FILTERING WITH ALIGNMENT FREE DISTANCES FOR HIGH THROUGHPUT DNA READS ASSEMBLY

Maria de Cola, Giovanni Felici,
Daniele Santoni, Emanuel Weitschek

Istituto di Analisi dei Sistemi ed Informatica
Consiglio Nazionale delle Ricerche
Roma

National Research Council of Italy

Background

- high throughput next generation sequencing (NGS) machine: large collection of short DNA fragments, or *reads* (40-200 bp)

- The DNA sequence assembly process is based on aligning and merging the reads for effectively reconstructing the real primary structure of the DNA sample sequence or reference genome.

Assembly Methods

Overlap Graph

- Each read and its complement correspond to a node
- the overlaps between pairs of reads are calculated with alignment methods (Needleman & Wunsch) and determine the weight of the arcs between nodes
- A **hamiltonian path** in the graph is a good assembly

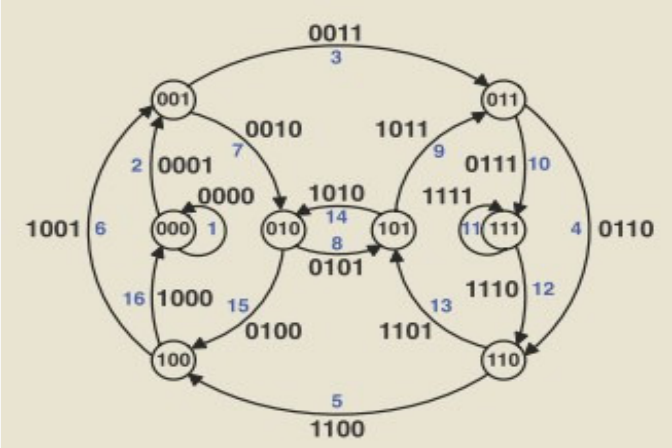
Drawbacks: alignment algorithm takes **O(kl)**, where **k** and **l** are the lengths of the sequences.

The number of possible alignments is **$O(n^2)$** where n is the number of sequences.

Most of the sequences do not overlap with each other in a satisfying manner.

De Bruijn Graphs [1]

- Reads are represented on a graph whose nodes and arcs are nucleotides subsequences.
- Assembly is found searching for an **eulerian** cycle in this graph and is represented by a sequence of arcs



Problem: fast filtering

- select in a fast way the pairs of reads which possibly give high score of the alignment, then use overlap graph on the selected pairs

Solution: alignment-free distances

- Similarity of two strings is assessed based only on a dictionary of substrings, irrespective of their relative position.

Dictionary of substrings D

- **$F(d_i)$, $d_i \in D$**: Frequency of each substring

  (% of the appearance of that substring in the sequence)
- Each string is represented with a **profile** over the dictionary D
- Two strings can be compared according to the distance between their profiles

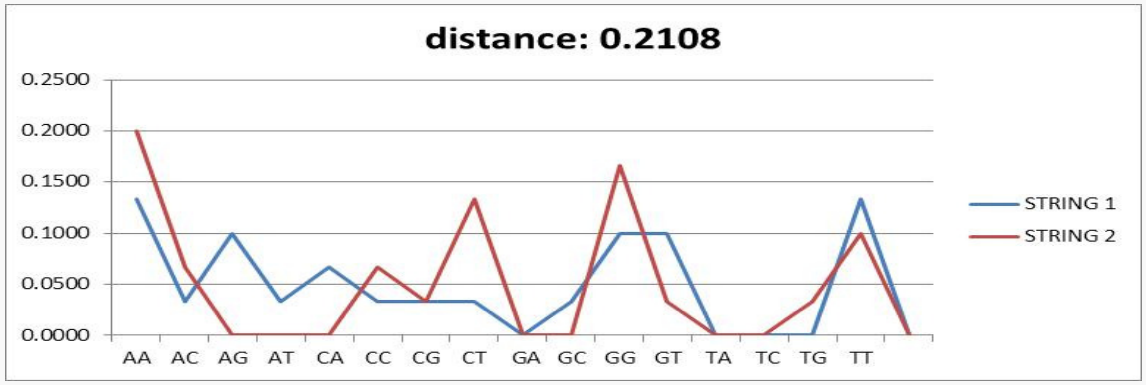
No need to align the two strings
Extremely fast (O(k)) and easy to parallelize

