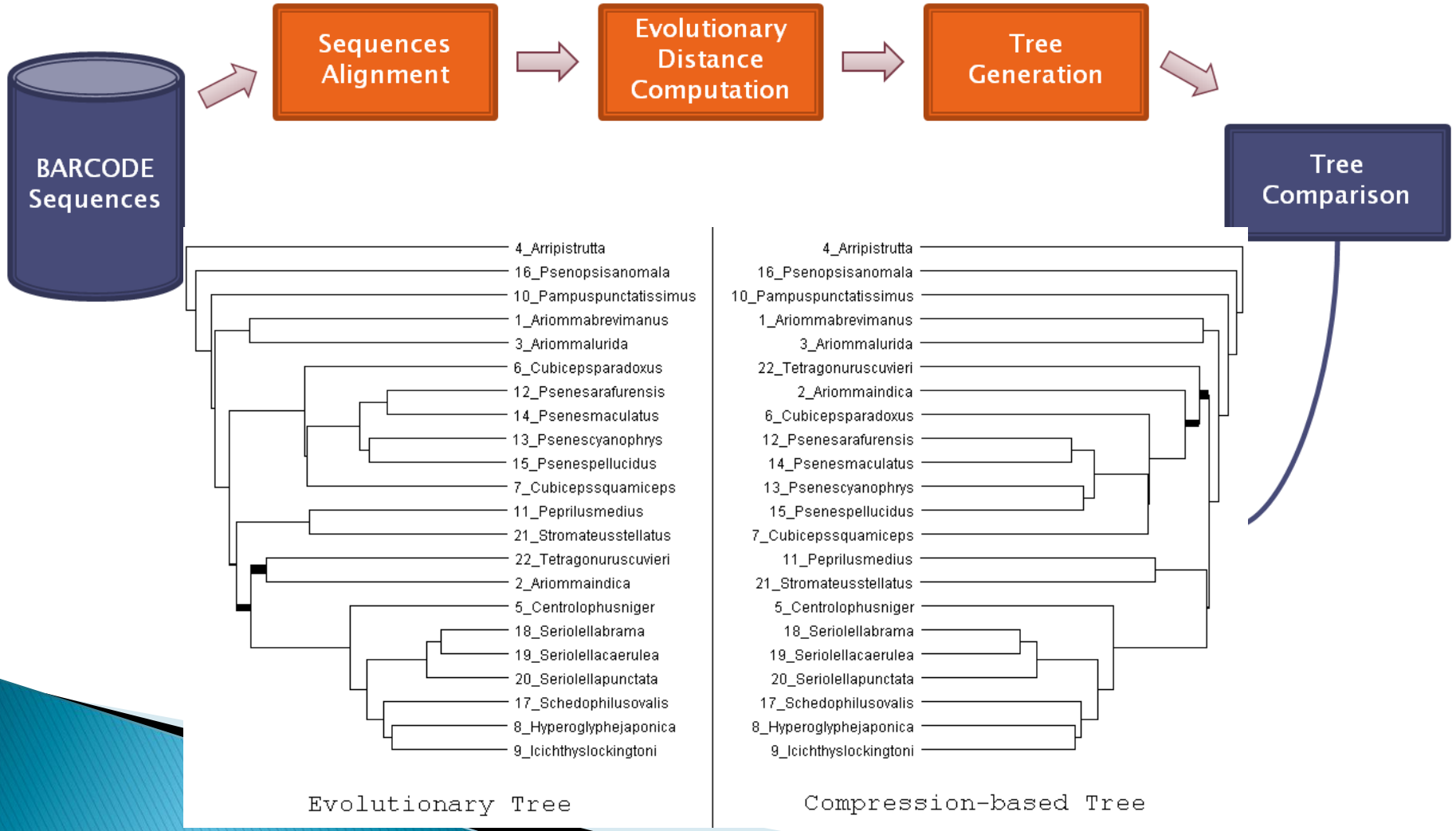




Evolutionary Tree

Neighbour Joining (NJ)
UPGMA

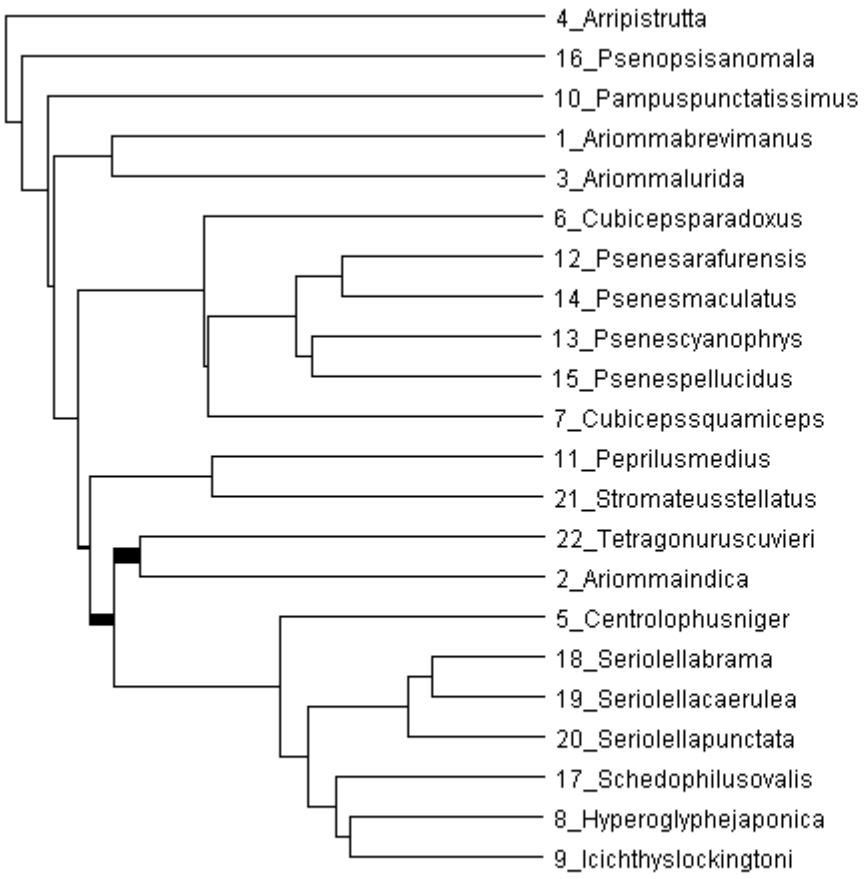
Proposed Methodology



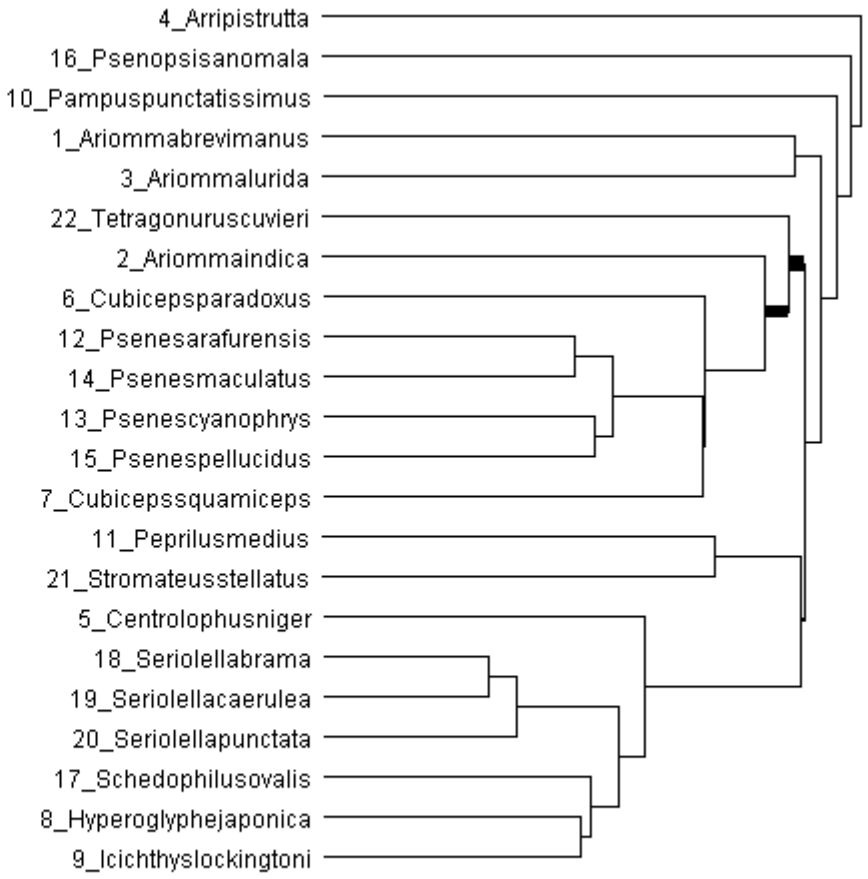
Tree Comparison

- Nye et al. (2006) algorithm
  - It considers topological features
  - It builds a sort of alignment between the two trees to compare
  - It compares the shared leaf nodes belonging to two corresponding partitions

