Test on Healthy tissues database

# Dataset

Dataset “Healthy tissues” [1] contains 102 instances with 10,387 attributes.

There are 35 classes defined in database. Several classes have one or two instances. All such classes were united into group ‘other’. As a result we have 19 individual classes and 1 combined class (see Table 1).

Table . Class definition

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Tissue | Number | Icon |  | Tissue | Number | Icon |
| BRAIN | 2 |  |  | BLADDER | 1 |  |
| ADRENAL | 3 |  |  | BREAST |
| BOWEL | 4 |  |  | BUFFYCOAT |
| COLON | 5 |  |  | CERVIX |
| ESOPHAGUS | 6 |  |  | DIAPHRAGM |
| FALLOPIAN | 7 |  |  | EPIDIDYMUS |
| HEART | 8 |  |  | GALLBLADDER |
| KIDNEY | 9 |  |  | MUSCLE |
| LIVER | 10 |  |  | PANCREAS |
| LUNG | 11 |  |  | PARATHYROID |
| LYMPHNODE | 12 |  |  | PERICARDIUM |
| OVARY | 13 |  |  | SEMINAL |
| PROSTATE | 14 |  |  | TESTES |
| SALIVARY | 15 |  |  | THYMUS |
| SPLEEN | 16 |  |  | UTERINE |
| STOCMACH | 17 |  |  | VAGINA |
| THYROID | 18 |  |  |  |  |  |
| TONSIL | 19 |  |  |  |  |  |
| UTERUS | 20 |  |  |  |  |  |

# Feature/record removing

Unfortunately database contains missing values. To resolve this problem I decide to use procedure of record/features removing with following algorithm:

1. For each record calculate fraction of missed values.
2. For each attribute calculate fraction of missed values.
3. Select record with maximal fraction of missed values .
4. Select feature with maximal fraction of missed values .
5. If we have only one feature with missed values then remove this feature. Stop.
6. If we have only one record with missed values then remove this record. Stop.
7. If then remove feature.
8. If then remove record.

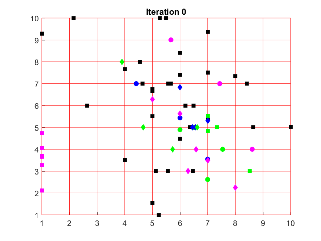
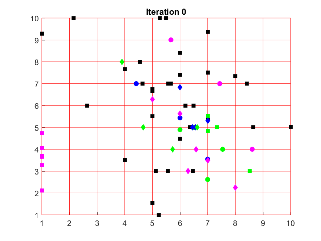
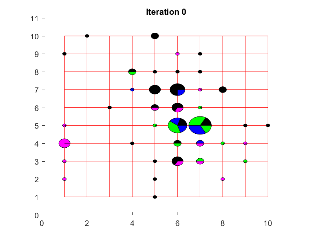
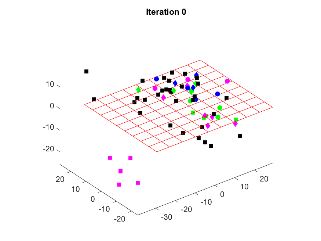
At the beginning we have missed values in all records and in 10,1090 features among 10,387. There is record with 33.3% of missed values and attribute with 24.5% of missed values.

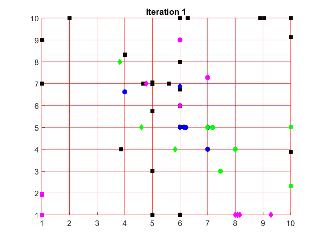
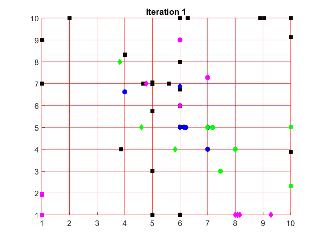
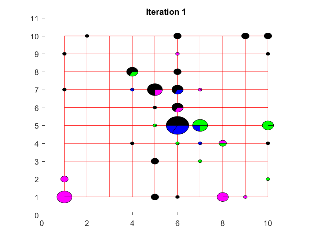
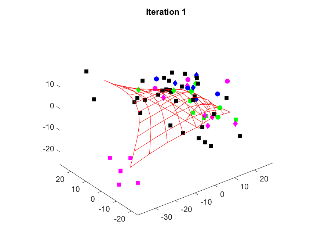
After de-gapping database we have 66 instances and 1,570 attributes. These means that proposed algorithm is not very appropriate for data cleaning because we lost too many data.

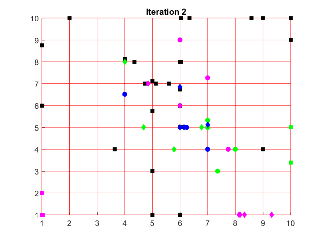
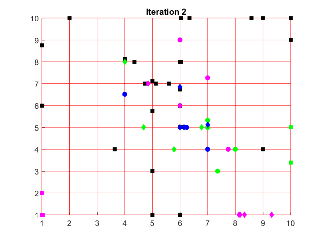
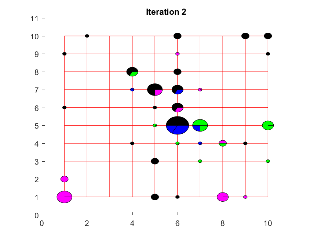
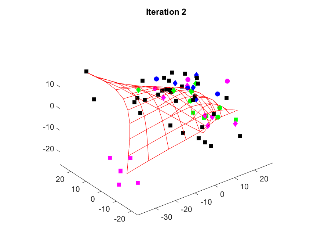
Classes for ‘cleaned’ database are presented in Table 2.

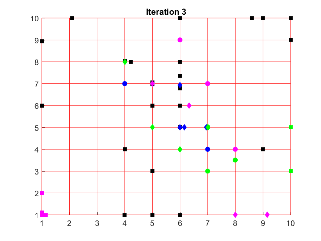
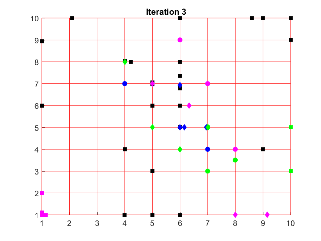
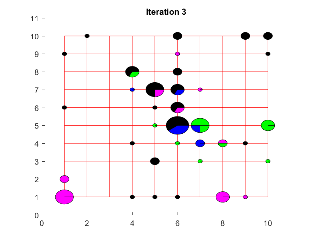
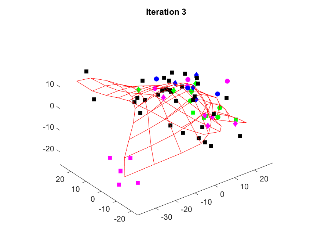
Table . Class definition

