

PAZAR XML DTD DOCUMENTATION

I – CONTACT INFORMATION

PAZAR is a software framework for the collection and maintenance of regulatory sequence annotations. Additional information on PAZAR may be obtained from <http://pazar.info>.

The current PAZAR team members are:

Wyeth W. Wasserman, PI
Jay R. Snoddy, PI
Stefan Kirov, Postdoctoral fellow
Elodie Portales-Casamar, Postdoctoral fellow
Jonathan Lim, Software developer

PAZAR is an open-source project hosted by <http://sourceforge.net>.

Any comment or criticism may be posted on the Open Discussion forum: https://sourceforge.net/forum/forum.php?forum_id=512784

Some help can also be asked for on the Help forum: https://sourceforge.net/forum/forum.php?forum_id=512785

II – BASIC CONCEPTS

This manual provides the detailed presentation of each element in the Document Type Definition (DTD) of the XML exchange format for the PAZAR database (version 1.0 – December 2005).

A regulatory sequence (reg_seq) can be of any size, from a single binding site to a several kb (for example sequences used in gene reporter assays). However, a reg_seq has to follow 3 constraints to be included in the database (see p.10 for the description of the 'reg_seq' element):

- It has to be linked to a gene through a Transcription Start Region (TSR) or to a marker. This is done through a hierarchical link in the XML schema.
- It has to have genomic coordinates and the coordinates used have to come from the latest assembly available in the Ensembl database (This strict limitation to genomes represented in Ensembl will hopefully be relaxed in the future).
- The sequence itself has to be stored.

A transcription factor is described in multiple layers:

- at the transcript level: a 'tf' element (see p.13) is a descendent of a 'transcript' element.
- at the protein level as a unit of a complex: a 'tf_unit' (see p.17) element calls a 'tf' element id.
- at the protein complex level: a 'funct_tf' element (see p.17) links all units of a functional transcription factor. The funct_tf is considered as the active transcription factor. Every transcription factor has to be described using the 3 layers even if it works as a monomere. The name of the transcription factor is actually stored in the 'funct_tf' element.

Internal IDs are used in this xml format in order to link elements together (pazar_id). Those IDs are not stored in the database. For each element, a format is suggested but any kind of ID can be used as long as it is unique.

III – PAZAR DTD CONTENT

1. Major document sections

The root node is a 'pazar' element (see below). It includes three child-elements:

- 'project' element: It consists of the user and project information (only one occurrence; see below).
- 'data' element: This section holds all the sequence and transcription factor information, as well as the analysis details (only one occurrence). These are the “parts” of the annotation and are embedded within a hierarchical structure (see description of the element on p.4 and special discussion 1 on p.7).
- 'analysis' element: This section is used to link the different 'data' elements together through specific analyses (as many occurrences as there are analyses to describe; see special discussion 2 on p.23).

2. Description of elements and attributes

ROOT NODE

Pazar Element

Comment

Straightforward.

Content Model

```
<!ELEMENT pazar (project, data, analysis*)>
```

PROJECT SECTION

Project Element

Comment

The 'project' element stores basic information to distinguish between different 'boutiques' within the database.

Content Model

```
<!ELEMENT project (user)>
```

Attributes

```
<!ATTLIST project
  pazar_id      ID          #REQUIRED
  project_name  CDATA       #REQUIRED
  edit_date     CDATA       #REQUIRED
  status        (restricted|published|open) #REQUIRED>
```

Notes

Attribute

pazar_id
status

Explanation

This ID has to be unique. Suggested format: p_0001, then auto-increment.
 'restricted' gives access to the project only to the user.
 'published' gives read access to everyone but write access only to the user .
 'open' gives read and write access to everyone.

Example

```
<project edit_date="13-12-05" pazar_id="p_0001" project_name="example_project" status="restricted">
<user .../>
</project>
```

User Element

Comment

The 'user' element stores basic information to identify the annotator.

Content Model

```
<!ELEMENT user EMPTY>
```

Attributes

```
<!ATTLIST user
  pazar_id      ID      #REQUIRED
  first_name    CDATA   #REQUIRED
  last_name     CDATA   #REQUIRED
  username      CDATA   #REQUIRED
  affiliation    CDATA   #IMPLIED>
```

Notes

Attribute

pazar_id

Explanation

This ID has to be unique. Suggested format: u_0001, then auto-increment

Example

```
<user affiliation="some_lab" first_name="first_name" last_name="last_name" pazar_id="u_0001"
username="lab_user"/>
```

DATA SECTION**A – ROOT ELEMENT****Data Element**

Comment

The 'data' element can hold any of its child-elements (description to follow), in any order and with zero to many occurrences.

Content Model

```
<!ELEMENT data
(homolog|gene_source|marker|construct|dataset|matrix|funct_tf|sample|cell|time|
condition|expression|interaction)*>
```

B – CHILD-ELEMENTS SHARED BY MULTIPLE ELEMENTS**Parameter Element**

Comment

The 'parameter' element allows the user to give an extra parameter to any other element. This provides flexibility to the system.

Content Model

```
<!ELEMENT parameter EMPTY>
```

Attributes

```
<!ATTLIST parameter
  tag      CDATA #REQUIRED
  value    CDATA #REQUIRED>
```

Example

```
<parameter tag="my_database_id" value="ID00123">
```

DB_source Element

Comment

The 'db_souce' element describes an external database.

Content Model

```
<!ELEMENT db_source (parameter*)>
```

Attributes

```
<!ATTLIST db_source
  db_name    CDATA #REQUIRED
  db_subset  CDATA #IMPLIED
  assembly   CDATA #REQUIRED>
```

Example

```
<db_source db_name="EnsEMBL" db_subset="homo_sapiens_36_35i" assembly="NCBI 35"/>
```

Coordinate Element**Comment**

The 'coordinate' element refers to genomic coordinates.