Example

Dictionary D = {AA,AC,AG,AT,CA,CC,CG,CT, GA,GC,GG,GT, TA, TC,TG,TT}

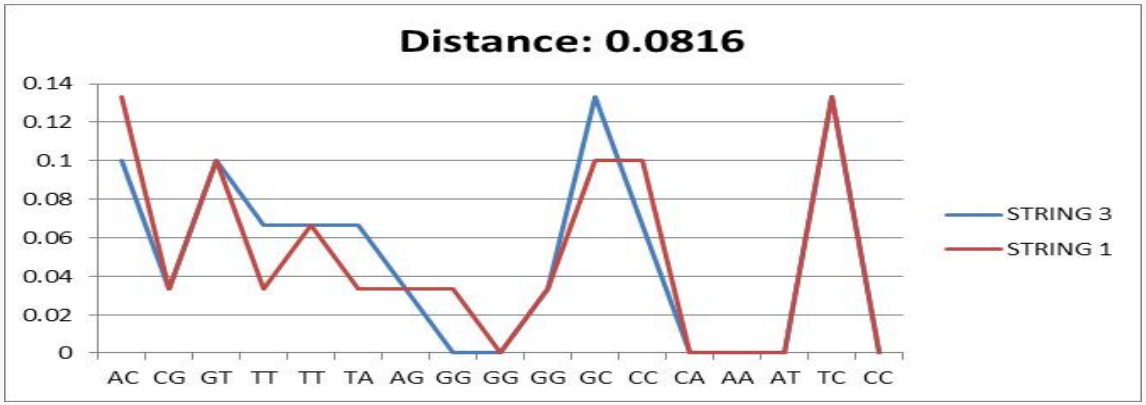
String 1: ACGTTTAAGGCCAATCTCAGGTTTAAAGGT

String 2: AAAAAACCTTTCTCTTCTGGGGGTAACCGG



String 3: ACGTTTAGGGGCCAATCCAGATTTAAAGGT

String 1: ACGTTTAAGGCCAATCTCAGGTTTAAAGGT



Distances between profiles

The distance d_{ij} between two profiles f_i and f_j measures their similarity.

- Euclidean distance: $$d_{ij} = \sum_{k=0}^{z} \sqrt{(f_{ik} - f_{jk})^2}$$

- Zero distance: $$\tilde{d}_{ij} = \sum_{k=0}^{z} D_{ij}$$

Where $$D_{ij} = \begin{cases} 1 \text{ if } \dfrac{| f_{ik} - f_{jk} |}{\frac{1}{2}(f_{ik} + f_{jk})} < t \\ 0 \text{ otherwise} \end{cases}$$

Alignment Free distances tuning

- Length of the words in the dictionary: substrings of length k, obtained with a sliding window
- Type of distance
- Frequency normalization: expected value of each word based on its substrings
- Low complexity regions

We test the use of AFD to filter good read pairs to be assembled

Very fast: the method operates in constant time in the string length

Positive Bias: if distance is large, the strings are different;

if distance is small, they may also be different:

$$D(S_1, S_2 \cup S_3) \approx d(S_1, S_3 \cup S_2)$$

Our experiments

We test the ability of AF distance to «approximate» other distances between strings that are more difficult to compute.

1. Take a set of reads from an organism;
2. Take all read-pairs
3. Compute distance beween each pair
4. Analyze the similarity of the two functions over the set of pairs, using:

   a) The **correlation** between the two functions
   b) The **ability to predict** a threshold value of one function using the threshold value of the other, as follows:

      A distance function F1 is used to predict a distance function F2; given α_1, α_2, we want to know how precise is the following rule:

$$IF\ (F1 < \alpha_1)\ THEN\ (F2 < \alpha_2)$$

Distances

AF: Alignment free — euclidean distance between the relative frequencies of the 256 4-mer (AAAA, AAAC, AACA,…,TTTT)

NW: Needleman-Wunsch — quality measure of the aligment that minimizes the Edit distance between the 2 strings, using also a substitution matrix and other tuning paramaters [2]

BT: Bowtie Distance — this is the **IDEAL** distance, as it is computed using the knowledge of the original sequence from which the reads have been sampled. How to compute it:

1. align the reads along the genome with Bowtie [3]
2. use as distance between two reads is the length of their intersection on the genome
3. If no intersection, then distance is maximum (1)

Motivation

- BT distance supports an alignment that returns the originating sequence
- It is used only for testing AF and NW to see how good they are
- If a distance function is strongly correlated with BT we can expect that it can be successfully used for DNA assembly in an Overlap Graph

Questions

1. Are we happy to filter out non promising pairs using AF before using NW in the overlap graph ?
2. Do we need at all to use the more time consuming NW distance?

