DiviK Software Manual

Dariusz Kuchta Dariusz.Kuchta@polsl.pl
Grzegorz Mrukwa Grzegorz.Mrukwa@polsl.pl
Michal Gallus Michal.Gallus@polsl.pl
Michal Wolny Michal.Wolny@polsl.pl
Sebastian Pustelnik Sebastian.Pustelnik@polsl.pl
Wojciech Wilgierz Wojciech.Wilgierz@polsl.pl

03.04.2017

Abstract

DiviK software is a tool for unsupervised segmentation of MALDI IMS data sets. It allows to either find molecularly diverse regions, or to compress such data with preservation of intra-sample heterogeneities. Achieved compression factors reach 99.87% of size reduction, while such representation can still be used for classifier training or inter-sample comparison.

Contents

Installation

MATLAB MCR

In order to make DiviK software fully operational, a MCR (MATLAB Compiler Runtime) has to be installed *(in current release - this will be removed soon)*.

However, due to the licensing by MathWorks, it is not available through project website. Nevertheless, we are still eligible to share it with you personally, so feel free to contact us by mail.

Please note, that it can be also downloaded through MathWorks website for free, but this way has not been tested yet.

DiviK software

DiviK software itself needs no installation. Once the files are unpacked from the `.zip` archive, it can be used without further delay.

Usage

Interface of the application allows to provide all the options that are *available* to be used with DiviK. There is however an established pipeline for MALDI IMS data processing and these options are available just to allow anyone extend our investigations.

Default setting provided on application start is adjusted for clustering purposes. There is a little work to switch to compression mode: it is enough to disable *Using Levels* checkbox (explained in section **Parameters**).

After setting all the parameters (or at least required ones) it is enough to press *Start DiviK* button.

Input specification

Input file is a text file constructed as follows:

1. Row with global metadata - it can contain anything (unused for now).
2. Row with global *m/z* axis - data has to be resampled before usage.
3. Data of each spectra, each consisting of two lines:
    1. Spatial coordinates of a spectrum (X, Y, and Z, separated with spaces). *Please note, that Z coordinate is not used yet, so it can be e.g. set to zero.*

2. Intensity values for each *m/z* value specified above, separated with spaces. Their number **must** be equal to the number of elements in *m/z* axis. This is similar to *imzML* format in *processed* form.

Artificial test data (*only for demonstration of this structure*):

```
in this line are global metadata, which is discarded for now
899.99 902.58 912.04
1 1 0
12 20 0
2 1 0
9 18 13
1 2 0
5 10 20
2 2 0
14 2 19
```

This data cannot be used for testing the program itself; it is just a reference, how to format data file.

Sample **real** data file is available here. The same data set was used in G. Mrukwa, G. Drazek, M. Pietrowska, P. Widlak and J. Polanska, "A Novel Divisive iK-Means Algorithm with Region-Driven Feature Selection as a Tool for Automated Detection of Tumour Heterogeneity in MALDI IMS Experiments," in International Conference on Bioinformatics and Biomedical Engineering, 2016, so anyone can compare results.

Output specification

Result of this hierarchical segmentation/compression is stored in the form of the tree, illustrating consecutive splits.

This tree keeps all of the information present at each division, in following fields:

1. **QualityIndex** - quality index of best division at particular level (for now it is Dunn's index)
2. **Centroids** - collection of centroids for each sebregion found
3. **Partition** - assignment to subregions at particular level
4. **AmplitudeThreshold** - threshold of log2 of feature amplitude used to remove low-abundance features
5. **AmplitudeFilter** - collection with boolean value of *true*/*false* for each feature. *True* means, that a feature was used in split.
6. **VarianceThreshold** - exactly the same as **4**, but variance of each feature is considered
7. **VarianceFilter** - exactly the same as **5**, but with respect to variance. Please **note**: all features marked *false* in **AmplitudeFilter** have been excluded from this filter.

8. **Merged** - this collection contains information about cluster assignment after all stages of clustering. Cluster indices are the same for spectra in the same cluster, and different from those from another. However, no assumption can be made, that they are consecutive integers.
9. **Subregions** - Collection with results of recursive segmenting of subregions, in the exactly same form, as this tree.

Such a tree is serialized to JSON format, which is supported by a variety of tools and environments.

