Supplementary Materials for paper: A Novel Soft-Coded Error-Correcting Output Codes Algorithm

Kai-Jie Feng, Xiao-Na Ye, Sze-Teng Liong, Kun-Hong Liu, Qing-Qiang Wu, Gui-Ming Dai, Yuna Okina

I. THE SUMMARY OF SYMBOLS USED IN THIS PAPER

Table SI lists the symbols used in this study.

Table. SII Symbols used in this study

N_M	The number of columns in codematrix.
N_c	The number of classes.
N_s	The number of samples
M_{col}	The col^{th} column
M^i	The i^{th} codeword(row)
M_j^i	The code of the i^{th} row and j^{th} column
x_i^l	The l-th feature of sample x_i
C_i	The i^{th} class
$G_{col}^{k,1}/G_{col}^{k,2}$	The positive/negative group in the k-th iteration of SFFS in the generation of the col^{th} column
$c_{col}^{k,1}/c_{col}^{k,2}$	The group centroid of $G_{col}^{k,1}/G_{col}^{k,2}$ in the k-th iteration of SFFS in the generation of the col^{th} column
$\|G_{col}^{k,i}\|$	The number of samples in $G_{col}^{k,i}$
$\|C_r\|$	The number of samples in C_r
$T_{col}^{k,i}$	The number of classes in $G_{col}^{k,i}$
r_{col}	The col^{th} regressor
ave_{col}^i	The average value of r_{col}'s outputs for samples in C_i
w_{col}	The weight of r_{col}
$O_{col}^{i,s}$	The output value of r_{col} for the s^{th} sample in C_i

II. ADDITIONAL EXPERIMENTAL RESULTS

A. Comparison of ensemble scale

The number of columns in $\boldsymbol{M}$ is the scale of an ECOC ensemble, and an excellent ensemble should be compact and accurate. Fig. S1 illustrates the average size of $\boldsymbol{M}$ for each algorithm on the UCI and microarray data sets. It is observed that the ensemble size of OVO is $N_C \times (N_C - 1)/2$, which is far larger than other ECOC algorithms. In order to limit the range of the y-axis, OVO is excluded in Fig. S1.

While four ECOC algorithms, OVA, DC-ECOC, D-ECOC, and SC-ECOC, require the same number of columns ($N_c - 1$). And their ensemble scales are the smallest among all ECOC algorithms. SR and DR are the data independent ECOC algorithms, which always have a fixed number of columns for data sets with the same N_c. They demand $15log_2(N_c - 1)$ and $10log_2(N_c - 1)$ learners respectively. While the ensemble size of ECOC-ONE changes on different data sets, and the ensemble scales of ECOC-ONE is the largest among these seven ECOC algorithms in most cases. SR and DR are ranked as the second and third largest respectively. In conclusion, our SC-ECOC performs the best with the smallest ensemble size in general.

Table SII
The average distance between predicted vectors and target codewords.

ID	HSC	SC-BA	SC
A	0.3901	0.1358	0.1307
B	1.1106	0.6515	0.4816
C	0.6968	0.5360	0.4275
D	0.8036	0.2920	0.2785
E	0.5967	0.4746	0.4091
F	1.5609	0.9941	0.833
G	0.5514	0.3338	0.2636
H	0.8384	0.4105	0.4015
I	0.6986	0.2847	0.2811
J	0.6691	0.4317	0.4166
K	0.5944	0.3262	0.3003
L	0.7437	0.3692	0.3400
M	0.6559	0.3757	0.3567
N	0.3706	0.1486	0.1228
O	0.9293	0.4728	0.3865
P	1.0921	0.6055	0.5153
Q	0.3537	0.3052	0.2958
R	1.0504	0.6452	0.5038
S	1.2726	0.7242	0.5403
T	0.6434	0.4657	0.3656
U	0.6394	0.591	0.5188
V	0.8946	0.7428	0.6177
W	0.6854	0.5511	0.4987
X	1.9271	1.5271	1.1547
Y	0.526	0.505	0.4404
Z	0.5233	0.4756	0.4425
AA	0.6943	0.5648	0.5188
AB	0.7012	0.5689	0.5431
AC	0.4541	0.3798	0.3197
AD	0.5624	0.5	0.4665

B. The distance comparisons of different coding schemes

The distance between the target codeword and the predicted vector is vital to the classification task, and the smaller

distances guarantee more accurate predictions in the decoding process. This subsection compares the average distance between predicted vectors and the target codewords based on the three coding schemes: HSC, SC-BA and SC, as shown in Table SII.

It is obvious that the average distances of the HSC method are much larger than those obtained by the SC method, while it is reasonable because the value of each element in the HSC method is larger than those of the SC method. But as mentioned above, The HSC method can still lead to satisfactory performance usually. On the other hand, the average distances of the SC method are smaller than those of the SC-BA method in all cases, showing that after the adjustments, the codematrices better fit the distribution of regressors' outputs. These results comply with previous observations.

C. The analysis of value changes of the target function

The typical changes of the values of the target function $E(G_{col}^{k,1}, G_{col}^{k,2})$ for generating the corresponding first columns are illustrated in Fig. S4. Here seven data sets are deployed for comparisons with different symbols. The x-axis represents the number of loops in the SFFS, and the y-axis represents the value of $E(G_{col}^{k,1}, G_{col}^{k,2})$.

From Fig. S4, it is obvious that the value of the target function increases fast at the first few loops. And when the target function reaches its peak value, SFFS would repeat one more loop, and it stops when the value can no longer be promoted. The more classes a data set has, the more loops SFFS requires in most cases. While it is found although the Movement data set (15 classes) contains the largest number of class among the seven data sets, it only requires 8 loops for splitting these classes to two groups. In contrast, the Texture data set needs 12 loops to separate its 11 classes. That is, the number of SFFS loops for determining a new column would be different according to the difficulty of class separation. While it is found that in our algorithm, the maximum number of loops of SFFS is less than $2 \times N_c$.