Content Model

```
<!ELEMENT coordinate (parameter*, location)>
```

Attributes

```
<!ATTLIST coordinate
  strand    CDATA #REQUIRED
  begin     CDATA #REQUIRED
  end       CDATA #REQUIRED
  length    CDATA #REQUIRED>
```

Example

```
<coordinate strand="+" begin="609283" end="609310" length="28">
<location .../>
</coordinate>
```

Location Element**Comment**

The 'location' element has to refer to a specific chromosome. As a consequence, the entries are limited to the species with sufficiently advanced annotation. Currently (December 2005), those species are, in the EnsEMBL database:

- Fugu rubripes
- Anopheles gambiae
- Xenopus tropicalis
- Homo sapiens
- Drosophila melanogaster
- Pan troglodytes
- Danio rerio
- Bos taurus
- Mus musculus
- Caenorhabditis elegans
- Gallus gallus
- Rattus norvegicus
- Apis mellifera
- Tetraodon nigroviridis

Content Model

```
<!ELEMENT location (parameter*, db_source)>
```

Attributes

```
<!ATTLIST location
  species   CDATA #REQUIRED
  chr       CDATA #REQUIRED
  band      CDATA #IMPLIED>
```

Notes

Attribute	Explanation
species	ex: Homo sapiens
chr	chromosome number
band	ex: p31.1

Example

```
<location species="Homo sapiens" chr="4" band="p16.3">
<db_source .../>
</location>
```

Method Element**Comment**

The 'method' element relates to the approach used to identify a characteristic (could be computational or laboratory driven).

Content Model

```
<!ELEMENT method (parameter*)>
```

Attributes

```
<!ATTLIST method
  method      CDATA #REQUIRED
  description  CDATA #IMPLIED>
```

Example

```
<method method="EMSA" description="in vitro gel shift assay"/>
```

Ref Element**Comment**

The 'ref' element stores a PubMed citation for an observation.

Content Model

```
<!ELEMENT ref (parameter*)>
```

Attributes

```
<!ATTLIST ref
  pmid          CDATA #REQUIRED>
```

Notes

Attribute	Explanation
pmid	pubmed id

Example

```
<ref pmid="11438531"/>
```

C – HIERARCHICAL LINKING OF ELEMENTS**##### Special Discussion 1 #####**

Implicit links are used that rest on a hierarchical structure:

```

  <homolog>
    <gene_source>
      <tsr>
        <reg_seq>
          <mutation_set>
            <mutation>
            </mutation>
          </mutation_set>
        </reg_seq>
      </tsr>
    <transcript>
      <tf>
      </tf>
    </transcript>
  </gene_source>
</homolog>
<marker>
  <reg_seq>
    <mutation_set>
      <mutation>
      </mutation>
    </mutation_set>
  </reg_seq>
</marker>
```

#####

Homolog Element

Comment

The 'homolog' element is for the user to group homologous genes together but it is not required. The 'gene_source' element can be at the root of this hierarchical tree. The 'db_source' child-element identifies the external database in which this homology relationship is reported.

Content Model

```
<!ELEMENT homolog (parameter*, db_source, gene_source+, conserved_el*)>
```

Attributes

```
<!ATTLIST homolog
  pazar_id          ID          #REQUIRED
  homology_type     (na|ortholog|paralog) #REQUIRED>
```

Notes

Attribute	Explanation
pazar_id	This ID has to be unique. Suggested format: ho_0001, then auto-increment

Example

```
<homolog pazar_id=" ho_0001" homology_type="ortholog">
<gene_source .../>
</homolog>
```

Gene_source Element

Comment

The 'gene_source' element provides the gene information. Currently, the supported gene ids are:

- Ensembl
 - Refseq
 - EntrezGene
 - Swiss prot
 - Genbank gene-specific sequences (usually cDNA)
- The id will be converted and stored as an Ensembl gene id in the database.

Content Model

```
<!ELEMENT gene_source (parameter*, db_source, tsr*, transcript*)>
```

Attributes

```
<!ATTLIST gene_source
  pazar_id          ID          #REQUIRED
  db_accn           CDATA      #REQUIRED
  description       CDATA      #IMPLIED>
```

Notes

Attribute	Explanation
pazar_id	This ID has to be unique. Suggested format: gs_0001, then auto-increment accession number from the external database described in the child-element 'db_source'.
db_accn	
description	full gene name

Example

```
<gene_source pazar_id="gs_0001" db_accn="ENSG00000129535" description="NRL">
<db_source .../>
</gene_source>
```

TSR Element

Comment

The 'tsr' (Transcription Start Region) element allows the annotator to define fuzzy start sites. The 'tsr' can be a unique start site though if fuzzy_start = fuzzy_end. A 'transcript' element can be included (hierarchical link) to assign an example transcript using this start site. This is based on the developers' view that few genes have a unique and absolute transcription start site.

Content Model

```
<!ELEMENT tsr (parameter*, transcript?, reg_seq+)>
```

Attributes

```
<!ATTLIST tsr
  pazar_id          ID          #REQUIRED
  fuzzy_start      CDATA      #REQUIRED
  fuzzy_end        CDATA      #REQUIRED
  predominant_start CDATA      #IMPLIED>
```

Notes

Attribute	Explanation
pazar_id	This ID has to be unique. Suggested format: tsr_0001, then auto-increment upstream most possible start in the tsr, genomic coordinate
fuzzy_start	
fuzzy_end	downstream most possible start in the tsr, genomic coordinate
predominant_start	most used tss within the tsr

Example

```
<tsr pazar_id="tsr_0001" fuzzy_start="609373" fuzzy_end="609383" predominant_start="609373">
<reg_seq .../>
</tsr>
```

Marker Element

Comment

If a 'reg_seq' element is not linked to one or more specific 'gene_source' (through 'tsr' elements), it has to be linked to a 'marker' element which can be any annotated feature with genomic coordinates and stable id. This allows users to work with sequences not yet linked to an annotated gene.