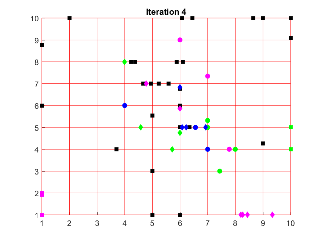
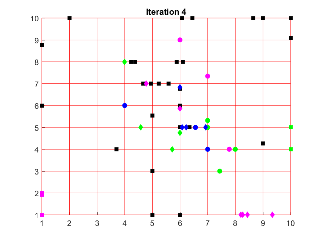
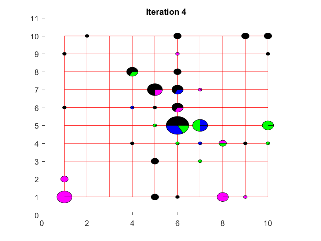
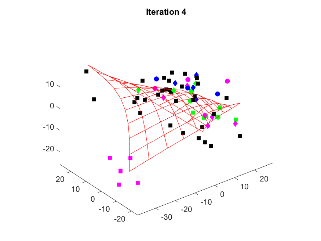
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Tissue | Number | Icon |  | Tissue | Tissue | Number | Icon |
| ADRENAL | 9 |  |  | BLADDER | MUSCLE | 1 |  |
| BRAIN | 2 |  |  | BOWEL | OVARY |
| ESOPHAGUS | 4 |  |  | BREAST | PARATHYROID |
| FALLOPIAN | 7 |  |  | CERVIX | SPLEEN |
| LUNG | 5 |  |  | COLON | STOCMACH |
| PROSTATE | 6 |  |  | DIAPHRAGM | TESTES |
| SALIVARY | 3 |  |  | HEART | THYMUS |
| THYROID | 8 |  |  | KIDNEY | TONSIL |
|  |  |  |  | LIVER | UTERINE |
|  |  |  |  | LYMPHNODE | UTERUS |

a 

b 

c 

d 

e 

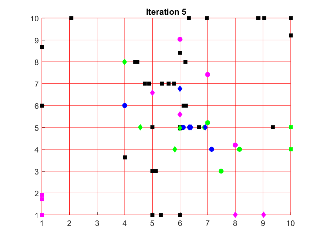
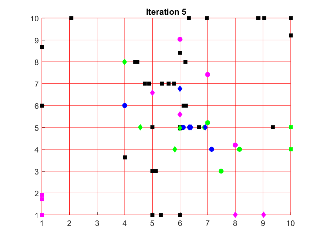
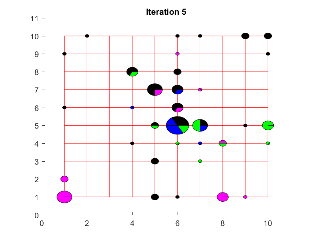
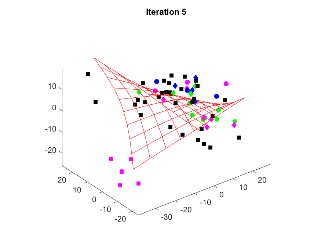
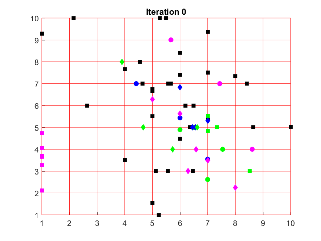
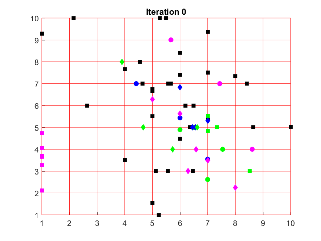
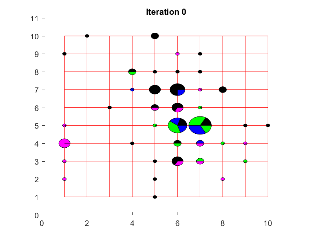
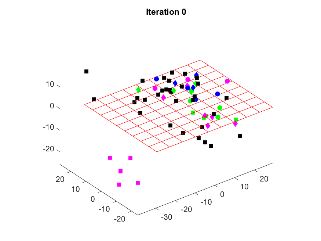
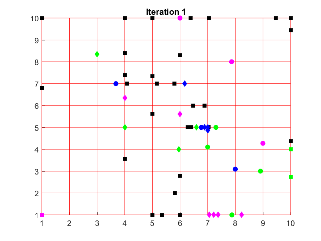
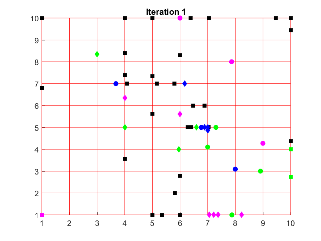
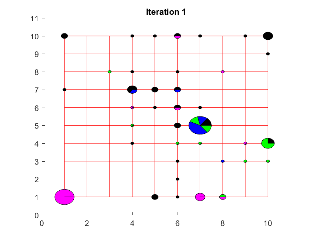
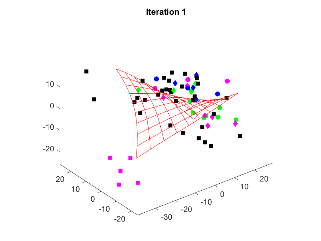
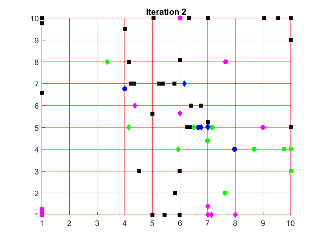
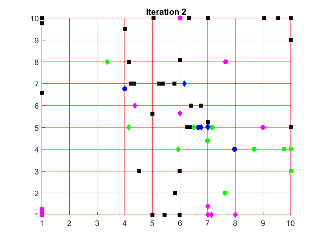
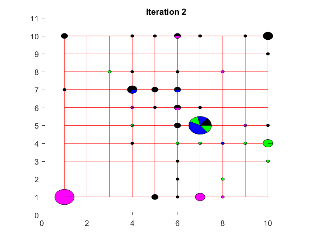
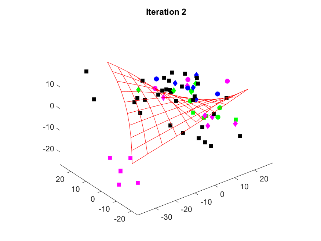
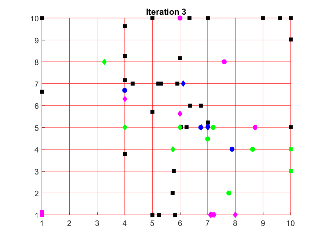
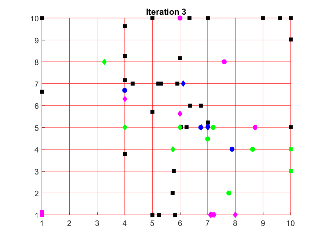
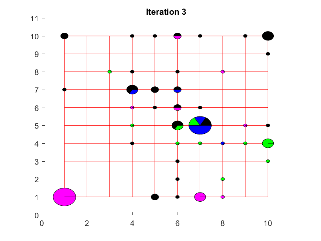
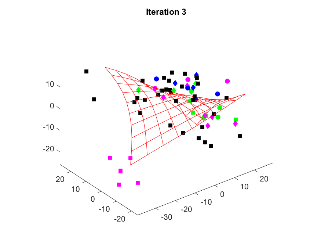
f 