Therefore, it is concluded that usually SFFS would search for an optimal solution for the target function $E(G_{col}^{k,1}, G_{col}^{k,2})$ within a limited number of loops.

D. The comparisons of runtime for different algorithms

This subsection compares the runtime of various algorithms on different data sets. Here SC-ECOC deploys SVR as the base learner. The experiment is carried out on a computer equipped with Intel Core i7-7700, 16 GB Memory, based on Windows 10 OS.

Table SVI-SVII listed the average runtime of different algorithms, and it is found that although SC-ECOC is not the fastest one, it doesn't require much longer runtime compared with other algorithms. When compared with GDBT and RF, its computational cost is much smaller.

Table SVIII-SIX compare the average runtime of different ECOC algorithms with SVR as the base learner. It is discovered that SC-ECOC requires the fewest runtime in most cases. It is accordant with the analysis of the time complexity. The main reason is that SC-ECOC demands only N_c-1 base learners. While other ECOC algorithms except for DC-ECOC uses more base learners, which leads to longer runtime when the training data set doesn't have a small sample size. Therefore, on the UCI data sets, our algorithm accomplishes a classification task faster than most of the algorithms. Although DC-ECOC is of the same ensemble scale as SC-ECOC, it needs to take extra effort on the class exchange process defined in its algorithm, which requires longer time on the UCI data set. While on the microarray data sets, the advantage of our algorithm is not so obvious due to the small training sample size. But in general, our algorithm doesn't need a lot of computation resources compared with other algorithms.

In short, our algorithm provides a solution for the multiclass problem with low computational requirements. So SC-ECOC is both effective and efficient.

E. The hypothesis tests on the results of various ECOC algorithms

To further compare the results, Friedman test and Nemenyi test [52] are used to test the performance of all the ECOC algorithms.

Friedman test is an improved statistical test, and it is based on the hypothesis that there is no significant difference in the overall distribution of multiple pairs of algorithms' mean ranks. Let K_j^i denote the rank of the j^{th} algorithm on the i^{th} data set, and K_j denote the mean rank of the j^{th} algorithm. The mean ranks for results on both UCI data sets and microarray data sets are calculated by Formula (1), where k is the number of algorithms to be compared, and D is the number of data sets deployed.

$$K_j = \frac{\sum_{i=1}^{D} R_j^i}{D}, \forall\, j \in [1, k] \tag{1}$$

Then the Friedman statistic is computed as follows:

$$\tau_{\chi^2} = \frac{12D}{k(k+1)}\left(\sum_{i=1}^{k} K_i^2 - \frac{k(k+1)^2}{4}\right) \tag{2}$$

As τ_{χ^2} was found to be too conservative, the improved version is given [53] by Formula (3):

$$\tau_F = \frac{(D-1)\tau_{\chi^2}}{D(k-1)-\tau_{\chi^2}} \tag{3}$$

To better understand the performance of SC-ECOC, the results of Friedman test and Nemenyi test are calculated on the UCI data sets and microarray data sets respectively. That is, D equals 20 when comparing results on the UCI data sets, and equals 10 for the microarray data sets. k equals 8 in both cases.

Table SX shows the Friedman test value for both UCI and microarray data sets. When α = 0.05, the critical value τ_F are 21.026 and 5.99 for the UCI and microarray data sets, respectively. As the corresponding results are all much higher than the critical value, the original hypothesis can be safely rejected. That is, there are significant differences among the results of different algorithms on both types of data sets.

Nemenyi test is calculated by Formula (4). It defines the critical difference (CD) value for the two methods that are significantly different with certain confidence (1-γ).

$$CD = q_\gamma \sqrt{\frac{k(k+1)}{6D}} \tag{4}$$

In Fig. S5, the mean rank of each algorithm is marked as a dot, and the horizontal bar across each dot shows the range of the Nemenyi value. As shown in Fig. , the average ranks of our algorithm range from 1 to 2. In Fig. (a)-(d), our algorithm always achieves the best average rank values. On the UCI data sets, the average ranks of our algorithm are close to 2 in both metrics. OVO achieves the second-best performance on the UCI data sets. Its average rank is close to 3. The other algorithms are much worse than our proposed algorithm or OVO.

On the other hand, SC-ECOC also achieves the best performance on the microarray data sets. The average ranks on the microarray data sets of the proposed algorithm change in the range [1, 2], which are both better than the corresponding results on the UCI data sets. As for other algorithms, their ranks change on different metrics, revealing that their performances are not balanced among classes. In short, we can safely conclude that our algorithm gains the highest rank on different data sets by different metrics.

F. The different performance with various base learners

This subsection compares the different performance of SC-ECOC obtained based on various types of base learners, including KNN, GDBT, RF, Gaussian Naive Bayes (GNB), MLP and SVR. Here base learners are all the regressors with default settings provided by the Scikit-learn toolbox.

Fig. S6 (a-b) illustrate that SC-ECOC achieves close performance with different base learners on the UCI data sets. When the base learners are very weak, such as GNB, the performance of SC-ECOC would be deteriorated. In general, with SVR and RF as base learners, SC-ECOC would perform better in the UCI data sets, while SVR and MLP tend to lead to high performance in the microarray data sets. Therefore, both the base learner and the data distribution affect the performance of our algorithm. Fig. S2 and Fig. S3 in the Supplementary Materials give the results of different base learners on the microarray data sets, and similar observations are obtained.

In addition, we compare the runtime of different algorithms in Table SVI-SIX, along with analysis in Subsection D in the Supplementary Materials. The comparisons show that our algorithm runs faster than most ECOC algorithms and is comparative to most of the elaborate algorithms.

