Institute for Biomedical Technologies
CNR - Bari, IT

INFN Bari, IT
Istituto Nazionale di Fisica Nucleare

Department of Computer Science,
University of Bari, IT

Computational annotation of UTR cis-regulatory modules using the EGEE (gLite) infrastructure

Domenica D'Elia

CNR – Institute for Biomedical Technologies - Bari - Italy

4th EGEE User Forum - OGF 25 and OGF Europe's 2nd International Event
March 2-6, 2009 - Catania, Italy

This work is carried out within the LIBI project exploiting the EGEE production infrastructure

From gene to protein

Gene

yes/not
..at what extend?

DNA

TRANSCRIPTION

mRNA

when/where
..at what extend?

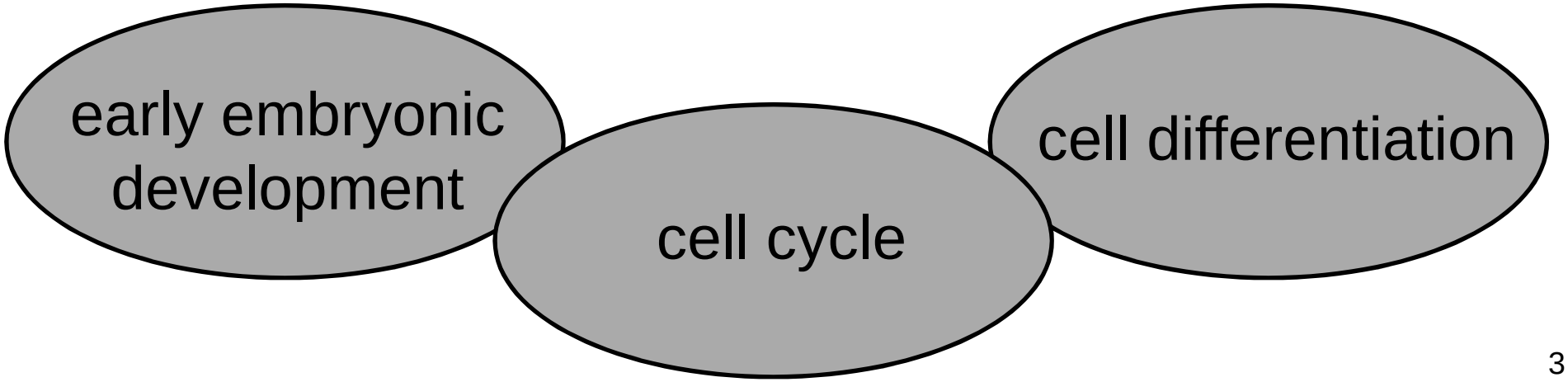
TRANSLATION

PROTEIN

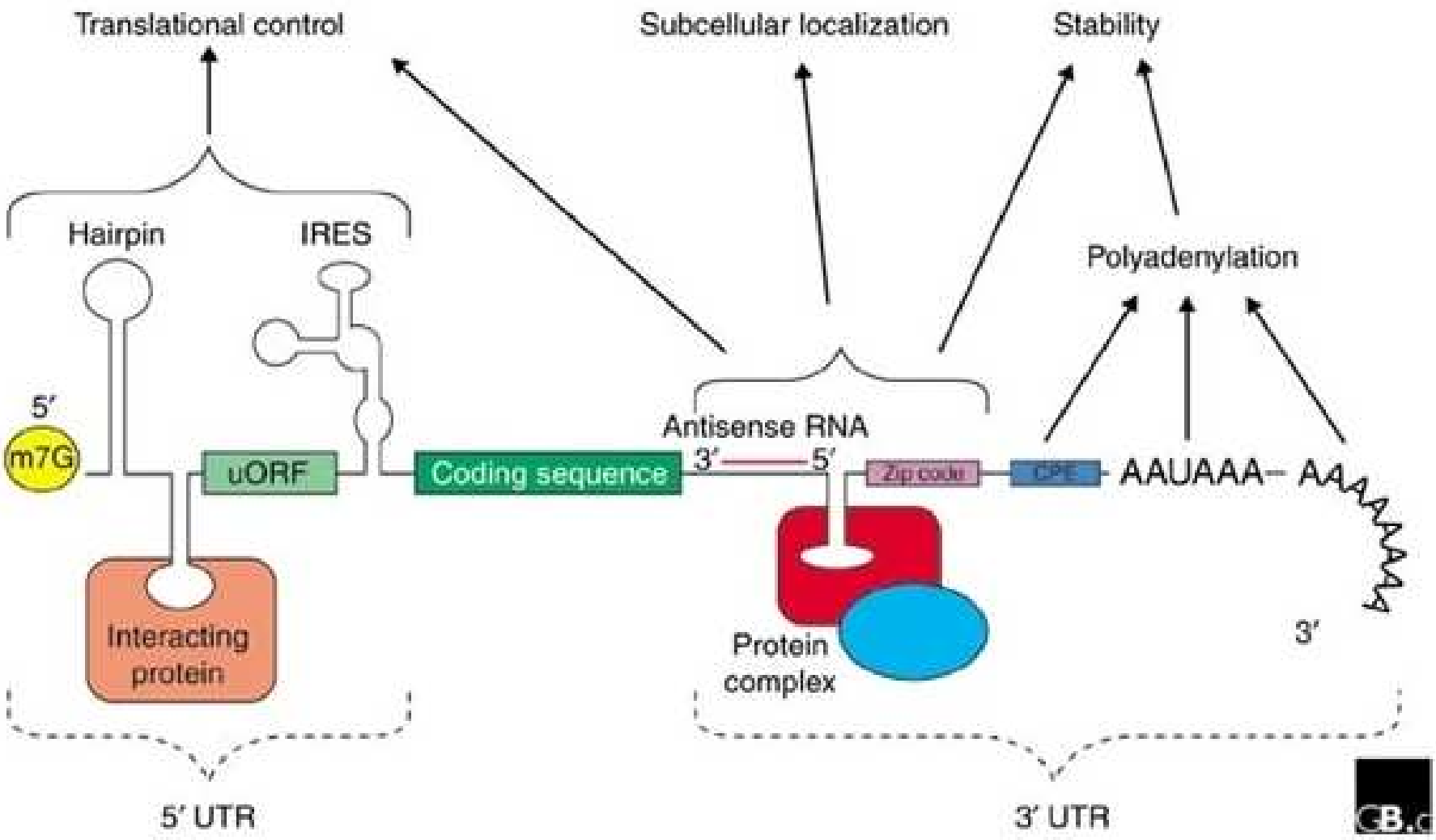
Translational control fine-tunes protein levels in time and space

Translation control modes:

- Global control
  - mainly occurs by the modification of translation-initiation factors
- mRNA specific-control
  - is driven by regulatory proteins that recognize functional motifs in the 5' and 3' UTRs



Different types of regulatory motifs involved in translational control (mRNA -> Protein)



Genome Biol. 2002; 3(3): reviews0004.1–reviews0004.10.
Published online 2002 February 28.

Research Goal

Annotation of UTRs cis-regulatory modules

The computational approach aim:

- extraction of implicit knowledge (relations) not explicitly annotated in current biological databases in order to discover **Frequent Sequential Patterns (FSP)** of Translation Regulatory Motifs **by using data mining techniques**

Biological Issue

The underlying biological assumption is that:

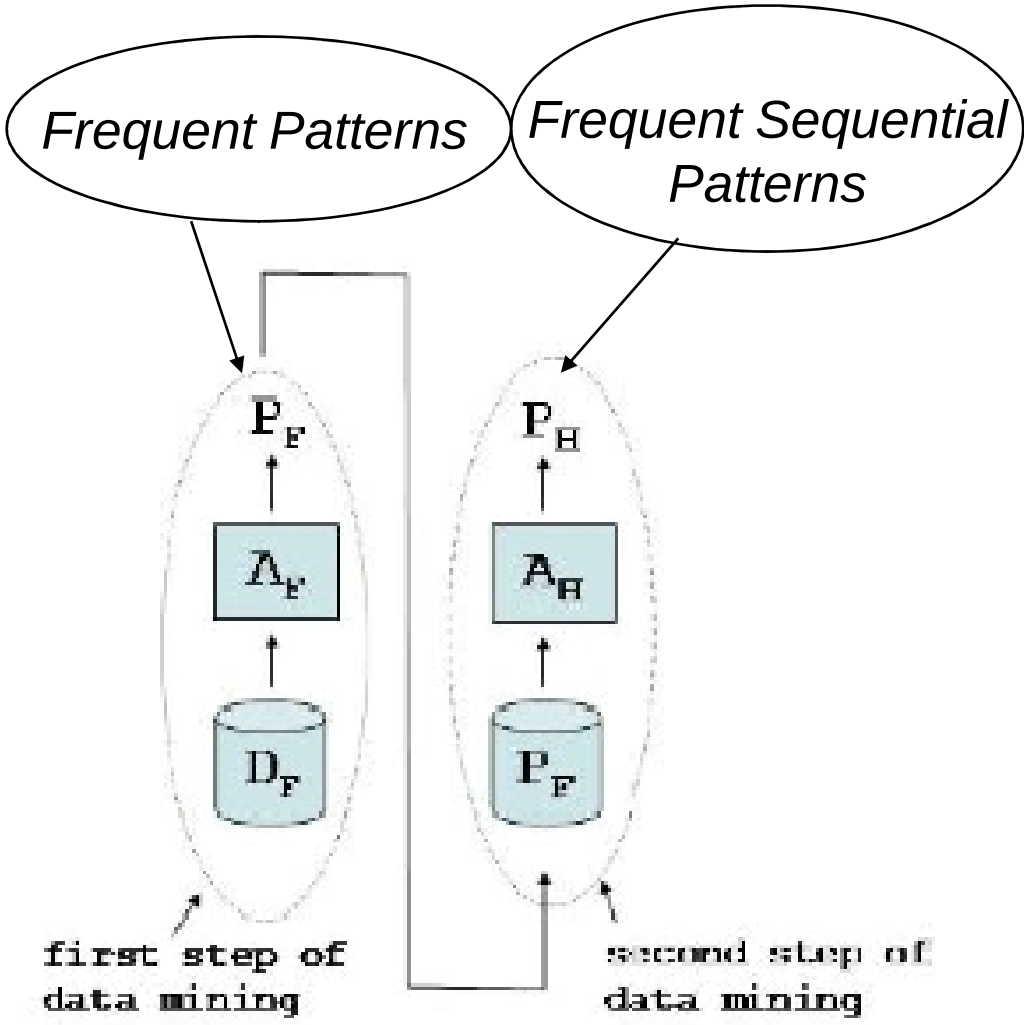
- a **pattern of regulatory motifs** which **appears with high frequency** in a set of biological sequences has been preserved for longer time during evolution and thus it **is expected to be biologically important** from a functional point of view

- **mutual distance** among motifs frequently co-occurring along a UTR sequence **may play a key role** for their functional interaction

Data Mining Approach

The approach is based on a two-stepped procedure using a **sequential pattern mining algorithm**

- The first step (A_F) generates frequent patterns of motifs (P_F) from a specific set of motifs previously mapped, without taking into account their spatial displacement
- The second step exploits a technique of sequence mining (A_H) on annotated motifs (P_F) to generate FSP of motifs by taking into account their spatial displacement and conservation of spacers



First mining step

Generation of Frequent Patterns (FPs):

- the search is based on the levelwise method by Mannila and Toivonen, i.e. a breadth-first search is performed in the lattice of sets of motifs
- starts from the smallest element (sets with a single motif) and proceeds from smaller to larger sets
- the set of motifs which are frequent at the i-th level are considered to generate candidate sets of motifs at the (i+1)-th level. Candidates are then evaluated against the input data in order to prune those that are infrequent

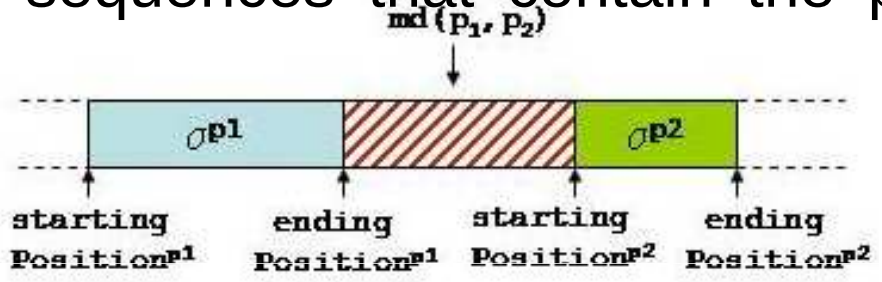
Advantage:

- timing the high computational complexity of the FSP generation process

Second mining step

Generation of Frequent Sequential Patterns (FSPs) by using GSP*:

- We are interested in **sequential patterns** which begin and end with a motif and have a greater support than an input threshold (*minsup*). The support of a sequential pattern is computed as the percentage of sequences that contain the pattern in the same order.