Content Model

```
<!ELEMENT marker (parameter*, db_source, reg_seq+)>
```

Attributes

```
<!ATTLIST marker
  pazar_id      ID          #REQUIRED
  db_accn       CDATA      #REQUIRED
  description    CDATA      #IMPLIED>
```

Notes

Attribute	Explanation
pazar_id	This ID has to be unique. Suggested format: ma_0001, then auto-increment marker accession number from the external database described in the child-element 'db_source'.
db_accn	
description	full marker name

Example

```
<marker pazar_id="ma_0001" db_accn="ENSG00000129535" description="NRL">
<db_source .../>
<reg_seq .../>
</marker>
```

Reg_Seq Element

Comment

The 'reg_seq' element is a key component of the database, the annotation of a regulatory sequence within a genome (e.g. a specific TFBS). See chapter II – BASIC CONCEPTS (p.1) for further explanations.

Content Model

```
<!ELEMENT reg_seq (parameter*, coordinate, mutation_set*)>
```

Attributes

```
<!ATTLIST reg_seq
  pazar_id      ID          #REQUIRED
  tfbs_name     CDATA      #IMPLIED
  sequence      CDATA      #REQUIRED
  quality       (na|conserved|tested|predicted) #REQUIRED>
```

Notes

Attribute	Explanation
pazar_id	This ID has to be unique. Suggested format: rs_0001, then auto-increment
tfbs_name	if the reg_seq is a binding site, give the name of the site
quality	conserved = orthologous to an experimentally verified reg_seq tested = experimentally verified predicted

Example

```
<reg_seq pazar_id="rs_0001" tfbs_name="NRE" sequence="ATTTGTAGGAGTGAGTCAGCTGACCCGC"
quality="tested">
<coordinate .../>
</reg_seq>
```

Mutation_Set Element

Comment

The 'mutation_set' element allows for the identification of mutations used in lab studies. This is a descendant of a 'reg_seq' element (the wild-type sequence); it can only be used in this context.

Content Model

```
<!ELEMENT mutation_set (parameter*, method, ref?, mutation+)>
```

Attributes

```
<!ATTLIST mutation_set
  pazar_id          ID          #REQUIRED
  mutant_name       CDATA      #REQUIRED
  mutated_seq       CDATA      #REQUIRED
  comments          CDATA      #IMPLIED>
```

Notes

Attribute	Explanation
pazar_id	This ID has to be unique. Suggested format: ms_0001, then auto-increment
mutant_name	Give the name/code for this particular mutant
mutated_seq	Paste here the sequence after the mutation(s) has(ve) been introduced. It should be the same sequence as in the ascendant 'reg_seq' element except for the mutations.
comment	Noteworthy features of the mutant

Example

```
<mutation_set pazar_id="ms_0001" mutant_name="somemutation"
mutated_seq="ATTTGTAGGAGTGTCTGAGCTGACCCGC" comments="4 central bases are mutated">
<method .../>
<mutation .../>
</mutation_set>
```

Mutation Element

Comment

The 'mutation' element describes one mutation from a 'mutation_set'. Thus, within a 'mutation_set' element, there should be as many 'mutation' child-elements as there are different mutations included in its sequence.

Content Model

```
<!ELEMENT mutation (parameter*)>
```

Attributes

```
<!ATTLIST mutation
  pazar_id ID #REQUIRED
  position CDATA #REQUIRED
  base CDATA #REQUIRED>
```

Notes

Attribute	Explanation
pazar_id	This ID has to be unique. Suggested format: mu_0001, then auto-increment
position	Position in the referring reg_seq. If the mutation is an insertion, the position should have the format 'pos before insertion'_'pos after insertion'
base	How is this position changed with respect to the original sequence. Create entries only for the altered positions. If the mutation is a deletion, put a whitespace here.

Example

```
<mutation pazar_id="mu_0001" position="14" base="t"/>
```

Transcript Element

Comment

The 'transcript' element provides the transcript information. Currently, the supported ids are:

- Ensembl
- Refseq
- Locus link
- Swiss prot
- Genbank cDNA

The id will be converted and stored as an Ensembl transcript id in the database.

Content Model

```
<!ELEMENT transcript (parameter*, db_source, tf?)>
```

Attributes

```
<!ATTLIST transcript
  pazar_id ID #REQUIRED
  db_accn CDATA #REQUIRED
  isoform CDATA #IMPLIED
  comments CDATA #IMPLIED>
```

Notes

Attribute	Explanation
pazar_id	This ID has to be unique. Suggested format: tr_0001, then auto-increment
db_accn	accession number from the external database described in the child-element 'db_source'.
isoform	if the transcript encodes a specific isoform of the gene, give the isoform name

Example

```
<transcript pazar_id="tr_0001" db_accn="ENST00000255622">
<db_source .../>
</transcript>
```

TF Element

Comment

The 'tf' (transcription factor) element is a descendant of a 'transcript' element (see the hierarchical structure drawn above). Each tf must be defined as a transcript prior to use. This is a complex piece of the system, that is called through its pazar_id in the 'tf_unit' element (see p.17), descendant of a 'funct_tf' element (see p.17). See chapter II – BASIC CONCEPTS (p.1) for further explanations.

Content Model

```
<!ELEMENT tf (parameter*)>
```

Attributes

```
<!ATTLIST tf
  pazar_id ID #REQUIRED
  class CDATA #IMPLIED
  family CDATA #IMPLIED>
```

Notes

Attribute	Explanation
pazar_id	This ID has to be unique. Suggested format: tf_0001, then auto-increment

Example

```
<tf pazar_id="tf_0001" class="bZIP" family="MAF"/>
```

D – OTHER ELEMENTS

Construct Element

Comment

The 'construct' element stores artificial sequences (any sequence, irregardless to presence in Ensembl). It can refer to one or more 'reg_seq' elements (through IDREFS) if the construct is a concatenation of genomic sequences.