Tree Comparison

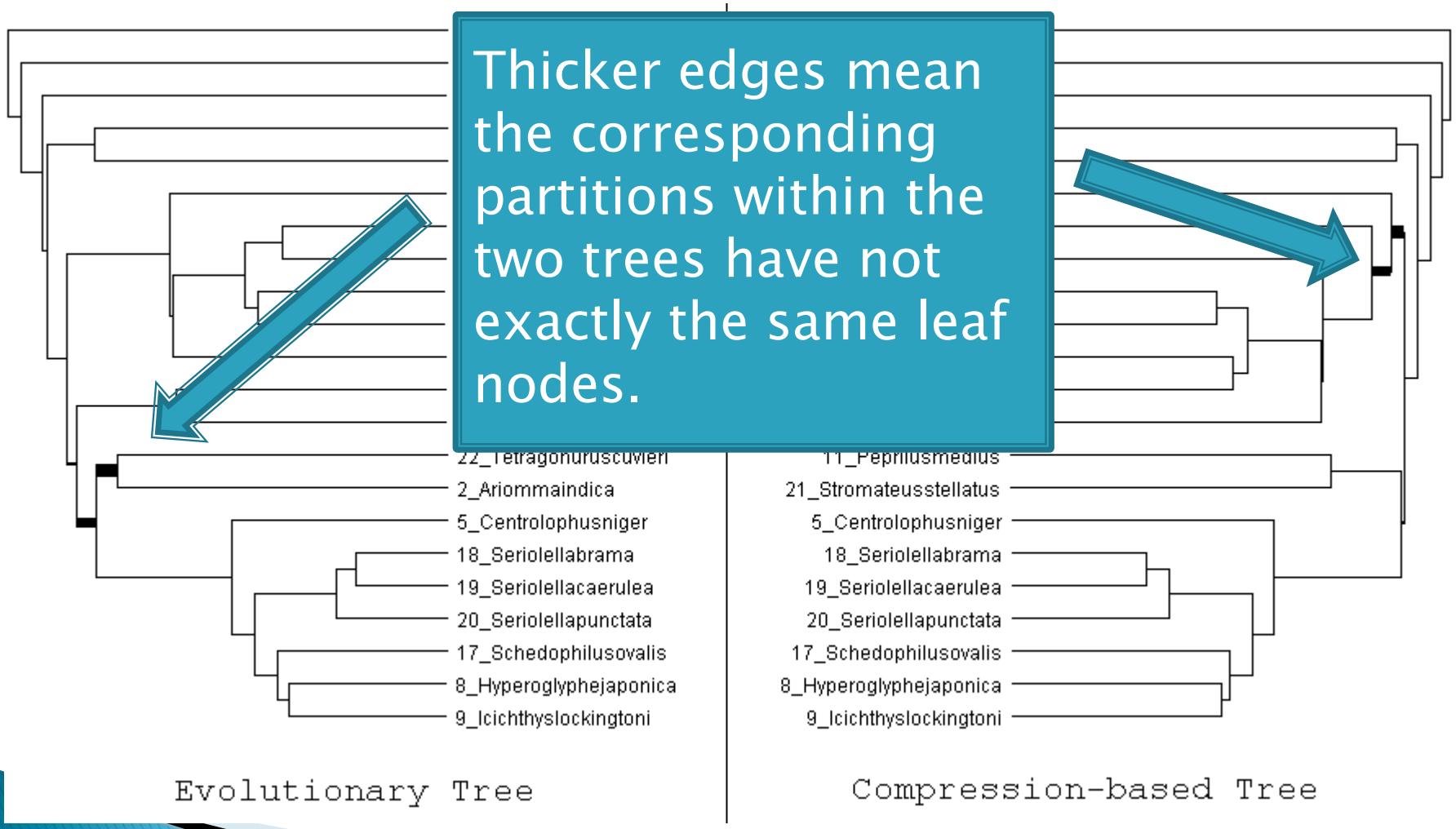


Evolutionary Tree

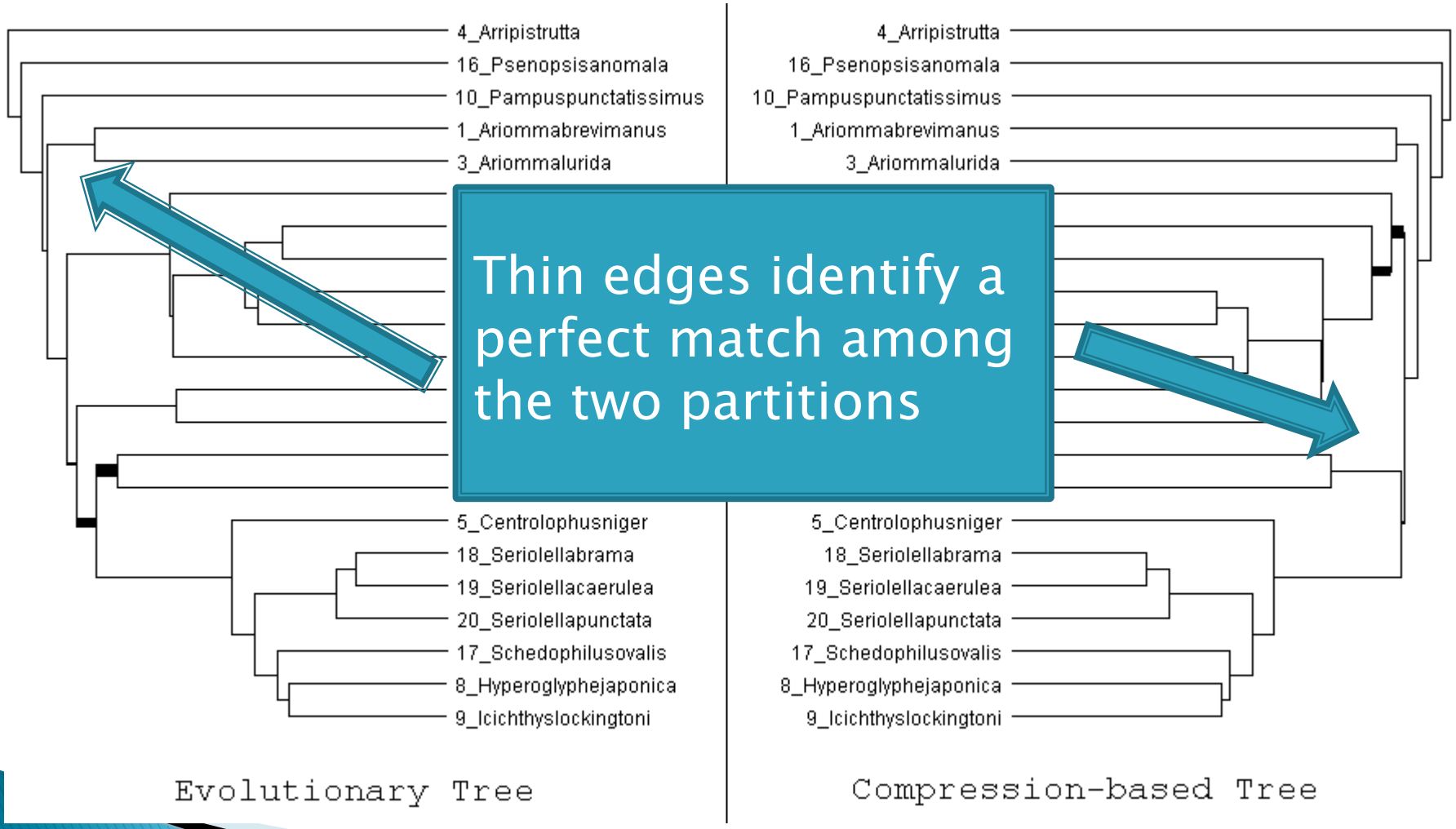


Compression-based Tree

Tree Comparison



Tree Comparison



Experimental Setup

- 30 input datasets from BOLD database

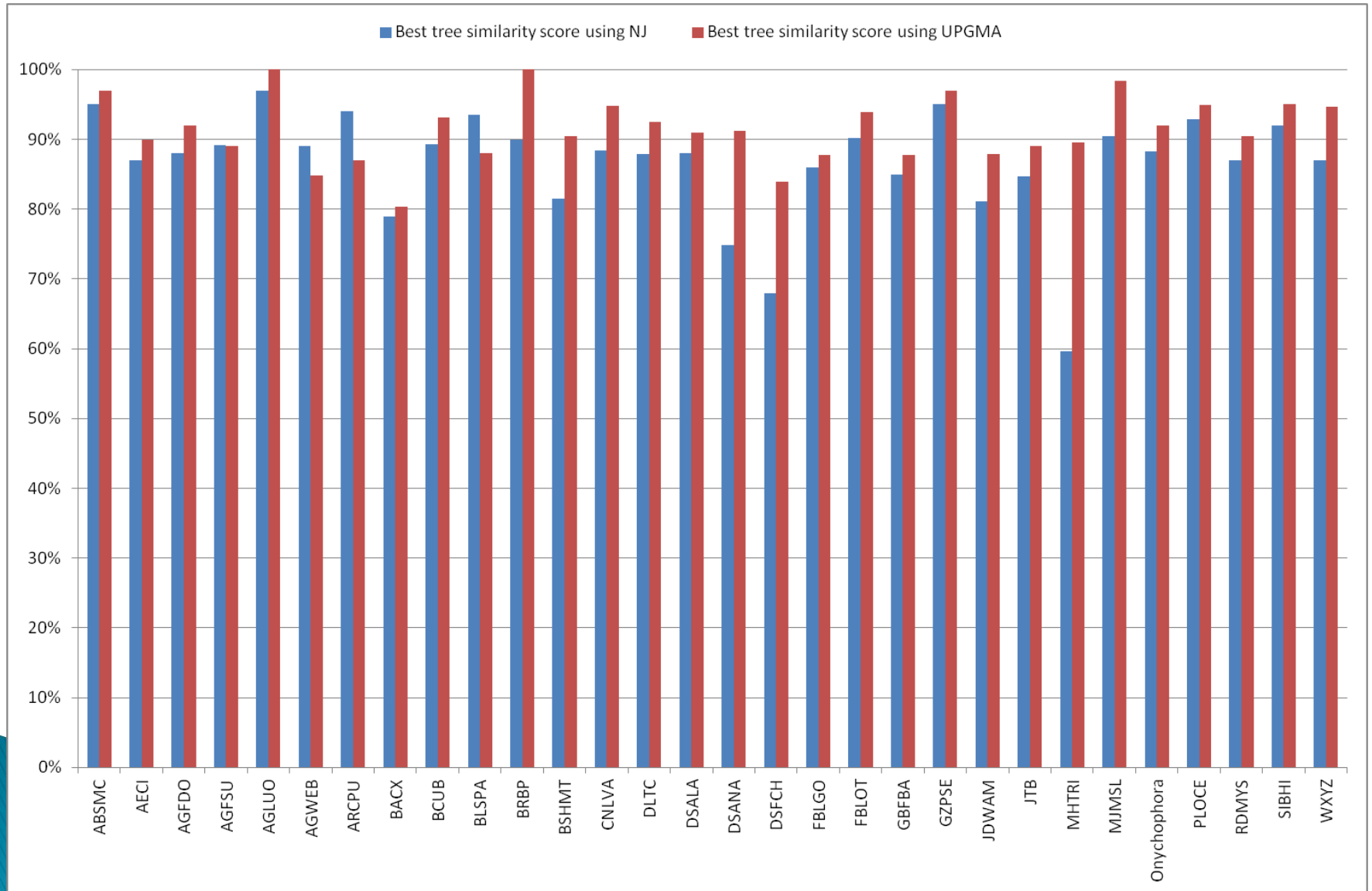
Table 2 Barcode datasets description.

DATASET	# Species	# Specimens	% Sequences with undefined bases	Sequence Length
ABSMC	46	72	1.3%	650-657
AECI	30	30	0.0%	605-679
AGFDO	22	22	0.0%	901
AGFSU	42	48	2.0%	633-639
AGLUO	38	46	2.1%	630
AGWEB	33	33	87.0%	900
ARCPU	28	52	5.0%	625-658
BACX	74	119	2.5%	616-657
BCUB	30	108	0.9%	657
BLSPA	86	86	4.0%	604-658
BRBP	17	106	0.0%	658
BSHMT	22	141	5.6%	645
CNLVA	33	73	5.0%	625-658
DLTC	40	67	1.5%	689-1821
DSALA	12	44	11.0%	649-651
DSANA	14	274	0.0%	652
DSFCH	17	173	3.4%	620-650
FBLGO	44	122	2.4%	580-658
FBLOT	34	64	3.0%	419-658
GBFBA	27	27	7.0%	669
GZPSE	23	78	7.7%	601-658
JDWAM	103	226	8.8%	620-650
JTB	53	225	0.4%	658-899
MHTRI	13	108	3.7%	620-650
MJMSL	76	198	4.5%	559-658
Onychophora	52	210	0.9%	451-884
PLOCE	33	102	0.0%	620-660
RDMYS	6	37	32.0%	636
SIBHI	38	85	0.0%	650-694
WXYZ	9	34	3.0%	650-680

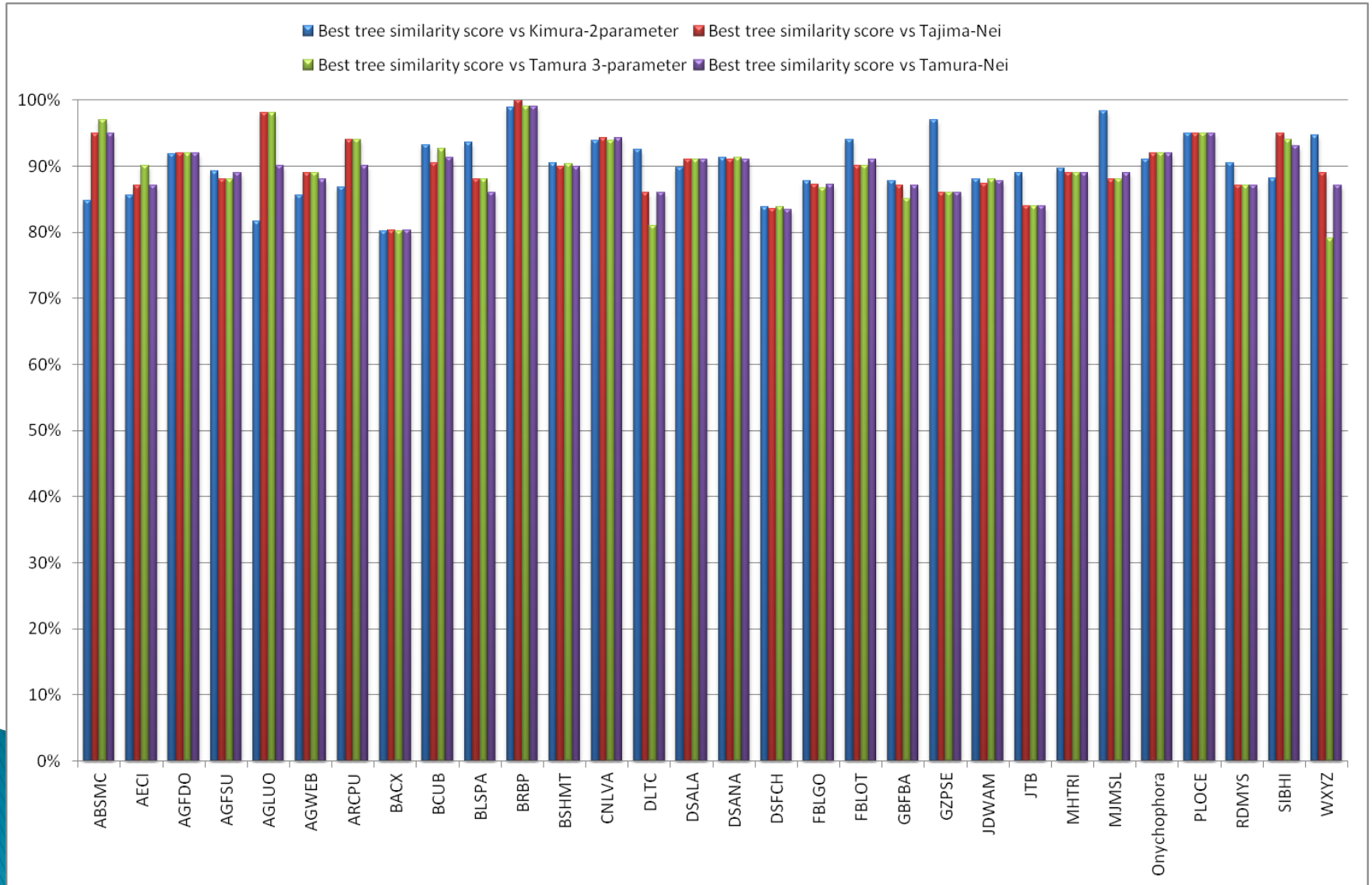
Experimental Setup

- Evolutionary distance by means of different models
  - Kimura 2-parameter
  - Tajima-Nei
  - Tamura 3-parameter
  - Tamura-Nei
- Evolutionary distances were computed using MEGA 5 software (http://www.megasoftware.net/mega.php)

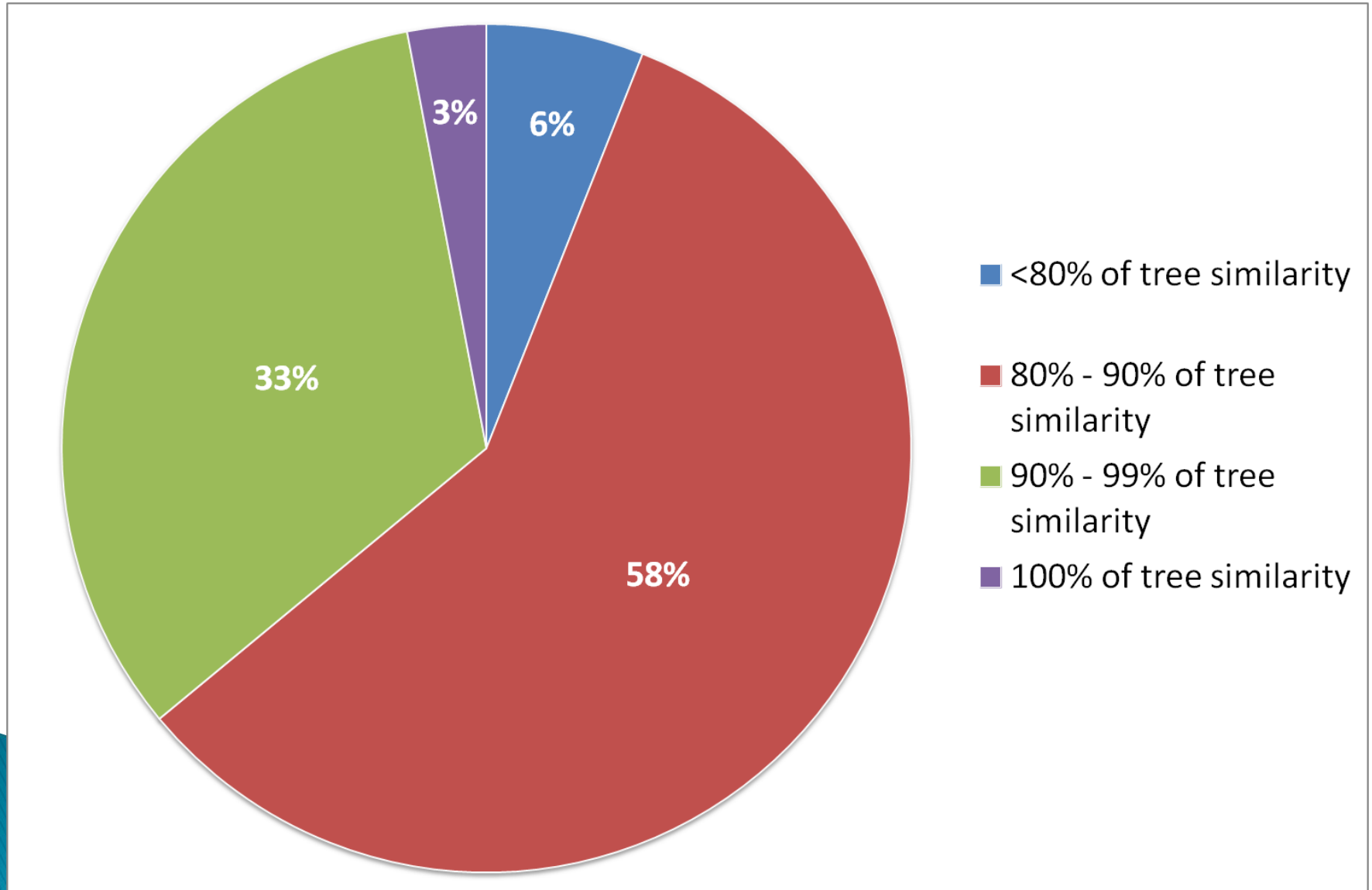
Experimental Results



Experimental Results



Experimental Results



Experimental Results: Details

- Trees obtained from compression-based methods are very similar (above 90%) to the ones built from classic distance
- The best results (>95%) with pure barcode datasets: about 650 sequence length and no undefined nucleotides
- Lower score (82%) for datasets with high percentage of undefined bases (noisy)
  - GenCompress works as a generic ASCII string compressor giving low compression ratios

Traditional vs Compression-Based Analysis

- No need of Sequence alignment (of course!)
- Parameterless approach
- Parallelizable approach
- USM, and NCD, define a distance metric and hold on strong theroretical aasumption
  - Information theory
  - Kolmogorov complexity
- It works with short barcode sequences

For further results and analysis

La Rosa *et al. BMC Bioinformatics* 2013, **14**(Suppl 7):S4
http://www.biomedcentral.com/1471-2105/14/S7/S4

BMC Bioinformatics

RESEARCH Open Access

Alignment-free analysis of barcode sequences by means of compression-based methods

Massimo La Rosa*, Antonino Fiannaca, Riccardo Rizzo, Alfonso Urso

From Ninth Annual Meeting of the Italian Society of Bioinformatics (BITS)
Catania, Sicily. 2-4 May 2012

Abstract

Background: The key idea of DNA barcode initiative is to identify, for each group of species belonging to different kingdoms of life, a short DNA sequence that can act as a true taxon barcode. DNA barcode represents a valuable type of information that can be integrated with ecological, genetic, and morphological data in order to obtain a more consistent taxonomy. Recent studies have shown that, for the animal kingdom, the mitochondrial gene cytochrome c oxidase I (COI), about 650 bp long, can be used as a barcode sequence for identification and taxonomic purposes of animals. In the present work we aims at introducing the use of an alignment-free approach in order to make taxonomic analysis of barcode sequences. Our approach is based on the use of two compression-based versions of non-computable Universal Similarity Metric (USM) class of distances. Our purpose is to justify the employ of USM also for the analysis of short DNA barcode sequences, showing how USM is able to correctly extract taxonomic information among those kind of sequences.

