Cell Reports Methods, Volume *3*

Supplemental information

A hybrid deep forest-based method for predicting synergistic drug combinations

Lianlian Wu, Jie Gao, Yixin Zhang, Binsheng Sui, Yuqi Wen, Qingqiang Wu, Kunhong Liu, Song He, and Xiaochen Bo

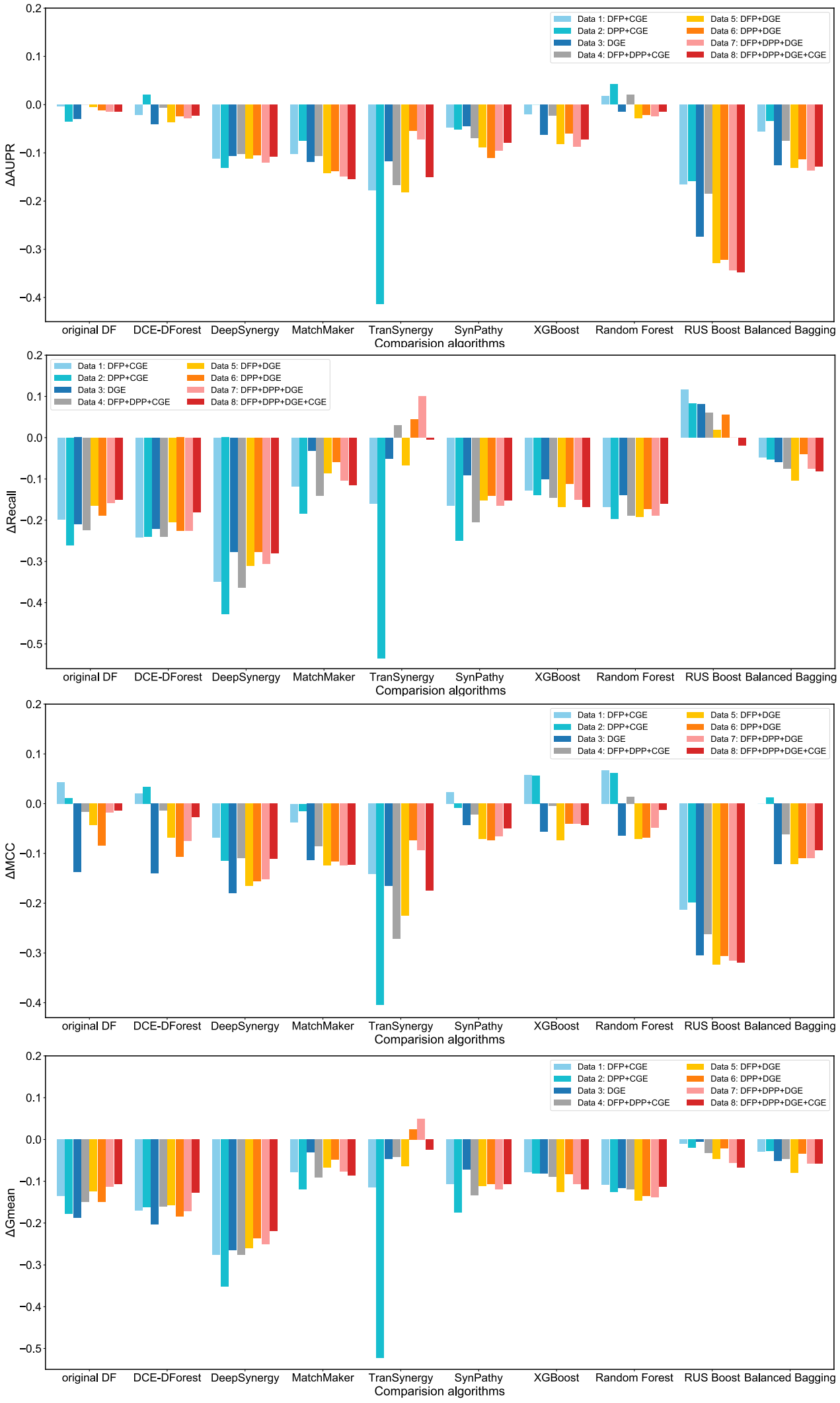

Figure S1. The performance of all algorithms under different datasets, related to Figure 2.