- Given 2 annotated motifs p_1 and p_2, they can be projected into a sequence of motifs ordered according to the values of the spacers between them $md(p_i, p_{i+1})$

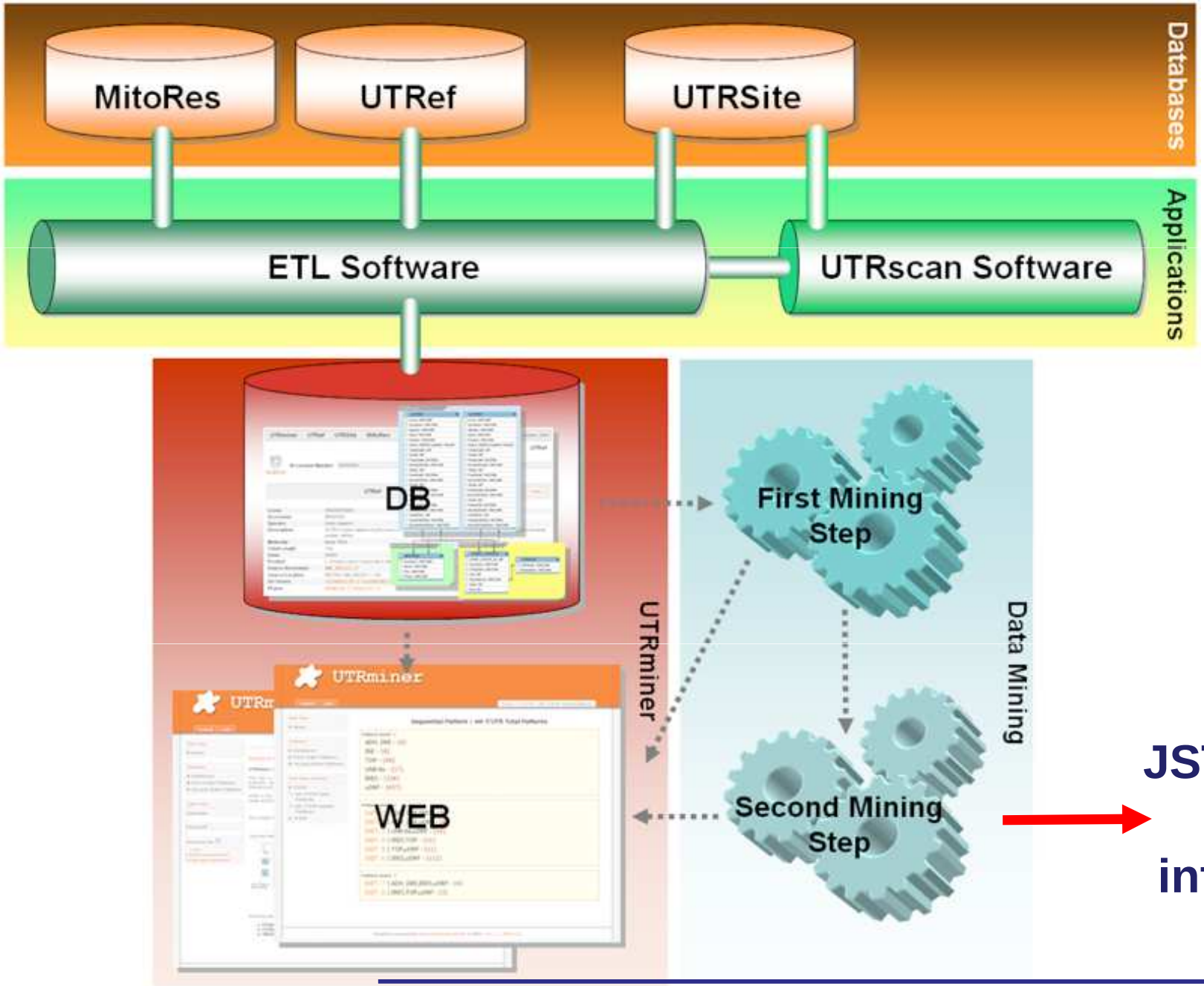
* *Generalized Sequential Pattern miner (Weka data mining tool)*

Second mining step

- the original numerical spacers (*Z) are mapped* into a finite set of bins/intervals (*Ψ = 6, 9, 12 …bins)* using an unsupervised equal frequency discretization method
- in this way each bin has a different width from each other and the initial values of each bin are distributed approximately in a uniform way
- at the end of the discretization each spacer is mapped into an interval and **GSP** can mine data and extract FSPs.