Content Model

```
<!ELEMENT construct (parameter*)>
```

Attributes

```
<!ATTLIST construct
  pazar_id      ID          #REQUIRED
  construct_name CDATA      #REQUIRED
  description   CDATA      #REQUIRED
  sequence      CDATA      #REQUIRED
  reg_seq_ids   IDREFS     #IMPLIED>
```

Notes

Attribute	Explanation
pazar_id	This ID has to be unique. Suggested format: co_0001, then auto-increment
reg_seq_ids	The IDREFS type refers to 'reg_seq' elements (pazar_id), separated with whitespaces.

Example

```
<construct pazar_id="co_0001" construct_name="FN-13A" description="random oligo"
sequence="gggtgagtcagcg"/>
```

Matrix Element

Comment

The 'matrix' element describes a tf binding profile.

It can be linked to multiple 'reg_seq' and/or 'construct' elements used to build it, through IDREFS. **THUS, THE 'REG_SEQ' AND/OR 'CONSTRUCT' ELEMENTS HAVE TO BE DECLARED BEFORE THE MATRIX ELEMENT IN ORDER TO HAVE THEIR IDS ALREADY RECORDED.**

The 'db_source' child-element identifies the external database where this matrix is reported.

When the data is stored in the database there will be some loss of data due to a compressing algorithm we use, but it should be negligible. For details see Bio::Matrix::PSM::SiteMatrix::_compress_array.

Content Model

```
<!ELEMENT matrix (parameter*, db_source, matrix_info?)>
```

Attributes

```
<!ATTLIST matrix
  pazar_id      ID          #REQUIRED
  name         CDATA       #REQUIRED
  db_accn      CDATA       #REQUIRED
  vectora      CDATA       #REQUIRED
  vectorc      CDATA       #REQUIRED
  vectorg      CDATA       #REQUIRED
  vectort      CDATA       #REQUIRED
  sequence_ids IDREFS     #IMPLIED
  description  CDATA       #IMPLIED>
```

Notes

Attribute	Explanation
pazar_id	This ID has to be unique. Suggested format: mx_0001, then auto-increment
name	profile name
db_accn	matrix accession number from the external database described in the child-element 'db_source'.
vectora	position frequency matrix (raw counts) for the letter A at each position
vectorc	position frequency matrix (raw counts) for the letter C at each position
vectorg	position frequency matrix (raw counts) for the letter G at each position
vectort	position frequency matrix (raw counts) for the letter T at each position
sequence_ids	The IDREFS type refers to 'reg_seq' or 'construct' elements and should list at least 2 pazar_id from these elements, separated with withspaces.

Example

```
<matrix pazar_id="mx_0001" name="my_matrix" db_accn="nb_01" vectora="2 2 6 0 0 10 0 1 2 3 0 0
9" vectorc="4 2 0 0 1 0 9 1 4 2 0 10 0" vectorg="2 1 4 0 9 0 1 3 3 0 8 0 1" vectort="2 5 0 10
0 0 0 5 1 5 2 0 0" sequence_ids="co_0001 co_0002 co_0003 co_0004 co_0005 co_0006 co_0007
co_0008 co_0009 co_0010">
<db_source .../>
</matrix>
```

Matrix_info Element

Comment

The 'matrix_info' element is not required. It is based on the annotations available in the JASPAR database (specific to creating compatibility between PAZAR and JASPAR). You can use it as well if you have reasons not to go through 'reg_seq' or 'construct' to assemble the matrix. This is not encouraged however.

Content Model

```
<!ELEMENT matrix_info (parameter*)>
```

Attributes

```
<!ATTLIST matrix_info
  pazar_id ID          #REQUIRED
  species  CDATA       #IMPLIED
  pubmed  CDATA       #IMPLIED
  exptype CDATA       #IMPLIED>
```

Notes

Attribute	Explanation
pazar_id	This ID has to be unique. Suggested format: mi_0001, then auto-increment

Example

```
<matrix_info pazar_id="mi_0001" species="Homo sapiens" pubmed="8413232" exptype="SELEX"/>
```

Dataset Element

Comment

The 'dataset' element allows the user to group subsets of sequences ('reg_seq' or 'construct') within a project. For instance, if the project deals with tissue-specific regulatory sequences, a dataset can be muscle-specific sequences.

Content Model

```
<!ELEMENT dataset (parameter*)>
```

Attributes

```
<!ATTLIST dataset
  pazar_id      ID          #REQUIRED
  dataset_name  CDATA       #REQUIRED
  sequence_ids  IDREFS     #REQUIRED>>
```

Notes

Attribute	Explanation
pazar_id	This ID has to be unique. Suggested format: ds_0001, then auto-increment
sequence_ids	The IDREFS type refers to 'reg_seq' or 'construct' elements and should list at least 2 pazar_id from these elements, separated with whitespaces.

Example

```
<dataset pazar_id="ds_0001" dataset_name="muscle-specific" sequence_ids="rs_0001 rs_0023 rs_0073"/>
```

Conserved Element

Comment

The 'conserved' element allows the user to define the link between at least two conserved sequences.

Content Model

```
<!ELEMENT conserved_el (parameter*)>
```

Attributes

```
<!ATTLIST conserved_el
  pazar_id      ID          #REQUIRED
  reg_seq_ids   IDREFS     #REQUIRED>
```

Notes

Attribute	Explanation
pazar_id	This ID has to be unique. Suggested format: coe_0001, then auto-increment
reg_seq_ids	The IDREFS type refers to 'reg_seq' elements and should list at least 2 pazar_id from this element, separated with withespaces.