Experiments have been designed to answer these questions.

Good Predictors

Recall that a distance function F1 is used to predict a distance function F2 as follows.
Given α_1, α_2: IF ($F1 < \alpha_1$) THEN ($F2 < \alpha_2$)

True Positive (**TP**): cases where ($F1 < \alpha_1$) AND ($F2 < \alpha_2$)
True Negative (**TN**): cases where ($F1 \geq \alpha_1$) AND ($F2 \geq \alpha_2$)
False Positive (**FP**): cases where ($F1 < \alpha_1$) AND ($F2 \geq \alpha_2$)
False Negative (**FN**): cases where ($F1 \geq \alpha_1$) AND ($F2 < \alpha_2$)

AN = all positive cases, AN = all negative cases.
The level of α_1 and α are sampled in 0-1 with step 0.01
F1 is **a good predictor** for ($F2$, α_2) if there exists α such that:

1. TP/AP > 80%
2. TN/AN > 80%
3. (FP+FN)/(AP+AN) < 10%

We would like to find many **good predictors** for all interesting values of α_2

Experiments on Ecoli Genome

Average Length of reads **234.54**
- Standard Deviation **9.82**

- Reads are aligned to the reference sequence with Bowtie
- After Alignment, 100.000 reads are sampled at random
- Reads are considered both forward and reversed for a total of 200k
- A total of 200.000^2 pairs are avaialble
- All pairs of reads with Bowtie distance < 1 are considered (**620,798**)
- Out of the remaining (100.000 x 100.000 – 620.798) pairs with BT distance 1, we sample at random **233,099** reads (less than 1%)