- The **number of FSPs** that can be detected **depends on:**
  - ✓input threshold set for minimum pattern support
  - ✓granularity of spacer length discretization
- **By varying** these **two parameters** it is possible to **flexible customize** the tool in order improve the quality of the computational analysis

The UTRminer resource



JST application LIBI GRID infrastructure

First mining step: experimentation and results

1. M-dataset (1944 5’UTR and 1952 3’UTR sequences from mitochondrial targeted mRNA of metazoan org.)

   FPs found:

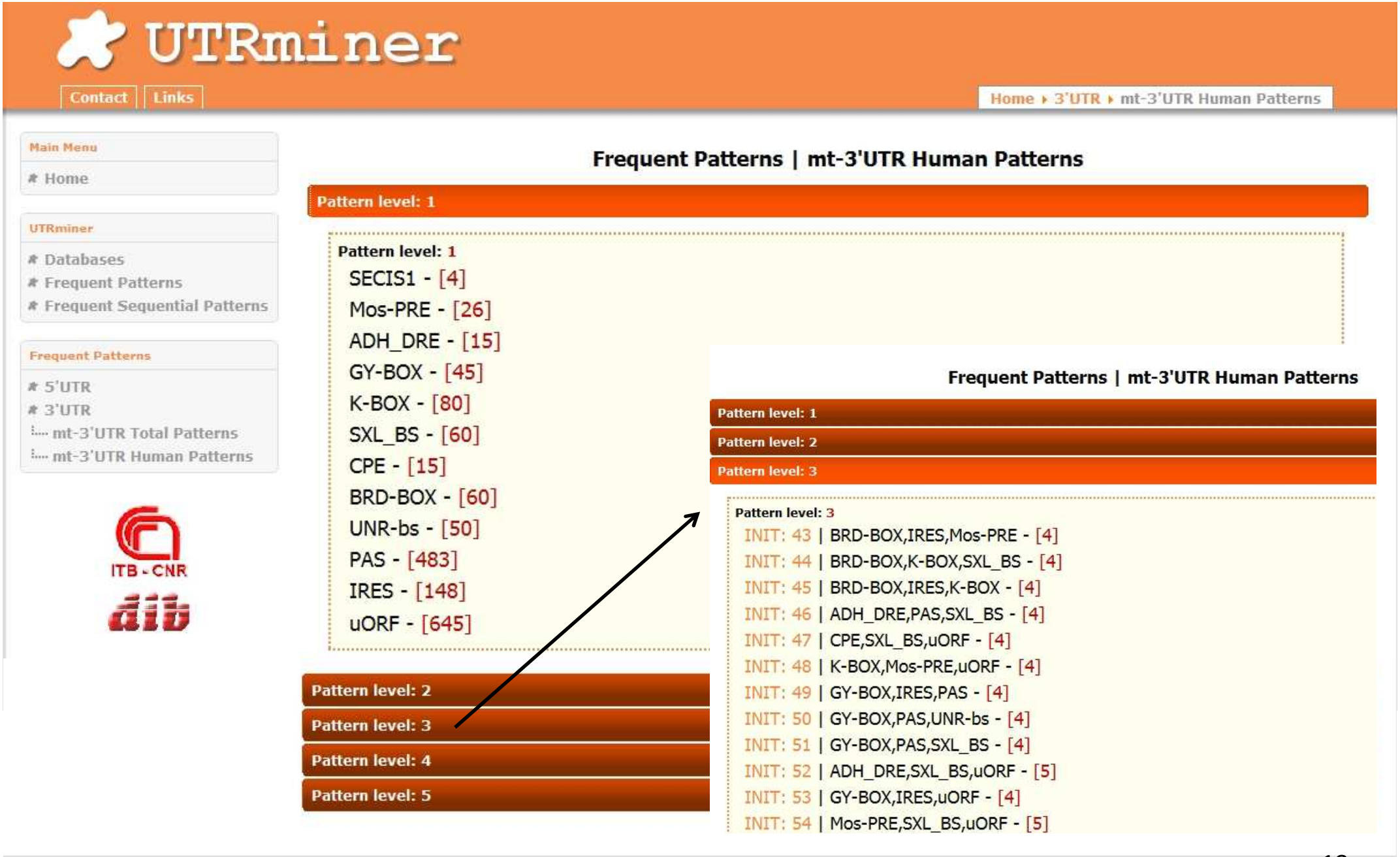
   - 8 up to three motifs in 5’UTR
   - 211 up to five motifs in 3’UTR