Example

```
<conserved_el pazar_id="coe_0001" reg_seq_ids="rs_0007 rs_0090"/>
```

Func_tf Element

Comment

The 'func_tf' element is a linking element REQUIRED to describe functional transcription factors including complexes. Even if the transcription factor is a monomere it has to be registered in this element in order to be used in the analysis section. See chapter II – BASIC CONCEPTS (p.1) for further explanations.

Content Model

```
<!ELEMENT func_tf (parameter*, tf_unit+, ref?)>
```

Attributes

```
<!ATTLIST func_tf
  pazar_id      ID          #REQUIRED
  func_tf_name  CDATA      #REQUIRED>
```

Notes

Attribute	Explanation
pazar_id	This ID has to be unique. Suggested format: fu_0001, then auto-increment
func_tf_name	name of the protein complexe (or monomere)

Example

```
<func_tf pazar_id="fu_0001" func_tf_name="NRL">
<tf_unit .../>
</func_tf>
```

Tf_unit Element

Comment

The 'tf_unit' element describes each protein which is part of the func_tf complex. See chapter II – BASIC CONCEPTS (p.1) for further explanations.

Content Model

```
<!ELEMENT tf_unit (parameter*)>
```

Attributes

```
<!ATTLIST funct_tf
  pazar_id      ID          #REQUIRED
  tf_id        IDREF       #REQUIRED
  modifications CDATA      #IMPLIED>
```

Notes

Attribute	Explanation
pazar_id	This ID has to be unique. Suggested format: tu_0001, then auto-increment
tf_id	The IDREF type refers to the pazar_id if one 'tf' element.
modifications	phosphorylations or other modifications needed for that TF to be functional

Example

```
<tf_unit pazar_id="tu_0001" tf_id="tf_0001" modifications="phosphorylation"/>
```

Sample Element

Comment

The 'sample' element describes the property of a biological sample (e.g. nuclear extract).

Content Model

```
<!ELEMENT sample (parameter*, cell, time)>
```

Attributes

```
<!ATTLIST sample
  pazar_id      ID          #REQUIRED
  sample_type   CDATA      #REQUIRED>
```

Notes

Attribute	Explanation
pazar_id	This ID has to be unique. Suggested format: sa_0001, then auto-increment

Example

```
<sample pazar_id="sa_0001" sample_type="nuclear extract">
<cell .../>
<time .../>
</sample>
```

Cell Element

Comment

The 'cell' element stores the annotation of types/lines of cells used in an experiment. This could change with future work on expression ontology.

Content Model

```
<!ELEMENT cell (parameter*)>
```

Attributes

```
<!ATTLIST cell
  pazar_id      ID          #REQUIRED
  name         CDATA       #IMPLIED
  tissue_ontology CDATA     #IMPLIED
  status       (na|primary|cell__line) #IMPLIED
  description  CDATA       #IMPLIED
  species      CDATA       #REQUIRED>
```

Notes

Attribute	Explanation
pazar_id	This ID has to be unique. Suggested format: ce_0001, then auto-increment

Example

```
<cell pazar_id="ce_0001" name="Y79" status="cell__line" description="retinoblastoma cell line"
species="Homo sapiens"/>
```

Time Element

Comment

The ‘time’ element stores temporal annotations (e.g. stages of development or experimental timepoints). This could change with future work on expression ontology.

Content Model

```
<!ELEMENT time (parameter*)>
```

Attributes

```
<!ATTLIST time
  pazar_id      ID          #REQUIRED
  name         CDATA       #IMPLIED
  description  CDATA       #IMPLIED
  range_start  CDATA       #IMPLIED
  range_end    CDATA       #IMPLIED
  scale        CDATA       #REQUIRED>
```

Notes

Attribute	Explanation
pazar_id	This ID has to be unique. Suggested format: ti_0001, then auto-increment

Example

```
<time pazar_id="ti_0001" range_start="E12" range_end="E15" description="from 12 to 15 days post-
coitum" scale="days of embryogenesis"/>
```

Condition Element

Comment

The 'condition' element stores the conditions used in biological studies. It can be the addition of any kind of molecule to an experiment. The condition element has to be referred to as an input in the input_output element. If the molecule is a transcription factor, the funct_tf element has to be also reported as an input of the experiment (see Example III in the Annexes p.28).

Content Model

```
<!ELEMENT condition (parameter*)>
```

Attributes

```
<!ATTLIST condition
  pazar_id      ID      #REQUIRED
  cond_type    CDATA  #REQUIRED
  molecule     CDATA  #REQUIRED
  description  CDATA  #IMPLIED
  concentration CDATA  #REQUIRED
  scale        CDATA  #REQUIRED>
```

Notes

Attribute	Explanation
pazar_id	This ID has to be unique. Suggested format: cd_0001, then auto-increment

Example

```
<condition pazar_id="cd_0001" cond_type="coexpression" molecule="transcription factor"
concentration="1:1" scale="ratio"/>
```

Expression Element

Comment

The 'expression' element stores the result/output of an experiment regarding a gene expression level.

Content Model

```
<!ELEMENT expression (parameter*)>
```

Attributes

```
<!ATTLIST expression
  pazar_id      ID      #REQUIRED
  qualitative   (highly__induced|induced|no__change|repressed|
                strongly__repressed|na)      #IMPLIED
  quantitative  CDATA  #IMPLIED
  scale        CDATA  #IMPLIED
  comments     CDATA  #IMPLIED>
```

Notes

Attribute	Explanation
pazar_id	This ID has to be unique. Suggested format: ex_0001, then auto-increment
quantitative	only a digit, no unit
scale	scale for the digit used in 'quantitative' (ex: percent, relative, absolute,...)