Results: We downloaded from Barcode of Life Data System (BOLD) database 30 datasets of barcode sequences belonging to different animal species. We built phylogenetic trees of every dataset, according to compression-based and classic evolutionary methods, and compared them in terms of topology preservation. In the experimental tests, we obtained scores with a percentage of similarity between evolutionary and compression-based trees between 80% and 100% for the most of datasets (94%). Moreover we carried out experimental tests using simulated barcode datasets composed of 100, 150, 200 and 500 sequences, each simulation replicated 25-fold. In this case, mean similarity scores between evolutionary and compression-based trees span between 83% and 99% for all simulated datasets.

Conclusions: In the present work we aims at introducing the use of an alignment-free approach in order to make taxonomic analysis of barcode sequences. Our approach is based on the use of two compression-based versions of non-computable Universal Similarity Metric (USM) class of distances. This way we demonstrate the reliability of compression-based methods even for the analysis of short barcode sequences. Compression-based methods, with their strong theoretical assumptions, may then represent a valid alignment-free and parameter-free approach for barcode studies.

Practice

- Sequence Compression
  - Parsefasta.py, Concatena.py, Gencompress.py
- NCD Computation
  - NCD.py
- Tree generation
  - MEGA software
- Tree comparison
  - Phylocore.jar

Parsefasta.py

parsefasta.py - D:\slides\tutorial\script\script_tutorial\parsefasta.py

File Edit Format Run Options Windows Help

```
def parsefasta(dataset):

    from Bio import SeqIO

##estre le sequenze come file di testo e li salva
    cont = 1
    for seq_record in SeqIO.parse(dataset, "fasta"):
        sequenza = seq_record.seq.tostring()
        sequenza = sequenza.lower()
        sequenza = sequenza.replace("-","")
        foutput = open("./sequenze_txt/sequenza"+str(cont)+".txt", 'w')
        foutput.write(sequenza)
        foutput.close()
        cont +=1
    size = cont-1
    return size
```

Concatena.py

concatena.py - D:\slides\tutorial\script\script_tutorial\concatena.py

File Edit Format Run Options Windows Help

```
##concatena due file di testo in un unico file di destinazione
def concatena(size):
##size = 11
    for cont in range(1, size+1):
        finput = open("./sequenze_txt/sequenza"+str(cont)+".txt")
        sequenza = finput.readline().strip()
        for cont2 in range(cont+1, size+1):
            finput2 = open("./sequenze_txt/sequenza"+str(cont2)+".txt")
            sequenza2 = finput2.readline().strip()
            sequenzatot = sequenza+sequenza2
            foutput = open("./sequenzeconc_txt/sequenza"+str(cont)+"+"+str(cont2)+".txt", 'w')
            foutput.write(sequenzatot)
            foutput.close()
```

Gencompress.py

gencompress.py - D:\slides\tutorial\script\script_tutorial\gencompress.py

File Edit Format Run Options Windows Help

```
##script che lancia gencompress
def gencompress():
    import os
    dirinput = os.listdir("./")
    size = 0

    for elem in dirinput:
        if elem.endswith("txt"):
            os.system("GenCompress.exe "+elem+"")
            size+=1
##    os.system("rm *.LOG")
    os.system("copy *.GEN ..\\sequenze_compr")
    os.system("del *.log")
    os.system("del *.GEN")
    return size
```

NCD.py

NCD.py - D:\slides\tutorial\script\script_tutorial\NCD.py

File Edit Format Run Options Windows Help

```
from numpy import *

def ncd(dataset,size):
## compute NCD distance among comressed barcode sequences
    import os
    seq = "./sequenze_compr/" ##path of compressed sequences
    seq_conc = "./sequenzeconc_compr/" ##path of concatenated compressed sequences

    ncd_matrix = zeros((size,size), dtype=double)

    for elem in range(ncd_matrix.shape[0]):
        for elem2 in range(elem+1,ncd_matrix.shape[0]):
            concat = os.path.getsize(seq_conc+"sequenza"+str((elem+1))+"+"+str((elem2+1))+".GEN")
            seq1 = os.path.getsize(seq+"sequenza"+str((elem+1))+".GEN")
            seq2 = os.path.getsize(seq+"sequenza"+str((elem2+1))+".GEN")
            ncd = concat - min(seq1,seq2)

            ncd = float(ncd)/float(max(seq1,seq2))
            ncd_matrix[elem][elem2] = ncd

    a =matrix(ncd_matrix)
    savetxt("./"+dataset+"_ncd.csv", a, fmt='%f', delimiter=",")
```

Sections

- 1) Traditional and Compression-Based Approaches

- 2) **Alignment-free Classification techniques**

Outline

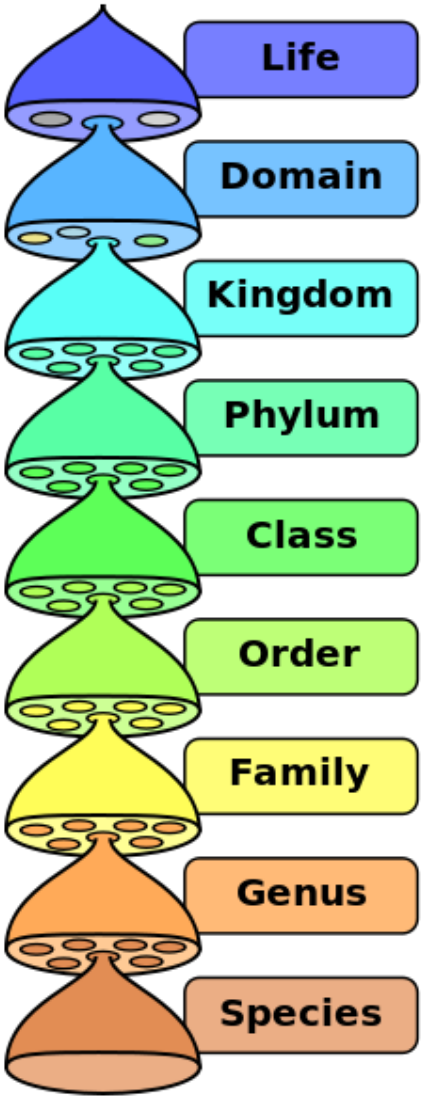
- DNA Sequence Approaches for Classification
  - Distance Based (phylogenetic tree)
  - Model Based (LDA)
  - Feature Based (Pattern/Vector)
    - Spectral Representation
- Feature Based Training Methods
  - Supervised (SVM)
  - Unsupervised (NG)
- A Spectral Representation + NG pipeline
  - Implementation in R environment

Biological Classification

Scientific classification in biology, is a method of scientific taxonomy used to group and categorize organisms into groups such as genus or species.

Taxonomic category:

Taxa classification is hierarchical. In a biological classification, **rank** is the level in a hierarchy.



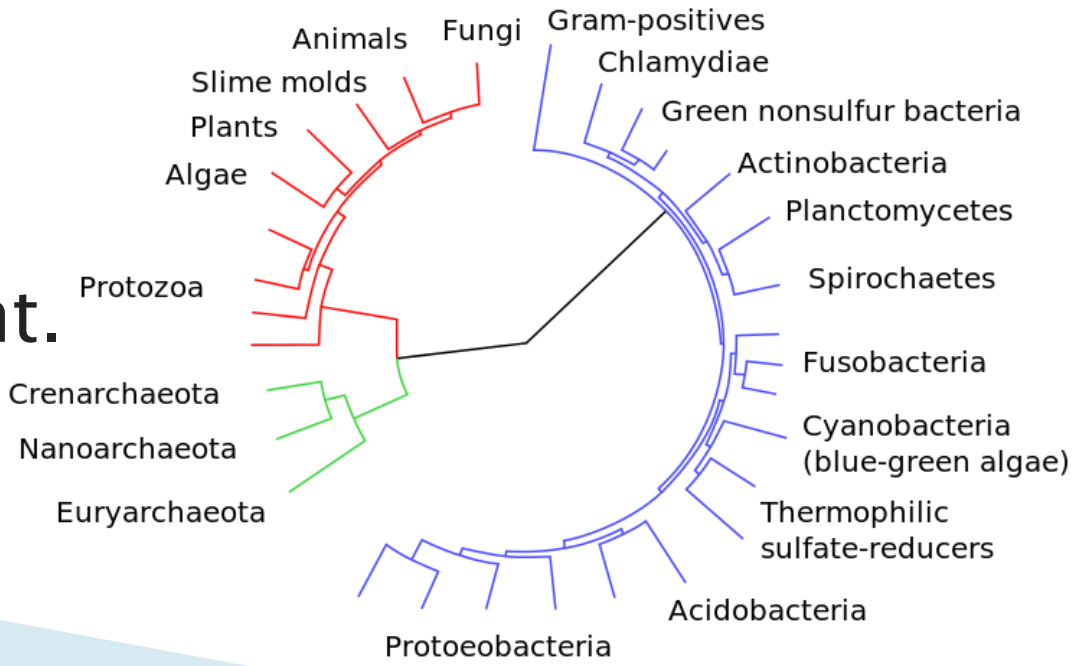
Sequence Classification Methods

- *Distance based:*
  - Define the distance function which measures the similarity between sequences; determines the quality of the classification significantly.
- *Model based:*
  - Use statistical and probabilistic methods to classify sequences, exploiting generative models.
- *Feature based:*
  - Find representative patterns or feature vector and then apply conventional classification methods. Feature selection plays an important role in this kind of methods.

Sequence distance based

Cladistics is a to biological classification approach based on unique characteristics of common ancestry.

Molecular systematics with evolutionary tree assumes that classification must correspond to phylogenetic descent.



Weakness of Sequence Distance based

- Errors in DNA sequence alignment to establish homology by nucleotide position (GAP, SNP).
- Lost of information when approximation is used for reconstruction of phylogenetic tree.
- Evolutionary distances are stochastic models that do not define a distance metric.

Model based

It assumes sequences in a class are generated by an underlying **model *M*.**

Given a class of sequences,
M models the probability distribution of the sequences in the class.

Training step => the parameters of *M are learned.*

Classification step => *a* new sequence is assigned to the class with the highest likelihood.

Model based: Naive Bayes Classifier

- The simplest generative model is the Naive Bayes sequence classifier.

Assumption:

"Given a class, the features in the sequences are independent of each other."

- The conditional probabilities of the features in a class are learned in the training step.

Model based: Probabilistic Topic Model

Generative model that allows sets of observations to be explained by unobserved groups that explain why some parts of the data are similar.

Latent Dirichlet Allocation

In Text Analysis:
each **document** is a mixture of a small number of **topics** and that each **word**'s creation is attributable to one of the document's topics.

Probabilistic Topic Models-Theory

Hypothesis :

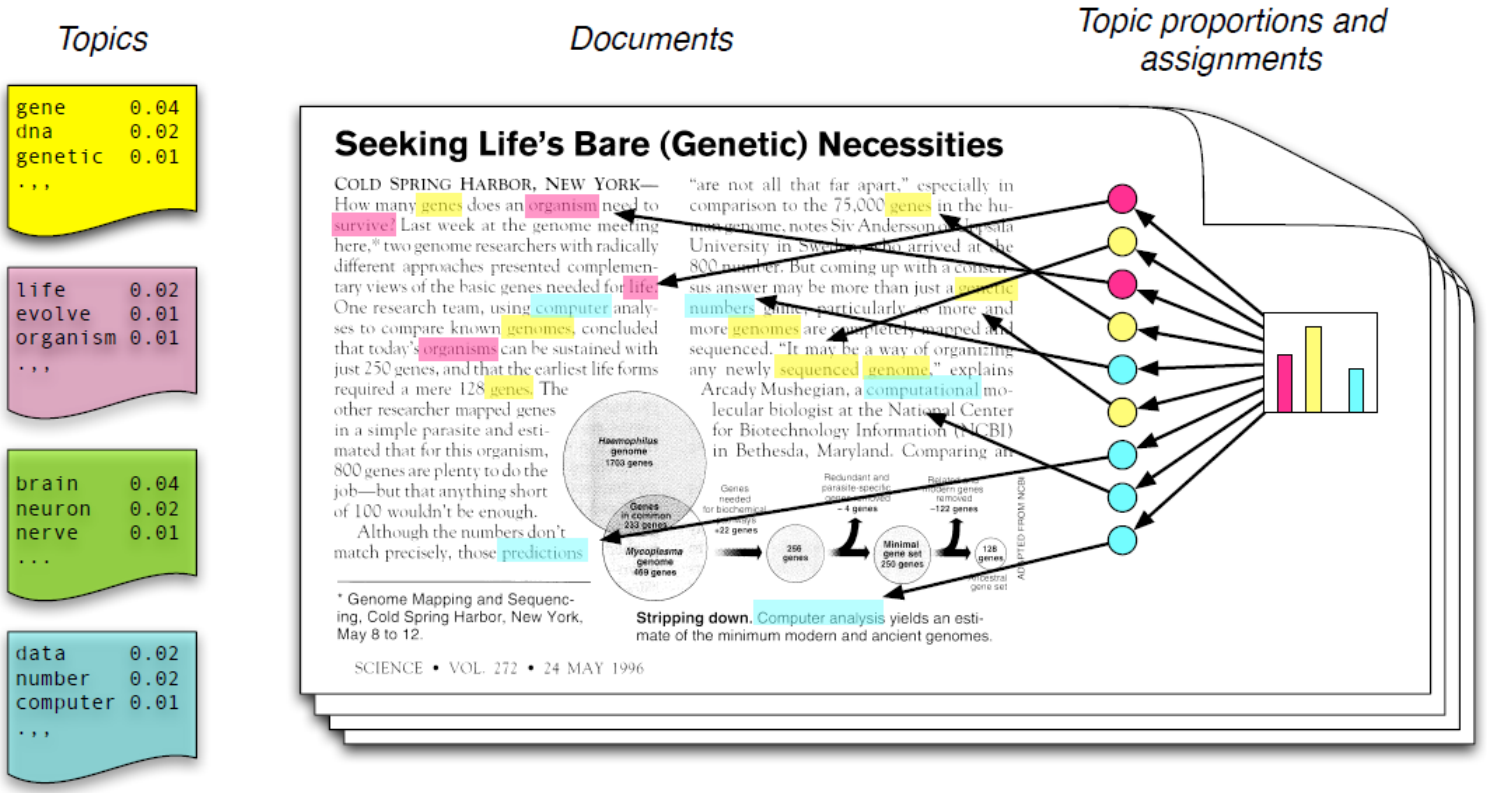
- Topics are probability distributions over a fixed dictionary (set of words)
- Topics represent recurring themes over documents
- Documents can be labeled according to their most recurring (probable) topics