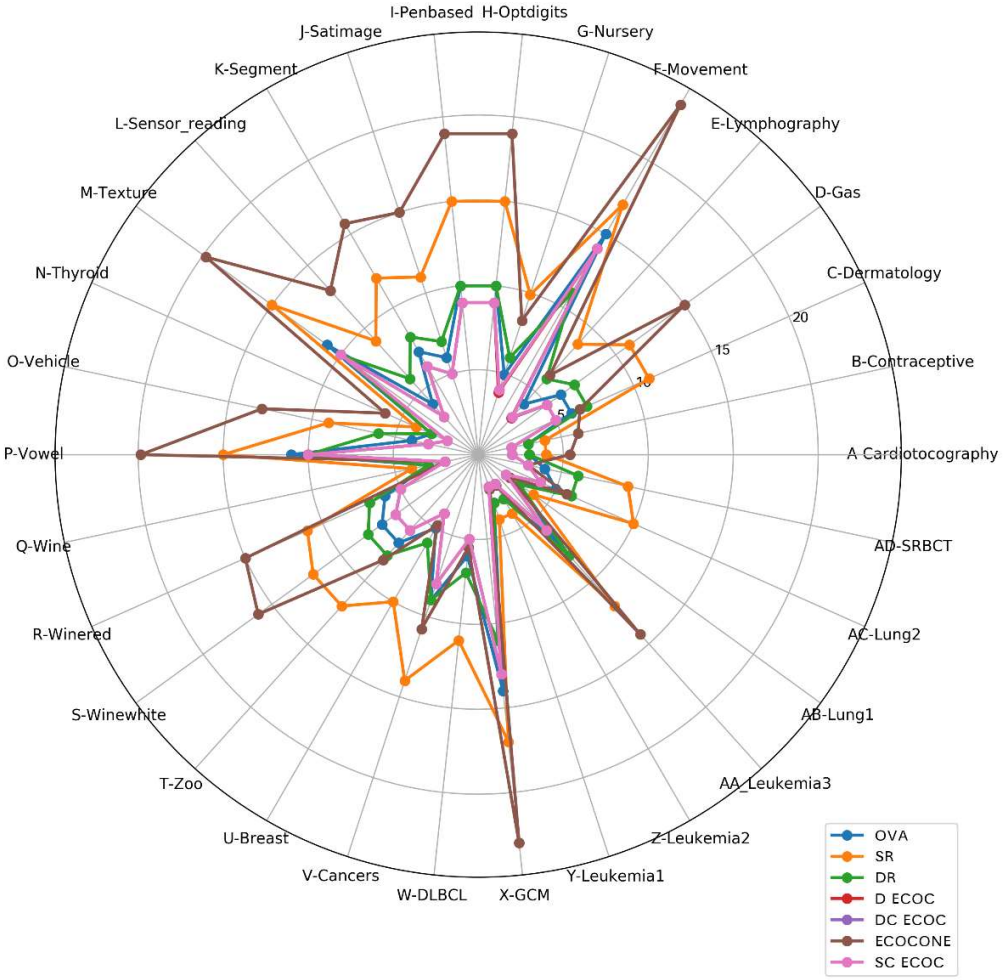


Fig. S1 The ensemble sizes of different algorithms.

Table SIII
The performance comparisons of different codewords.

	ACCURACY			F-SCORE		
	HSC	SC-BA	SC	HSC	SC-BA	SC
U	63.14±17.25	83.53±6.34	85.29±8.82	63.42±9.27	78.84±6.37	83.85±9.23
V	87.08±5.09	84.69±6.47	87.50±2.80	80.46±6.72	80.00±4.56	84.37±3.60
W	93.33±4.16	98.33±2.55	96.67±2.72	90.16±7.47	97.61±4.01	95.59±3.8
X	47.19±4.94	48.95±5.67	53.16±3.29	31.78±4.29	34.25±4.23	38.39±3.83
Y	92.89±5.14	95.33±3.06	96.67±3.33	90.20±8.01	94.32±4.68	97.50±2.51
Z	95.56±3.98	98.67±2.67	97.33±3.27	95.06±4.47	98.79±2.42	97.34±3.27
AA	93.13±3.07	92.73±1.32	93.64±2.23	91.09±4.30	89.99±0.78	92.87±3.42
AB	82.33±7.04	87.50±2.50	86.00±7.00	64.40±6.43	67.83±6.12	73.16±13.56
AC	95.61±2.39	93.66±1.62	96.34±2.25	92.91±4.29	86.71±3.57	94.77±3.50
AD	100.00±0.00	100.00±0.00	100.00±0.00	100.00±0.00	100.00±0.00	100.00±0.00
MEAN	85.03±5.31	88.34±3.22	89.26±3.57	79.95±5.53	82.83±3.67	85.78±4.67

Table SIV
The F1score indices of various algorithms on the UCI data sets.