2. H-dataset (728 5’UTR and 728 3’UTR sequences from the human subset)

   FPs found:

   - 5 up to three motifs in 5’UTR
   - 135 up to five motifs in 3’UTR

First mining step results



First mining step results

UTRminer

Contact | Links

Home ▸ 3'UTR ▸ mt-3'UTR Human Patterns

Main Menu
- Home

UTRminer
- Databases
- Frequent Patterns
- Frequent Sequential Patterns

Frequent Patterns
- 5'UTR
- 3'UTR
  - mt-3'UTR Total Patterns
  - mt-3'UTR Human Patterns

Frequent Pattern | mt-3'UTR Human Patterns | INIT: 88

Pattern level: 3

INIT: 88 | IRES,PAS,uORF - Support items: 111

Chi-square: 6.622 | Significant (alpha=0.05)

COMPACT VIEW | NORMAL VIEW | FULL VIEW

POSITIONAL PATTERNS: COMPACT VIEW

- IRES uORF PAS (1)
- uORF IRES PAS (107)
- uORF IRES uORF PAS (3)

Pattern level: 3

INIT: 88 | IRES,PAS,uORF - Support items: 111

Chi-square: 6.622 | Significant (alpha=0.05)

COMPACT VIEW | NORMAL VIEW | FULL VIEW

POSITIONAL PATTERNS: NORMAL VIEW

- IRES uORF PAS (1)
- uORF IRES PAS (24)
- uORF IRES uORF PAS (1)
- uORF[2] IRES PAS (12)
- uORF[3] IRES PAS (10)
- uORF[4] IRES PAS (8)
- uORF[5] IRES PAS (5)

Pattern level: 3

INIT: 88 | IRES,PAS,uORF - Support items: 111

Chi-square: 6.622 | Significant (alpha=0.05)

COMPACT VIEW | NORMAL VIEW | FULL VIEW

POSITIONAL PATTERNS: FULL VIEW

- uORF (81:-52..28) 30 | uORF (111:57..167) 19 | uORF (216:185..400) 88 | uORF (135:487..621) 88 | uORF (351:708..1058) 142 | uORF (129:1199..1327) 12 | uORF (120:1338..1457) 2 | uORF (93:1458..1550) 166 | uORF (225:1715..1939) 24 | uORF (87:1962..2048) 33 | uORF (117:2080..2196) 66 | uORF (69:2261..2329) 18 | uORF (72:2346..2417) 6 | uORF (129:2422..2550) 9 | uORF (78:2558..2635) 30 | uORF (108:2664..2771) 171 | uORF (108:2941..3048) 38 | uORF (165:3085..3249) 46 | uORF (114:3294..3407) -5 | IRES (95:3401..3495) -35 | PAS (37:3459..3495) [CR144407]
- uORF (207:33..239) 11 | IRES (96:249..344) -20 | PAS (22:323..344) [CR655030]
- uORF (123:4..126) -99 | IRES (103:26..128) -22 | PAS (24:105..128) [CR036012]

ITB - CNR

Second mining step: experimentation

- 346 **FPs** analised
- varying **number of bins (intervals)** for spacer length discretization (6, 9 and 12 bins)
- varying **the threshold** (minsup=0.2 and 0.3) for **FSP** support
- **6 runs for each set of sequences containing a FP**

Second mining step: computational issue

- the second step of mining is quite **CPU intensive**
- it is not possible to solve the problem using typical department computational resources
- in order to solve this problem **we have used the EGEE Grid infrastructure**
- since the application was written in Java, **the Java Virtual Machine was required** on each WN
- the **running time** of a single job is **quite long** (it could be also > 60 hours)