Aim:

Given a number of fixed *a priori* topics

Extract topics from a corpus of documents

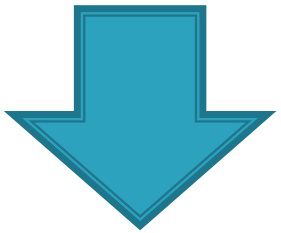
Probabilistic Topic Models-Theory



From "*Blei, D.M.: Probabilistic Topic Models. Communication of the ACM 55(4), 2012*"

Probabilistic Topic Models-Use

- Each document has a probability of showing a specific topic
- Given a fitted model, trained on a corpus of tagged documents, we can obtain the topics of an unknown document



Document Classification

Topic Models and DNA sequence

Premises:

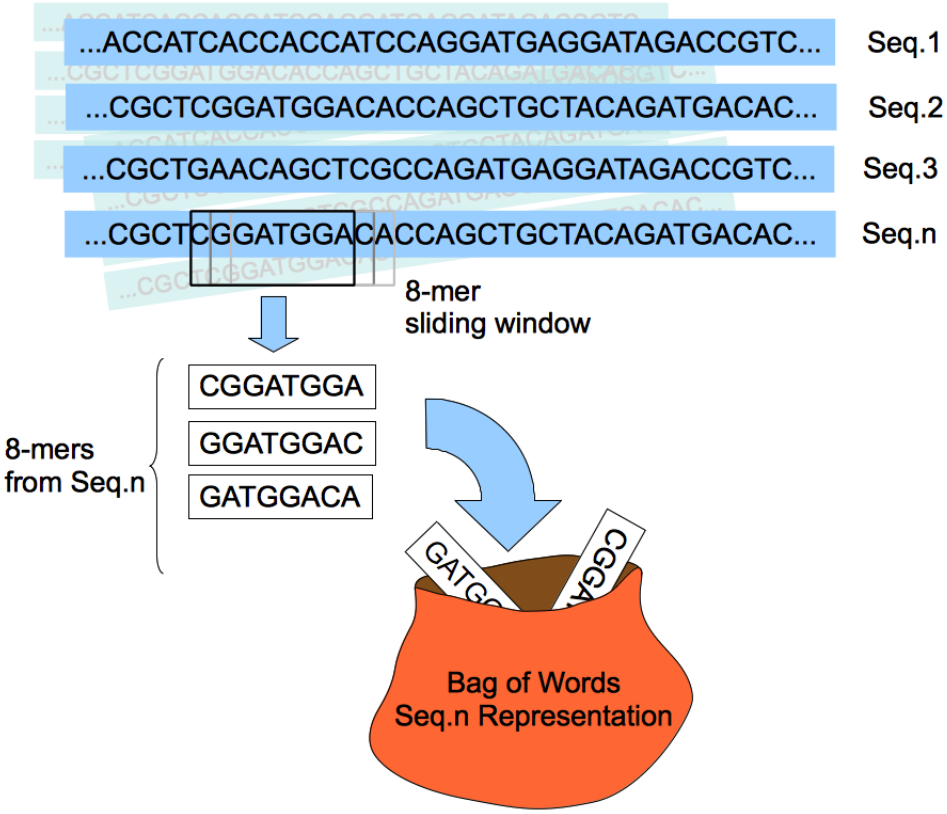
- A DNA sequence as a document
- Words as DNA fragments (e.g., k-mers)

K-mer: a k-base long sequence (k-tuple) of DNA

Topic Models and DNA sequence

Premises:

- A DNA sec
- Words as l



Topic Models and DNA sequence

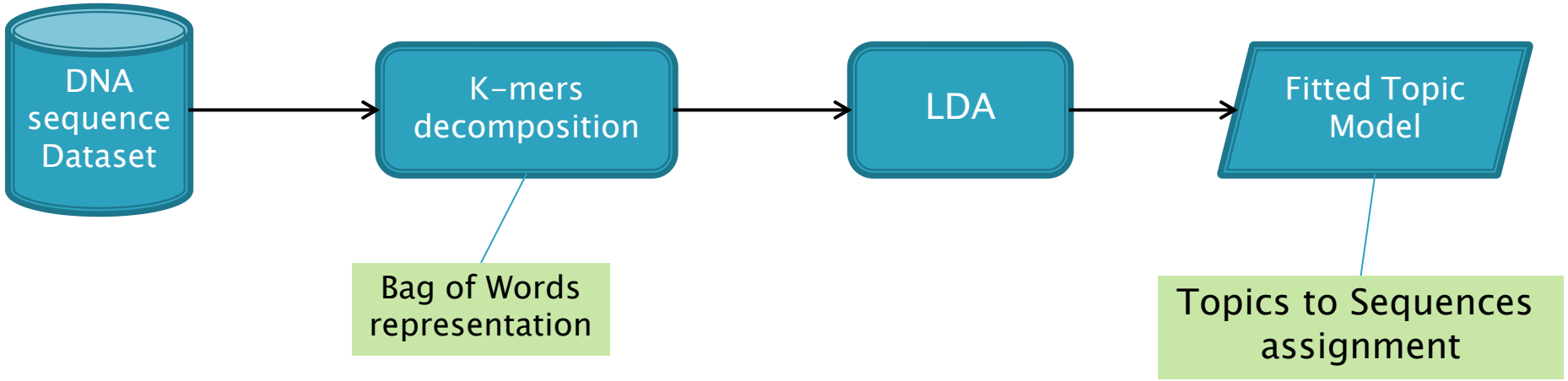
Premises:

- A DNA sequence as a document
- Words as DNA fragments (e.g., k-mers)
- Extract the set of topics from a dataset of sequences

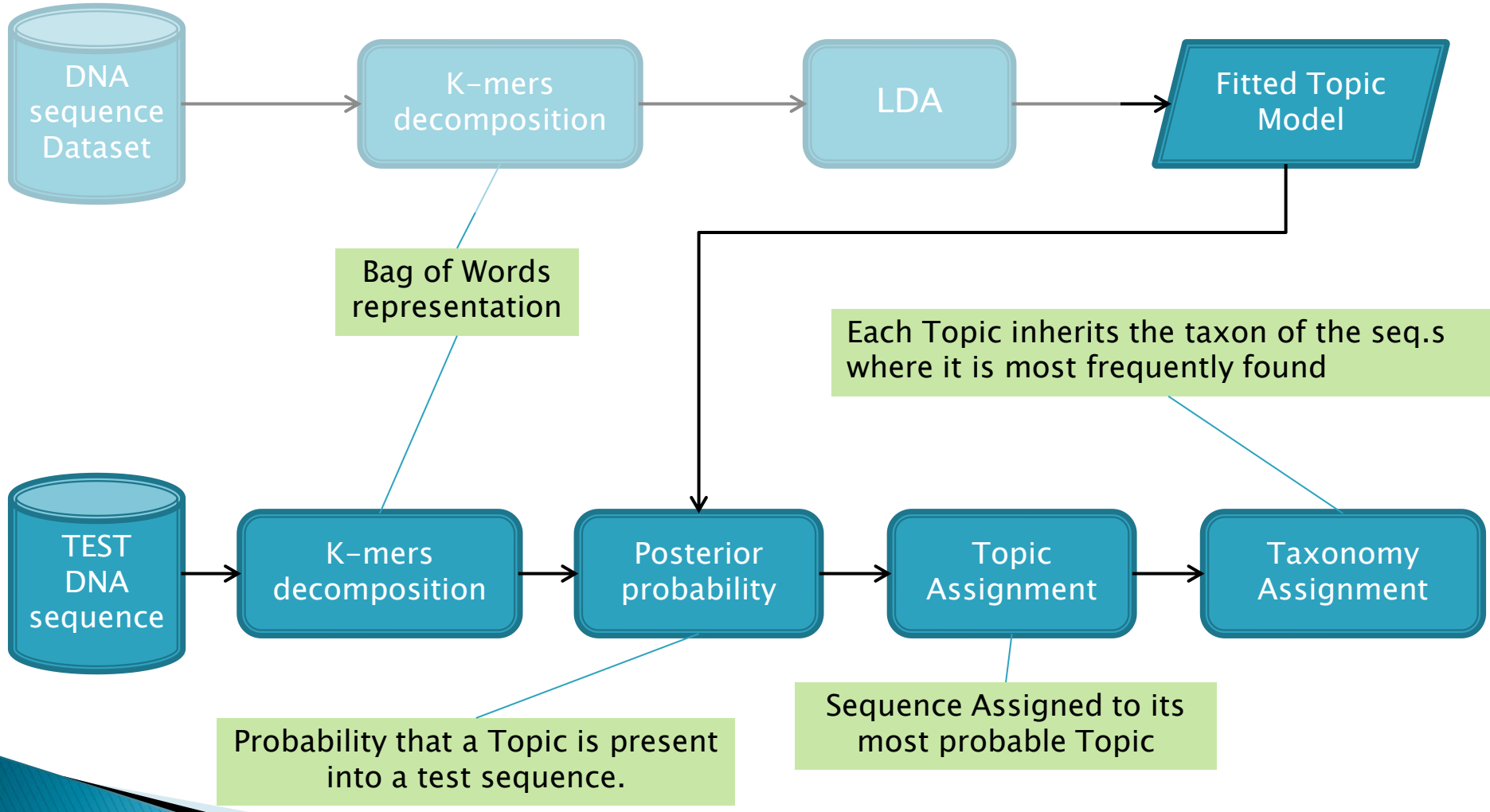
Thesis:

- **Sequences sharing the same topics belongs to the same taxa**

Training Phase



Testing Phase



Feature based techniques

Existing methods differ from each other on the following aspects:

- Which *criteria* should be used for selecting features, such are distinctiveness, frequency and length?
- In which *scope* does feature selection reflect the sequential nature of a sequence, local or global?
- Should feature selection be *integrated within the process* of constructing the classifier or a separate pre-processing step?

Pattern-based feature selection method

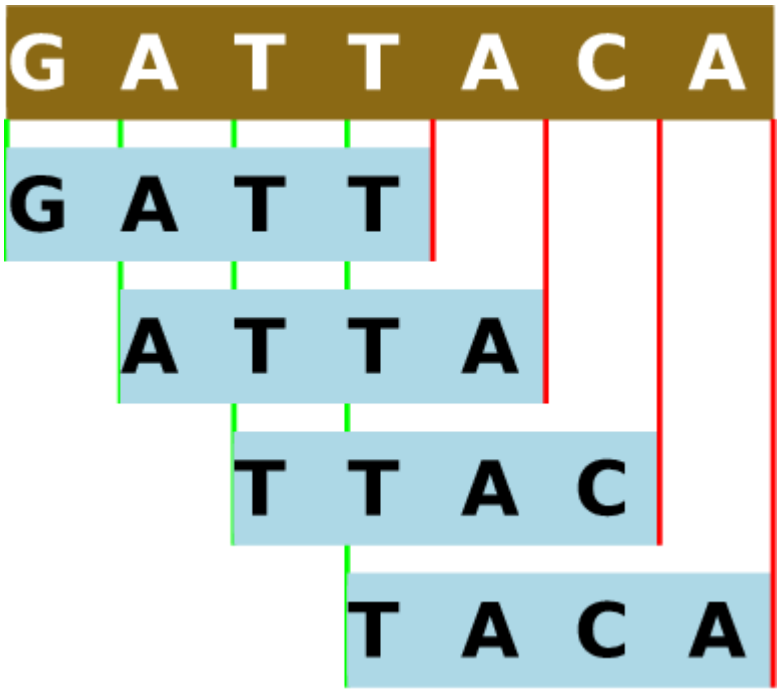
- The features are short DNA sequence segments which satisfy the following criteria:
  (1) frequent in at least one class;
  (2) distinctive in at least one class;
  (3) not redundant.

- **Cons**: they describe the local properties of a long DNA sequence.

Vector-based feature selection method

- Given a set of *k-mers, a sequence can be represented as a vector of the* presence and the absence of the *k-mers or as a vector of* the frequencies of the *k-mers.*

Counting *k*-mers with Sliding Window, step=1



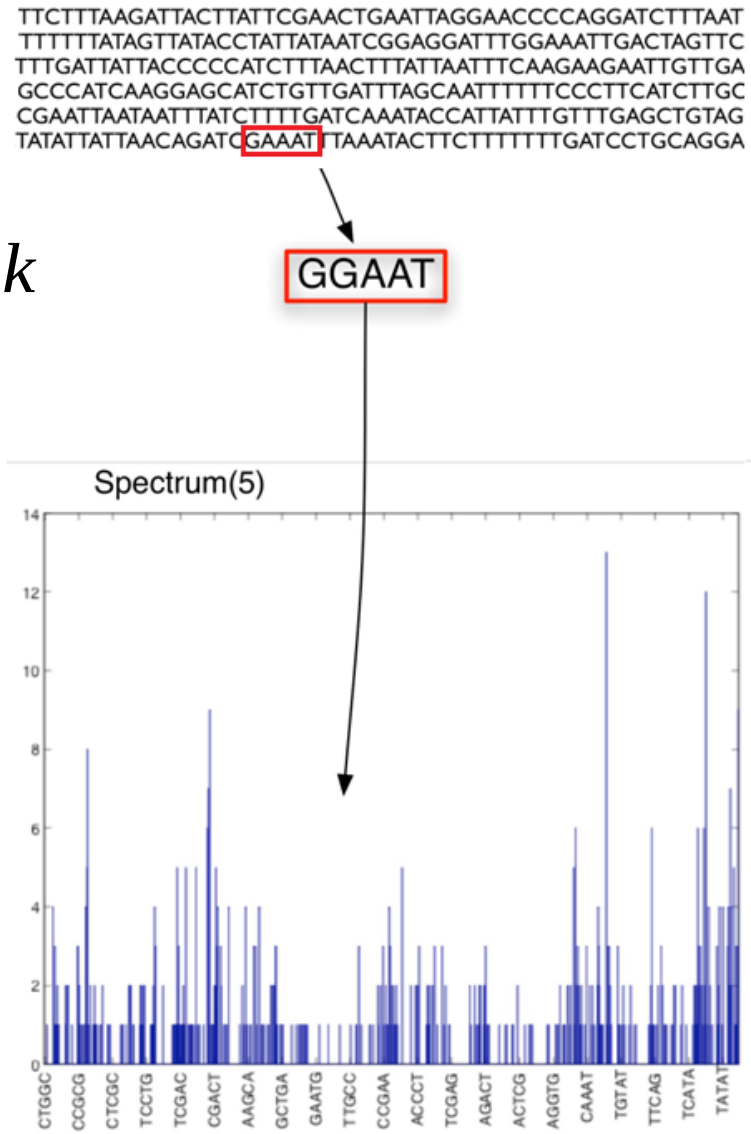
***Cons**: it does not save information about sequence of k-mers.*