- The data set is finally composed of **853,897** pairs of reads

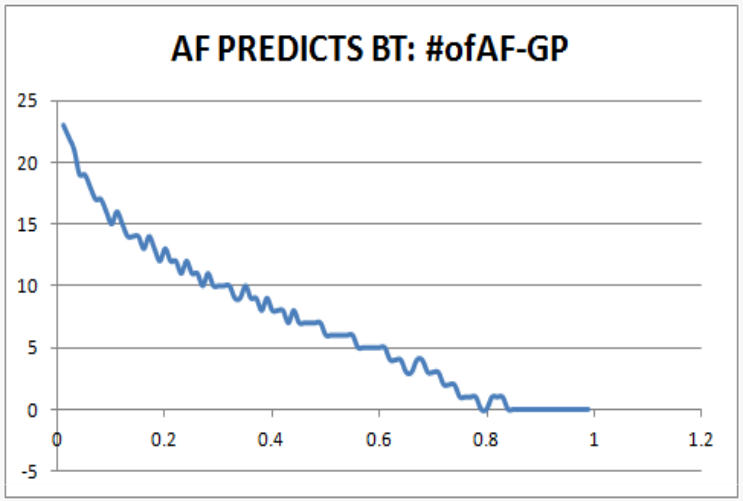
Correlation

AF-BT	AF-NW	NW-BT
0.761	0.721	0.702

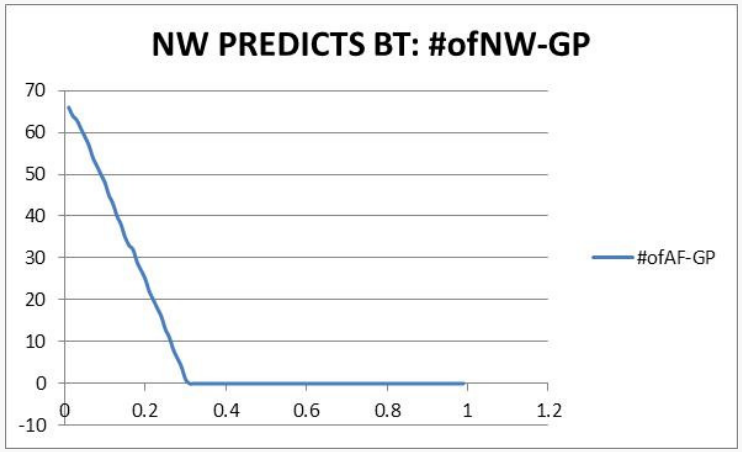
Good Predictors

Experiments on Ecoli Genome

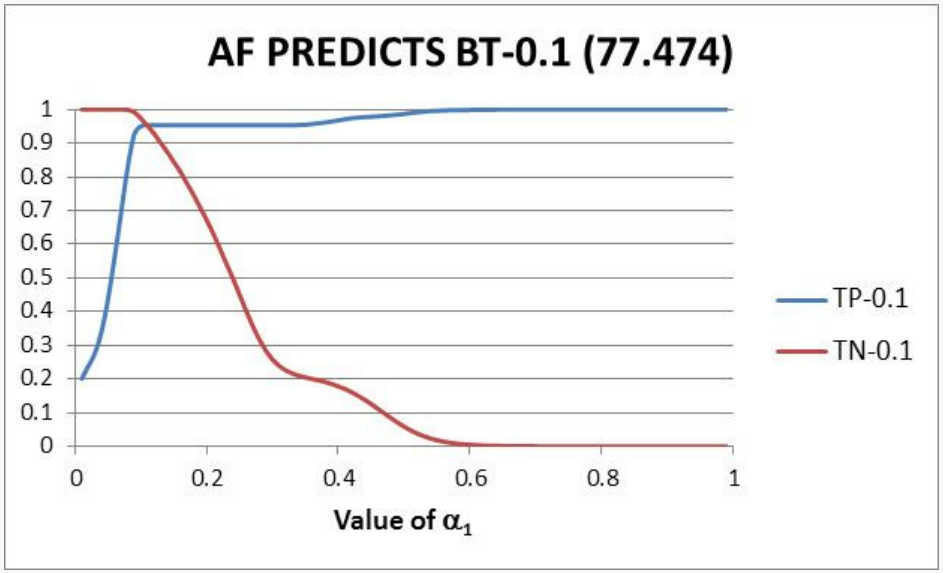
How many good AF predictors are there for any α of BT ?

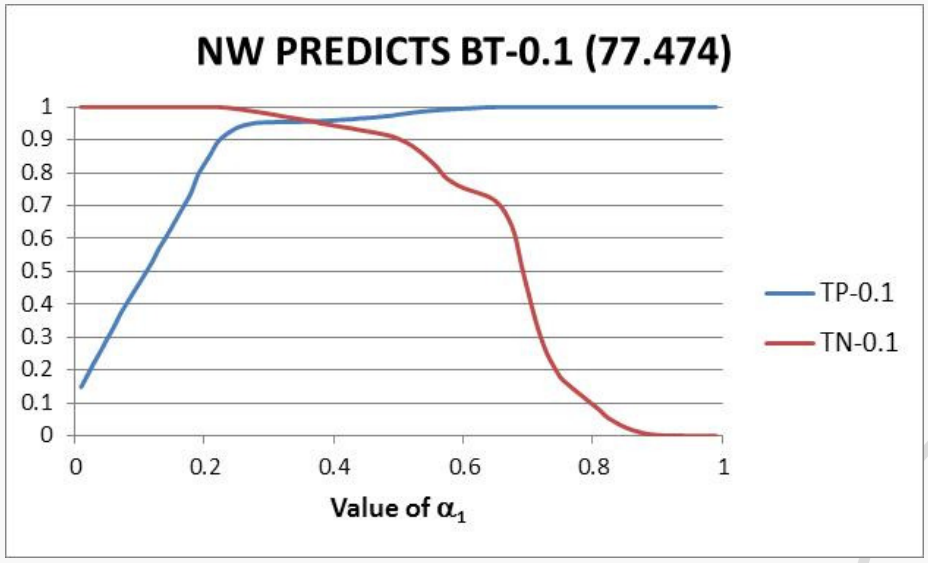


How many good NW predictors are there for any α of BT ?