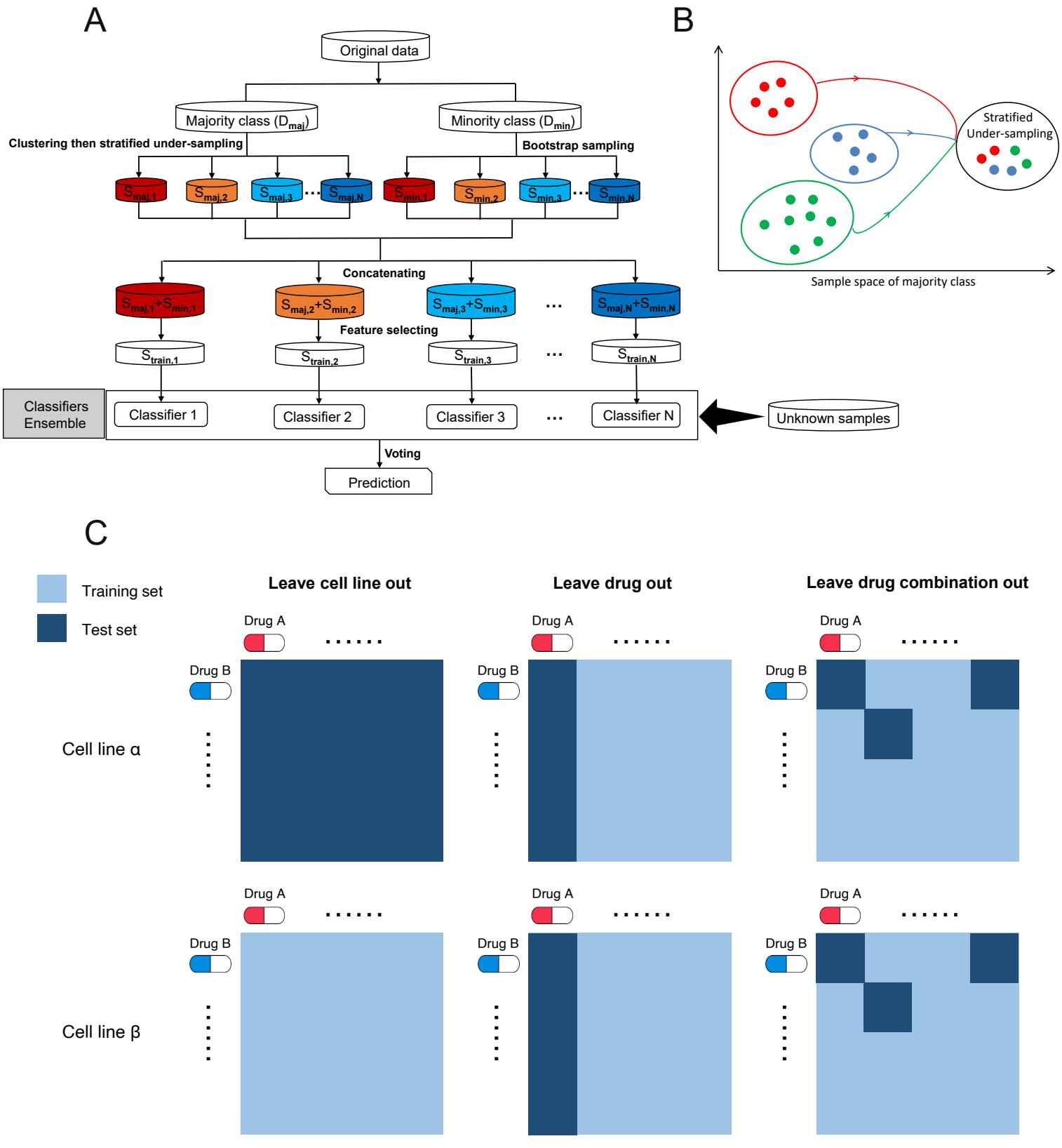

Figure S2. The diagram of ForSyn's random forest unit and three cross-validation strategies, related to STAR Methods.
(A) The framework of random forest unit based on clustering and stratified under-sampling.
(B) The diagram of stratified under-sampling based on clustering.
(C) The diagram of three cross-validation strategies, i.e., leave-cell-line-out, leave-drug-out and leave-drug-combination-out cross-validation.

Table S1. Performance comparison of all algorithms based on F1-score, related to Figure 2.

	ForSyn	Original DF	DCE-DForest	Deep Synergy	Match Maker	Tran Synergy	SynPathy	XGBoost	Random Forest	RUS Boost	Balanced Bagging	DeepDDS-GCN	DeepDDS-GAT
Data 1	0.499(3.5)	0.487 (6.0)	0.451 (8.0)	0.330 (10.0)	0.459 (7.0)	0.367 (9.0)	0.491 (5.0)	0.528 (1.0)	0.518 (2.0)	0.269 (11.0)	0.499 (3.5)	/	/
Data 2	0.496 (4.0)	0.451 (7.0)	0.473 (5.0)	0.262 (10.0)	0.466 (6.0)	0.076 (11.0)	0.445 (8.0)	0.533 (1.0)	0.514 (2.0)	0.282 (9.0)	0.510 (3.0)	/	/
Data 3	0.519 (1.0)	0.328 (8.0)	0.315 (9.0)	0.248 (10.0)	0.427 (4.0)	0.384 (7.0)	0.451 (2.0)	0.442 (3.0)	0.411 (6.0)	0.247 (11.0)	0.415 (5.0)	/	/
Data 4	0.529 (2.5)	0.493 (6.0)	0.486 (7.0)	0.357 (9.0)	0.470 (8.0)	0.278 (11.0)	0.500 (4.0)	0.534 (1.0)	0.529 (2.5)	0.283 (10.0)	0.497 (5.0)	/	/
Data 5	0.568 (1.0)	0.483 (2.0)	0.445 (8.0)	0.324 (10.0)	0.455 (5.5)	0.367 (9.0)	0.475 (3.0)	0.466 (4.0)	0.452 (7.0)	0.263 (11.0)	0.455 (5.5)	/	/