Sample output is presented below:

```
{
  "QualityIndex": 3.8840405390582151,
  "Centroids": [ ... ],
  "Partition": [ ... ],
  "AmplitudeThreshold": 9.6377174422348748,
  "AmplitudeFilter": [ ... ],
  "VarianceThreshold": 26.383680633770286,
  "VarianceFilter": [ ... ],
  "Merged": [ ... ],
  "Subregions": [
    {
      "QualityIndex": 1.0056280394007902,
      "Centroids": [ ... ],
      "Partition": [ ... ],
      "AmplitudeThreshold": "NaN",
      "AmplitudeFilter": null,
      "VarianceThreshold": 26.450276742883155,
      "VarianceFilter": [ ... ],
      "Merged": null,
      "Subregions": null
    },
    {
      "QualityIndex": 2.6253359243037786,
      "Centroids": [ ... ],
      "Partition": [ ... ],
      "AmplitudeThreshold": "NaN",
      "AmplitudeFilter": null,
      "VarianceThreshold": 26.585733127398438,
      "VarianceFilter": [ ... ],
      "Merged": null,
      "Subregions": null
    }
  ]
}
```

Dots `...` represent content of an array; the array is closed with `]` sign and each entry is separated by a comma `,`. Instances of tree are enclosed into curly braces `{...}`.

Full sample output file is present here.

Parameters

1. **Input path** - must point to the file with properly formatted data.
2. **Output path** - this is the directory, where result will be stored.
3. **MaxK** - is the limit of number of clusters checked by k-means algorithm.
4. **Level** - limit of recursion nesting, when using level criterion as stop condition.
5. **Using Levels** - specifies whether to stop recursive splitting of data at particular level (specified above), or to use minimal subregion size ratio.
6. **Amplitude** - if checked, amplitude-based global filtration of features is performed at top level. This removes low-abundance features which are noisy.
7. **Variance** - if checked, variance-based local filtration of features is performed at each stage of splitting. This selects most informative features for every division.
8. **Percent size limit** - specifies, what is the lowest rate of subregion size to starting size of whole preparation, below which no segmentation is performed. Applicable only when not using levels as stop condition for recursion.
9. **Feature preservation limit** - specifies, how much features *must* be preserved regardless the filtration applied (to not to leave e.g. single feature)
10. **Metric** - method to measure dissimilarity between spectra.
11. **Plotting partitions** - when true, dumps plots regarding top-level and merged split, into output directory.
12. **Plotting recursively** - when true, dumps above plots for each of the subregions, too.
13. **Plotting decomposition** - when true, dumps plots regarding amplitude and variance filtering at the topmost level.
14. **Plotting decomposition recursively** - as above, but for decomposition plots.
15. **Max decomposition components** - the maximal number of components used to decompose amplitude/variance and create filter.
16. **Cache path** - path to directory where partial results may be stored. This speeds up repeated calculations if trying to chech several variants of nesting, etc. It can grow large. Should be specified the same for all analyses, to be able to resolve already computed results. Applicable only if **Caching** is enabled.
17. **Caching** - if true, partial results are saved on disk to avoid unnecessary

computational overhead in case of repeated calculations.

18. **Max iterations for K-means** - maximal number of iterations, after which k-means algorithm stops.
19. **Pretty print** - if true, output JSON file is printed with visually indented rows. Otherwise no additional white characters are introduces, to reduce the number of characters.

Final notes

In case of any questions, do not hesitate to contact us by mail.

References

This software is part of contribution made by Data Mining Group of Silesian University of Technology, rest of which is published here.

- Marczyk M, Polanska J, Polanski A: Comparison of Algorithms for Profile-Based Alignment of Low Resolution MALDI-ToF Spectra. In Advances in Intelligent Systems and Computing, Vol. 242 of Man-Machine Interactions 3, Gruca A, Czachorski T, Kozielski S, editors. Springer Berlin Heidelberg 2014, p. 193-201 (ISBN: 978-3-319-02308-3), ICMMI 2013, 22-25.10.2013 Brenna, Poland
- P. Widlak, G. Mrukwa, M. Kalinowska, M. Pietrowska, M. Chekan, J. Wierzgon, M. Gawin, G. Drazek and J. Polanska, “Detection of molecular signatures of oral squamous cell carcinoma and normal epithelium - application of a novel methodology for unsupervised segmentation of imaging mass spectrometry data,” Proteomics, vol. 16, no. 11-12, pp. 1613-21, 2016
- M. Pietrowska, H. C. Diehl, G. Mrukwa, M. Kalinowska-Herok, M. Gawin, M. Chekan, J. Elm, G. Drazek, A. Krawczyk, D. Lange, H. E. Meyer, J. Polanska, C. Henkel, P. Widlak, “Molecular profiles of thyroid cancer subtypes: Classification based on features of tissue revealed by mass spectrometry imaging,” Biochimica et Biophysica Acta (BBA)-Proteins and Proteomics, 2016
- G. Mrukwa, G. Drazek, M. Pietrowska, P. Widlak and J. Polanska, “A Novel Divisive iK-Means Algorithm with Region-Driven Feature Selection as a Tool for Automated Detection of Tumour Heterogeneity in MALDI IMS Experiments,” in International Conference on Bioinformatics and Biomedical Engineering, 2016
- A. Polanski, M. Marczyk, M. Pietrowska, P. Widlak and J. Polanska, “Signal partitioning algorithm for highly efficient Gaussian mixture modeling in mass spectrometry,” PloS one, vol. 10, no. 7, p. e0134256, 2015