Figure . Fitting of ElMap without restart for (top down) with (left to right) 3D image, projection onto nodes, onto edges and onto faces; a – without training; b – 'stretch', 0.01, 'bend', 0.1; c – 'stretch', 0, 'bend', 0.1; d – 'stretch', 0, 'bend', 0.01; e – 'stretch', 0, 'bend', 0.2; f – 'stretch', 0, 'bend', 0.5

a 

b 

c 

d 

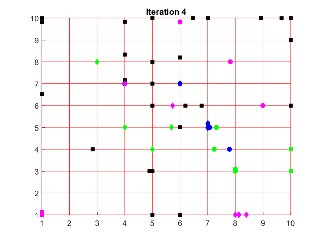
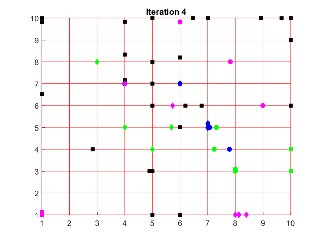
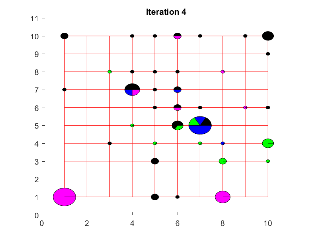
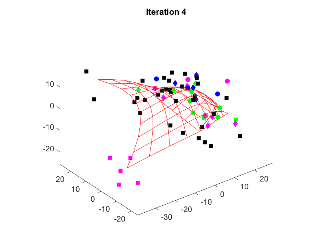
e 

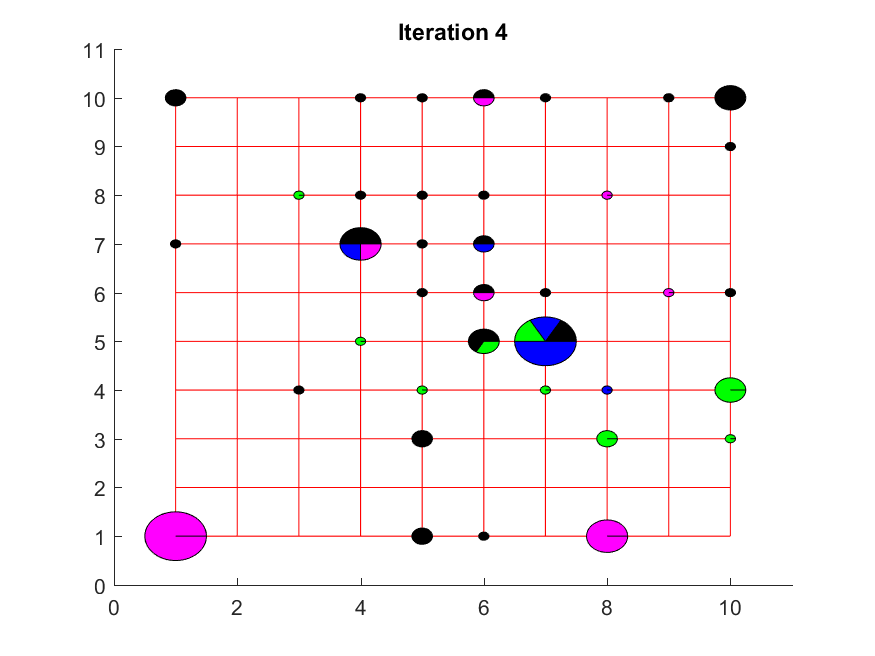
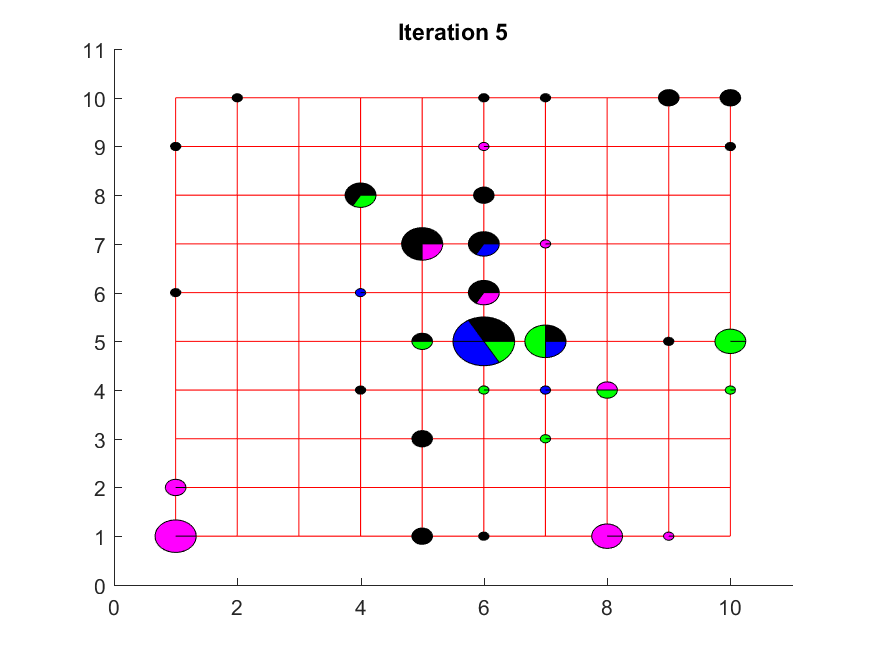
Figure . Fitting of ElMap without restart for with (left to right) 3D image, projection onto nodes, onto edges and onto faces; a – without training; b – 'stretch', 0.01, 'bend', 1; c – 'stretch', 0, 'bend', 1; d – 'stretch', 0, 'bend', 0.5; e – 'stretch', 0, 'bend', 0.1

Spend time is presented in Table 3.