Data 6	0.551 (1.0)	0.419 (8.0)	0.381 (9.0)	0.320 (10.0)	0.447 (6.0)	0.494 (2.0)	0.452 (5.0)	0.485 (3.0)	0.436 (7.0)	0.264 (11.0)	0.456 (4.0)	/	/
Data 7	0.564 (1.0)	0.502 (2.0)	0.431 (9.0)	0.338 (10.0)	0.447 (8.0)	0.476 (4.0)	0.470 (5.0)	0.493 (3.0)	0.467 (6.0)	0.264 (11.0)	0.463 (7.0)	/	/
Data 8	0.572 (1.0)	0.509 (2.0)	0.488 (5.0)	0.383 (10.0)	0.444 (8.0)	0.395 (9.0)	0.490 (4.0)	0.486 (6.0)	0.506 (3.0)	0.257 (11.0)	0.472 (7.0)	/	/
Graph Data	/	/	/	/	/	/	/	/	/	/	/	0.474	0.426
Average Rank	(1.9)	(5.1)	(7.5)	(9.9)	(6.6)	(7.8)	(4.5)	(2.8)	(4.4)	(10.6)	(5.0)	/	/

Note: The performance of the first 11 algorithms is evaluated on Data 1-8 in tabular formats. Deep-GCN and Deep-GAT are the graph neural network-based algorithms and are tested on an additional dataset, Graph Data. The value in parentheses represents the ranking value of the corresponding performance. Taking Data 8 as an example, the ForSyn on this dataset has the best performance (0.572) and is assigned a ranking value of 1.0; the performance of Random Forest is the second (0.521) and is assigned a ranking value of 2.0; the performance of cascade forest and XGBoost are the same (0.515), they occupy the third and fourth positions respectively, so their ranking values are uniformly assigned 3.5 ((3.0+4.0)/2). The average rank of each algorithm is defined as the average of its ranks on all datasets.