ID	AdaBoost	GDBT	RF	Bagging	MLP	SVM	SC-ECOC
A	0.7959	0.9824	0.9678	0.7315	0.8784	0.6804	**0.9828**
B	0.5230	0.5560	0.4964	0.4709	0.5494	0.3969	0.5312
C	0.9056	0.9195	0.9505	0.8834	0.9505	0.9552	**0.9834**
D	0.9779	0.9883	**0.9925**	0.5592	0.8802	0.8838	0.9772
E	0.3169	0.6298	0.3739	0.2296	0.6151	0.3169	**0.8908**
F	0.6485	0.6154	0.7963	0.5772	0.6829	0.7277	**0.8084**
G	0.6739	0.7471	0.7608	0.4605	0.7594	0.6227	0.7228
H	0.9867	0.9731	0.9812	0.7498	0.9805	**0.9885**	0.9854
I	0.9931	0.9834	0.9897	0.8402	0.9873	0.9954	**0.9955**
J	**0.8911**	0.8774	0.8883	0.7803	0.7495	0.8599	0.8783
K	0.9251	0.9722	**0.9741**	0.7611	0.9405	0.8273	0.9413
L	0.8379	**0.9942**	0.9873	0.5490	0.8893	0.8555	0.8895
M	0.9888	0.9789	0.9772	0.7736	0.9906	0.9880	**0.9919**
N	0.9161	0.9340	0.9226	0.8814	0.4044	0.7503	**0.9607**
O	0.6521	0.7222	0.7519	0.3322	0.5861	0.4251	**0.7802**
P	0.8920	0.8746	0.9475	0.6396	0.8098	0.5172	**0.9631**
Q	0.7635	0.9460	0.9425	0.9770	0.1961	0.7755	**0.9978**
R	0.1900	0.3268	0.3533	0.3296	0.2485	0.1837	0.3288
S	0.2779	0.3276	**0.4188**	0.2655	0.2468	0.1027	0.3221
T	0.0967	0.6905	0.7520	0.8701	0.6786	0.0920	**0.9531**
Mean	0.7126	0.8020	0.8112	0.6331	0.7012	0.6472	**0.8442**
Rank	4.3	3.35	2.7	5.85	4.55	5.15	**2**

Table SV
The F1score indices of various algorithms on the microarray data sets.

ID	AdaBoost	GDBT	RF	Bagging	MLP	SVM	SC-ECOC
U	0.5657	0.6581	0.6381	0.4596	0.6558	0.4596	**0.8385**
V	0.7444	0.8031	0.8114	0.7814	0.7029	0.8056	**0.8437**
W	0.7905	0.5500	0.7905	0.6889	0.9346	0.8148	**0.9559**
X	0.3942	0.4889	0.5265	**0.6415**	0.4374	0.3974	0.5299
Y	0.9441	0.9188	0.9441	0.9441	0.8222	0.9441	**0.9750**
Z	0.9328	0.7000	0.9221	**1.0000**	0.9441	**1.0000**	**1.0000**
AA	0.6476	0.7884	0.7523	0.4742	0.7235	0.3277	**0.9287**
AB	0.6889	0.5048	0.5462	0.4630	0.4375	0.5462	**0.7316**
AC	0.8813	0.8734	0.6891	0.7539	0.8813	0.8420	**0.9477**
AD	0.9083	0.9053	**1.0000**	**1.0000**	**1.0000**	0.9222	**1.0000**
Mean	0.7498	0.7191	0.7620	0.7207	0.7539	0.7060	**0.8751**
Rank	4.4	4.8	3.5	4	4.2	4.1	**1.1**

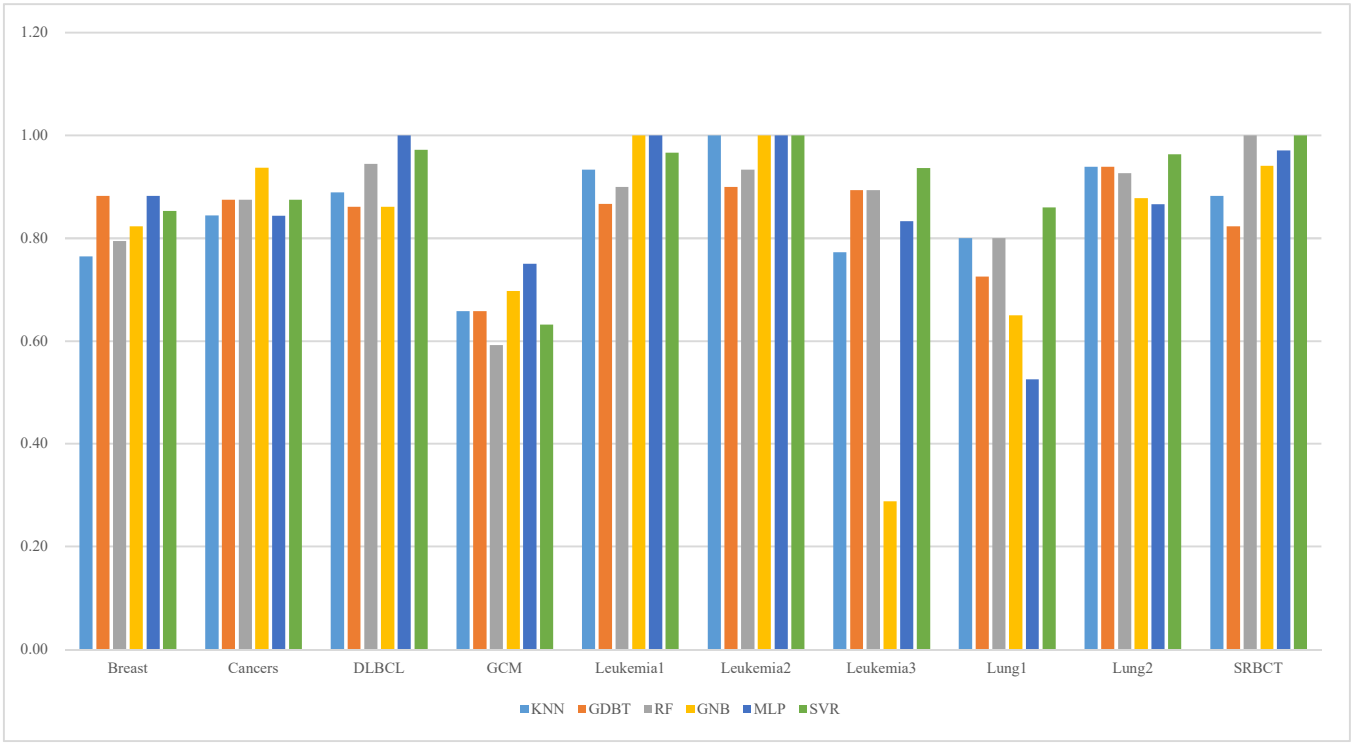


Fig. S2 The Accuracy comparison of SC-ECOC results based on various base classifiers on the microarray datasets.