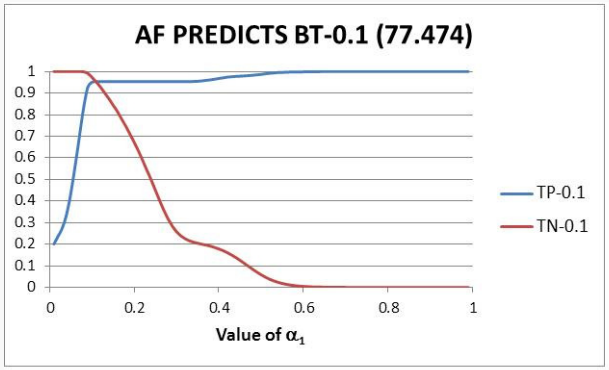
Experiments on Ecoli Genome

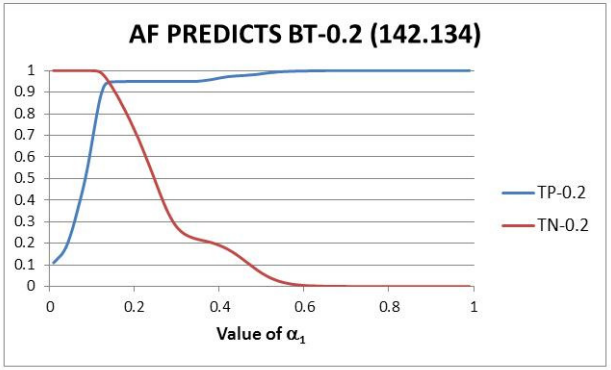


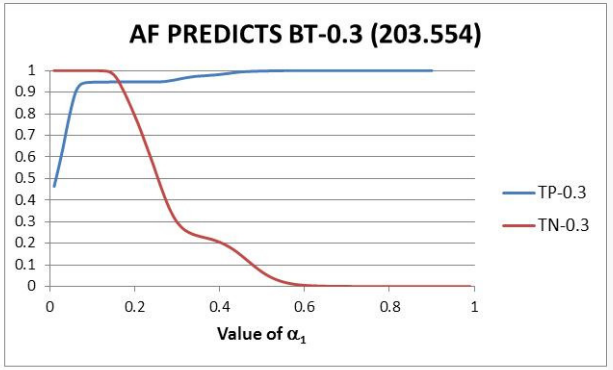


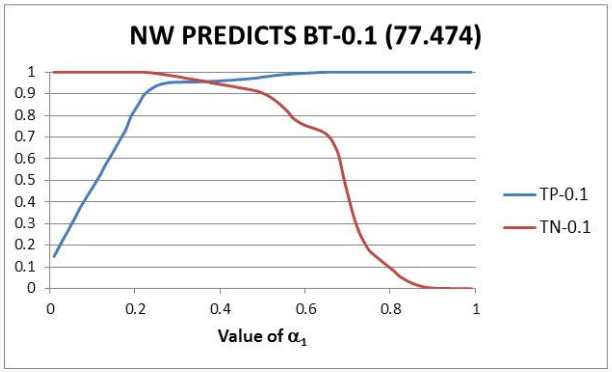
AF predicts BT

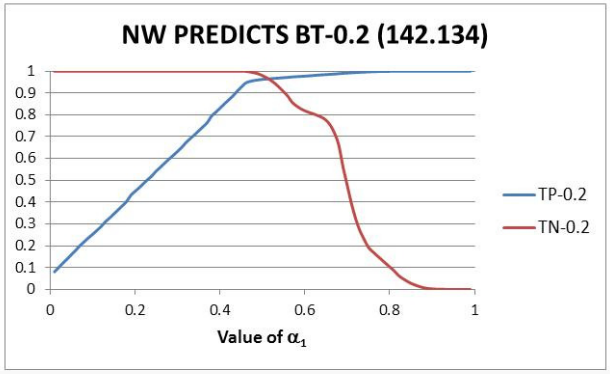
Experiments on Ecoli Genome

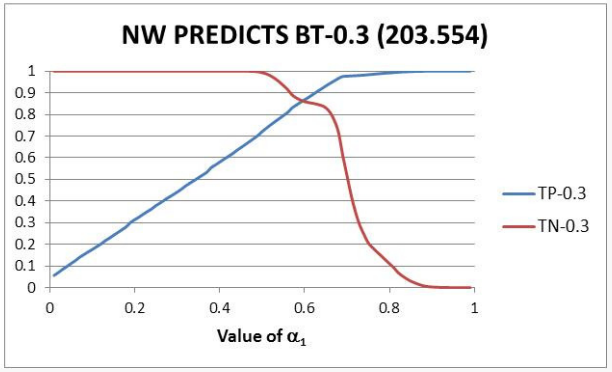












NW predicts BT

Experiments on Human Genome

Length of reads **46**

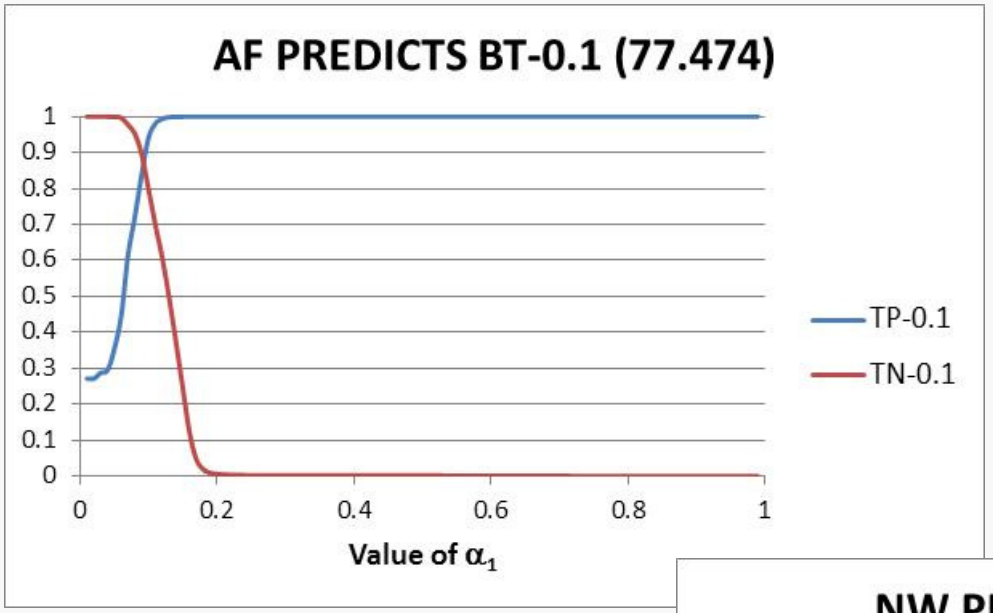
- Reads are aligned to the reference sequence with Bowtie