Job Submission Tool

- In order to simplify the jobs submission we have used a tool developed by the LIBI project called Job Submission Tool (JST).
- JST allows to:
  - ✓ track the status of a large amount of jobs
  - ✓ automatic resubmit the failing jobs
  - ✓ collect the output files produced by all the jobs

*Poster: “A web interface to Job Submission Tool (JST)”

(Poster Board: 30)

Second mining step: issues

Using Java Virtual Machine:

- **many failures** due to missing or wrong Java configuration

  - we decided to **install** our Java Virtual Machine **on the fly** just before running each job

    ✓ this increased the job success rate by a factor 2 (more than 90%)

- **large memory required** by the application
- **many sites do not support long jobs**
- only a **small set of resources** were **usable**

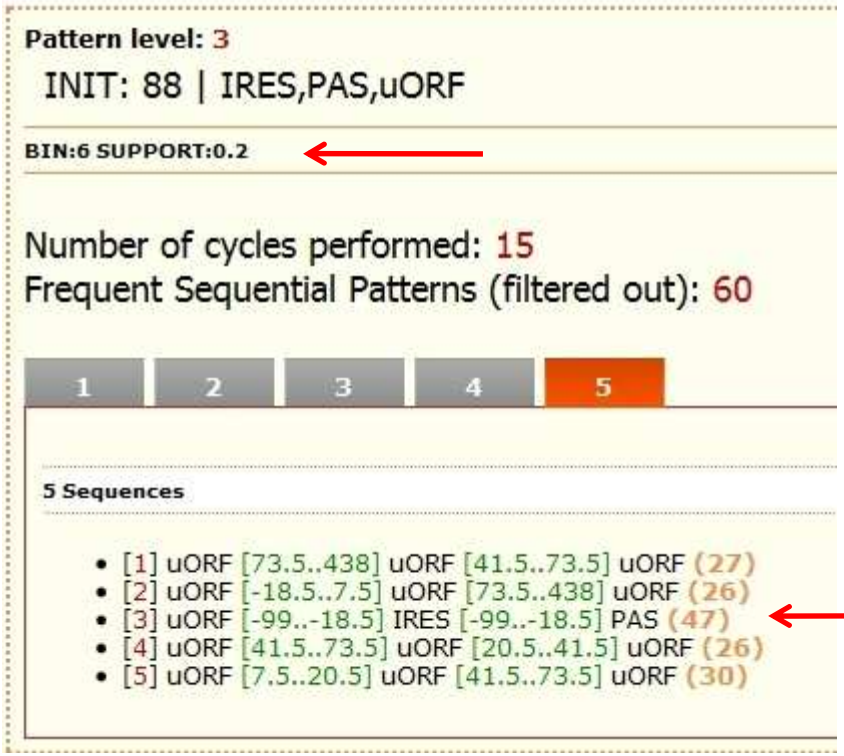
Automatic resubmission in case of failure was of great help!

Second mining step: results

- **Results** from the second mining step on FPs may **refine the prediction** by more than **two folds** compared to the first step of the mining process, where no evaluation about structural features of spacers is done.

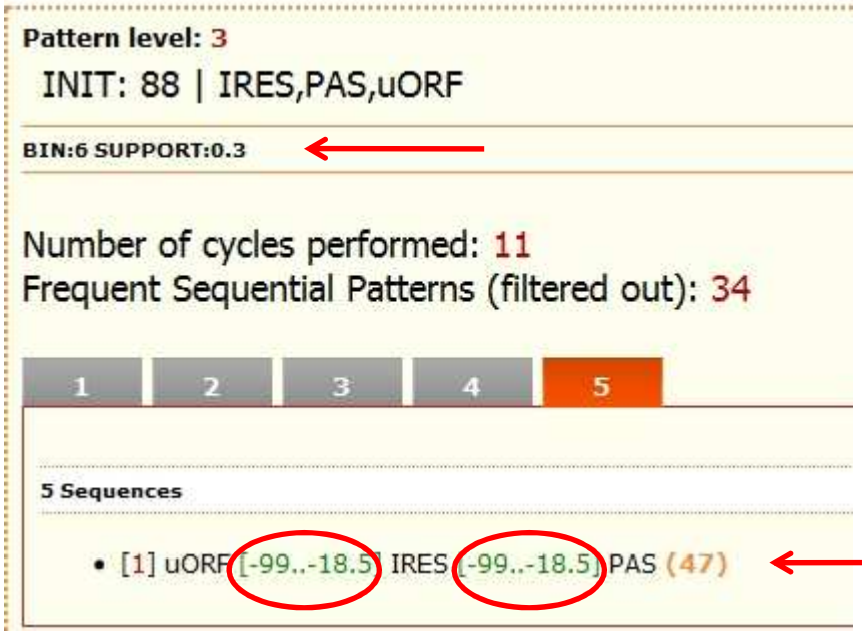
- **Grid execution numbers**:
  - ✓ jobs: 2076
  - ✓ time required on a single CPU: ~ 7 years
  - ✓ run time: about 10 days (EGEE Grid infrastructure)
  - ✓ speedup factor: 255

