

Image Signatures

<http://cvs.openmicroscopy.org.uk/tiki/tiki-index.php?page=Computing+Image+Signatures+in+OME>

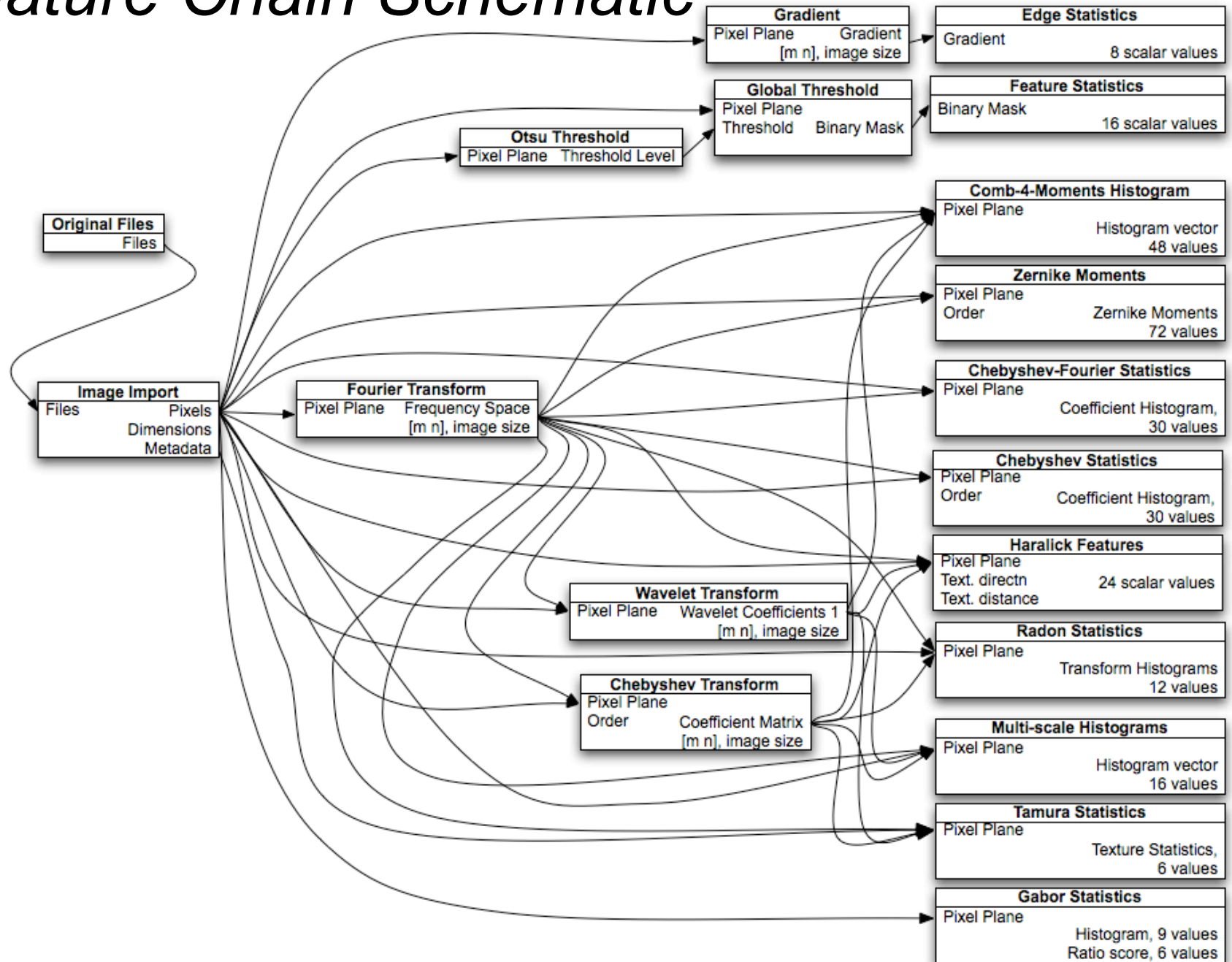
Image Signatures are numerical descriptors of image content