Example

```
<expression pazar_id="ex_0003" quantitative="420" scale="percent"/>
```

Interaction Element

Comment

The 'interaction' element stores the result/output of an experiment regarding the quality/quantity of an interaction.

Content Model

```
<!ELEMENT interaction (parameter*)>
```

Attributes

```
<!ATTLIST interaction
  pazar_id      ID              #REQUIRED
  qualitative   (saturation|good|marginal|poor|none|na) #IMPLIED
  quantitative  CDATA          #IMPLIED
  scale        CDATA          #IMPLIED
  comments     CDATA          #IMPLIED>
```

Notes

Attribute	Explanation
pazar_id	This ID has to be unique. Suggested format: in_0001, then auto-increment
quantitative	only a digit, no unit
scale	scale for the digit used in 'quantitative' (ex: percent, relative, absolute,...)

Example

```
<interaction pazar_id="in_0001" qualitative="good"/>
```

ANALYSIS SECTION**##### Special Discussion 2 #####**

This section holds all the experiment information linking sequences and transcription factors (inputs) to the result of the experiment (output – mostly interaction or expression).

Implicit links are used that rest on a hierarchical structure:

```

<analysis>
  <method>
  </method>
  <ref>
  </ref>
  <input_output>
    <input>
    </input>
    <output>
    </output>
  </input_output>
</analysis>

```

The IDREFS type in the 'input' and 'output' elements refer to any element described in the data section.

ALWAYS USE A 'FUNCT_TF' ELEMENT INSTEAD OF A 'TF' ELEMENT.

Examples (see more examples in the annexes):

1. A tf binding to a reg_seq:
input = funct_tf_id + reg_seq_id
output = interaction_id

2. A reg_seq able to induce a specific expression:
input = reg_seq_id
output = expression_id

#####

Analysis Element

Comment

The 'analysis' element delineates an experiment, the child nodes providing the specific details.

Content Model

```
<!ELEMENT analysis (parameter*, evidence, method, ref?, input_output+)>
```

Attributes

```

<!ATTLIST analysis
  name      CDATA #REQUIRED
  cell      IDREF #IMPLIED
  time      IDREF #IMPLIED
  comments  CDATA #IMPLIED>

```

Evidence Element

Comment

The 'evidence' element distinguishes between data of different quality (curated vs predicted).

Content Model

```
<!ELEMENT evidence (parameter*)>
```

Attributes

```
<!ATTLIST evidence
  type_evid      (curated|ADMC|prediction)          #REQUIRED
  status_evid    (approved|provisional|archivable|removable) #REQUIRED>
```

Notes

Attribute	Explanation
type_evid	curated = manual annotation ADMC = Natural Language processing (not yet available) prediction
status_evid	approved = curated by two independent sources provisional = initial state archivable = conflicting data removable = bad data

Example

```
<evidence status_evid="provisional" type_evid="curated"/>
```

Input-Output Element

Comment

The 'input_output' element groups one or more inputs (reg_seq, funct_tf, construct,...) to one or more outputs (interaction, expression,...).

Content Model

```
<!ELEMENT input_output (input, output, parameter*)>
```

Input Element

Comment

List of inputs.

Content Model

```
<!ELEMENT input EMPTY>
```

Attributes

```
<!ATTLIST input
  inputs      IDREFS      #REQUIRED>
```

Output Element**Comment**

List of outputs.

Content Model

```
<!ELEMENT output EMPTY>
```

Attributes

```
<!ATTLIST output
  outputs      IDREFS      #REQUIRED>
```

ANNEXES : XML FILE EXAMPLES

I – EXEMPLE 1

Comment

This example describes an interaction between a transcription factor and a binding site located upstream a gene, and a set of mutations affecting this interaction.

XML format

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE pazar SYSTEM "http://www.pazar.info/pazar.dtd">
<pazar>
  <project edit_date="13-12-05" pazar_id="p_0001"
    project_name="example_project" status="restricted">
    <user affiliation="CMMT" first_name="first_name"
      last_name="last_name" pazar_id="u_0001" username="cmmt_user"/>
  </project>
  <data>
    <gene_source db_accn="ENSG00000133256" description="PDE6B"
pazar_id="gs_0001">
      <db_source db_name="Ensembl" assembly="NCBI 35"/>
      <tsr fuzzy_end="609373" fuzzy_start="609373" pazar_id="tsr_0001">
        <transcript db_accn="ENST00000255622" pazar_id="tr_0001">
          <db_source db_name="Ensembl" assembly="NCBI 35"/>
        </transcript>
        <reg_seq tfbs_name="NRE" pazar_id="rs_0001" quality="tested"
sequence="ATTTGTAGGAGTGAGTCAGCTGACCCGC">
          <coordinate begin="609283" end="609310" length="28" strand="+">
            <location band="p16.3" chr="4" species="Homo sapiens">
              <db_source db_name="Ensembl" assembly="NCBI 35"/>
            </location>
          </coordinate>
          <mutation_set comments="none" mutant_name="sometmutation"
mutated_seq="ATTTGTAGGAGGTCTGAAGCTGACCCGC" pazar_id="ms_0001">
            <method method="EMSA"/>
            <mutation base="g" pazar_id="mu_0001" position="12"/>
            <mutation base="t" pazar_id="mu_0002" position="13"/>
            <mutation base="c" pazar_id="mu_0003" position="14"/>
            <mutation base="t" pazar_id="mu_0004" position="15"/>
            <mutation base="g" pazar_id="mu_0005" position="16"/>
            <mutation base="a" pazar_id="mu_0006" position="17"/>
          </mutation_set>
        </reg_seq>
      </tsr>
    </gene_source>
    <gene_source db_accn="ENSG00000129535" description="NRL"
pazar_id="gs_0002">
      <db_source db_name="Ensembl" assembly="NCBI 35"/>
      <transcript db_accn="ENST00000250471" pazar_id="tr_0002">
        <db_source db_name="Ensembl" assembly="NCBI 35"/>
        <tf class="bZIP" family="MAF" pazar_id="tf_0001"/>
      </transcript>
    </gene_source>
  </data>
</pazar>
```