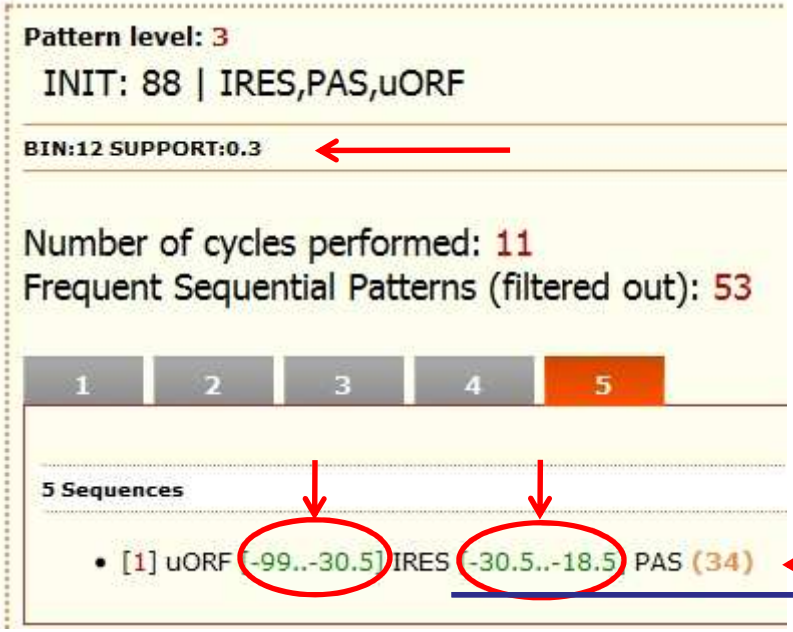
Second mining step: results on INIT 88



FP support = 111

bin 6, minsup 0.2 -> FSP support **= 47**
bin 6, minsup 0.3 -> FSP support **= 47**





bin 12, minsup 0.3 -> FSP support at **= 34**

Conclusions

- **Sequential pattern miners** allow to overcome some limits of other competitive pattern finder algorithms, i.e.:
  - need of a priori knowledge
  - pattern size
  - fixed width of spacers

  ..but are CPU expensive

- The **gLite infrastructure was a key element** for this preliminary study

- The **main problems** we faced were:
  - few resources for the execution of long job
  - non uniform Java environment
  - memory reservation on WNs

- By a **biological** point of view **preliminary results are promising** since they support experimental evidences reported in literature

- **Future developments** will include the improvement of the FSP mining process and the enlargement of analysis to whole transcript collections of different organisms for comparative analysis

Reference and links

UTRminer: http://utrminer.ba.itb.cnr.it/

Job Submission Tool: http://webcms.ba.infn.it/cms.software/index.html/index.php/Main/JobSubmissionTool

JST Web Page: http://webcms.ba.infn.it/~pierro/JST/

Acknowledgements

Co-workers

Antonio Turi Eliana Salvemini Corrado Loglisci Donato Malerba	Department of Computer Science, University of Bari, IT
Giorgio Grillo	Institute for Biomedical Technologies CNR, Bari, IT
Giacinto Donvito Giorgio Maggi	INFN, Bari, IT

Many thanks for your attention!

Bari, IT

Institute for Biomedical Technologies
CNR - Bari, IT

Department of Computer Science,
University of Bari, IT