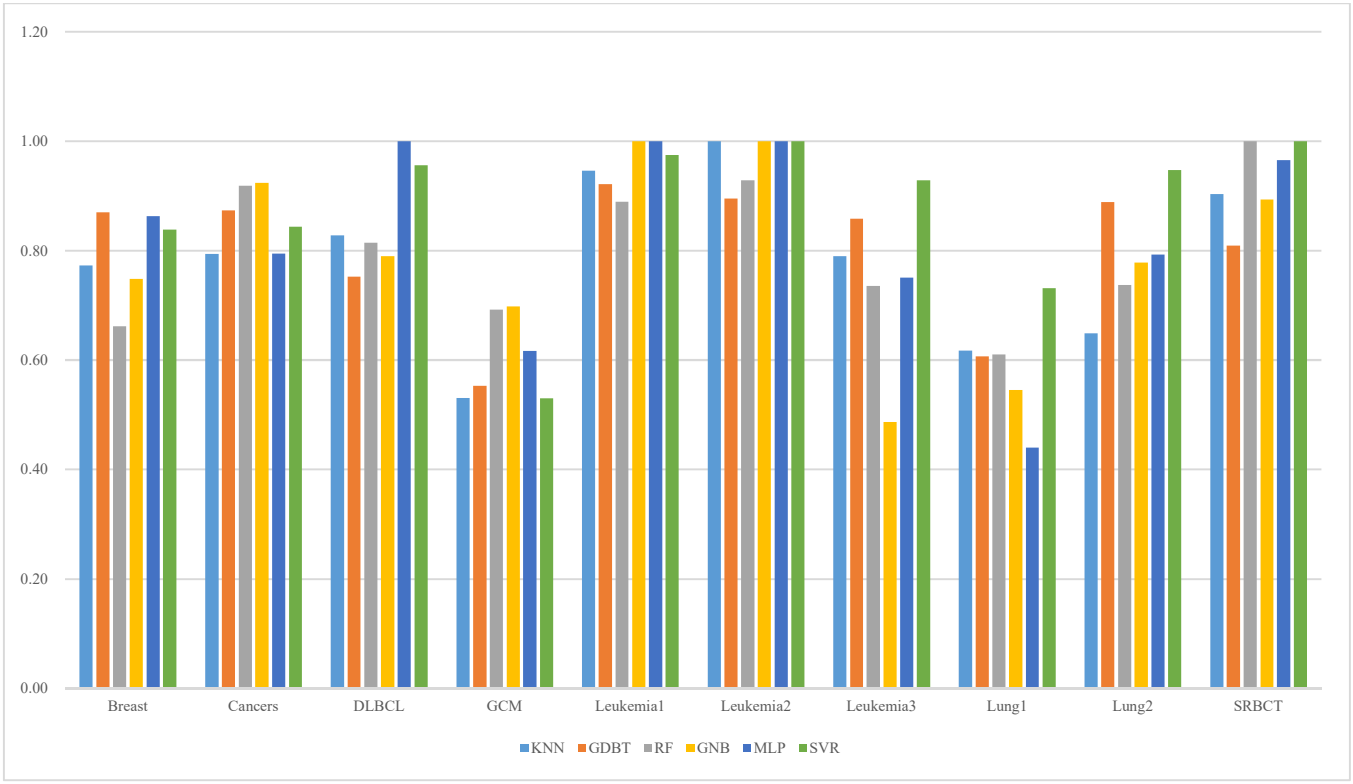


Fig. S3 The F1score comparison of SC-ECOC results based on various base classifiers on the microarray data sets.

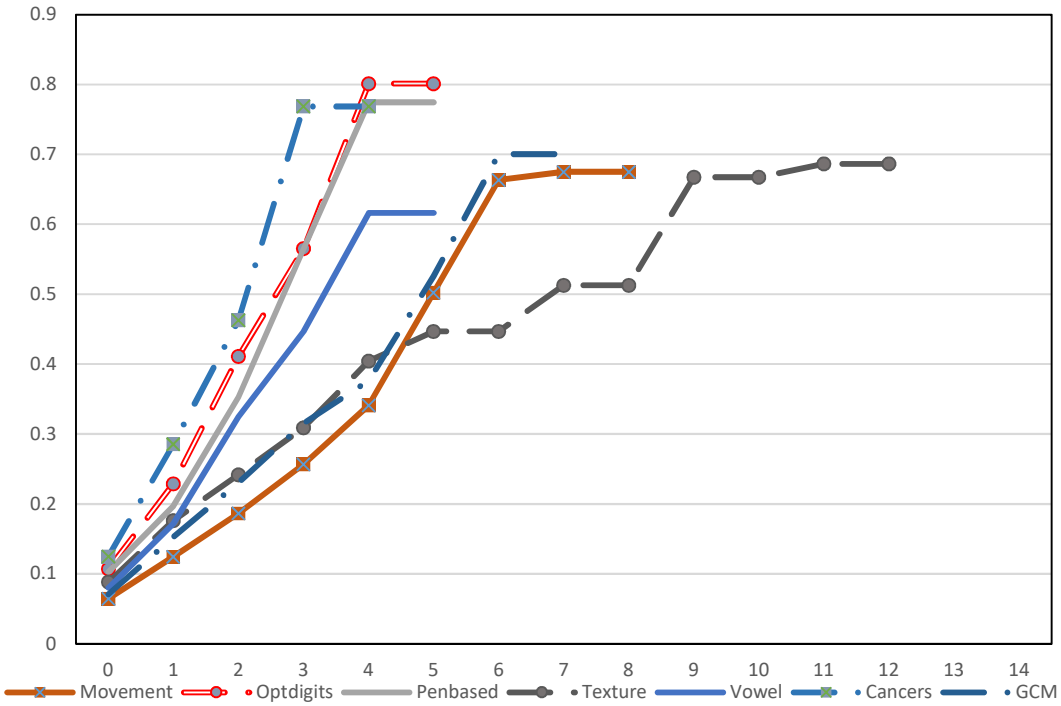


Fig. S4 The changes of the values of target function on different data sets.