- After Alignment, 50,000 reads are sampled at random
- Reads are considered both forward and reversed for a total of 100k
- A total of $100{,}000^2$ pairs are avaialble
- All pairs of reads with Bowtie distance < 1 are considered (**994,904**)
- Out of the remaining (100,000 x 100,000 – **994,904**) pairs with BT distance 1, we sample at random **53,670** reads (less than 1%)

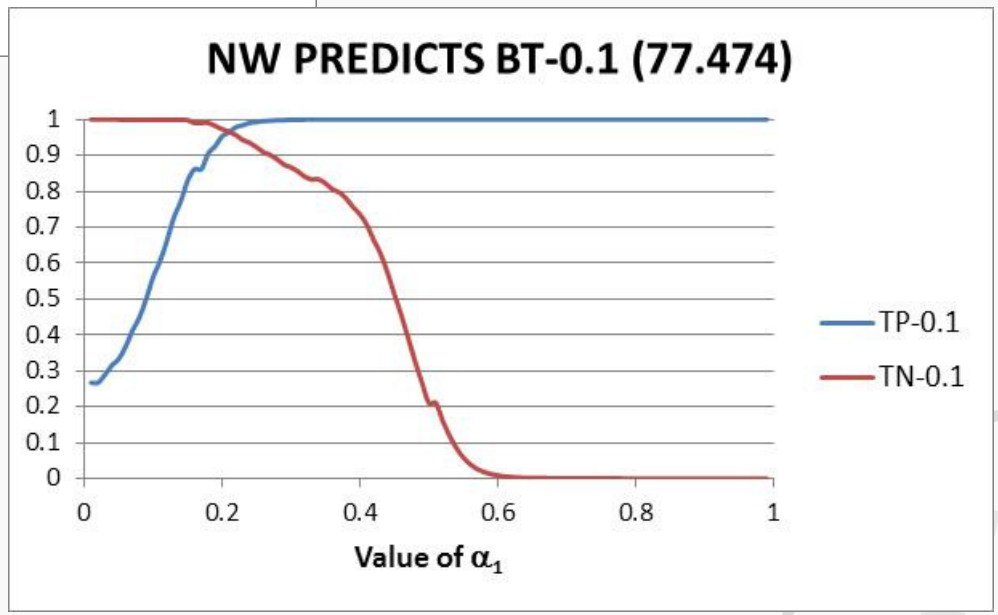
- The data set is finally composed of **1,048,574** pairs of reads