- They describe image properties such as
 - Edges, Shapes, Textures, Regions
- We use AI machines to interpret the image signatures to judge image content
- Image Signatures are the basis for our lab's research on Automated Image classification and scoring of High-Throughput High-Content Image screens

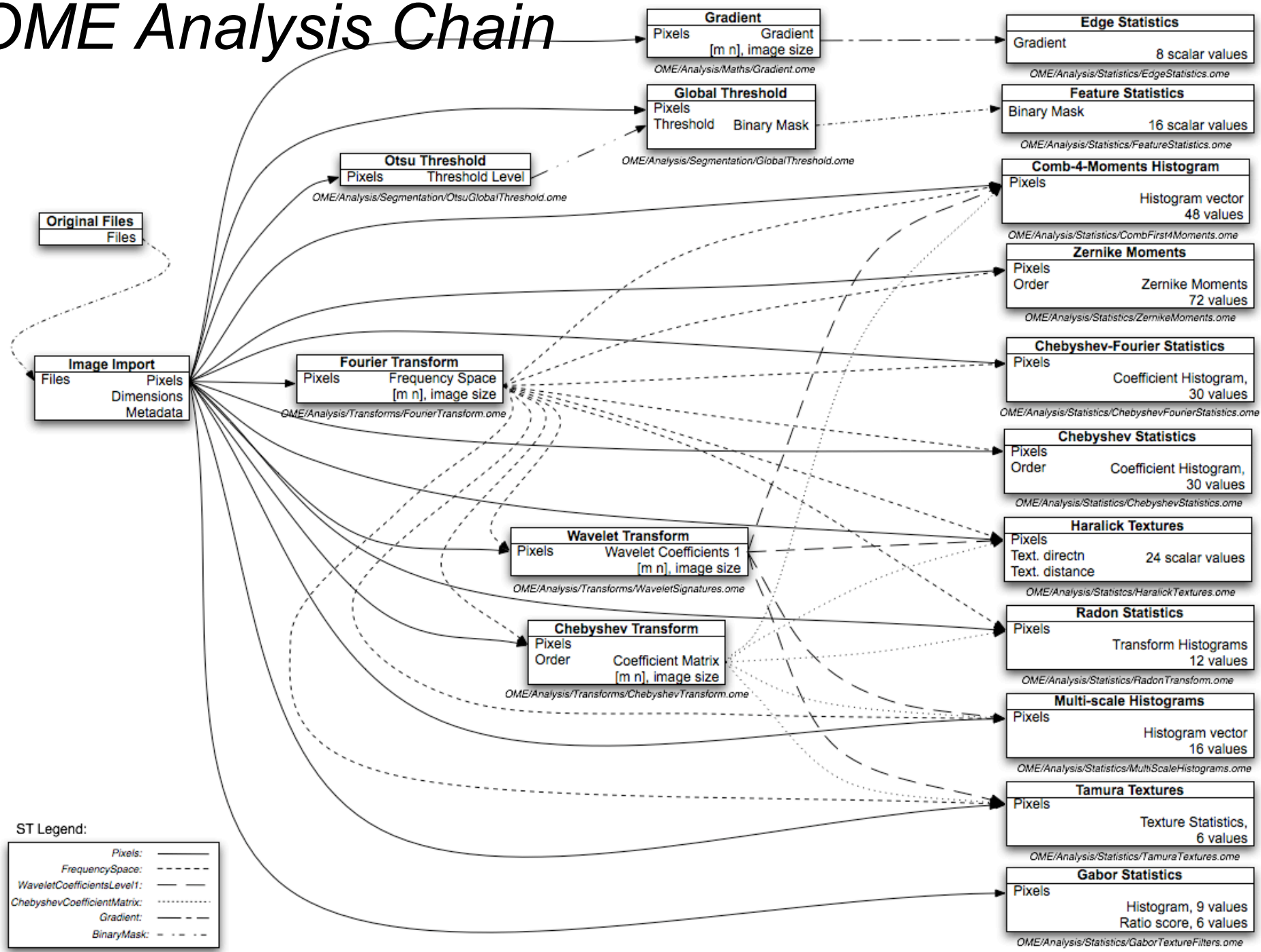
Computing Image Signatures

- There already exist state-of-the-art Computer Vision algorithms for measuring various image properties:
 - Edges, Shapes, Textures, Regions
- We've collected MATLAB implementations of a dozen such algorithms
- *Use-Case:* run our algorithms on hundreds of images to feed our AI machines.