Table . Times for separate operations

|  |  |  |
| --- | --- | --- |
| Operation | Time map 1 | Time map 2 |
| Initialisation | 0.0204 | 0.0164 |
| Learning b | 0.0112 | 0.0149 |
| Learning c | 0.0108 | 0.0131 |
| Learning d | 0.0062 | 0.0096 |
| Learning e | 0.0072 | 0.0088 |
| Learning f | 0.0082 | ‑ |

Final maps of two approaches are more or less the same:



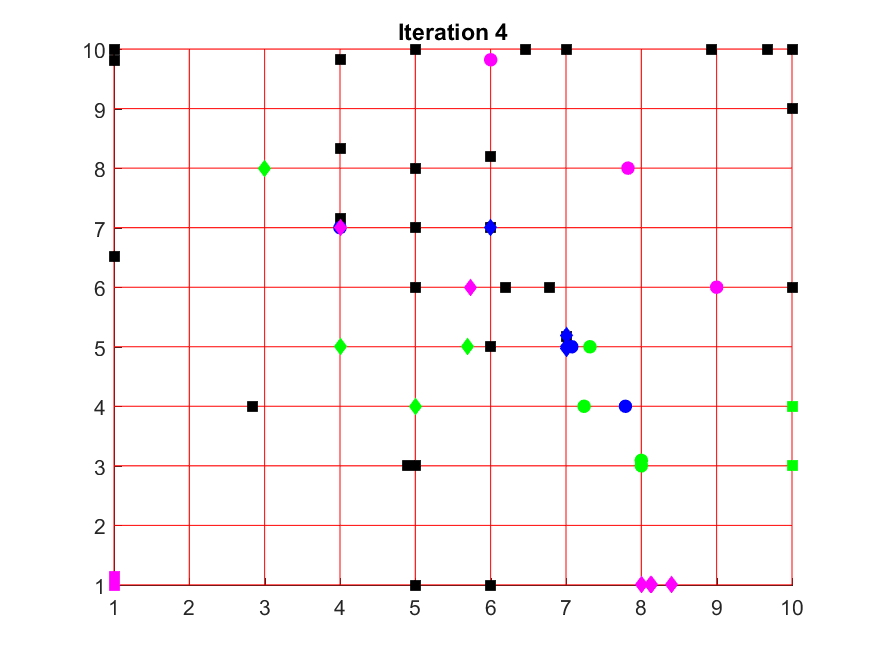
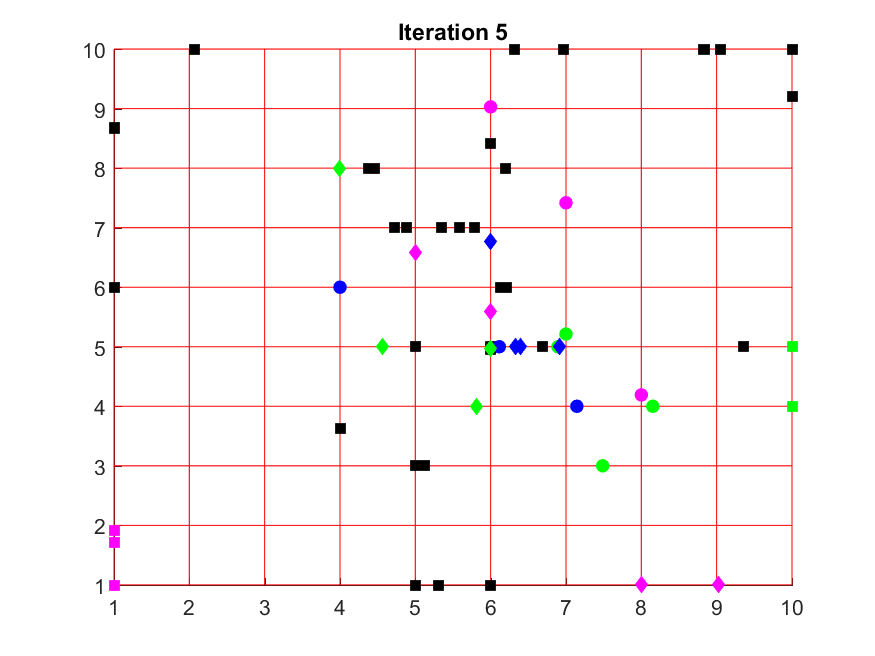


Figure . Final maps: map 1 (left) and map 2 (right)

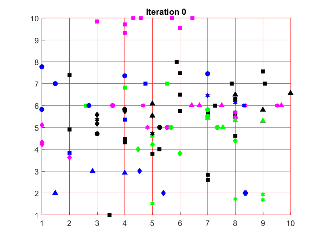
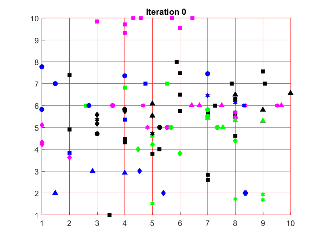
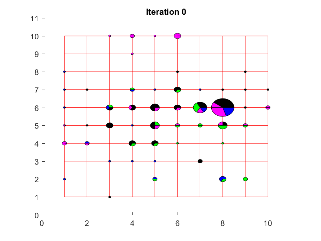
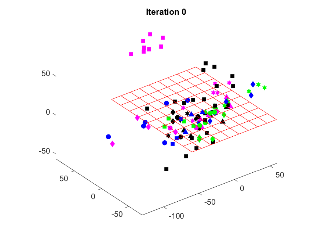
# Data repairing

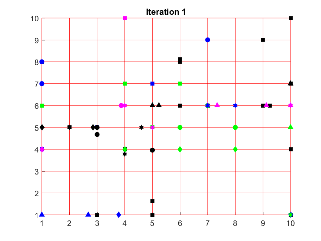
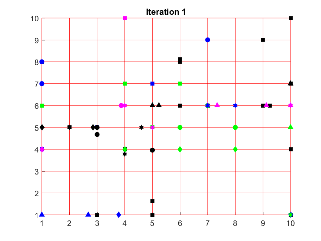
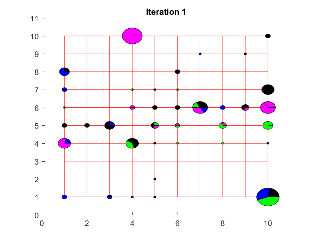
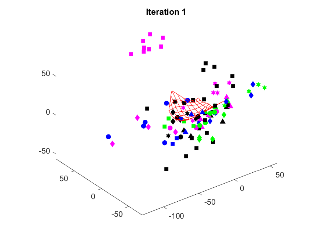
There are many methods of data imputation and data repairing. In this case we use restricted SVD method [2] with intervals defined as three standard deviation out of mean. Fraction of missing values is 8.51%. Data completing time is 332.929 sec.

