A dockerized string analysis workflow for Big Data

Maria Kotouza
PhD candidate
Aristotle University of Thessaloniki, Greece
maria.kotouza@issel.ee.auth.gr

AUTh

Introduction

- **Data Science:** manipulation of data using mathematical and algorithmic methods to solve complex problems in an analytical way
- **Data of various types**: biological data, documents, energy consumption, etc.
  Big data + lack of generalized methods -> machine learning in large-scale infrastructures
- **Challenges**: high dimensionality, complexity and diversity of the data, limited resources, varying structures of the available analytic tools
- **Scientific workflows:** combine heterogeneous components to solve problems characterized by data diversity and high computational demands
- **Cloud computing:** a popular way of acquiring computing and storage resources on demand through virtualization technologies

Data Transformation into Strings

- Diversity of data
  - Need for expressing them in a common format
- We select to transform the input data into strings
  - Easy to handle them
  - Makes the whole process quicker
  - Lossy compression (in some cases)
    – controlled by the user
- Dockerization
  - Big data cannot fit in a single machine

Numeric Vectors

	Pos 1	Pos 2	..	Pos L
1	0.95	0.15	..	0.86
2	0.98	0.28	..	0.87
3	0.95	0.51	..	0.02
..	..	..	..	..
N	0.99	0.54	..	0.01

Character Vectors

	Pos 1	Pos 2	..	Pos L
1	A	R	..	F
2	A	R	..	F
3	A	K	..	Y
..	..	..	..	..
N	A	K	..	Y

Or

Strings

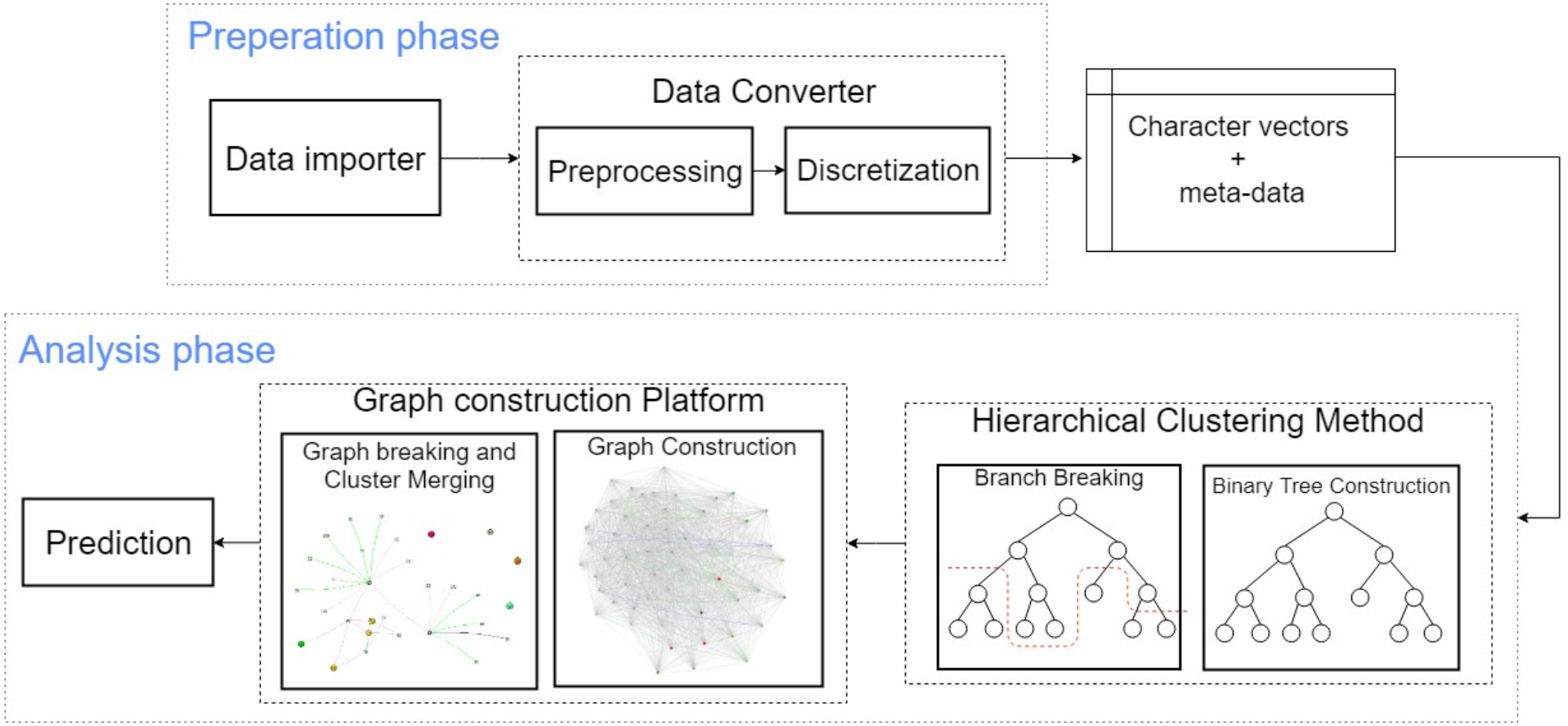
	Sequences
1	ARAYDFWSGYLF
2	ARVYDFWSGYLF
3	AKSGAIAAAGDY
..	..
N	AKSGTIAAAGDY