Table SVI
The average runtime of various algorithms on the UCI data sets (in second).

ID	AdaBoost	GDBT	RF	Bagging	MLP	SVM	SC-ECOC
A	0.0189	0.8687	0.1755	0.0010	0.8128	0.0509	0.0309
B	0.0090	0.2862	0.1496	0.0010	0.5406	0.0419	0.0239
C	0.0060	0.4688	0.0977	0.0010	0.3131	0.0070	0.0119
D	0.4269	172.6657	5.3248	0.0279	2.1901	7.7934	4.8899
E	0.0020	0.1795	0.0898	0.0010	0.1297	0.0010	0.0030
F	0.0050	5.6369	0.1606	0.0010	0.3291	0.0130	0.0469
G	0.1326	2.4556	0.3650	0.0030	7.7972	0.6602	0.5884
H	0.4408	12.6345	0.4967	0.0060	2.6010	0.3251	0.1520
I	0.1526	12.9295	0.8198	0.0040	4.1449	0.2863	1.1788
J	0.1566	8.3987	0.7031	0.0030	2.3128	0.3191	0.4897
K	0.0239	3.3810	0.2404	0.0030	1.5568	0.0603	0.0020
L	0.1526	6.2065	0.5745	0.0070	3.5066	0.3910	0.2763
M	0.0987	27.6194	1.0053	0.0040	3.5435	0.2204	0.7521
N	0.0010	0.1496	0.0908	0.0010	0.0239	0.0010	0.0020
O	0.0060	0.5785	0.1416	0.0010	0.2773	0.0170	0.0180
P	0.0060	2.0386	0.1685	0.0020	0.6572	0.0259	0.0798
Q	0.0010	0.1925	0.0918	0.0010	0.1326	0.0010	0.0020
R	0.0090	1.1150	0.2095	0.0010	0.7430	0.0608	0.0718
S	0.0359	3.3171	0.5187	0.0020	0.8996	0.5226	0.5575
T	0.0010	0.2922	0.0888	0.0010	0.1147	0.0010	0.0030
Mean	0.0843	13.0707	0.5756	0.0036	1.6313	0.5399	0.4590

Table SVII
The average runtime of various algorithms on the microarray data sets (in second).

ID	AdaBoost	GDBT	RF	Bagging	MLP	SVM	SC-ECOC
U	0.0618	34.3919	0.1755	0.0379	1.1745	0.1127	0.0668
V	0.2244	176.1308	0.3971	0.1137	2.1393	0.4059	0.5108
W	0.0229	18.3457	0.1336	0.0120	0.3241	0.0529	0.0289
X	0.0060	7.5888	0.1446	0.0030	0.0808	0.0140	0.0010
Y	0.0010	0.1646	0.0878	0.0010	0.0209	0.0010	0.0010
Z	0.0010	0.1855	0.0888	0.0010	0.0209	0.0010	0.0010
AA	0.6951	124.4581	0.5058	0.2304	3.0279	1.6218	1.4930
AB	0.0539	17.7532	0.1716	0.0319	1.1998	0.0688	0.0289
AC	0.3102	132.5764	0.4807	0.1366	1.5060	0.4299	0.3680
AD	0.0120	6.4481	0.1157	0.0050	0.6249	0.0209	0.0136
Mean	0.1388	51.8043	0.2301	0.0572	1.0119	0.2729	0.2513

Table SVIII
The average runtime of various ECOC algorithms on the UCI data sets (in second).

ID	OVA	OVO	SR	DR	D-ECOC	DC-ECOC	ECOCONE	SC-ECOC
A	0.1407	0.1389	0.1806	0.1477	0.1039	0.1050	0.0399	0.0309
B	0.0952	0.0933	0.1237	0.0951	0.0643	0.0697	0.0296	0.0239
C	0.0504	0.1126	0.0891	0.0550	0.0399	0.0444	0.0090	0.0119
D	10.1283	12.3800	11.6773	10.7270	10.1904	10.3106	4.0908	4.8899
E	0.0148	0.0220	0.0460	0.0224	0.0111	0.0114	0.0013	0.0030
F	0.1161	0.7791	0.1302	0.0886	0.1120	0.1416	0.8993	0.0469
G	1.9169	3.0185	3.1183	2.1490	1.6460	1.6837	0.5294	0.5884
H	1.7102	5.4495	2.1777	1.7598	1.5628	1.6030	1.3215	0.1520
I	2.7181	9.6871	4.1712	2.9337	2.4842	2.4960	1.6048	1.1788
J	1.0733	2.1268	1.6604	1.2065	0.9498	0.9631	0.3361	0.4897
K	0.3503	0.9278	0.5557	0.3933	0.3068	0.2981	0.1070	0.0020
L	0.6876	0.8847	1.2087	0.9003	0.5830	0.5861	0.2080	0.2763
M	1.4085	5.9123	1.7967	1.3466	1.3834	1.3803	1.8963	0.7521
N	0.0146	0.0144	0.0222	0.0148	0.0100	0.0098	0.0025	0.0020
O	0.0723	0.1033	0.2875	0.1068	0.0576	0.0574	0.0234	0.0180
P	0.2187	1.0533	0.2991	0.1999	0.2160	0.2087	0.1881	0.0798
Q	0.0125	0.0122	0.0186	0.0126	0.0087	0.0089	0.0028	0.0020
R	0.1924	0.4552	0.3500	0.2250	0.1603	0.1610	0.0640	0.0718
S	0.7335	1.9646	1.2194	0.8380	0.6319	0.6398	0.3645	0.5575
T	0.0207	0.0563	0.0320	0.0225	0.0169	0.0182	0.0050	0.0030
MEAN	1.0838	2.2596	1.4582	1.1622	1.0270	1.0398	0.5862	0.4590

Table SIX
The average runtime of various algorithms on the microarray data sets (in second).