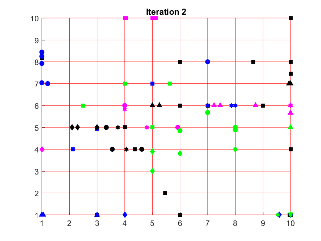
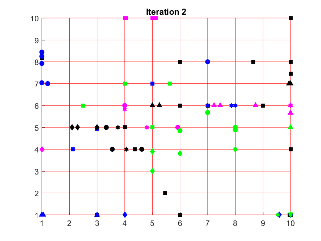
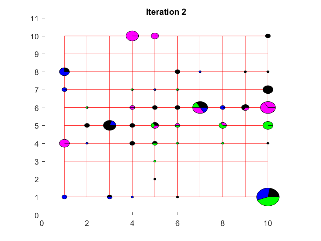
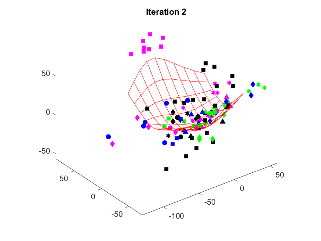
Time spent for two different maps is presented in Table 4.

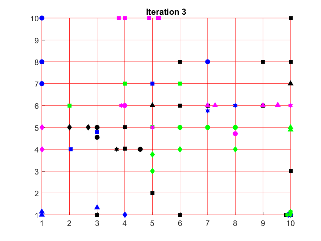
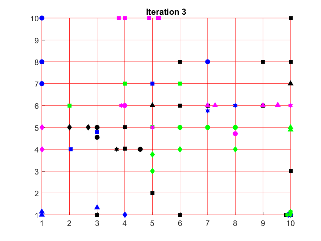
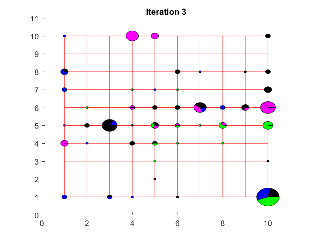
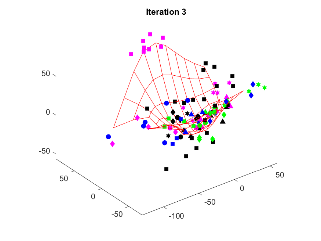
Table . Times for separate operations

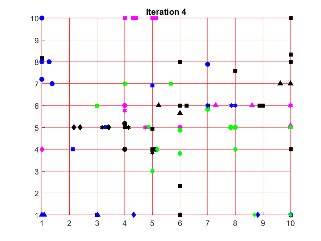
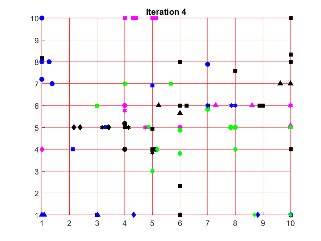
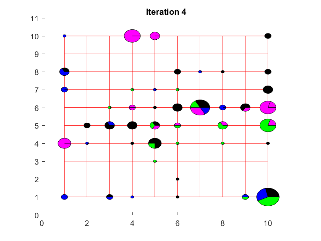
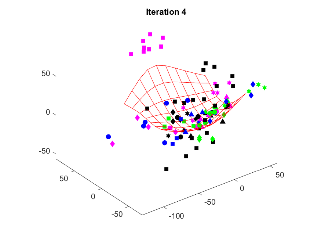
|  |  |  |
| --- | --- | --- |
| Operation | Time map 1 | Time map 2 |
| Initialisation | 0.2600 | 0.2207 |
| Learning b | 0.0878 | 0.0424 |
| Learning c | 0.0596 | 0.0500 |
| Learning d | 0.0354 | 0.0330 |
| Learning e | 0.0268 | 0.0349 |
| Learning f | 0.0406 |  |

a 

b 

c 

d 

e 

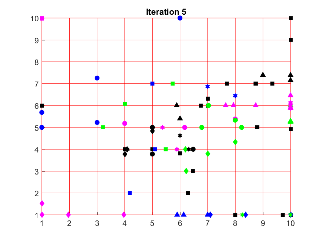
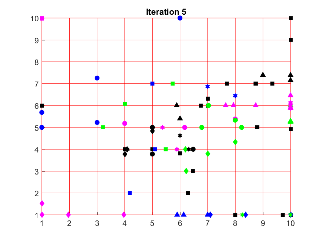
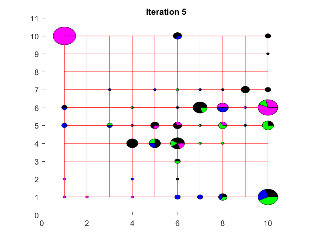
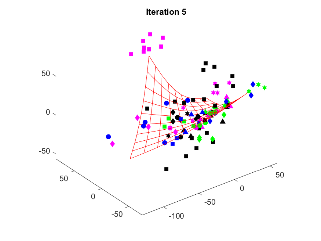
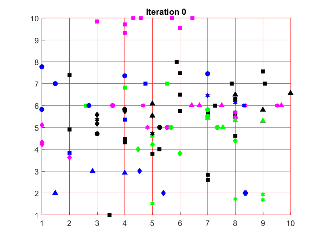
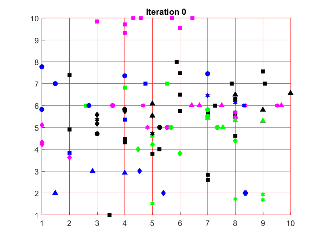
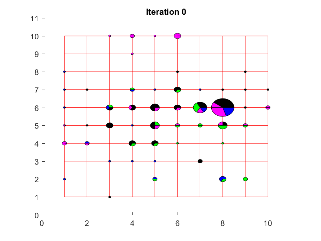
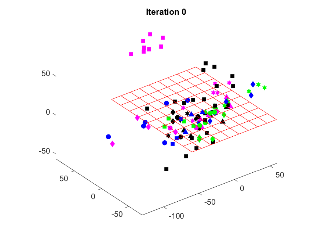
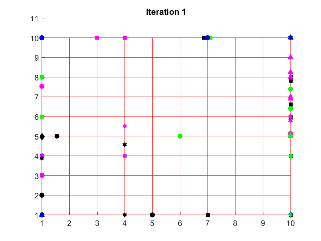
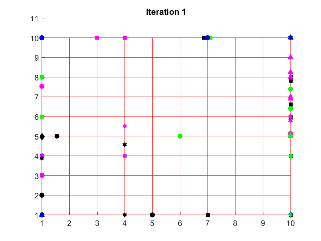
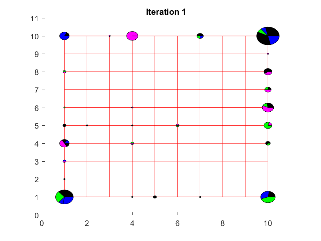
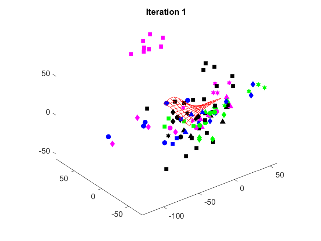
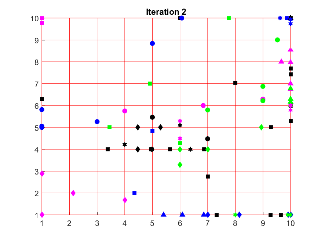
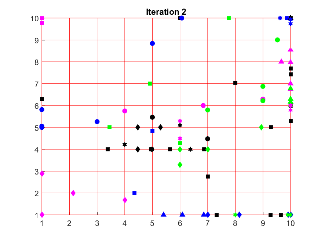
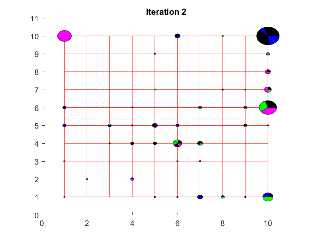
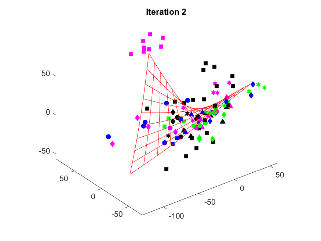
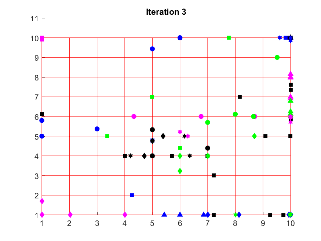
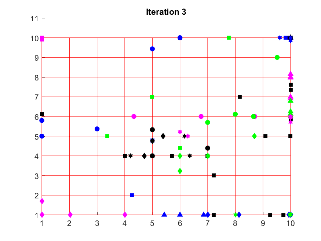
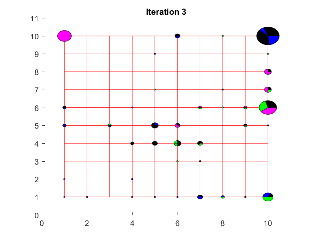
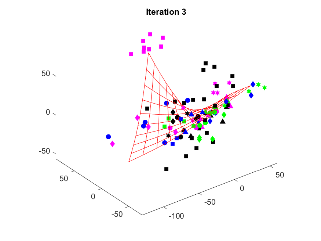
f 