```

    </transcript>
  </gene_source>
  <funct_tf funct_tf_name="NRL" pazar_id="fu_0001">
    <tf_unit pazar_id="tu_0001" tf_id="tf_0001"/>
  </funct_tf>
  <interaction pazar_id="in_0001" qualitative="good"/>
  <interaction pazar_id="in_0002" qualitative="none"/>
  <cell name="Y79" pazar_id="ce_0001" species="Homo sapiens"
status="cell__line"/>
</data>
<analysis name="analysis_example1" cell="ce_0001">
  <evidence status_evid="provisional" type_evid="curated"/>
  <method method="EMSA"/>
  <ref pmid="11438531"/>
  <input_output>
    <input inputs="fu_0001 rs_0001"/>
    <output outputs="in_0001"/>
  </input_output>
  <input_output>
    <input inputs="fu_0001 ms_0001"/>
    <output outputs="in_0002"/>
  </input_output>
</analysis>
</pazar>

```

II – EXEMPLE 2

Comment

This example describes a SELEX experiment with an heterodimer transcription factor and the matrix built from the sequences.

XML format

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE pazar SYSTEM "http://www.pazar.info/pazar.dtd">
<pazar>
  <project edit_date="06-12-05" pazar_id="p_0002"
    project_name="example_project2" status="restricted">
    <user affiliation="CMMT" first_name="first_name"
      last_name="last_name" pazar_id="u_0001" username="cmmt_user"/>
  </project>
  <data>
    <construct construct_name="FN-13A" description="random oligo"
sequence="gggtgagtcagcg" pazar_id="co_0001"/>
    <construct construct_name="JN-1B" description="random oligo"
sequence="taatgacacatca" pazar_id="co_0002"/>
    <construct construct_name="FN-1B" description="random oligo"
sequence="ccgtgactcagca" pazar_id="co_0003"/>
    <construct construct_name="FN-5B" description="random oligo"
sequence="ttatgacgccgca" pazar_id="co_0004"/>
    <construct construct_name="FM-1B" description="random oligo"
sequence="atgtgactgtgca" pazar_id="co_0005"/>
    <construct construct_name="JN-3B" description="random oligo"
sequence="atgtgacgtttca" pazar_id="co_0006"/>
  </data>
</pazar>

```

```

    <construct construct_name="JN-4B" description="random oligo"
sequence="caatgactgtgca" pazar_id="co_0007"/>
    <construct construct_name="FN-6B" description="random oligo"
sequence="gcatcacggtgca" pazar_id="co_0008"/>
    <construct construct_name="FN-3B" description="random oligo"
sequence="ctatgactacgca" pazar_id="co_0009"/>
    <construct construct_name="FN-2B" description="random oligo"
sequence="ctatgaccatgca" pazar_id="co_0010"/>
    <matrix name="my_matrix" db_accn="nb_01" vectora="2 2 6 0 0 10 0 1 2
3 0 0 9" vectorc="4 2 0 0 1 0 9 1 4 2 0 10 0" vectorg="2 1 4 0
9 0 1 3 3 0 8 0 1" vectort="2 5 0 10 0 0 0 5 1 5 2 0 0"
sequence_ids="co_0001 co_0002 co_0003 co_0004 co_0005 co_0006 co_0007 co_0008
co_0009 co_0010" pazar_id="mx_0001">
        <db_source db_name="my_database" assembly="na"/>
    </matrix>
    <gene_source db_accn="ENSG00000129535" description="NRL"
pazar_id="gs_0001">
        <db_source db_name="Ensembl" assembly="NCBI 35"/>
        <transcript db_accn="ENST00000250471" pazar_id="tr_0001">
            <db_source db_name="Ensembl" assembly="NCBI 35"/>
            <tf class="bZIP" family="MAF" pazar_id="tf_0001"/>
        </transcript>
    </gene_source>
    <gene_source db_accn="ENSG00000170345" description="FOS"
pazar_id="gs_0002">
        <db_source db_name="Ensembl" assembly="NCBI 35"/>
        <transcript db_accn="ENST00000303562" pazar_id="tr_0002">
            <db_source db_name="Ensembl" assembly="NCBI 35"/>
            <tf class="bZIP" family="FOS" pazar_id="tf_0002"/>
        </transcript>
    </gene_source>
    <funct_tf funct_tf_name="NRL-FOS heterodimer" pazar_id="fu_0001">
        <tf_unit pazar_id="tu_0001" tf_id="tf_0001"
modifications="phosphorylation"/>
        <tf_unit pazar_id="tu_0002" tf_id="tf_0002"/>
    </funct_tf>
    <interaction pazar_id="in_0001" quantitative="16" scale="percent"/>
    <interaction pazar_id="in_0002" quantitative="11" scale="percent"/>
    <interaction pazar_id="in_0003" quantitative="14" scale="percent"/>
    <interaction pazar_id="in_0004" quantitative="24" scale="percent"/>
    <interaction pazar_id="in_0005" quantitative="21" scale="percent"/>
    <interaction pazar_id="in_0006" quantitative="19" scale="percent"/>
    <interaction pazar_id="in_0007" quantitative="15" scale="percent"/>
</data>
<analysis name="analysis_example2">
    <evidence status_evid="provisional" type_evid="curated"/>
    <method method="SELEX"/>
    <ref pmid="7936637"/>
    <input_output>
        <input inputs="fu_0001 co_0001"/>
        <output outputs="in_0001"/>
    </input_output>
    <input_output>
        <input inputs="fu_0001 co_0002"/>
        <output outputs="in_0002"/>
    </input_output>
    <input_output>