Dockerized String Analysis workflow (DSA)

The main objectives of DSA are:

1. **Transform** input data into internal format, considering domain specific features
2. Create **custom pipelines** based on the user preferences
3. Provide **analytics services** integrating new scalable tools
4. Provide **visualization services** that can support decision-making
5. Be available in both **script-based format** and in a **graphical interface**
6. Be suitable for **cloud infrastructures**

The DSA workflow architecture



docker

Takes into account: a) Domain-specific characteristics

b) User preferences

Preparation phase

The preparation phase includes **data importing** and **transformation**, in order for the input data to be reformatted as a set of

Character vectors + meta-data

Data importer: Acquire the data to be analyzed in specific supported formats based on their domain

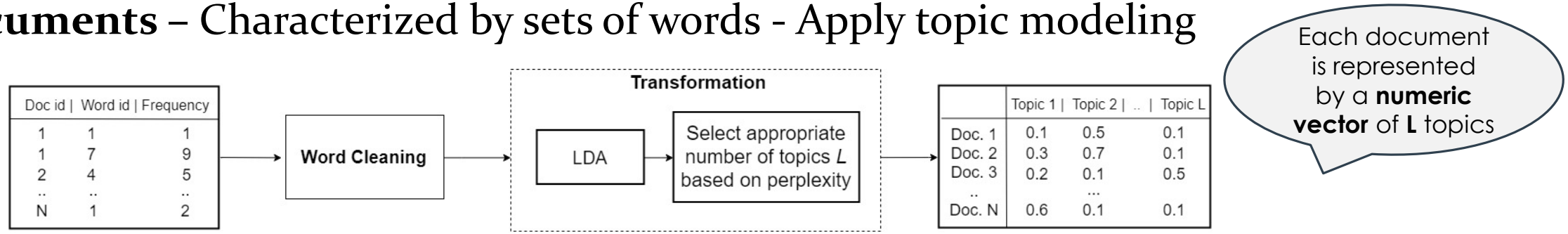
Preprocessing module: Clean the input data and transform them into a general format which is required by the analysis phase. Data are transformed into vectors of values accompanied with the appropriate meta-data depending on the domain.

Discretization module:

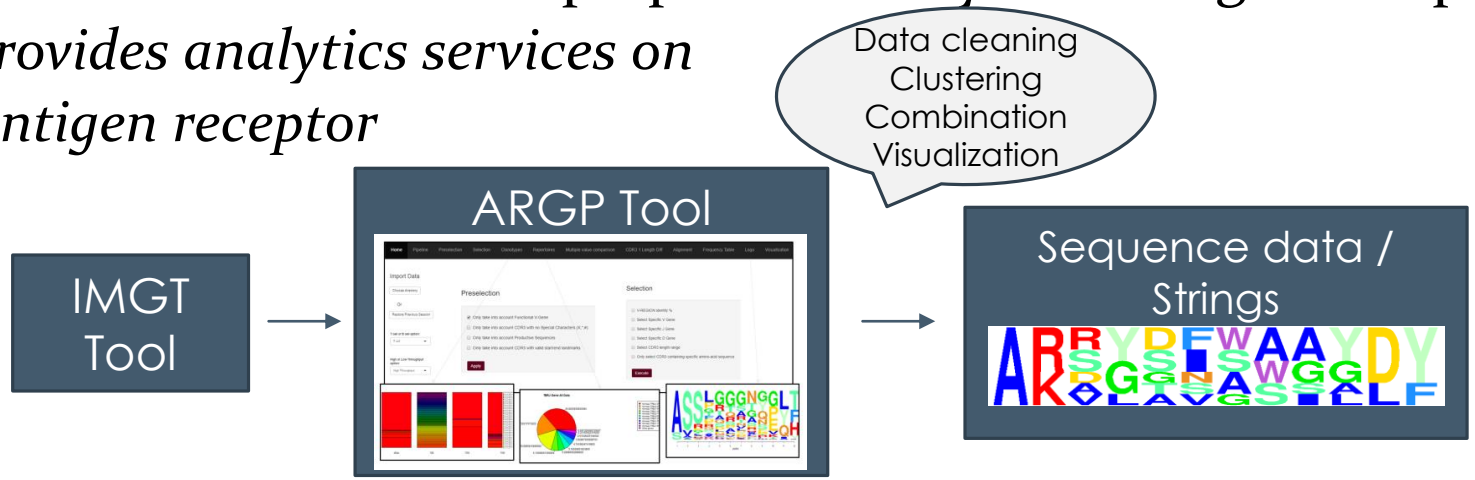
- The numeric vectors are discretized into partitions of **length *B*** by assigning each value into a **bin** based on the closed interval where it belongs to
- By making use of letters to represent the bins, the numeric vectors are converted into **strings**

Preprocessing module per domain

Documents – Characterized by sets of words - Apply topic modeling



Gene sequence data - Data are preprocessed by the Antigen receptor gene profiler (ARGP) - *Provides analytics services on antigen receptor*



Time series data - Data cleaning, normalization, missing value handling etc.

Analysis phase

Clustering module: A new scalable multi-metric algorithm for hierarchical clustering is applied. It is a Frequency Based Clustering (FBC) algorithm [1]

It consists of:

Binary Tree Construction + Branch Breaking

Algorithms

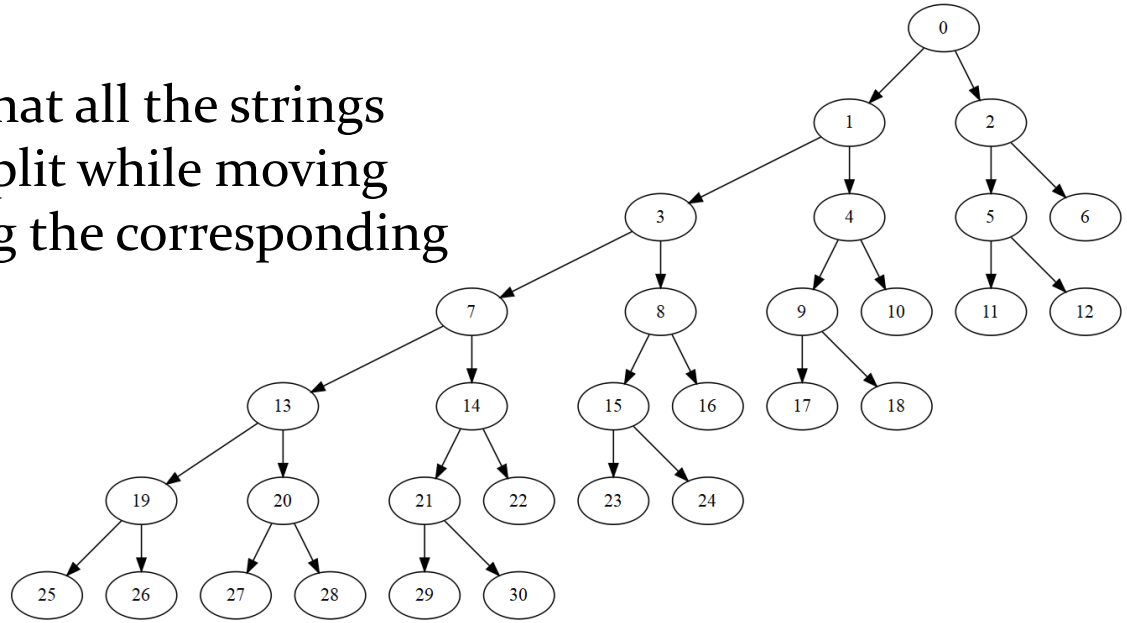
Graph mining module: Using clustering results in combination with graph construction techniques, we provide information about the data relationships in a graphical interactive environment. Graph mining metrics and graph clustering algorithms for sub-graph creation are also utilized.