ID	OVA	OVO	SR	DR	D-ECOC	DC-ECOC	ECOCONE	SC-ECOC
U	0.0681	0.0825	0.0866	0.0695	0.1264	0.4773	0.0787	0.0668
V	0.2884	0.4628	0.2968	0.2951	1.0193	3.6645	1.6504	0.5108
W	0.0583	0.1030	0.0875	0.0693	0.1130	0.3507	0.0619	0.0289
X	0.1614	1.0364	0.1758	0.1220	0.2266	0.2874	1.9488	0.0010
Y	0.0143	0.0123	0.0174	0.0125	0.0088	0.0094	0.0018	0.0010
Z	0.0133	0.0130	0.0226	0.0205	0.0132	0.0124	0.0016	0.0010
AA	0.9112	1.0453	0.9299	0.9700	1.4593	2.3205	2.7785	1.4930
AB	0.0440	0.0403	0.0545	0.0446	0.0425	0.1349	0.0330	0.0289
AC	0.4393	0.4628	0.5337	0.4973	0.7160	1.5458	0.6277	0.3680
AD	0.0313	0.0418	0.1518	0.0451	0.0335	0.0979	0.0174	0.0136
MEAN	0.2029	0.3300	0.2357	0.2146	0.3759	0.8901	0.7200	0.2513

Table SX
The Friedman test results for various ECOC algorithms across all data sets.