 - Lots of Images per Collection (hundred)
 - Lots of Image Collections (dozen)
 - Many algorithms, computationally intensive, and arranged in a complex workflow
- *Goal:* We need the signatures to be “easily” and “quickly” (tools generate results quickly)
 - Easily: It is easy to teach biologists how to compute signatures
 - Quickly: The tools are computationally efficient
 - it goes without saying: Validation!
- How are we accomplishing this? Using OME Technology of course!

Signature Chain Schematic

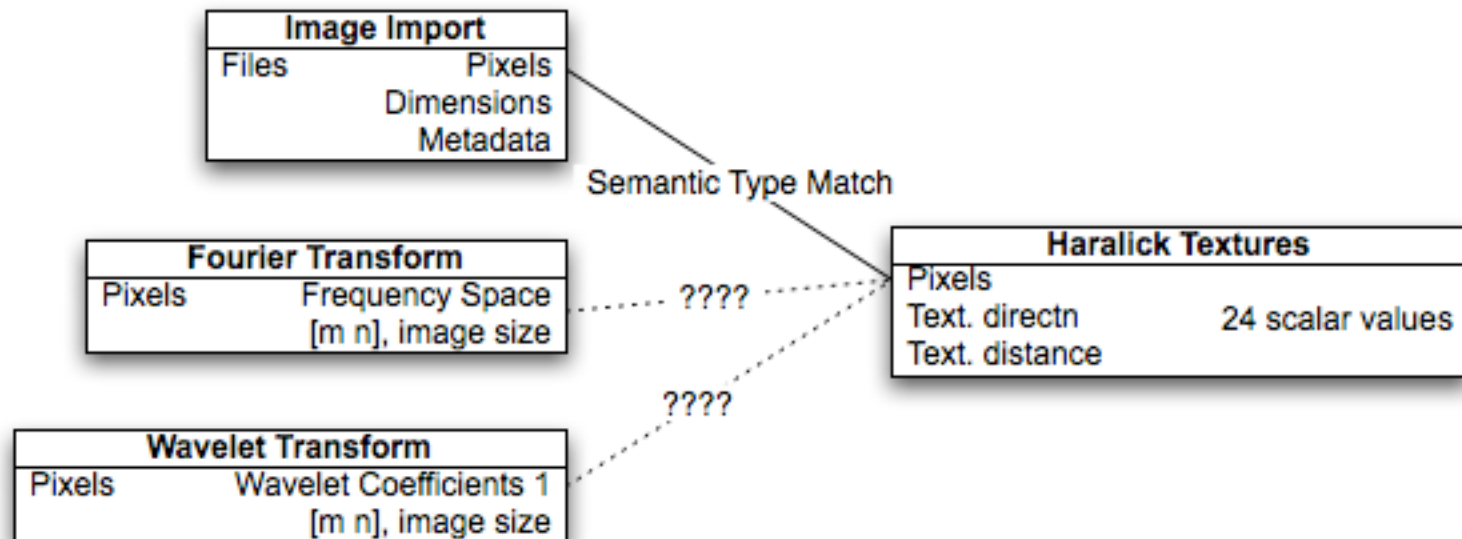


OME Analysis Chain



Data Modeling

- Our use-case requires both strong semantic typing and generality; e.g.:



- Outputs from *Fourier Transform* and *Wavelet Transform* must be stored in OMEIS, and should be treated (sometimes) as Pixels. However these outputs must not be modeled as Pixels ST.
- This has motivated our home-brewed method of inheritance: *Type-Casters* and *Parental Types*

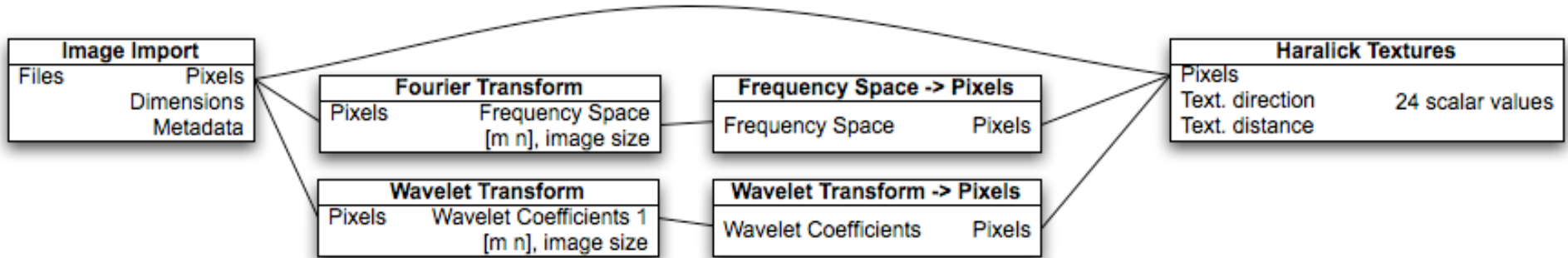
Parental Types

- *Fourier Transform* and *Wavelet Transform* ST definitions:

```
<SemanticType Name="FrequencySpace" AppliesTo="F"> <SemanticType Name="WaveletCoefficientsLevel1" AppliesTo="F">
  <Element                                     <Element
    Name="Parent"                               Name="Parent"
    DBLocation="FOURIER.MAG_AND_PHASE"          DBLocation="WAVELET_COEFFICIENTS1.PARENT"
    DataType = "reference"                     DataType = "reference"
    RefersTo = "Pixels" />                   RefersTo = "Pixels" />
</SemanticType>                               </SemanticType>
```

- Parental Types are a convention supported by the MATLAB Handler and Typecaster Modules
(src/perl2/OME/Analysis/Modules/PPM)

Typecaster Modules



- Typecaster Modules are defined in XML by specifying Formal Inputs and Outputs. There's a general Perl package that serves as a general Typecaster module implementation.

`src/perl2/OME/Analysis/Modules/PPM/Typecast2Pixels.pm`

ROIs

- Although image signatures are typically used to describe the whole image's content, it is equally valid to compute signatures just on an image region.
- Our lab's use-cases require computing signatures per-channel and per image tile. Hence we have developed modules (*ROI constructors*) in Perl for programmatically fragmenting the images. The signature chain is data-modeled to compute signatures per ROI.
- <tiling figure>
- Let us agree on a common ROI data-model that is usable by manual ROI tools (e.g. Java ROI UI tool), automated ROI modules (e.g. ROI Constructors), and OME Analysis Chains that need to operate on them (e.g. Signature Analysis Chain).

ROI Data model

- Whole image or ROI?
- Our ROI constructors take an image's Pixels and generate Feature Granularity attributes of *ST DerivedPixels*.
- We provide three ROI Constructor modules (Whole Image ROI, 2D Image ROI, 2D Tiled Image ROI). These modules are defined in Perl (`src/perl2/OME/Analysis/Modules/ROI`) and programmatically generate ROIs in various ways.
- ROI constructor modules generate Features called e.g. *Whole Image ROI*, *2D Image Tile 0 0*, *2D Image Tile 0 1*, *2D Image Tile 1 1* etc .
- DerivedPixels (F) ST has Parent Pixels reference & extents.