Correlation

AF-BT	AF-NW	NW-BT
0.785	0.796	0.757

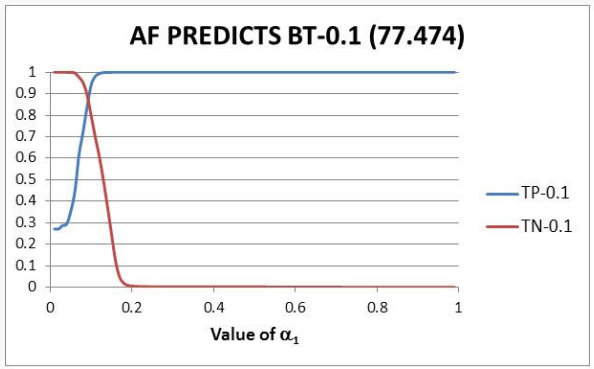
Experiments on Human Genome

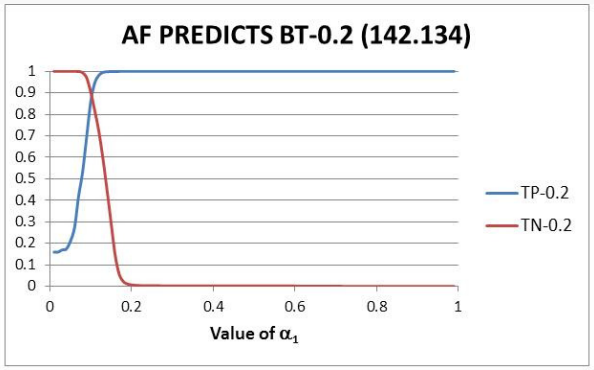


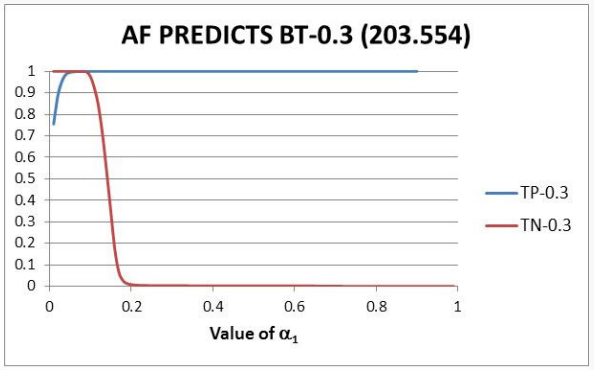


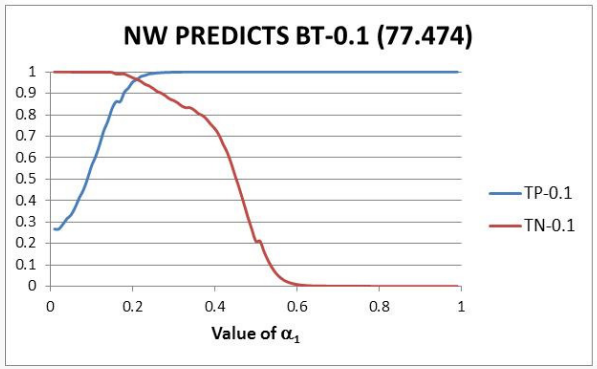
Experiments on Human Genome

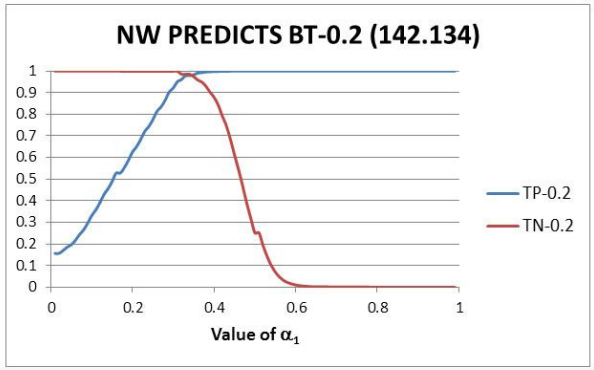
AF predicts BT

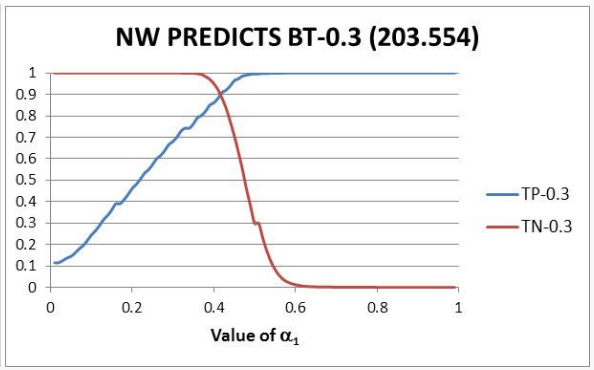












NW predicts BT

Conclusions

- AF is a very **good threshold predictor for BT** for the considered data
- If performs **better or equivalently** than the more complex NW edit distance when its ability to support a threshold predictor is considered
- There is evidence that AF **can be used for read filtering** in DNA assembly algorithms
- The results seem slightly **more robust on Ecoli than on Human**, likely due to the different read length

Future Work

- **Refine** AF distance
- Test on **larger** samples
- **Reinforce** results with statistical tests
- Experiment on **assembly** methods (on going)

The authors are partially supported by the FLAGSHIP ”InterOmics” project (PB.P05) funded by the Italian MIUR and CNR institutions, and by the cooperative programme 2010–2012 between the National Research Council of Italy (CNR) and the Polish Academy of Sciences (PAN).

Istituto di Analisi dei Sistemi ed Informatica
Consiglio Nazionale delle Ricerche

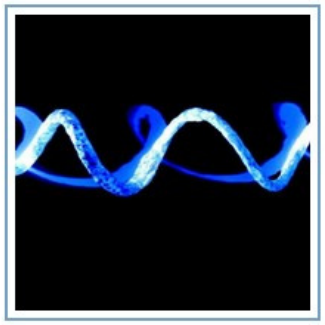
DMB Data Mining Big

ROMA TRE UNIVERSITÀ DEGLI STUDI

Home | Software | Publications | Collaborations | Downloads | About Us | FAQ | News

DMB (Data Mining Big)

DMB (Data Mining Big) is a set of data analysis tools.
DMB contains a collection of software tools that perform knowledge extraction from data. The methods adopted are based on several models and algorithms that have been developed by a team of researchers, most of them members of the computational and system biology research group. The description of the methods is available in different papers listed in the publication page, while the code can be executed remotely from this website on the servers of IASI - CNR - a research institute of the Italian Research Council. Results are sent by email or visualized on the web interface.