```

```

    <input inputs="fu_0001 co_0003"/>
    <output outputs="in_0003"/>
</input_output>
<input_output>
    <input inputs="fu_0001 co_0004"/>
    <output outputs="in_0004"/>
</input_output>
<input_output>
    <input inputs="fu_0001 co_0005"/>
    <output outputs="in_0005"/>
</input_output>
<input_output>
    <input inputs="fu_0001 co_0006"/>
    <output outputs="in_0006"/>
</input_output>
<input_output>
    <input inputs="fu_0001 co_0007"/>
    <output outputs="in_0006"/>
</input_output>
<input_output>
    <input inputs="fu_0001 co_0008"/>
    <output outputs="in_0001"/>
</input_output>
<input_output>
    <input inputs="fu_0001 co_0009"/>
    <output outputs="in_0007"/>
</input_output>
<input_output>
    <input inputs="fu_0001 co_0010"/>
    <output outputs="in_0003"/>
</input_output>
</analysis>
</pazar>

```

III – EXEMPLE 3

Comment

This example describes a gene reporter assay and the influence of co-expression with a transcription factor.

XML format

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE pazar SYSTEM "http://www.pazar.info/pazar.dtd">
<pazar>
  <project edit_date="13-12-05" pazar_id="p_0001"
    project_name="example_project3" status="restricted">
    <user affiliation="CMMT" first_name="first_name"
      last_name="last_name" pazar_id="u_0001" username="cmmt_user"/>
  </project>
  <data>
    <gene_source db_accn="ENSG00000133256" description="PDE6B"
      pazar_id="gs_0001">
      <db_source db_name="Ensembl" assembly="NCBI 35"/>
    </data>
  </pazar>

```

```

<tsr fuzzy_end="609373" fuzzy_start="609373" pazar_id="tsr_0001">
  <transcript db_accn="ENST00000255622" pazar_id="tr_0001">
    <db_source db_name="Ensembl" assembly="NCBI 35"/>
  </transcript>
  <reg_seq pazar_id="rs_0001" quality="tested"
sequence="gagtgagtcagctgacccgccccggggttcctaatactcactaagaaagactttgctgatgacaggggtt
tcctgggagtcctatgcgtgcctggagcagcagcgtctccagggacaggcagccacc">
    <coordinate begin="609291" end="609415" length="125" strand="+">
      <location band="p16.3" chr="4" species="Homo sapiens">
        <db_source db_name="Ensembl" assembly="NCBI 35"/>
      </location>
    </coordinate>
  </reg_seq>
  <reg_seq pazar_id="rs_0002" quality="tested"
sequence="gttcctaatactcactaagaaagactttgctgatgacaggggttcctgggagtcctatgcgtgcctggagc
agcagcgtctccagggacaggcagccacc">
    <coordinate begin="609318" end="609415" length="98" strand="+">
      <location band="p16.3" chr="4" species="Homo sapiens">
        <db_source db_name="Ensembl" assembly="NCBI 35"/>
      </location>
    </coordinate>
  </reg_seq>
</tsr>
</gene_source>
<gene_source db_accn="ENSG00000105392" description="CRX"
pazar_id="gs_0002">
  <db_source db_name="Ensembl" assembly="NCBI 35"/>
  <transcript db_accn="ENST00000221996" pazar_id="tr_0002">
    <db_source db_name="Ensembl" assembly="NCBI 35"/>
    <tf class="Homeobox" pazar_id="tf_0001"/>
  </transcript>
</gene_source>
<funct_tf funct_tf_name="CRX" pazar_id="fu_0001">
  <tf_unit pazar_id="tu_0001" tf_id="tf_0001"/>
</funct_tf>
<expression pazar_id="ex_0001" quantitative="100" scale="percent"/>
<expression pazar_id="ex_0002" quantitative="23" scale="percent"/>
<expression pazar_id="ex_0003" quantitative="420" scale="percent"/>
<expression pazar_id="ex_0004" quantitative="50" scale="percent"/>
<cell name="Y79" pazar_id="ce_0001" species="Homo sapiens"
status="cell_line"/>
  <cell name="embryonic head" tissue_ontology="head" description="ex vivo
dissected xenopus embryonic heads" pazar_id="ce_0002" species="Xenopus
tropicalis" status="primary"/>
  <time range_start="24" range_end="28" pazar_id="ti_0001" scale="stages of
embryogenesis"/>
  <condition pazar_id="cd_0001" cond_type="coexpression"
molecule="transcription factor" concentration="1:1" scale="ratio"/>
</data>
<analysis name="analysis_example3_1" cell="ce_0001">
  <evidence status_evid="provisional" type_evid="curated"/>
  <method method="luciferase gene reporter assay"/>
  <ref pmid="11943774"/>
  <input_output>
    <input inputs="rs_0001"/>
    <output outputs="ex_0001"/>
  </input_output>

```

```

<input_output>
  <input inputs="rs_0002"/>
  <output outputs="ex_0002"/>
</input_output>
<input_output>
  <input inputs="rs_0001 fu_0001 cd_0001"/>
  <output outputs="ex_0003"/>
</input_output>
<input_output>
  <input inputs="rs_0002 fu_0001 cd_0001"/>
  <output outputs="ex_0004"/>
</input_output>
</analysis>
<analysis name="analysis_example3_2" cell="ce_0002" time="ti_0001">
  <evidence status_evid="provisional" type_evid="curated"/>
  <method method="luciferase gene reporter assay"/>
  <ref pmid="11943774"/>
  <input_output>
    <input inputs="rs_0001"/>
    <output outputs="ex_0001"/>
  </input_output>
  <input_output>
    <input inputs="rs_0002"/>
    <output outputs="ex_0002"/>
  </input_output>
</analysis>
</pazar>

```