K-mers of a DNA sequence S

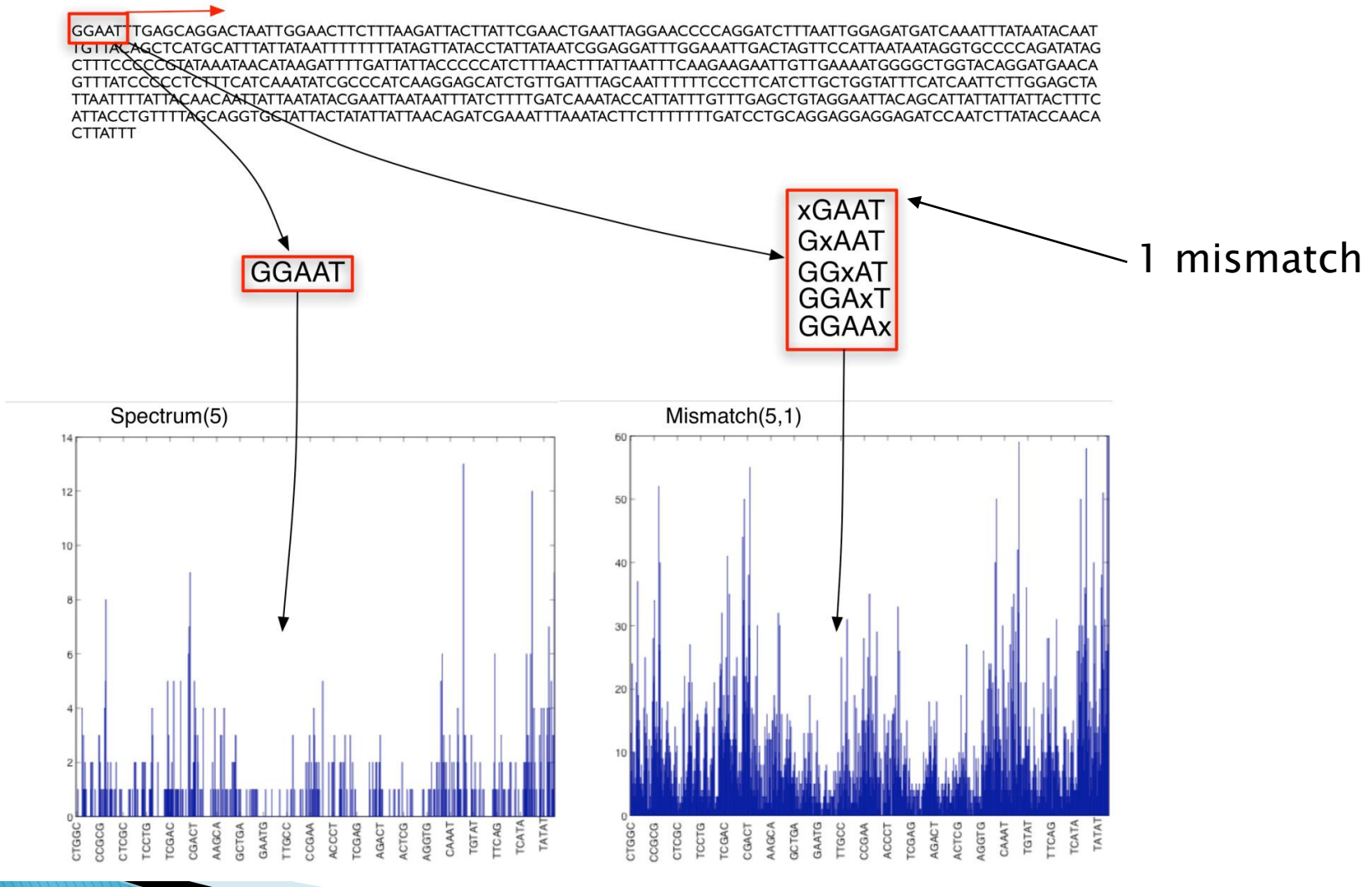
- **K-mer:** a k-base long sequence (k-tuple) of DNA $(a_1, a_2, \ldots, a_k), a_i \in S, i = 1, 2, \ldots, k$

K = 5 => 4^5 words= 1024 words

- **Spectrum (K-mer feature vector):** constructed using a frequency *f* of each k-mer in a DNA sequence

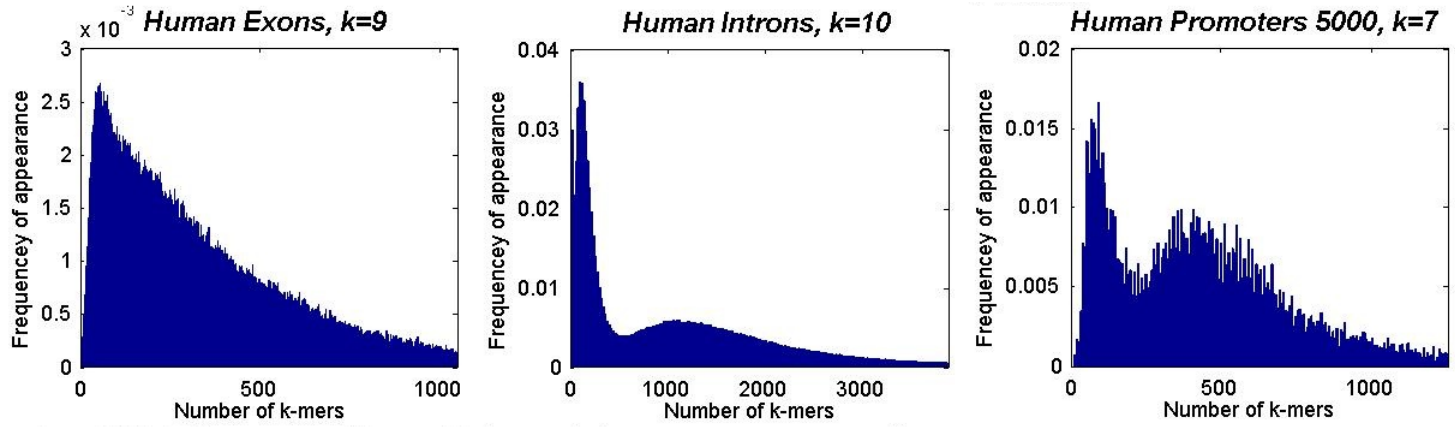


DNA Spectrum with 1-mismatch



Research on DNA k-mers

- Examine the properties of these DNA words and how their distributions vary between different species or genome elements.
- Study on the whole genome for examine models and modalities for different species.
- Words with extreme frequencies, namely, either missing or rare *k*-mers, or those with very high frequencies.



DNA fragment Nucloetide Frequencies

Test set:
From whole genome sequencing, test 10kb fragment of 65 bacteria and 6 eukaryotes from NCBI dataset.

Conclusion:
-dinucleotide frequency is unable to bin sequence fragments into well-clustered species groups;
-increasing order of oligonucleotide frequency may deteriorate the assignment of DNA sequences to classes.

Clustering Tecnique:
SOM + PCA preprocessing

Clustering evaluation:
F-measure

Results

Dinucleotide	0.94
Trinucleotide	0.98
Tetranucleotide	0.98
Pentanucleotide	0.99

Training Methods

- Supervised
- Unsupervised
- (Semi-supervised)

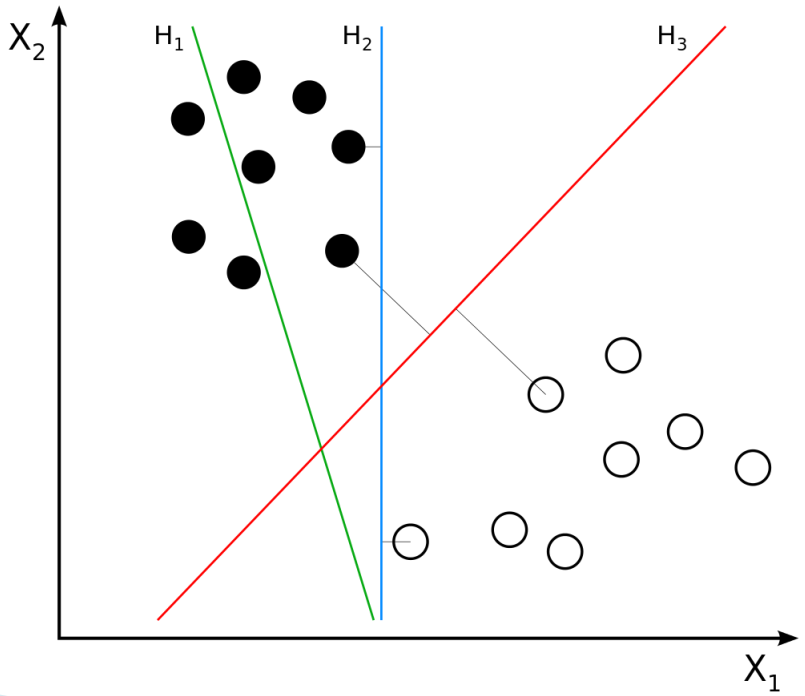
Supervised classification methods

- Desired output must be provided for each input used in the training.
  - Inputs are processed and compared with its actual outputs against the expected outputs.
  - The process is repeated until the errors are minimized.
- Objective: error minimization of the number of misclassifications.

Support Vector Machine

The SVM procedure reconstructs separating hyperplanes in Euclidean space to classify real-valued vectors.

There are many hyperplanes that might classify the data:
it finds the **maximum-margin** hyperplane:
where distance from it to the nearest
data point on each side is maximized.



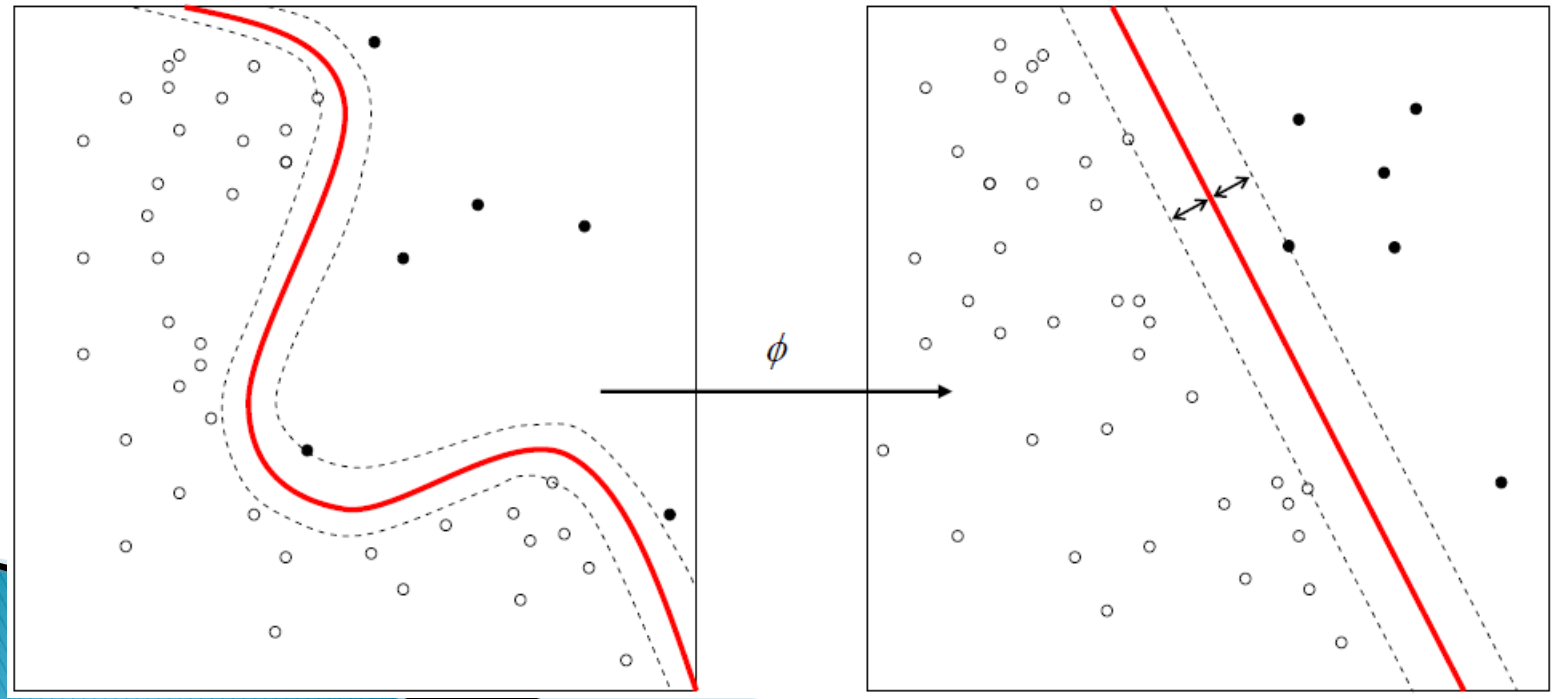
Double Objective at the time:

- Maximize hyperplanes margin
- Minimize training set errors

Support Vector Machine

If non-linear separation ⇒ Feature space transformation

General idea: the original input space can always be mapped to some higher-dimensional feature space where the training set is separable.



Weakness of SVM

- **Sensitive to noise**
  - A relatively small number of mislabeled examples can dramatically decrease the performance.
- **It only considers two classes at a time**
  - to do multi-class classification, it learns *n* SVM's.
- **Choice of kernel**
  - Gaussian or polynomial kernel is default;
  - if ineffective, more elaborate kernels are needed.

Unsupervised classification methods

- Training samples contain only input patterns
  - No desired output is given (teacher-less)

- Learn form classes/clusters of sample patterns according to similarities among them
  - Sequences in a cluster would have similar features;
  - No prior knowledge as what features are important for classification, and how many classes are there.

Prototype-Based Methods

- The prototypes represent the usually **a-priori** fixed number of clusters by representatives, and the cluster assignment takes place based on the similarity to the cluster prototype.