Figure . Fitting of ElMap without restart for (top down) with (left to right) 3D image, projection onto nodes, onto edges and onto faces; a – without training; b – 'stretch', 0.01, 'bend', 0.1; c – 'stretch', 0, 'bend', 0.1; d – 'stretch', 0, 'bend', 0.01; e – 'stretch', 0, 'bend', 0.2; f – 'stretch', 0, 'bend', 0.5

a 

b 

c 

d 

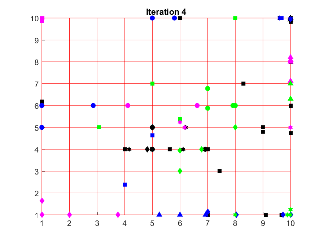
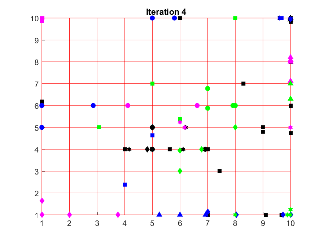
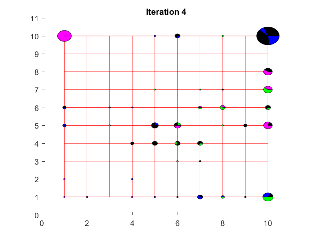
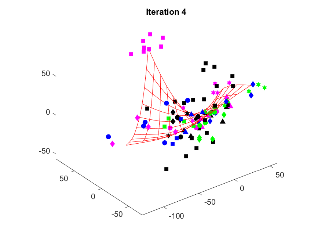
e 

Figure . Fitting of ElMap without restart for with (left to right) 3D image, projection onto nodes, onto edges and onto faces; a – without training; b – 'stretch', 0.01, 'bend', 1; c – 'stretch', 0, 'bend', 1; d – 'stretch', 0, 'bend', 0.5; e – 'stretch', 0, 'bend', 0.1

Final maps of two approaches are very different:

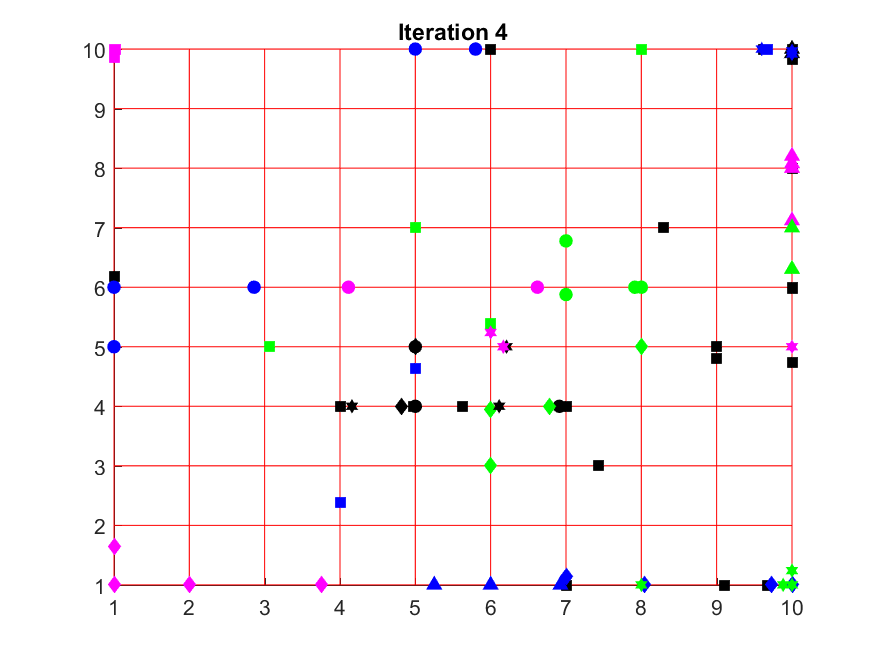
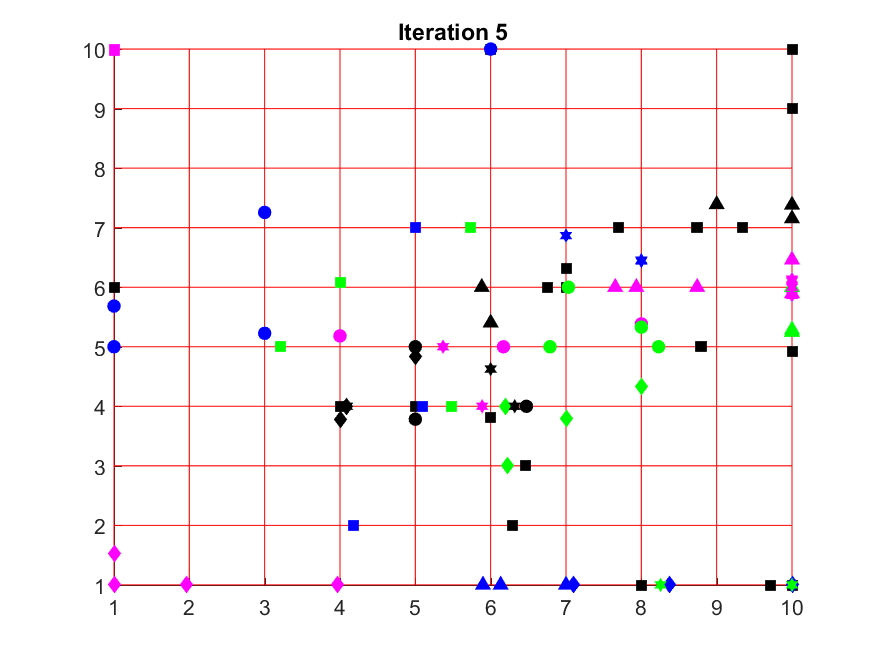
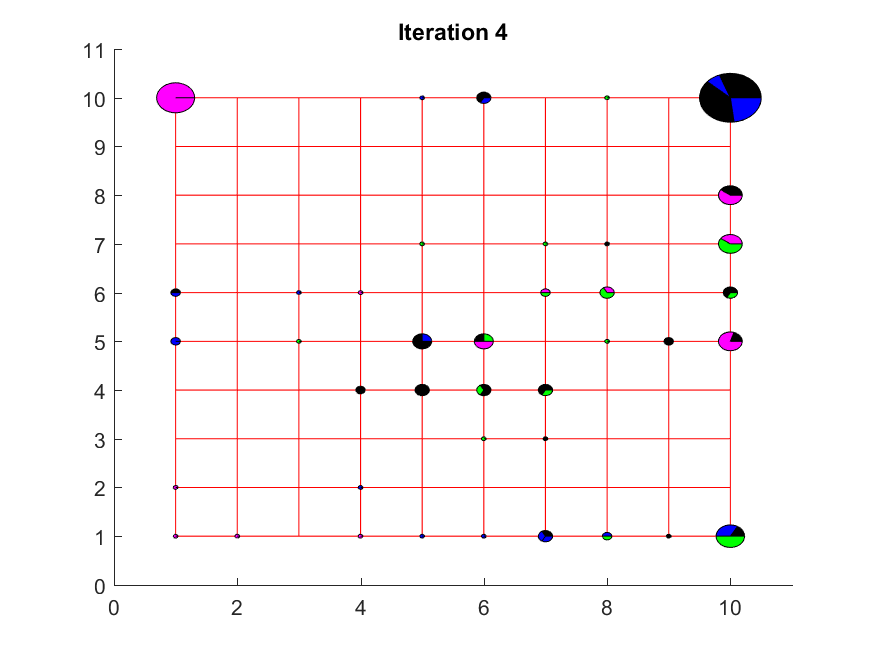
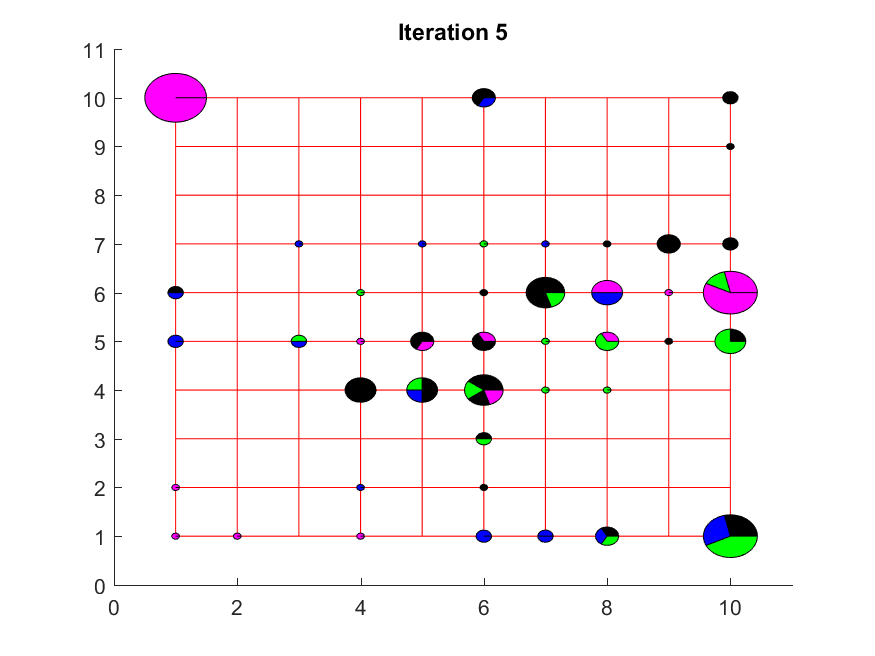
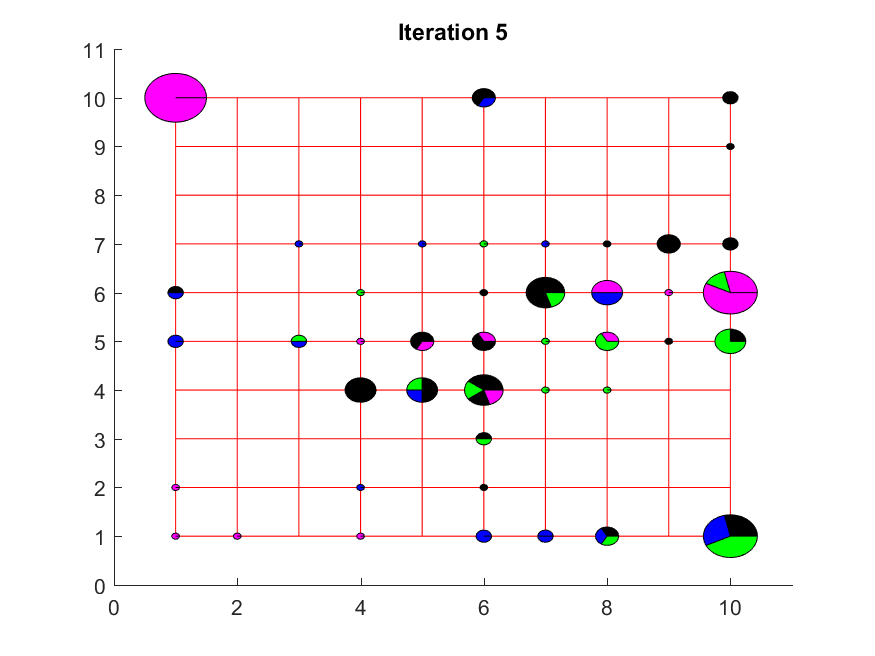
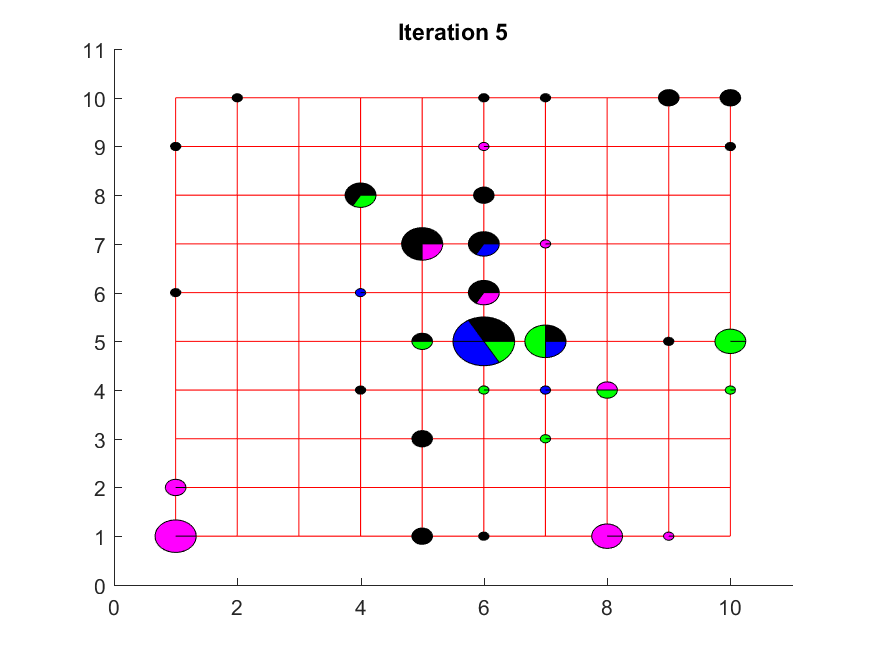


