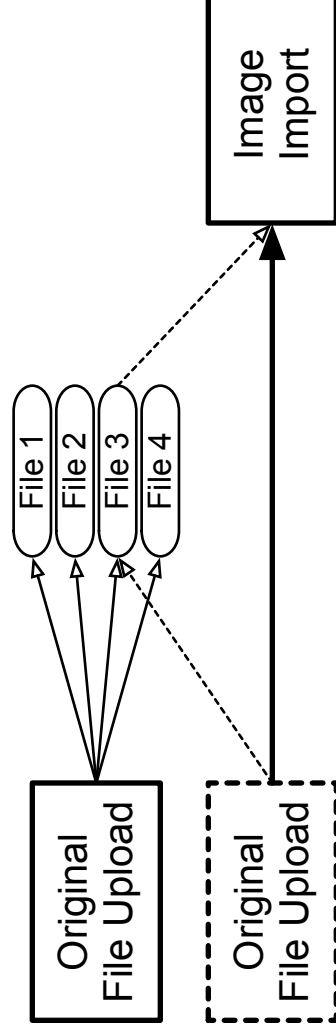


Analysis Engine Oddities

Virtual MEX

- Allows a MEX to refer to an output actually produced by something else.
- Used for sub-selecting a result for use as input into a module



Universal Node Execution

- A Node execution that is shared by every chain execution needing that node.
- A hack to allow the analysis engine to reuse MEXs of global granularity, such as File Upload and Image Import
- Would be unnecessary if we made MEX reuse dependent on inputs, rather than containers.

MEX Granularity

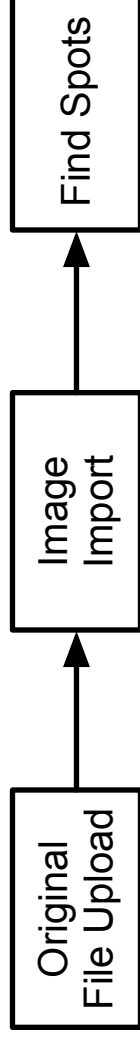
- Helps determine module reuse
- Determined by granularity of upstream inputs. The trump order is: D, I, G.

Analysis Engine Oddities

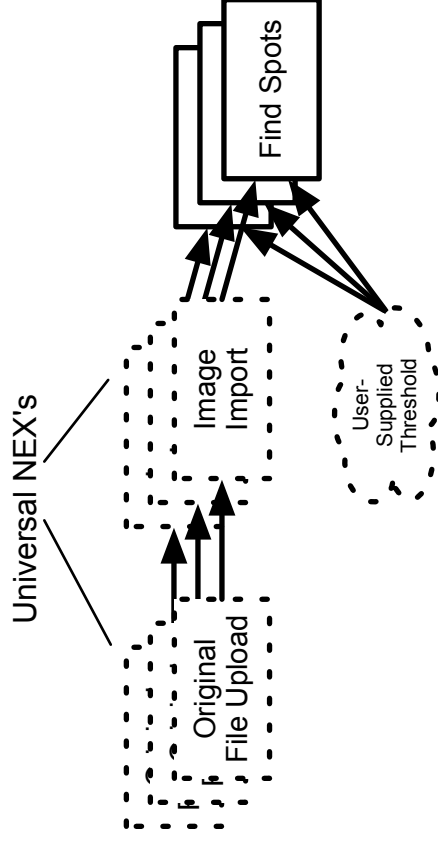
MEX Granularity & Arity Mismatch

- Depending on granularity of upstream MEXs, a module may be executed per image in the dataset, or once for the whole dataset.
- This plays havoc with arity claims if it produces one output per image.

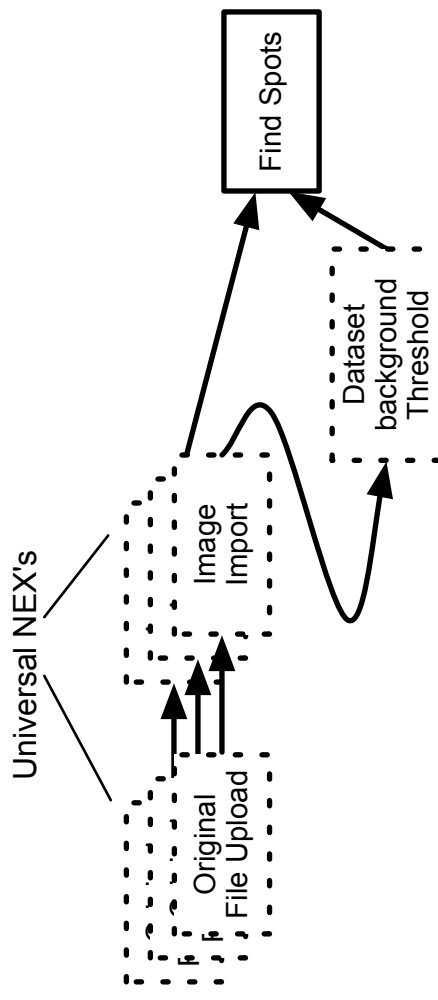
Chain



Chain Execution, threshold typed in by user



Chain Execution, threshold derived from averaging background of whole dataset



Limitations of Analysis Engine

Want to tie constant parameters to an Analysis Chain

Problem: A chain that makes predictions using a classifier object (e.g. Bayesian Network) needs a specific set of image descriptors calculated for the images. The set of image descriptors are encoded in the chain, but the classifier is free-floating.

Problem: Manually adjusted parameters that become part of a protocol

Lack of Untyped Formal Inputs

Problem 1: Machine learning software is indifferent to semantics of inputs, and different runs will use different sets of inputs

Workaround: "Signature Stitcher" module specific to each set of unique STs. It strips semantic information while maintaining audit trail.

Problem 2: Recording one-off external analyses, include audit trail do not require predeclaration of analysis.

Preferred solution: Untyped formal inputs, (AKA, typed at run-time) similar to Untyped formal outputs

No Multiple Formal Outputs of the same ST

Problem: Wavelet signatures that compute multiple levels of wavelet decompositions must specify a new ST for each additional level. A recent publication from the Kovacevic group at CMU showed a different usage of wavelets can improve performance of generic classifiers. We want to readily support this.

No iteration over Arity mismatch

Arity: How many instances are produced by a given formal output. e.g. Produce one fourier transform of a pixel set, Produce many PlaneMeans for a pixel set.

Problem 1: Described on previous slide.

Problem 2: Want to supply a list of thresholds to a segmentation module, and have one run per list element.

Workaround: Avoid & ignore