Table S2. Performance comparison of all algorithms based on AUPR, related to Figure 2.

	ForSyn	Original DF	DCE-DForest	Deep Synergy	MatchMaker	TranSynergy	SynPathy	XGBoost	Random Forest	RUS Boost	Balanced Bagging	DeepDDS-GCN	DeepDDS-GAT
Data 1	0.566 (2.0)	0.562 (3.0)	0.545 (5.0)	0.454 (9.0)	0.465 (8.0)	0.389 (11.0)	0.518 (6.0)	0.546 (4.0)	0.583 (1.0)	0.401 (10.0)	0.511 (7.0)	/	/
Data 2	0.551 (3.5)	0.516 (6.0)	0.571 (2.0)	0.420 (9.0)	0.477 (8.0)	0.138 (11.0)	0.500 (7.0)	0.551 (3.5)	0.593 (1.0)	0.392 (10.0)	0.517 (5.0)	/	/
Data 3	0.521 (1.0)	0.492 (3.0)	0.481 (4.0)	0.415 (7.0)	0.402 (9.0)	0.404 (8.0)	0.477 (5.0)	0.459 (6.0)	0.506 (2.0)	0.247 (11.0)	0.395 (10.0)	/	/
Data 4	0.581 (2.5)	0.581 (2.5)	0.575 (4.0)	0.479 (8.0)	0.475 (9.0)	0.414 (10.0)	0.512 (6.0)	0.559 (5.0)	0.601 (1.0)	0.397 (11.0)	0.506 (7.0)	/	/
Data 5	0.579 (1.0)	0.574 (2.0)	0.543 (4.0)	0.468 (7.0)	0.438 (9.0)	0.397 (10.0)	0.491 (6.0)	0.498 (5.0)	0.551 (3.0)	0.251 (11.0)	0.448 (8.0)	/	/
Data 6	0.565 (1.0)	0.553 (2.0)	0.541 (4.0)	0.461 (7.0)	0.428 (10.0)	0.511 (5.0)	0.455 (8.0)	0.505 (6.0)	0.544 (3.0)	0.244 (11.0)	0.452 (9.0)	/	/
Data 7	0.589 (1.0)	0.575 (2.0)	0.561 (4.0)	0.469 (8.0)	0.440 (10.0)	0.517 (5.0)	0.494 (7.0)	0.502 (6.0)	0.565 (3.0)	0.246 (11.0)	0.452 (9.0)	/	/
Data 8	0.591 (1.0)	0.577 (2.0)	0.568 (4.0)	0.483 (7.0)	0.437 (10.0)	0.441 (9.0)	0.512 (6.0)	0.519 (5.0)	0.576 (3.0)	0.244 (11.0)	0.463 (8.0)	/	/
Graph Data	/	/	/	/	/	/	/	/	/	/	/	0.508	0.464
Average Rank	(1.6)	(2.8)	(3.9)	(7.8)	(9.1)	(8.6)	(6.4)	(5.1)	(2.1)	(10.8)	(7.9)	/	/

Table S3. Performance comparison of all algorithms based on Recall, related to Figure 2.