	UCI data sets	microarray data sets
Accuracy	85.94	46.64
F-score	100.63	58.98

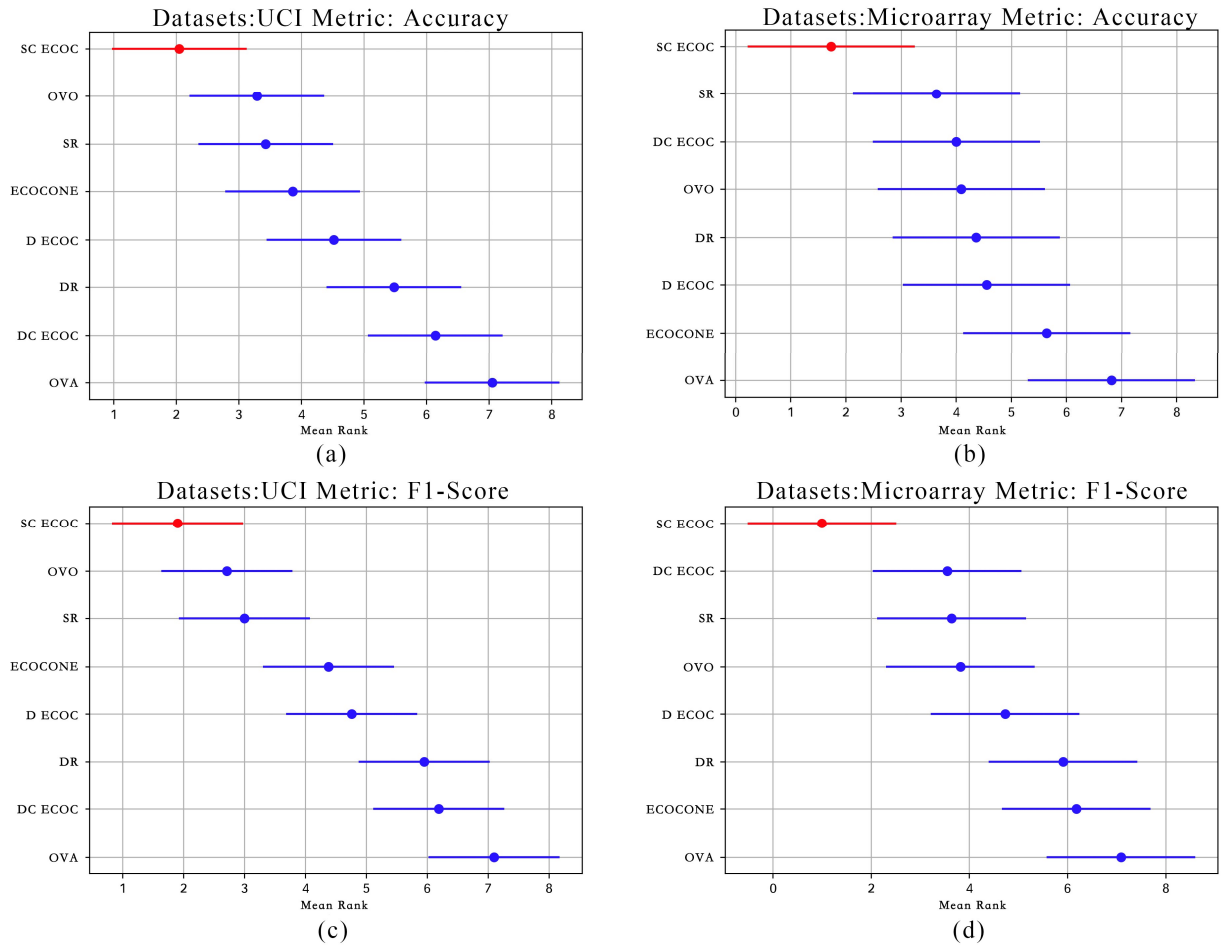
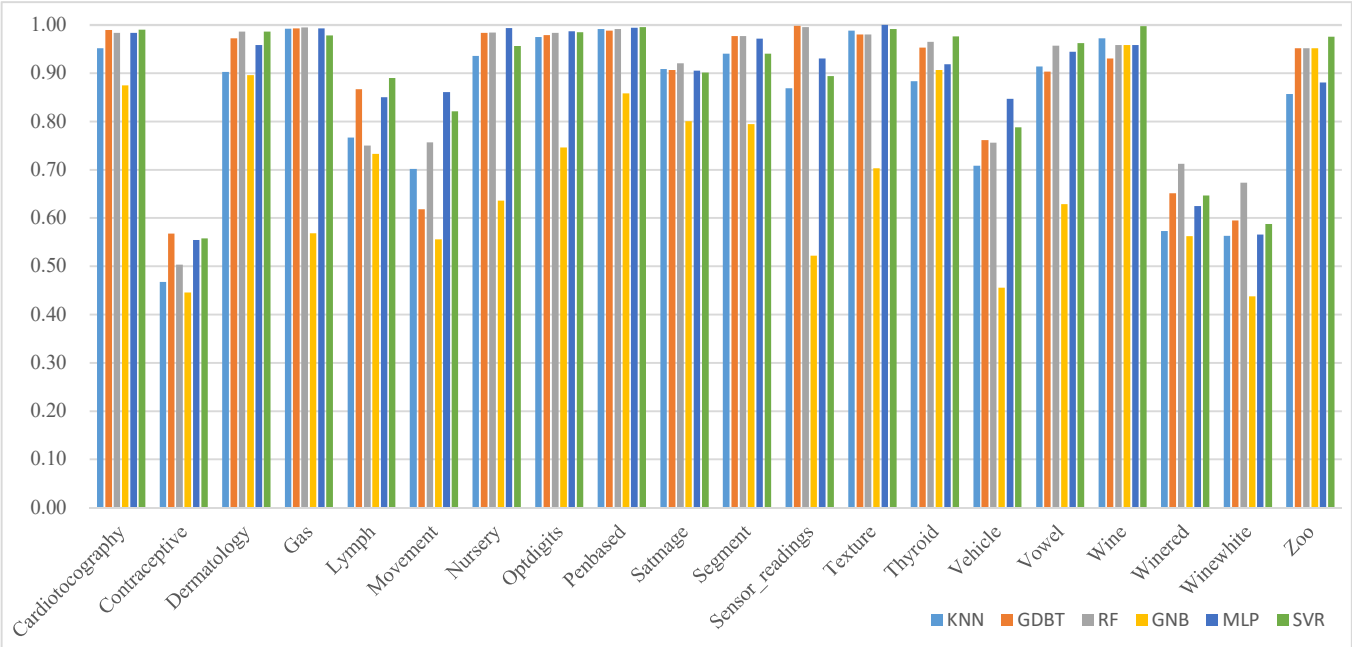
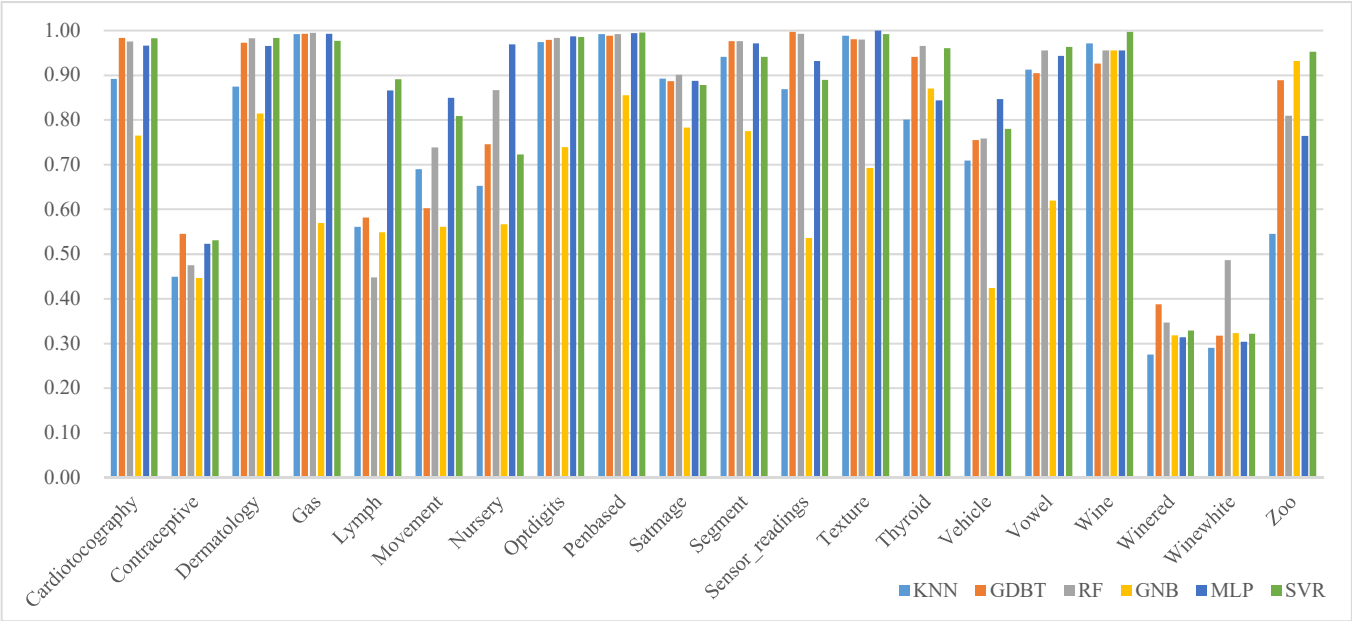


Fig. S5 The Nemenyi Test with different data sets and different metrics.



(a) The Accuracy results



(b) The F1score results

Fig. S6 The comparisons of SC-ECOC results based on various base learners on the UCI datasets.