- Decomposition of the given data set into clusters can be **fuzzy** or **crisp** :
  - k-means,
  - fuzzy-clustering,
  - neural gas,
  - self-organizing map

K-Centroids Cluster Analysis

- ***Cluster problem***
  - estimate a "good" number of *C* clusters for a data set

$$\mathrm{X_N} = \{x_1, \ldots, x_N\},\ x \in \text{variable space}$$

- ***K-centroids cluster problem***
  - *find a set of centroids C_k for fixed K such that the average distance of each point to the* closest centroid is minimal.
- **M*aximum radius cluster problem***
  - *find the minimal K such that a set of centroids C_k exists,* where

$$\max_{n=1,..,N} D\left(x_n, C(x_n)\right) \le r$$

for a given radius

$$r = \sqrt{\frac{\sum_{i=1}^{NK} (x_{ik} - C_k)^2}{NK}}$$

K-Centroids Cluster Analysis

- Representing clusters by centroids has computational advantages when predicting cluster membership for new data.
- For radius calculation one needs only comparison with the *K centroids*, *whereas for diameter calculation* one needs pairwise comparison with all *N data points.*

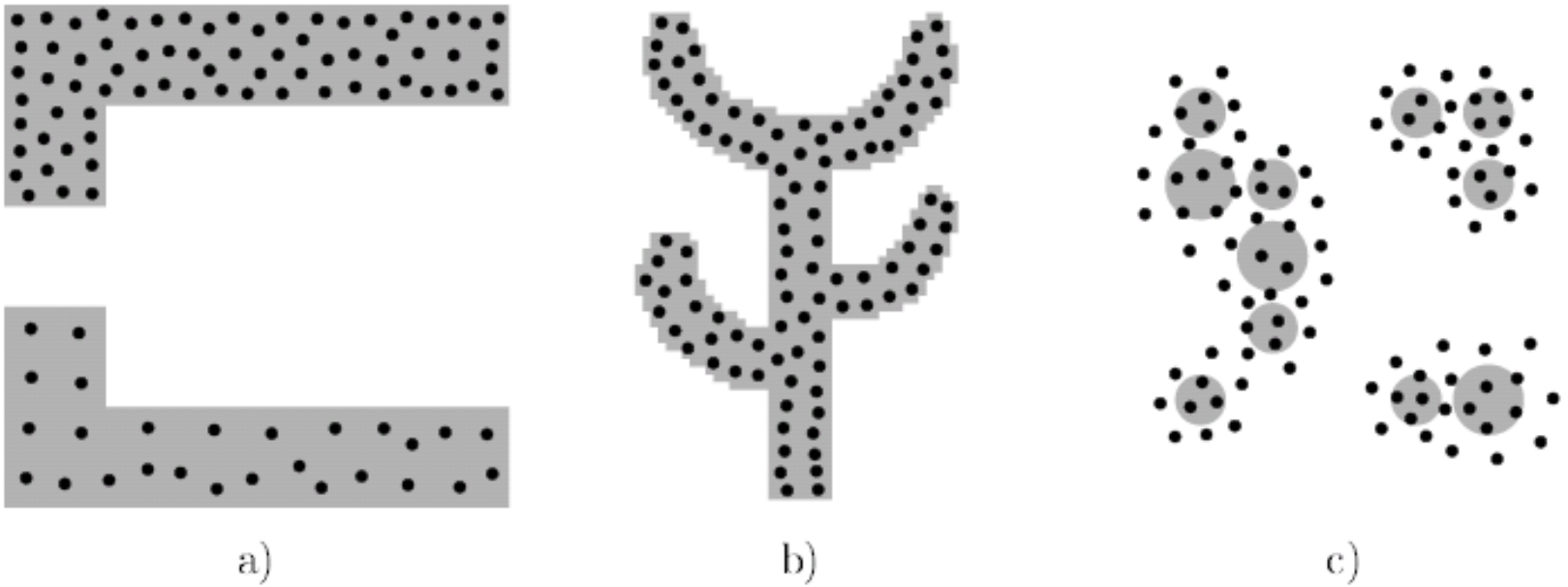
Neural Gas

- NG provides input space representations by constructing data summaries (via prototypical vectors).
- It's a gradient descent procedure imitating gas dynamics within data space to calculate the prototypes.
- Soft Competitive Learning (WTM):
  - not just the winning neuron adjusts its prototype, but all other cluster prototypes have the opportunity to be adapted based on how proximate they are to the input pattern.

Neural Gas Algorithm

1. Initialize a set of prototype vectors $\mathbf{W} = \{ \mathbf{w}_1 , \mathbf{w}_2 , \ldots , \mathbf{w}_k \}$ (*randomly*);
2. Present an input pattern **x** to the network. Sort the index list in order, from the prototype vector with the smallest Euclidean distance from **x** to the one with the greatest distance from **x** ;
3. Adjust the prototype vectors using the learning rule: $w_k^{t+1} = w_k^t + \eta \cdot e^{-k/\lambda} \cdot (x - w_k^t)$
   - learning rate $\eta = \eta_i \bullet (\eta_f / \eta_i)^{(iter/iter\_max)}$
   - decay constant $\lambda = \lambda_i \bullet (\lambda_f / \lambda_i)^{(iter/iter\_max)}$
4. Repeat steps 2 and 3 until the maximum number of iterations is reached.

NG Application



NG Examples from Bernd Fritzke, Ruhr Univercity

Draft 5 April 1997, p22

NG Comments

1. Ideally, when learning stops, each w_j is close to the centroid of a group/cluster of sample input vectors.
2. To stabilize w_j, the learning rate η may be reduced slowly toward zero during learning, e.g., $\eta(t+1) \leq \eta(t)$
3. # of output nodes:
   - too few: several clusters may be combined into one class
   - too many: over classification
4. Initial:
   - learning results depend on initial weights (node positions)
   - training samples known to be in distinct classes, provided such info is available
   - random (bad choices may cause anomaly)
5. Results also depend on sequence of sample presentation

A pipeline for

Analysis of DNA Barcode Sequences

using

Spectral Representation

(feature based representation)

and

Neural Gas

(unsupervised learning)

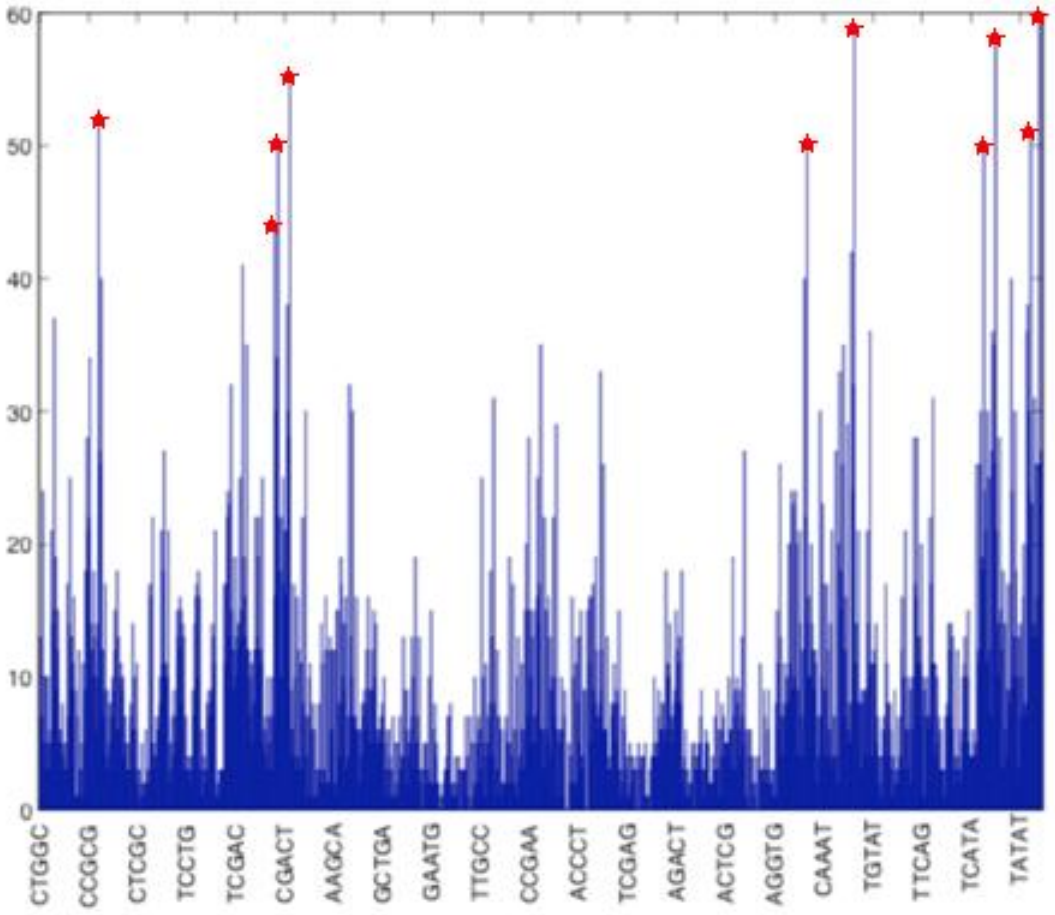
Selection of High Frequency Words from DNA Spectrum

Basic Assumption: Similar spectra ~ high similarity score

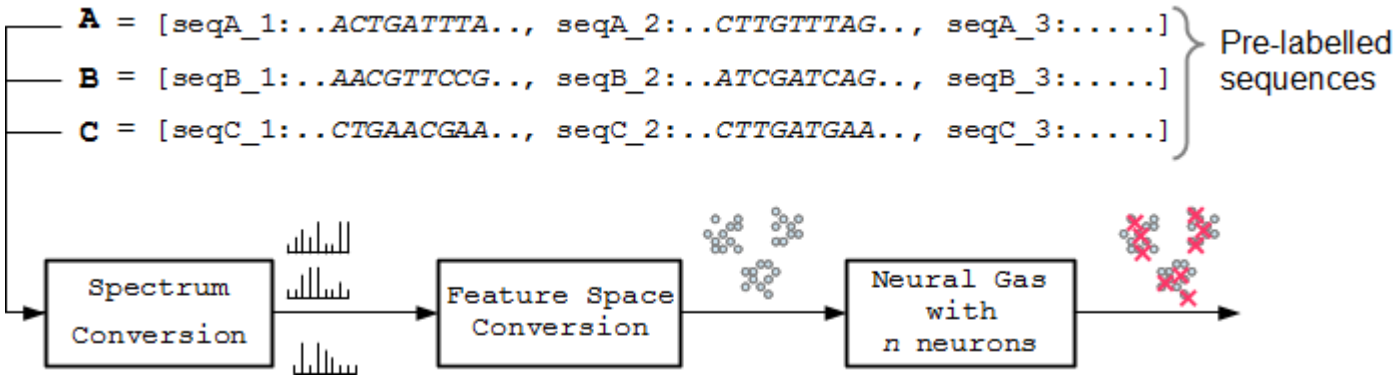
K = 5 => 1024 words

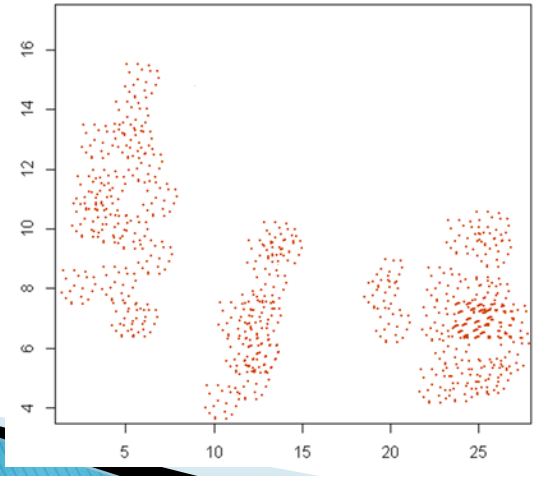
For each reference spectra, we select *n* HFW.

These *n* words represents a fingerprint for all DNA sequences, whose spectra are clustered with this reference spectrum.

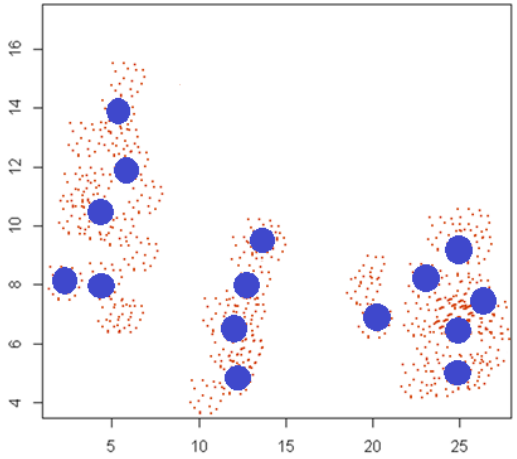


Training Phase



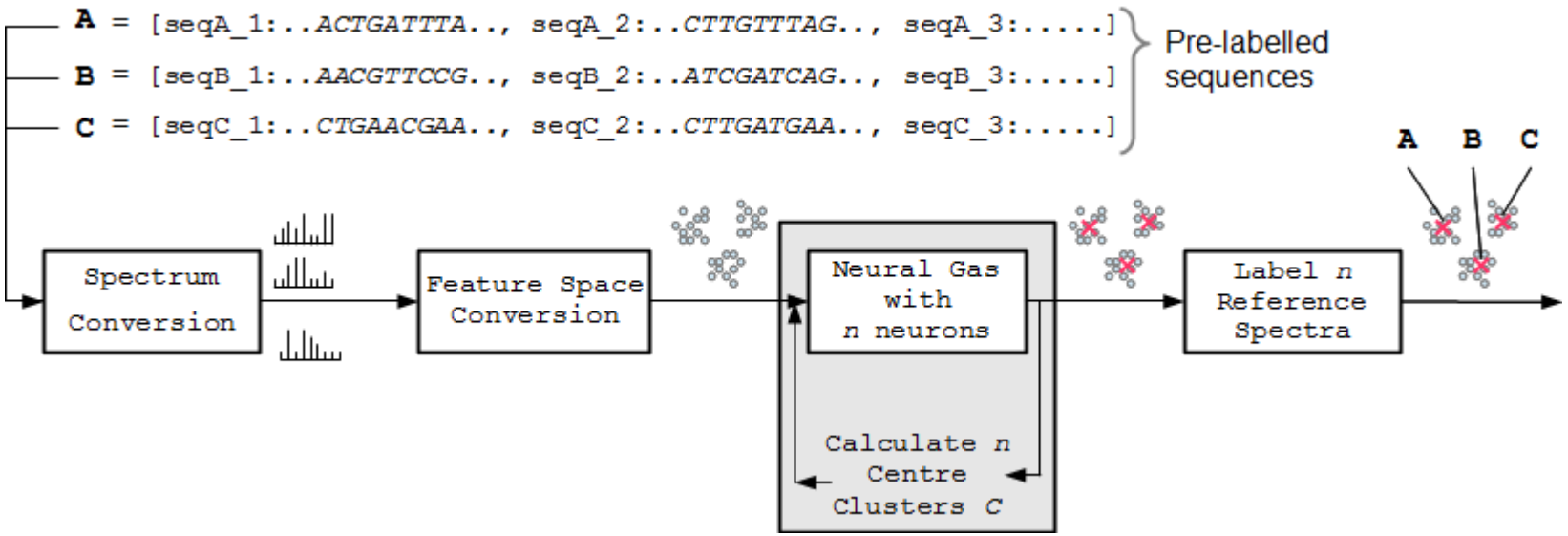


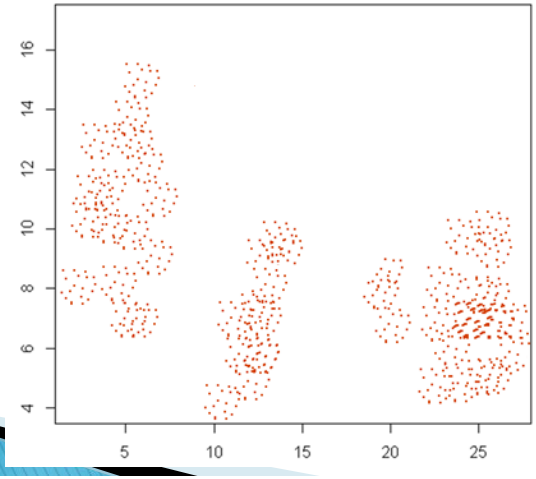
Feature Space @1024 words



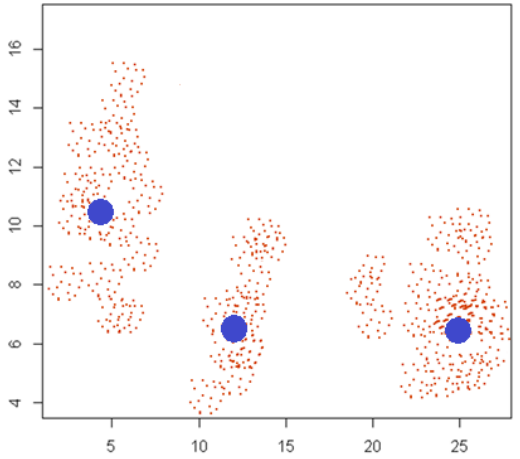
Neural Gas with 15 neurons (15 centroids)

