Build your own classifier with WND-CHRM and OMERO

Simon Li
<table>
<thead>
<tr>
<th>Outline</th>
</tr>
</thead>
<tbody>
<tr>
<td>Automated classification: What and why</td>
</tr>
<tr>
<td>How does this differ from OMERO.searcher?</td>
</tr>
<tr>
<td>WND-CHRM: How does it work?</td>
</tr>
<tr>
<td>Live demo</td>
</tr>
<tr>
<td>Future development</td>
</tr>
</tbody>
</table>
A way of automatically annotating images within a limited domain, such as categorical tags—could be biological, experimental, image modality, etc. E.g. Automatic tagging of metadata, automated cancer grading, looking for abnormal samples.

Some similarity to OMERO.searcher in the sense of labelling patterns, but could also be used for other tasks such as automated cancer grading.

Situations where we have a lot of images and want a way to group them.
High throughput screens mean we’ve got more data than we can handle. Large number of samples, large number of dimensions. Also saves time if we can make use of automated analysis. Should be more reliability and consistent, ideally unbiased though in practice that depends on how it’s been trained. Better accuracy means subtle phenotypes can be detected. A way to explore data- if there are discernible differences between groups maybe it’s worth investigating further. Not a complete substitute for human analysts.
OMERO.searcher is targeted at general database-wide image search subject to the domain of the image feature set, i.e. searching a whole database for a particular class of image based on a small number of examples. It is based on the idea of pairwise image similarity and does not require image labeling (though this is useful for testing), and is dynamically updated with new data. Furthermore it should ideally be usable by someone with no technical knowledge, i.e. it “just works”. In contrast WND-CHRM is designed to be used to solve a particular well-posed biological image classification problem, such as distinguishing between different types of cancer. It therefore requires a training process involving significant user input, primarily to provide labelled example images of each class/type. WND-CHRM learns from these training images, and the resulting classifier can be used to make predictions on new images.
WND-CHRM

Image features: these include standard features (mean intensity, variance, several texture descriptors), and also more complex features which should be uncorrelated.
1. Arrange the training images into separate Datasets, one per category, all within a single Project.
2. Select the Project, run the Pycharm_Extract_Features_Multichannel.py script. This will create a HDF5 FileAnnotation on each Dataset containing the calculated features for the images in that Dataset.
3. Select the Project, run the Pycharm_Build_Classifier.py script. This will create three HDF5 FileAnnotations on the Project which hold the classifier parameters. It will also create a Tag-set named after the Project from Step 1, containing with one tag for each class (corresponding to the names of the Datasets in Step 1).
1. Run the `Pychrm_Extract_Feature_Multichannel.py` script on the Dataset(s) or Project(s) containing the images to be labelled
2. Select the Dataset(s) or Project(s) to be labelled, run the `Pychrm_Predict.py` script manually entering the Project-ID of the training project
3. By default the classifier will tag images with the predicted label, and also add a comment containing the probabilities of each class.
4. Predictions can also be browsed from the Tags view in the Insight or Web clients.
Aim is to have a tool which is useful for researchers, and can produce scientifically sound results.
Unified API, so OMERO.searcher can use WND-CHRM features and vice-versa, and also other 3rd party features
Continuous predictions, e.g. expected survival time based on a cancer biopsy
Cluster support, speed up training stage, especially feature calculation, allow real-time predictions
Algorithm developers, instead of being a place to release algorithms maybe OMERO should support people who want to develop directly against OMERO, e.g. providing support for algorithm evaluation metrics such as cross-validation tools