# Publicly Available Implementation

All the codes are implemented in a few publicly available Java projects:

- Project *VMGenericTools* provides static methods with general applicability used throughout other Java projects[1].
- Project *VMMetricSpaceTechniquesImpl* defines various techniques for the similarity search. The codes use interfaces for reading and storing all pieces of information including data from arbitrary source[2]. This project also defines the CRANBERRY algorithm in a java class:
  - vm.search.impl.multiFiltering.CranberryAlgorithm
- Project *VMFSMetricSpace* provides full implementations including main classes. It reads and stores all pieces of information from/into a file system in a predefined system of folders, and allows quick and simple usage of the implemented techniques for data stored in different formats[3]. Examples of relevant classes for the CRANBERRY algorithm are:
  - vm.fs.main.datatools.voronoiPartitioning.FSVoronoiPartitioningMain
    - performs and stored the Voronoi partitioning of a given dataset
  - vm.fs.main.objTransforms.learning.FSLearnGHPSketchingMain
    - learns the sketching transformation
  - vm.fs.main.objTransforms.apply.FSApplyGHPSketchingMain
    - transforms dataset and pivots into binary sketches
  - vm.fs.main.search.filtering.learning.LearnSecondaryFilteringWithGHPSketchesMain – learns mapping of the Hamming distances on sketches to the distances in the original search space
  - vm.fs.main.objTransforms.learning.FSLearnSVDMain
    - learns and stores the Singular Value Decomposition which is used to perform the Principal Component Analysis (PCA)
  - vm.fs.main.objTransforms.apply.FSApplyPCAMain
    - transforms and stores vectors by the PCA. Possibly, just the prefixes of the PCA are computed and stored.
  - vm.fs.main.search.filtering.learning.LearnTOmegaThresholdsOrigSISAPMain – learns and stores the simRel filtering
- Finally, project *VMTrials* serves for the developing. It now contains the package:
  - vm.vmtrials.tripleFiltering_Challenge

  The package now includes a class Main which learns and builds the CRANBERRY Algorithm and also evaluates the 10NN similarity queries. The main class relies on the following precomputed information that must be stored in advance, using the VMFSMetricSpace project:
  - Learnt sketching technique.
  - Input dataset shrunk by the PCA to a given length. We use the length of 256. Possibly, just prefixes of length 24 floats are needed.

---

[1] https://github.com/VladimirMic/VMGenericTools
[2] https://github.com/VladimirMic/VMMetricSpaceTechniquesImpl
[3] https://github.com/VladimirMic/VMFSMetricSpace

- "tOmegaThresholds" for the simRel learned on the PCA vectors of the full length, i.e., 256 floats. Due to the main memory difficulties, the current Main class does not learn this information during the index build but assumes it is already stored in a proper csv file.