The methods adopted are based on several optimization models and algorithms. The algorithms used for the DMB System are highly accurated, efficient and innovative. Every single data is checked and examinated accurately by our analysis programs to guarantee an adeguate classification.

The main characteristic of the DMB system is that it extracts knowledge in form of logic rules. DMB takes an input a matrix with the elements and their attributes, plus a class label for each element and related labels. Regardless of the form of the input data - rational or discrete numbers, qualitative attributes, binary or logic values, it always returns as output an explanation of the type `if (X is in Ax) and (Y is in Ay) or (Z is in Az) then CLASS = C`, where `X`, `Y`, `Z` are attributes of the elements, `Ax` (`Ay`, `Az`) are set of possible values that the attributes can take, and `C` is one of the classes in which the elements are partitioned.

The different modules of DMB are composed to solve different types of classification problems. Each composition of modules that performs a complete analysis is called a "flow". In the software menu are listed the different flows currently implemented, and a more detailed explanation for their use.

Latest News

NEW SOFTWARE AVAILABLE
July 2012

The Cytoscape plugin BiNAT (Biological Networks Analysis Tool) has been developed by Fabio Cumbo.

NEW PUBLICATIONS ADDED
March 2012

New journal articles have been published.

NEW DOWNLOADS ADDED
March 2012

In downloads section you can find offline versions and data sets.

Read more news

[1] How to apply de Bruijn graphs to genome assembly, P. E. C. Compeau, P. A. Pevzner, G. Tesler, NatureBiotechnology, 29, 987–991 (2011)

[2] SB Needleman, CD Wunsch: A general method applicable to the search for similarities in the amino acid sequence of two proteins; Journal of molecular biology, 1970; Java implementation available at biojava.org

[3] Langmead B, Trapnell C, Pop M, Salzberg SL. Ultrafast and memory-efficient alignment of short DNA sequences to the human genome, Genome Biol. 2009;10(3)