`src/xml/OME/Analysis/ROI/DerivedPixels.ome`

Discuss Signature Data-Model

MATLAB + OME

more info:

Macura T. J. et.al. The Open Microscopy Environment MATLAB Handler: Combining a Bioinformatics Data & Image Repository with a Quantitative Analysis Environment. *Rejected IEEE ISBI 2006.*

http://users.openmicroscopy.org.uk/~tmacur1/Macura_OME_MatlabHandler.pdf

contact person:

Tom Macura tmacura@nih.gov

http://users.openmicroscopy.org.uk/~tmacur1/tomasz_goes_to_washington.jpg

MATLAB

- MATLAB is a high-level, interactive programming environment for numerical computation. With more than 1000 built-in mathematical, statistical, and engineering functions, including statistical, image analysis, and machine learning tools, MATLAB is especially useful for quantitative bioinformatics. Our lab uses MATLAB as:
 - an interactive environment for data analysis and visualization leading to algorithm development
 - a programming language that allows us to implement numerical analysis algorithms much faster than with traditional programming languages such as C, C++, or Fortran.

OME/MATLAB coupling

We designed the OME/MATLAB coupling to support both MATLAB usage methods:

- Developing algorithms in MATLAB using OME data (omeis-http)
<http://www.openmicroscopy.org.uk/api/omeis/matlab-mex.html>
- Using MATLAB for parallel execution of high-throughput workflows defined in OME (MATLAB Handler)
<http://www.openmicroscopy.org.uk/api/xml/AML/matlab.html>
- Recipe for using OME to compute image signatures
<http://cvs.openmicroscopy.org.uk/tiki/tiki-index.php?page=Computing+Image+Signatures+in+OME>

OME/MATLAB Install

Optional OME-MATLAB Setup

(All verbose information logged in /var/tmp/OME/install/MatlabTask.log)

Testing MATLAB configuration

_ Checking if path to MATLAB .m files is stale [**FAILURE**].

Path was stale. Reset /home/tmacurl/OME-Hacking/OME/src/matlab to /home/tmacurl/Fresh/OME/src/matlab

_ Checking if user apache is licensed to run /usr/local/bin/matlab [**SUCCESS**].

MATLAB Perl API configuration:

Install MATLAB Perl API?: **yes**

MATLAB User: **apache**

MATLAB Path: **/usr/bin/MATLAB7SP2**

MATLAB Exec: **/usr/local/bin/matlab**

MATLAB Exec Flags: **-nodisplay -nojvm**

Config MATLAB for dev?: **yes**

MATLAB .m files Path: **/home/tmacurl/Fresh/OME/src/matlab**

Are these values correct ? [y/n]: **y**

_ Double checking if user apache is licensed to run /usr/local/bin/matlab [**SUCCESS**].

_ Gathering information about your MATLAB installation

Matlab Vers: **7.0.4.352**

Matlab Arch: **glnx86**

Matlab Root: **/usr/bin/MATLAB7SP2**

Include: **-I/usr/bin/MATLAB7SP2/extern/include**

Lib: **-L/usr/bin/MATLAB7SP2/bin/glnx86 -lmx -leng -lut -lmat -licudata -licui18n -licuuc -lustdio -lz**

Installing MATLAB Perl API

_ Configuring [**SUCCESS**].

_ Compiling [**SUCCESS**].

_ Testing [**SKIPPING**].

_ Installing [**SUCCESS**].

Installing MEX Interface to omeis-http

_ Configuring omeis-http with MATLAB bindings [**SUCCESS**].

_ Compiling omeis-http with MATLAB bindings [**SUCCESS**].

Installing OME MATLAB .m files

_ Compiling MEX files [**SUCCESS**].

_ Installing [**SUCCESS**].

<http://www.openmicroscopy.org.uk/install/matlab-connector.html>