Prediction module: Integrates the results from the previous modules to train a model that can make predictions for missing connections of data and classify new items.

[1] Kotouza, M., Vavliakis, K., Psomopoulos, F., & Mitkas, P. (2018, December). A hierarchical multi-metric framework for item clustering. In 2018 IEEE/ACM 5th International Conference on Big Data Computing Applications and Technologies (BDCAT) (pp. 191-197). IEEE.

Binary Tree Construction Algorithm (Overview)

- A top down hierarchical clustering method
- It is based on the usage of a matrix that contains the frequencies for each position of the target strings (FM)
- At the beginning of the process, it is assumed that all the strings belong to a single cluster, which is recursively split while moving along the different levels of the tree, by splitting the corresponding FM
- Metrics:
  - Identity
  - Entropy
  - Bin Similarity



Theoretical basis

- *Frequency Matrix:* $FM \rightarrow B \times L$

Each element (i,j) of the matrix corresponds to the number of times bin i is present in positions j for all the strings

- *Identity:* $I = \sum_{j=1}^{L} id_j / L \quad , \quad id_j = \begin{cases} 100\%, & if\ max(FM_{ij}) = 100\% \\ 0\%, & if\ max(FM_{ij}) \neq 100\% \end{cases}$

The percentage of sequences with an exact alignment

- *Entropy:* $H_j = -\sum p_i \log(p_i)$

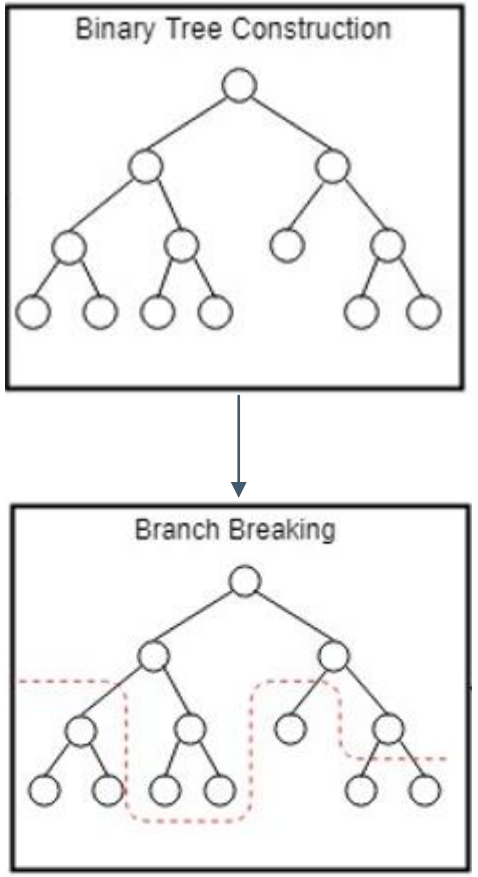
Represents the diversity of the column

- *Bin Similarity:* $BS = \sum_{j=1}^{L} BSM_j / L$, *BSM is a weighted version of FM*

Calculated using the similarities of the bins that participate in each topic

Branch Breaking Algorithm

- Asymmetric tree, the number of items that each cluster consists of varies
  -> the tree cannot be cut by selecting a unique level for the overall tree
  -> for each branch, the appropriate level to be cut is examined
- The parent cluster is compared to its two children clusters recursively as one goes down through the path of the tree branch
- The comparison is applied using the metrics that have been computed for each cluster C_i (I_i, H_i, BS_i) and user selected thresholds for each metric (*thrI*, *thrH*, *thrBS*)



Analysis phase (2)

Clustering module: A new scalable multi-metric algorithm for hierarchical clustering is applied.

It consists of the *Binary Tree Construction* and the *Branch Breaking* algorithms.

Graph mining module: Using clustering results in combination with graph construction techniques, we provide information about the data relationships in a graphical interactive environment. Graph mining metrics and graph clustering algorithms for sub-graph creation are also utilized.

Prediction module: Integrates the results from the previous modules to train a model that can make predictions for missing connections of data and classify new items.