Figure . Final maps: map 1 (left) and map 2 (right)

# Comparison of two approaches

Two final maps for two approaches are presented in Figure 7. As we can see maps are very different and the second approach provide essentially richer figure.



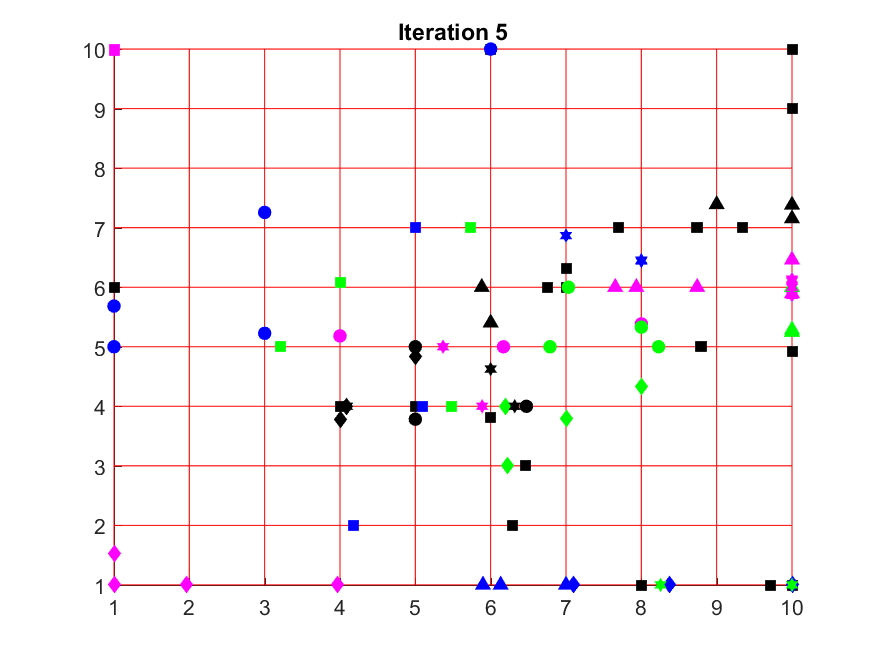
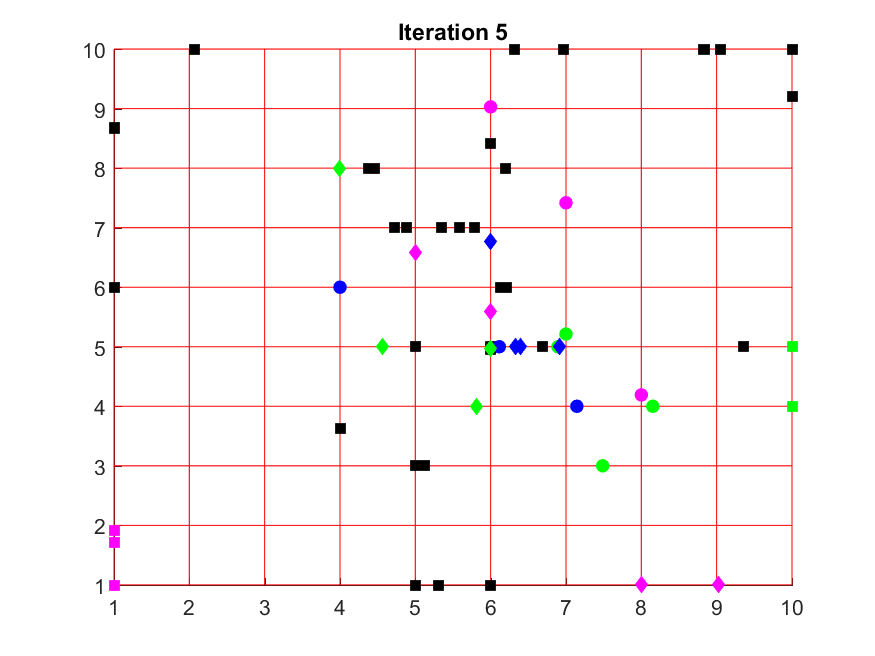


Figure . Final maps for two approaches

# Reference

1. Dataset “Healthy tissues”, <http://www.ihes.fr/~zinovyev/princmanif2006/>.
2. Data imputation by kNN and SVD <https://github.com/Mirkes/DataImputation>

[1] https://www.researchgate.net/publication/299414114\_Missing\_data\_imputation\_toolbox\_for\_MATLAB