

Q: What is the “Java heap size” error? How do I avoid it?

A: This means Java heap runs out of memory. When launching BIOCAT, it is better to specify a heap size of JVM to override the default heap size (which is small). For example, you can use the following command to set the max heap size by:

```
java -Xmx4G -jar biocat.jar
```

This assumes you have a 64 system with 4G+ RAM. You can adjust the parameter based on your hardware.

In addition, if BIOCAT runs out of memory in the stage of classification, the user may choose to include a feature selector to reduce the size of feature set for each image (e.g. pick the top 500 features).

Q: The model selection process is slow. Anything I can do to improve the speed?

A: Some pattern recognition algorithms do take a while to finish on large data sets. If you are comparing multiple algorithm chains, it may take longer. The table below describes the characteristics of the algorithm modules which may be helpful for you to understand the process. It is suggested that the user first test the algorithm in “single” mode before trying the extensive comparison with “multiple” mode.

Q: Can BIOCAT classify non-biological images?

A: BIOCAT is designed to allow working with a specific channel of florescence images, or working with gray-scale images. So RGB non-biological images will need to be converted to gray scale images in order to work with BIOCAT.

Characteristics of Some Feature Extractors in BIOCAT

	Running Time	Size of Feature Set	What if images (or ROI) in the set are of different size?
HAAR Wavelet	Medium	As big as image size.	Resize to make it the same as the first image
2D Hu Moments	Fast	8	No resizing.
Object Territory	Slow	1	No resizing.

Object Statistics	Slow	7	No resizing.
FeatureJ Statistics	Fast	14	No resizing.
FeatureJ Laplacian	Fast	As big as image size.	Resize to make it the same as the first image.
FeatureJ Edges	Fast	As big as image size.	Resize to make it the same as the first image.
FeatureJ Derivative	Fast	As big as image size.	Resize to make it the same as the first image.
FeatureJ Structure	Fast	As big as image size.	Resize to make it the same as the first image.
FeatureJ Hessian	Fast	As big as image size.	Resize to make it the same as the first image.
Zernike Moments	Medium	20	No resizing.
3D Hu Moments	Fast	5	No resizing.
3D anisotropic wavelet	Medium	Size of one image slice	Exception

Additional Notes:

- For feature extraction in BIOCAT, the memory used for processing each image is reused by subsequent images/slices.
- The bigger the feature set size, the more likely BIOCAT may encounter memory problem in later steps of feature selection/classification. In general, assuming a 4G Java heap size on a PC, feature size * set size is suggested to be below 1G.
- For a big image set (about 1000 images or more), it is suggested to conduct feature selection to reduce memory footprint, while feature selector is optional for small image sets. Examples of algorithm chains:

HAAR + Fisher Feature Selector to get 100 Features + some classifier

Hu Moments Features + Object Statistics Features + some classifier.