	ForSyn	Original DF	DCE-DForest	Deep Synergy	MatchMaker	TranSynergy	SynPathy	XGBoost	Random Forest	RUS Boost	Balanced Bagging	DeepDDS-GCN	DeepDDS-GAT
Data 1	0.559 (2.0)	0.361 (9.0)	0.317 (10.0)	0.210 (11.0)	0.442 (4.0)	0.400 (6.0)	0.395 (7.0)	0.432 (5.0)	0.392 (8.0)	0.676 (1.0)	0.511 (3.0)	/	/
Data 2	0.585 (2.0)	0.325 (9.0)	0.345 (7.0)	0.157 (10.0)	0.402 (5.0)	0.050 (11.0)	0.335 (8.0)	0.447 (4.0)	0.389 (6.0)	0.667 (1.0)	0.533 (3.0)	/	/
Data 3	0.425 (2.0)	0.215 (9.0)	0.205 (10.0)	0.149 (11.0)	0.394 (3.0)	0.375 (4.0)	0.335 (6.0)	0.325 (7.0)	0.287 (8.0)	0.506 (1.0)	0.367 (5.0)	/	/
Data 4	0.595 (3.0)	0.371 (9.0)	0.355 (10.0)	0.233 (11.0)	0.454 (5.0)	0.625 (2.0)	0.390 (8.0)	0.450 (6.0)	0.406 (7.0)	0.655 (1.0)	0.520 (4.0)	/	/
Data 5	0.517 (2.0)	0.352 (7.0)	0.312 (10.0)	0.207 (11.0)	0.432 (4.0)	0.450 (3.0)	0.365 (6.0)	0.350 (8.0)	0.325 (9.0)	0.535 (1.0)	0.414 (5.0)	/	/
Data 6	0.481 (3.0)	0.292 (9.0)	0.255 (10.0)	0.205 (11.0)	0.422 (5.0)	0.525 (2.0)	0.340 (7.0)	0.370 (6.0)	0.309 (8.0)	0.536 (1.0)	0.441 (4.0)	/	/
Data 7	0.525 (2.5)	0.367 (7.0)	0.300 (10.0)	0.220 (11.0)	0.422 (5.0)	0.625 (1.0)	0.360 (8.0)	0.375 (6.0)	0.336 (9.0)	0.525 (2.5)	0.451 (4.0)	/	/
Data 8	0.537 (1.0)	0.387 (6.0)	0.357 (10.0)	0.258 (11.0)	0.422 (5.0)	0.533 (2.0)	0.385 (7.0)	0.370 (9.0)	0.377 (8.0)	0.519 (3.0)	0.456 (4.0)	/	/
Graph Data	/	/	/	/	/	/	/	/	/	/	/	0.442	0.364
Average Rank	(2.2)	(8.1)	(9.6)	(10.9)	(4.5)	(3.9)	(7.1)	(6.4)	(7.9)	(1.4)	(4.0)	/	/

Table S4. Performance comparison of all algorithms based on MCC, related to Figure 2.