Training Phase



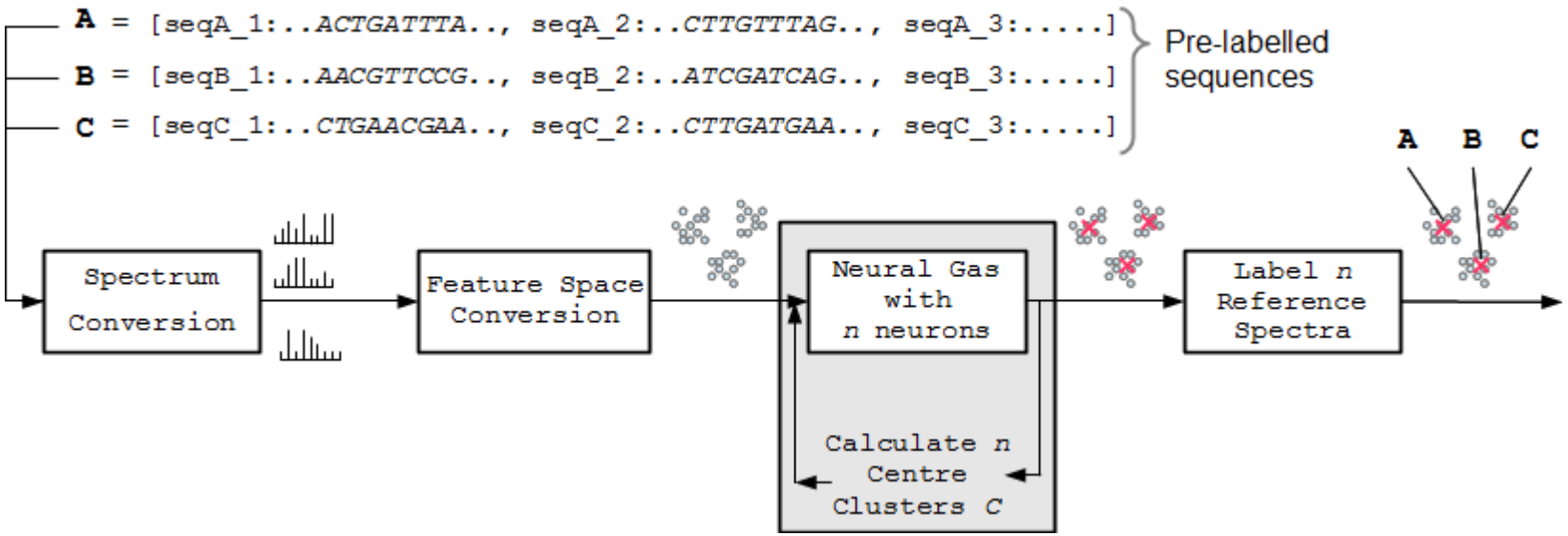


Feature Space @1024 words



Neural Gas with 3 neurons (3 centroids)

Training Phase

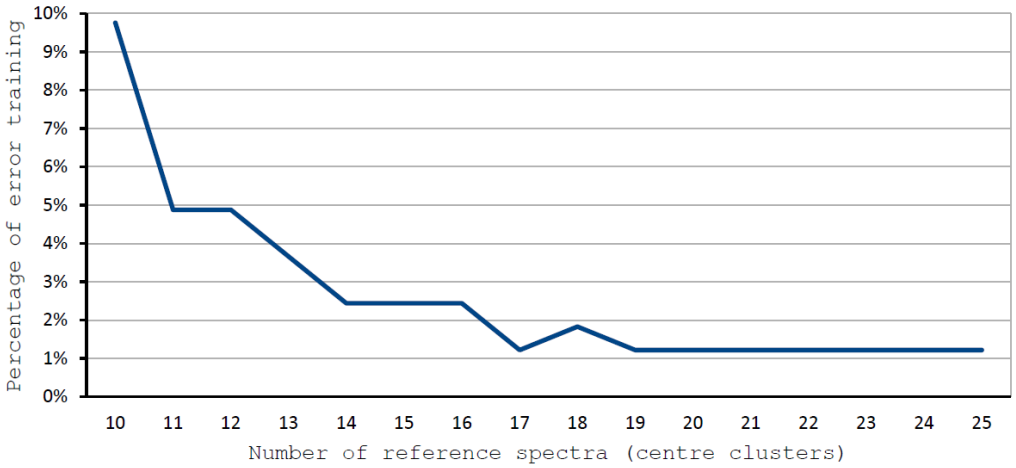


Minimum Number of Center, maximizing Overall Accuracy

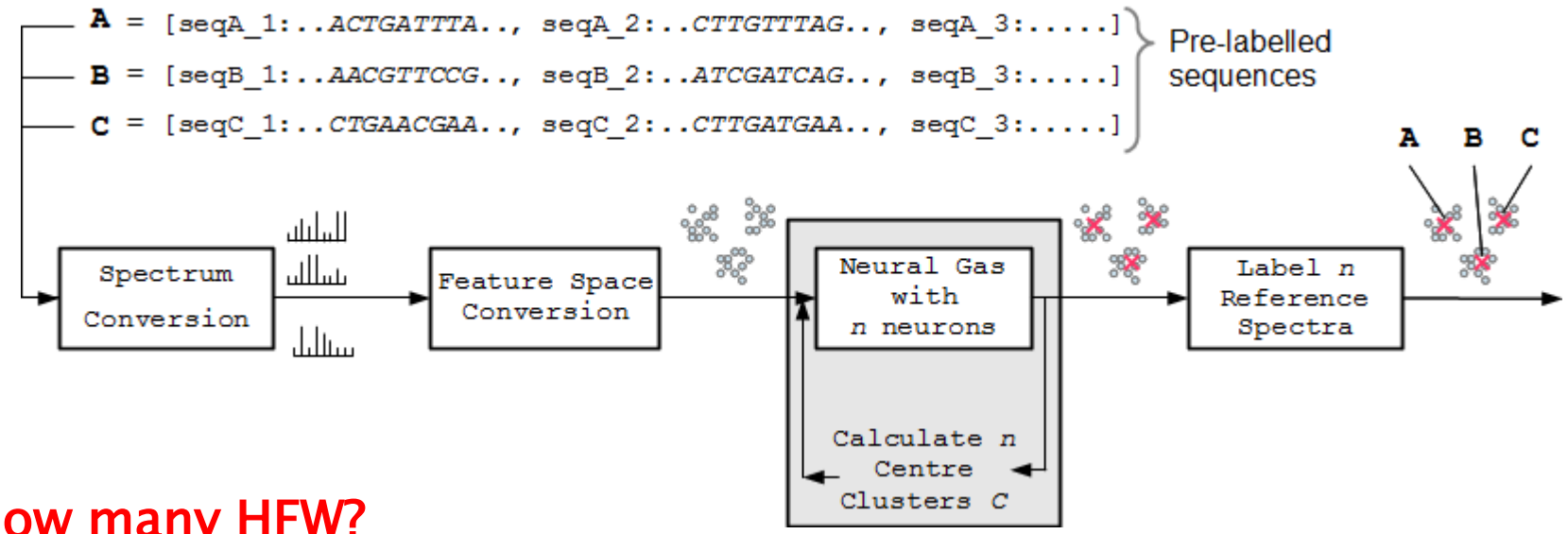
$$n_s = \underset{n}{\operatorname{argmin}}\left\{ \max\{OA(n)\} \right\}$$

Error Training

$$e_{t_{min}} = e_t(n_s) = 1 - \max\left\{OA(n_s)\right\}$$



Testing Phase

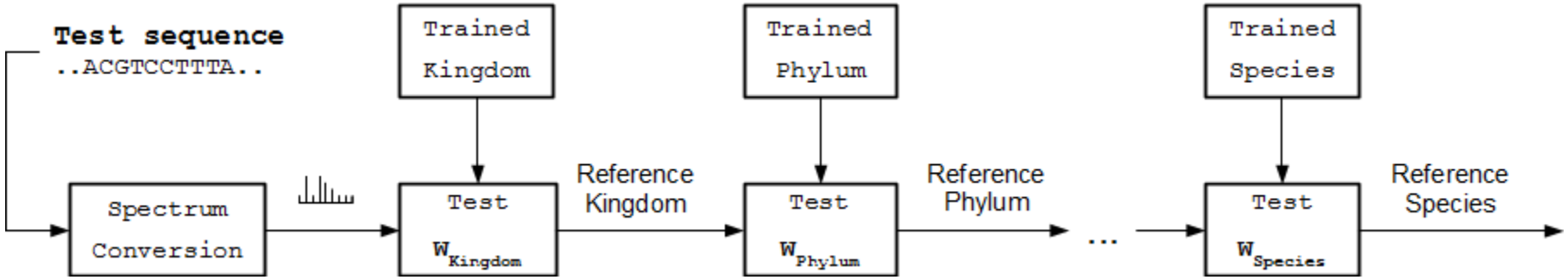


How many HFW?

Jaccard distance between HFWs in test sequence and centroids set.

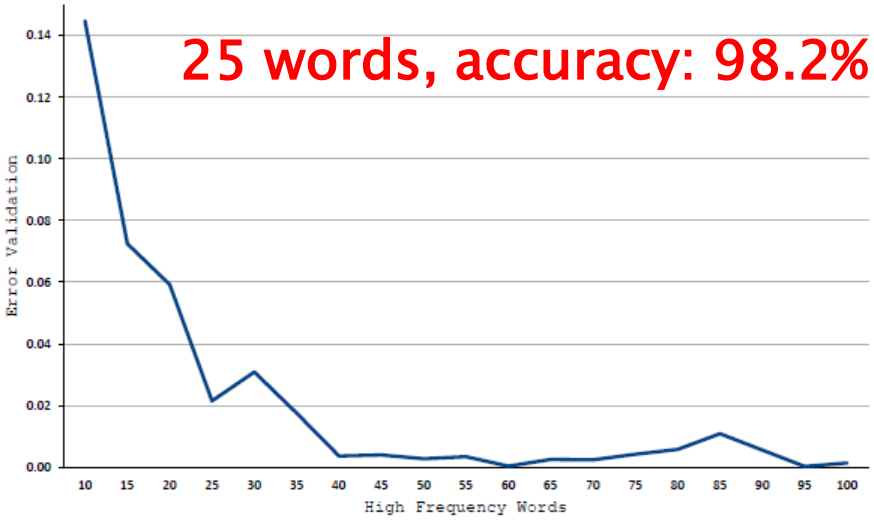
Validation Error

$$e_v(w) = \min\left\{ J_\delta \Big|_{HFW=w} \left(\overline{s^*}_{seq}, \overline{c^*}_i \right) \right\}$$

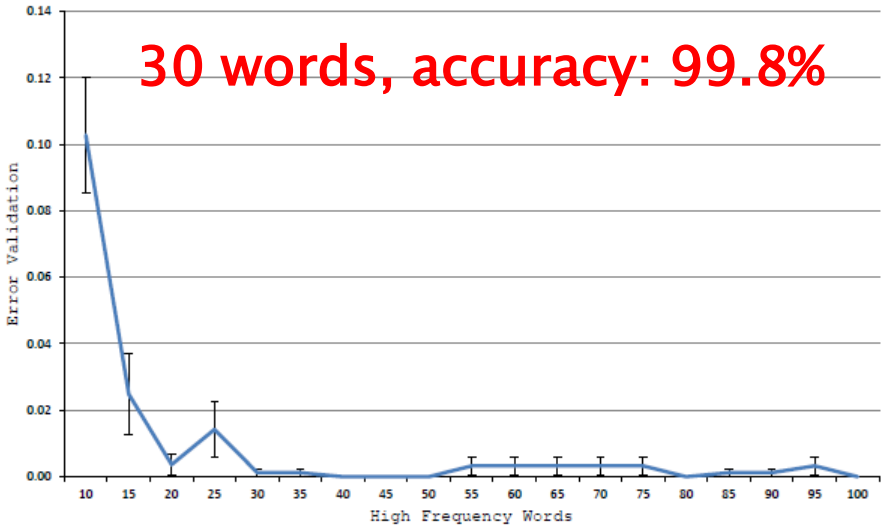


2212 Barcode Sequences (with cross-validation techniques)

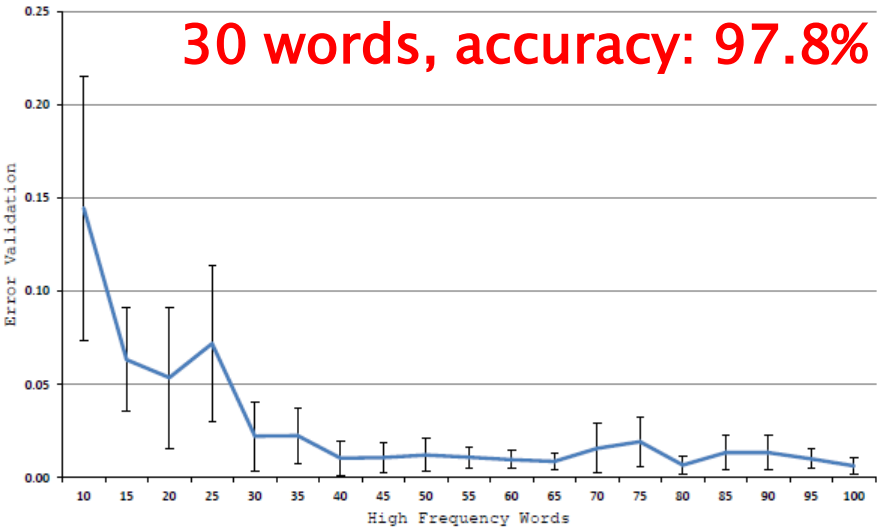
(a) Phylum Classification: 10-fold cross-validation