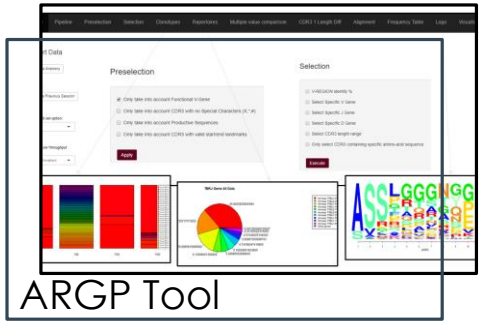
Network embedding – Application of Machine Learning techniques

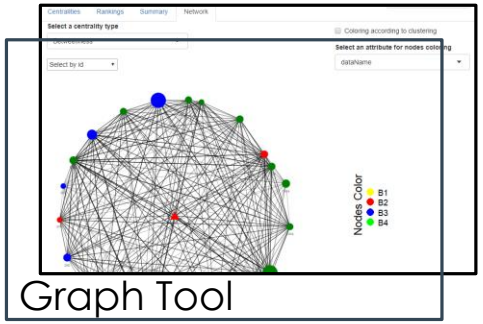
Software Implementation

- The modules are available in
  - **Script-based format**:
    - Command line interface
    - Faster execution
  - **Graphical user interface**:
    - For domain experts with limited technical experience

- The workflow components are dockerized
  -> able to run in cloud infrastructures

- All the modules are combined and described together using the Common Workflow Language (CWL)





Results

- *Case study 1*: Documents
- *Case study 2*: Gene sequence data
- *Case study 3*: Time series data -- in progress

Case study 1: Documents [1]

- We used benchmark data provided by the popular MovieLens 20M dataset:
  - 27,000 movies
- We created 20-length item vectors after applying LDA on the documents
- The item vectors were then discretized in 10 bins represented by alphabetic letters from A (90-100%) to J (0-10%)
- The groups of similar bins that were used are non-overlapping and are given by pairing bins in descending order i.e. <A,B>, <C,D>, <E,F>, <G,H>
- The results of the FBC algorithm were compared with those obtained by a Baseline Divisive Hierarchical Clustering (BHC) algorithm

#C	Algorithm	I	H	BS
23	BHC	13.696	0.167	85.769
	FBC	**74.783**	**0.081**	**93.264**
53	BHC	35.189	0.139	89.847
	FBC	**80.849**	**0.066**	**94.237**
125	BHC	53.080	0.120	92.886
	FBC	**90.600**	**0.038**	**96.981**

Performance results:
98% reduction in memory usage
99.4% reduction in computational time

[1] Kotouza, M., Vavliakis, K., Psomopoulos, F., & Mitkas, P. (2018, December). A hierarchical multi-metric framework for item clustering. In 2018 IEEE/ACM 5th International Conference on Big Data Computing Applications and Technologies (BDCAT) (pp. 191-197). IEEE.

Case study 2: Gene sequence data [2]

- We aimed to identify groups of patients based on a biologically important gene region of immunoglobulin
- Real-world dataset comprising 123 amino acid sequences of length 20, from patients with chronic lymphocytic leukemia
- The dataset was preprocessed using the ARGP tool
- FBC produced a binary tree with 19 levels
- The clustering results were assessed using the biological groups each sequence came from

Biological group	#Group seq/ #Cluster seq	Success rate	Level	Cluster
Subset #4	93/101	**92%**	4	13
Subset #4-34/20-1	2/2	**100%**	9	57
Subset #4-34-16	3/4	**75%**	5	31

[2] Tsarouchis, S. F., Kotouza, M. T., Psomopoulos, F. E., & Mitkas, P. A. (2018, May), A Multi-metric Algorithm for Hierarchical Clustering of Same-Length Protein Sequences, In IFIP International Conference on Artificial Intelligence Applications and Innovations (pp. 189-199). Springer, Cham.

Conclusion

- We present a workflow of **scalable algorithmic modules** that
  - Transforms the source data into strings, considering domain specific features
  - Provides big data analytic services
  - Provides fast execution of custom pipelines
  - Is dockerized

- Most of the modules of the workflow were applied on two practical case studies, showing promising results in terms of **efficiency** and **performance**

Future steps

- Adding further functionality on the graph mining module
- Development of the prediction module
- Further expansion of the work in more application fields, emphasizing in the source data transformation and the accurate representation of them
  - Time-series data
  - Data characterized by both numerical and verbal features

Thank you for your attention!

Maria Th. Kotouza
Fotis E. Psomopoulos
Pericles A. Mitkas

Electrical and Computer Engineering,
Aristotle University of Thessaloniki, Greece

maria.kotouza@issel.ee.auth.gr