	ForSyn	Original DF	DCE-DForest	Deep Synergy	MatchMaker	TranSynergy	SynPathy	XGBoost	Random Forest	RUS Boost	Balanced Bagging	DeepDDS-GCN	DeepDDS-GAT
Data 1	0.464 (6.5)	0.507 (3.0)	0.485 (5.0)	0.395 (9.0)	0.427 (8.0)	0.323 (10.0)	0.488 (4.0)	0.522 (2.0)	0.531 (1.0)	0.251 (11.0)	0.464 (6.5)	/	/
Data 2	0.464 (6.0)	0.475 (5.0)	0.498 (3.0)	0.349 (9.0)	0.449 (8.0)	0.059 (11.0)	0.455 (7.0)	0.521 (2.0)	0.526 (1.0)	0.265 (10.0)	0.477 (4.0)	/	/
Data 3	0.509 (1.0)	0.372 (7.0)	0.369 (8.0)	0.329 (10.0)	0.396 (5.0)	0.344 (9.0)	0.466 (2.0)	0.453 (3.0)	0.445 (4.0)	0.204 (11.0)	0.388 (6.0)	/	/
Data 4	0.525 (2.0)	0.508 (5.0)	0.511 (4.0)	0.415 (9.0)	0.440 (8.0)	0.253 (11.0)	0.503 (6.0)	0.521 (3.0)	0.539 (1.0)	0.262 (10.0)	0.463 (7.0)	/	/
Data 5	0.548 (1.0)	0.505 (2.0)	0.479 (3.0)	0.382 (9.0)	0.424 (8.0)	0.323 (10.0)	0.477 (4.5)	0.474 (6.0)	0.477 (4.5)	0.225 (11.0)	0.427 (7.0)	/	/
Data 6	0.533 (1.0)	0.449 (6.0)	0.426 (7.0)	0.377 (10.0)	0.417 (9.0)	0.459 (4.5)	0.459 (4.5)	0.493 (2.0)	0.464 (3.0)	0.227 (11.0)	0.423 (8.0)	/	/
Data 7	0.541 (1.0)	0.523 (2.0)	0.466 (6.0)	0.389 (10.0)	0.417 (9.0)	0.447 (7.0)	0.475 (5.0)	0.500 (3.0)	0.492 (4.0)	0.225 (11.0)	0.431 (8.0)	/	/
Data 8	0.535 (1.0)	0.521 (3.0)	0.508 (4.0)	0.425 (8.0)	0.413 (9.0)	0.360 (10.0)	0.486 (6.0)	0.492 (5.0)	0.522 (2.0)	0.216 (11.0)	0.441 (7.0)	/	/
Graph Data	/	/	/	/	/	/	/	/	/	/	/	0.450	0.414
Average Rank	(2.4)	(4.1)	(5.0)	(9.3)	(8.0)	(9.1)	(4.9)	(3.3)	(2.6)	(10.8)	(6.7)	/	/

Table S5. Performance comparison of all algorithms based on Gmean, related to Figure 2.

	ForSyn	Original DF	DCE-DForest	Deep Synergy	MatchMaker	TranSynergy	SynPathy	XGBoost	Random Forest	RUS Boost	Balanced Bagging	DeepDDS-GCN	DeepDDS-GAT
Data 1	0.729 (1.0)	0.594 (9.0)	0.559 (10.0)	0.453 (11.0)	0.651 (4.5)	0.615 (8.0)	0.622 (6.0)	0.651 (4.5)	0.621 (7.0)	0.718 (2.0)	0.700 (3.0)	/	/
Data 2	0.743 (1.0)	0.565 (9.0)	0.581 (7.0)	0.391 (10.0)	0.623 (5.0)	0.221 (11.0)	0.568 (8.0)	0.661 (4.0)	0.618 (6.0)	0.724 (2.0)	0.715 (3.0)	/	/
Data 3	0.646 (1.0)	0.459 (9.0)	0.442 (10.0)	0.381 (11.0)	0.616 (3.0)	0.600 (4.0)	0.573 (6.0)	0.565 (7.0)	0.529 (8.0)	0.64 (2.0)	0.595 (5.0)	/	/
Data 4	0.752 (1.0)	0.602 (9.0)	0.591 (10.0)	0.477 (11.0)	0.660 (6.0)	0.710 (3.0)	0.619 (8.0)	0.663 (5.0)	0.632 (7.0)	0.719 (2.0)	0.705 (4.0)	/	/
Data 5	0.711 (1.0)	0.587 (7.0)	0.554 (10.0)	0.451 (11.0)	0.643 (4.0)	0.647 (3.0)	0.599 (6.0)	0.586 (8.0)	0.565 (9.0)	0.664 (2.0)	0.631 (5.0)	/	/
Data 6	0.685 (2.0)	0.536 (9.0)	0.501 (10.0)	0.448 (11.0)	0.636 (5.0)	0.710 (1.0)	0.578 (7.0)	0.602 (6.0)	0.549 (8.0)	0.664 (3.0)	0.651 (4.0)	/	/
Data 7	0.714 (2.0)	0.601 (7.0)	0.543 (10.0)	0.464 (11.0)	0.637 (5.0)	0.764 (1.0)	0.594 (8.0)	0.607 (6.0)	0.575 (9.0)	0.658 (3.0)	0.657 (4.0)	/	/
Data 8	0.722 (1.0)	0.616 (6.0)	0.594 (10.0)	0.503 (11.0)	0.636 (5.0)	0.697 (2.0)	0.615 (7.0)	0.603 (9.0)	0.609 (8.0)	0.654 (4.0)	0.665 (3.0)	/	/
Graph Data	/	/	/	/	/	/	/	/	/	/	/	0.649	0.586
Average Rank	(1.3)	(8.1)	(9.6)	(10.9)	(4.7)	(4.1)	(7.0)	(6.2)	(7.8)	(2.5)	(3.9)	/	/