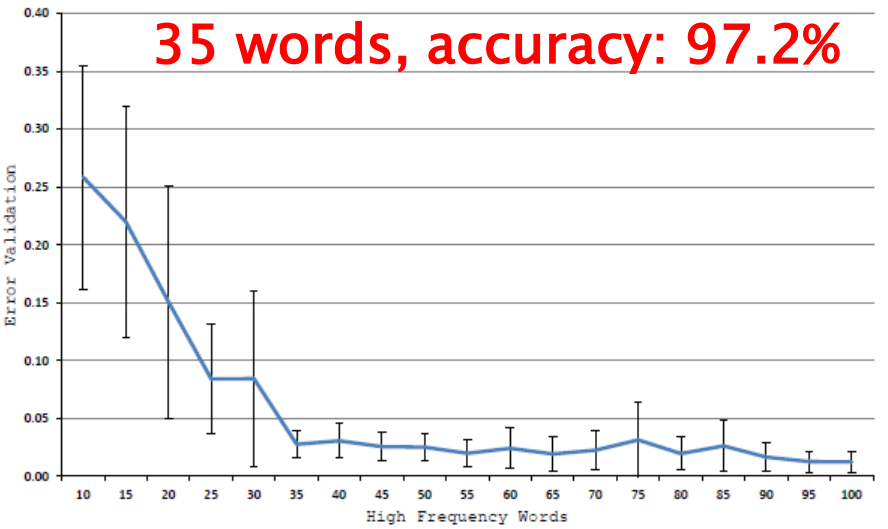
(b) Familia Classification: leave-1-out cross-validation



(c) Genus Classification: leave-1-out cross-validation



(d) Species Classification: leave-1-out cross-validation



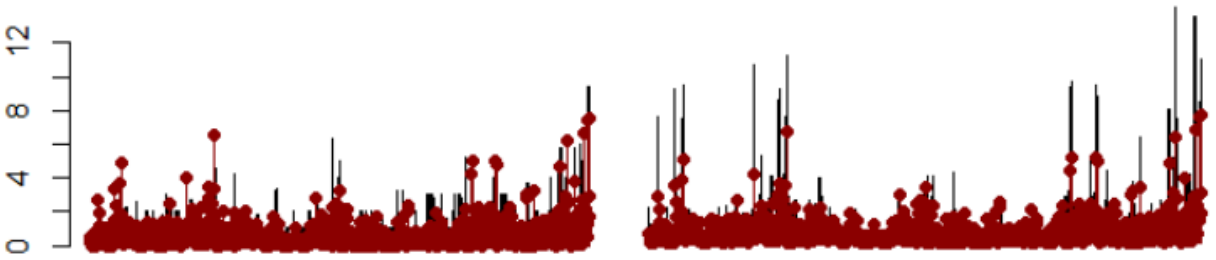
Visualization: Barplot and Neighbourhood Graph

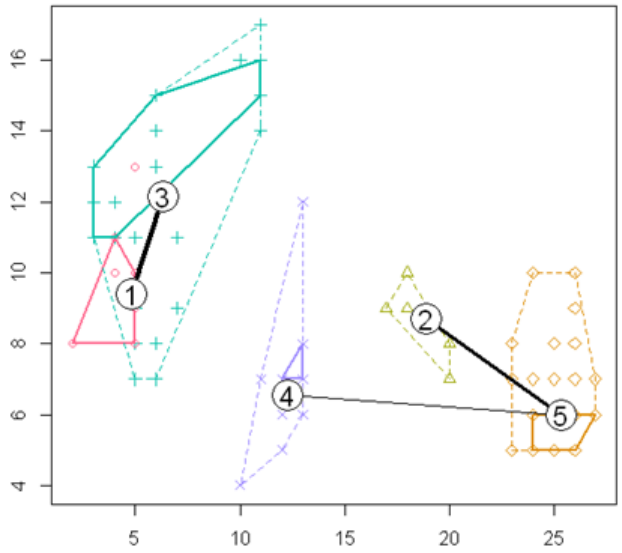
The visualization of the cluster structure is important in order to investigate the relationships between clusters.

The data is divided into artificial subsets where the relationship between clusters plays an important role.

The ***Neighborhood Graph*** can be used to display distances between clusters for centroid-based cluster solutions.

Barplots (max and means) of 2 centroids





An implementation of the previous pipeline for the Classification of DNA Barcode Sequences in R environment

Introduction to R environment

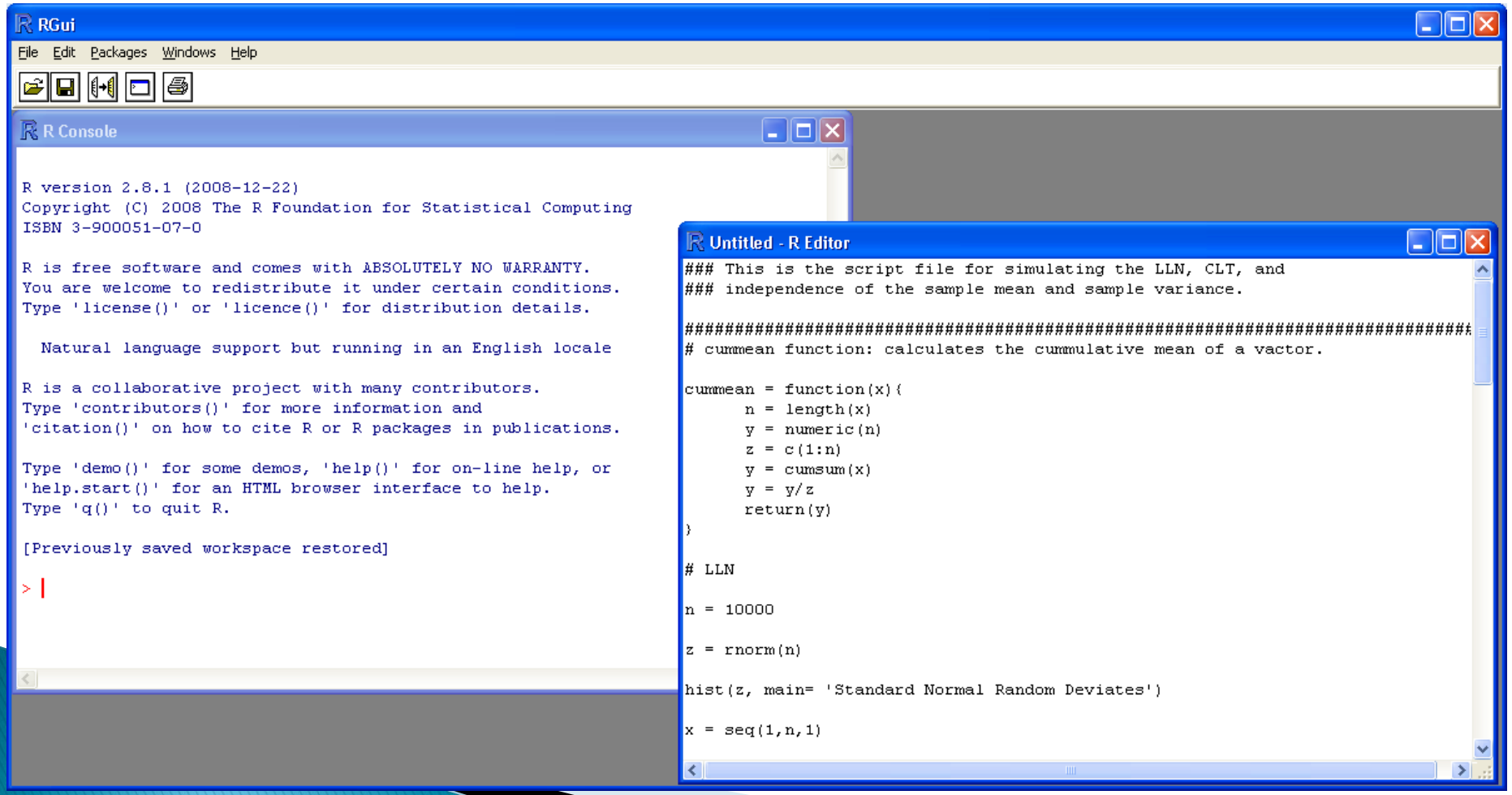
R is a free software programming language and software environment for statistical computing and data analysis.

R is an interpreted language; users typically access it through a command-line interpreter.

R supports procedural programming with functions and, for some functions, object-oriented programming with generic functions.

R's data structures include scalars, vectors, matrices, data frames and lists.

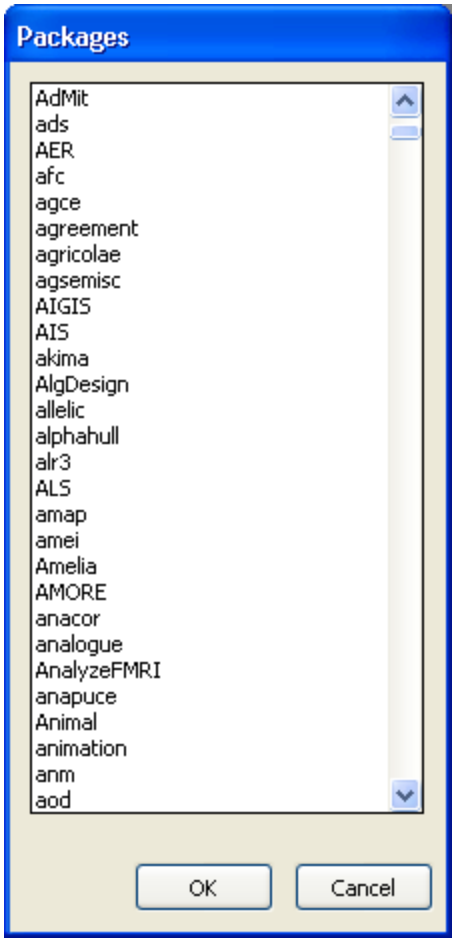
R Software


RGui
File Edit Packages Windows Help
R Console
R version 2.8.1 (2008-12-22)
Copyright (C) 2008 The R Foundation for Statistical Computing
ISBN 3-900051-07-0
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.
Natural language support but running in an English locale
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.
[Previously saved workspace restored]
>
Untitled - R Editor
This is the script file for simulating the LLN, CLT, and
independence of the sample mean and sample variance.
# cummean function: calculates the cummulative mean of a vactor.
cummean = function(x){
n = length(x)
y = numeric(n)
z = c(1:n)
y = cumsum(x)
y = y/z
return(y)
}
# LLN
n = 10000
z = rnorm(n)
hist(z, main= 'Standard Normal Random Deviates')
x = seq(1,n,1)

R Packages

- There are many contributed packages that can be used to extend R.

- These libraries are created and maintained by the authors.



R Packages for Bioinformatics

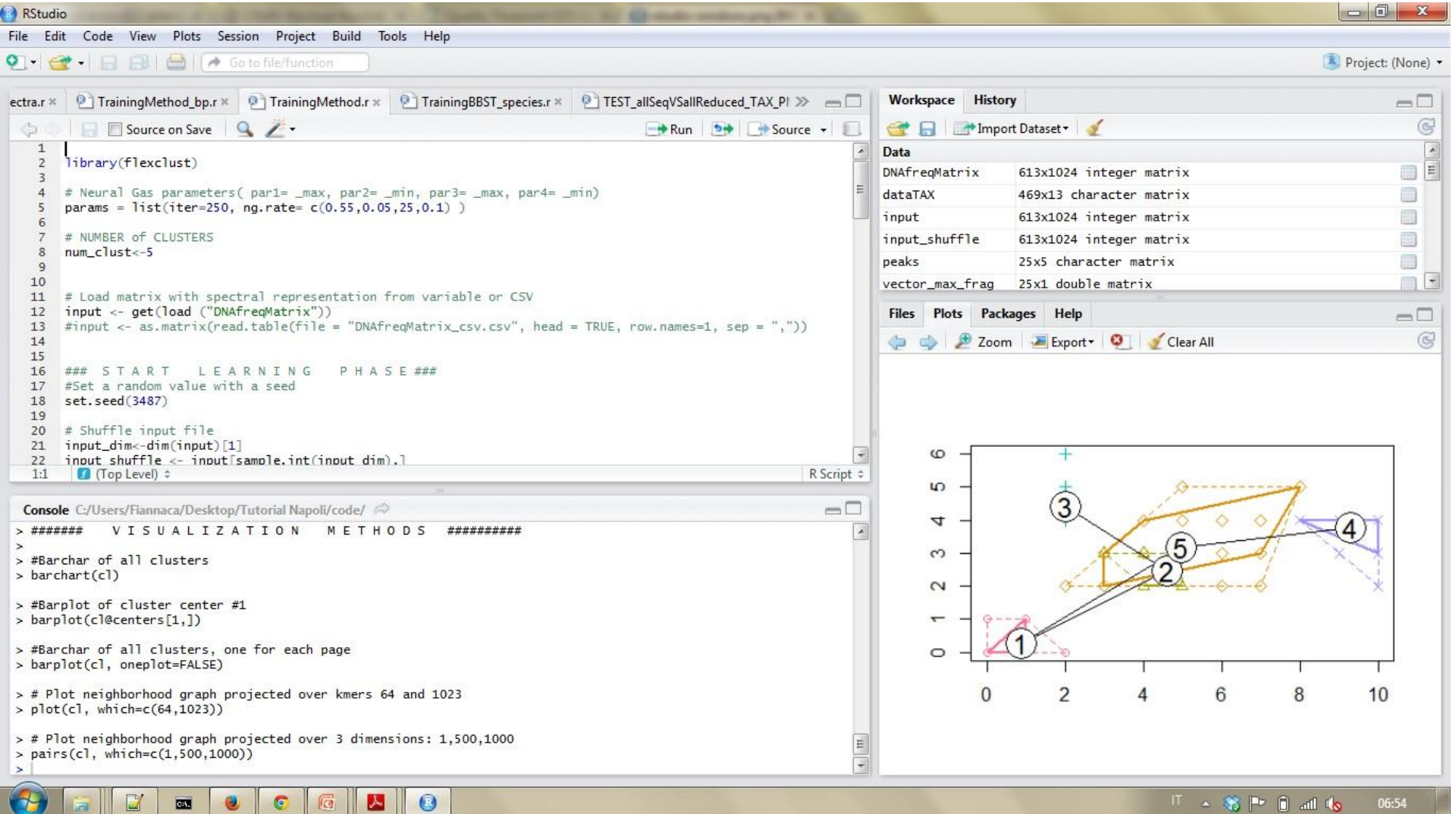
- Bioconductor (www.bioconductor.org)
  - contains several packages with many R functions for the analysis and comprehension of genomic data generated by wet lab experiments in molecular biology.
- Seqinr (pbil.univ-lyon1.fr/software/seqinr/home.php?lang=eng)
  - contains R functions for obtaining sequences from DNA and protein sequence databases, and for analysing DNA and protein sequences.

R Packages for KCCA

- FlexClust (http://cran.r-project.org/web/packages/flexclust/)
  - implements a general framework for k-centroids cluster analysis supporting arbitrary distance measures and centroid computation.
  - Cluster methods: k-means, hard competitive learning and neural gas clustering.
  - There are numerous visualization methods for cluster results (neighborhood graphs, barcharts of centroids, etc.)

RStudio IDE

A free and open source integrated development



Examples

1. Create a spectral representation of DNA Barcode sequences.
2. Train a neural gas for DNA sequences classification.
3. Select the best number of K centroids in training process.
4. Test classifier with cross-validation.

5 Datasets (613 Sequences) from BOLD

Code	Dataset Title	Specimens	Taxa
AHNFE	WG1.8 Marine Bio-Surveillance	133	Spe(3)
BBST	Bee Barcoding Initiative	164	Spe(10)
FUCUB	Marine Life (MarBOL)	111	Spe(3)
LHSMI	Human Pathogens and Zoonoses	103	Spe(11)
PMF	All Birds Barcoding Initiative	102	Fam(6); Gen(7); Spe(8)