Table S6. Critical values for the two-tailed Nemenyi test in this study, related to STAR Methods.

#Num of algorithms	2	3	4	5	6	7	8	9	10	11
$q_{0.05}$	1.960	2.343	2.569	2.728	2.850	2.949	3.031	3.102	3.164	3.219
$q_{0.10}$	1.645	2.052	2.291	2.459	2.589	2.693	2.780	2.855	2.920	2.978

Table S7. The performance of all models on the leave-cell-line-out, leave-drug-out, and leave-drug-combination-out cross validation, related to STAR Methods.

	F1-score	AUPR	Recall	MCC	Gmean
Leave-cell-line-out					
ForSyn	**0.361**	**0.454**	0.328	**0.311**	0.387
Original DF	0.201	0.398	0.181	0.165	0.243
DCE-DForest	0.198	0.435	0.134	0.165	0.254
DeepSynergy	0.107	Nan	0.091	0.072	0.160
Matchmaker	0.256	Nan	0.263	0.170	0.349
TranSynergy	0.334	0.447	0.310	0.181	0.352
SynPathy	0.141	Nan	0.092	0.135	0.211
XGBoost	0.135	0.354	0.091	0.139	0.177
Random Forest	0.163	0.385	0.114	0.119	0.215
RUS Boost	0.201	0.238	**0.355**	0.062	0.294
Balanced Bagging	0.265	0.263	0.265	0.135	0.366
DeepDDS-GCN	0.237	Nan	0.221	0.149	**0.394**
DeepDDS-GAT	0.128	Nan	0.137	0.082	0.295
Leave-drug-out					
ForSyn	**0.306**	**0.356**	0.247	**0.339**	0.467
Original DF	0.299	0.319	0.208	0.263	0.429
DCE-DForest	0.297	0.344	0.203	0.329	0.422
DeepSynergy	0.233	0.328	0.148	0.284	0.362
Matchmaker	0.301	0.315	0.246	0.264	0.471
TranSynergy	0.303	0.352	0.312	0.275	**0.513**
SynPathy	0.274	0.332	0.189	0.281	0.410
XGBoost	0.183	0.296	0.114	0.221	0.315
Random Forest	0.284	0.333	0.203	0.292	0.417
RUS Boost	0.196	0.151	**0.333**	0.092	0.491
Balanced Bagging	0.255	0.254	0.257	0.193	0.455
DeepDDS-GCN	0.252	0.340	0.230	0.215	0.447
DeepDDS-GAT	0.187	0.248	0.147	0.186	0.301
Leave-drug-combination-out					
ForSyn	**0.381**	**0.438**	0.332	**0.428**	0.559
Original DF	0.352	0.425	0.236	0.400	0.470
DCE-DForest	0.372	0.406	0.250	0.356	0.487
DeepSynergy	0.303	0.369	0.191	0.379	0.423
sMatchmaker	0.356	0.381	0.296	0.343	0.519
TranSynergy	**0.382**	0.432	0.337	0.390	**0.599**
SynPathy	0.367	0.409	0.253	0.411	0.494

XGBoost	0.308	0.374	0.206	0.341	0.448
Random Forest	0.362	0.419	0.238	0.425	0.477
RUS Boost	0.209	0.187	**0.431**	0.153	0.581
Balanced Bagging	0.332	0.346	0.328	0.299	0.552
DeepDDS-GCN	0.317	0.406	0.278	0.305	0.502
DeepDDS-GAT	0.231	0.319	0.190	0.250	0.358

Table S8. Ablation experiment of deep forest based on Data 8 and F1-score metric, related to STAR Methods.

Configuration	Description	F1-score
$DF^{(ADA*1+BAG*1+GBC*1+RF\text{-}CUS*1+ETF\text{-}DR*1)}$	Five different units are placed on each cascade layer.	0.562
$DF^{(ADA*1+BAG*1+GBC*1+RF\text{-}CUS*1)}$	Remove ETF-DR unit.	0.532
$DF^{(ADA*1+BAG*1+GBC*1+ETF\text{-}DR*1)}$	Remove RF-CUS unit.	0.537
$DF^{(ADA*1+BAG*1+RF\text{-}CUS*1+ETF\text{-}DR*1)}$	Remove GBC unit.	0.553
$DF^{(ADA*1+GBC*1+RF\text{-}CUS*1+ETF\text{-}DR*1)}$	Remove BAG unit.	0.558
$DF^{(BAG*1+GBC*1+RF\text{-}CUS*1+ETF\text{-}DR*1)}$	Remove ADA unit.	0.554

Table S9. The top eight synergistic drug combinations predicted by ForSyn, related to Figure 3.

Drug A	Drug B	Cell Line	Verified
Erlotinib Hydrochloride	Azd1775	HT29	√
Erlotinib Hydrochloride	Mk-5108	HT29	√
Erlotinib Hydrochloride	Mk-2206	HT29	
Etoposide	Gefitinib	HT29	√
Erlotinib Hydrochloride	Dinaciclib	HT29	√
Erlotinib Hydrochloride	Pd325901	HT29	
Erlotinib Hydrochloride	Bez-235	HT29	
Azd1775	Vemurafenib	SW620	

Table S10. The genes involved in the top contributing DGE features in four cell lines, related to Figure 4.

Index	A549 cell line	HT29 cell line	MCF7 cell line	PC3 cell line
1	CCND3	PLOD3	PGM1	UFM1
2	TSPAN4	CAMSAP2	SPRED2	SIRT3
3	CLPX	INSIG1	EIF4G1	RPA1
4	PRUNE1	GNA15	CEMIP2	NSDHL
5	ELAVL1	SKIV2L	RALA	AKAP8L
6	LPAR2	MINDY1	HIF1A	GRWD1
7	HK1	PCBD1	TPM1	NCK1
8	GNA11	FOXO3	BNIP3L	COG4
9	NT5DC2	EPHA3	FKBP14	ABHD4
10	PYGL	MTF2	GOLT1B	ATF5

Data S1. The algorithm of random forest by clustering and stratified under-sampling, related to STAR Methods.

Input:

D: a binary training set, which is composed of the majority class (D_{maj}) and the minority class (D_{min}), $D = D_{maj} \cup D_{min}$

d: the feature dimension

T: the number of decision trees in the Random Forest

Output:

The ensemble of decision trees.

Algorithm process:

1:**For** t=1,2…T **Do**

2: Step 1 Perform AP clustering on the majority class (D_{maj}).

3: Step 2 Perform stratified sampling on the clustering result of Step 1, to obtain the dataset S_{maj}, where $|S_{maj}|=|D_{min}|$.

4: Step 3 Perform bootstrap sampling on the minority class (D_{min}) to obtain the dataset S_{min}, where $|S_{min}|=|D_{min}|$.

5: Step 4 $D_{train} = S_{maj} \cup S_{min}$.

6: Step 5 Randomly select $\sqrt{d}$ features in D_{train}, to get the dataset S_{train}.

7: Step 6 Use S_{train} to train a decision tree.

8:**End For**

Data S2. The algorithm of extreme tree forest based on data complexity dimension reduction, related to STAR Methods.

Input:

D: the overall dataset

d: the feature dimension

T: the training set

V: the validation set //D=T ∪ V

R: the number of iteration //R = 5 in the experiment

s: the step size of the greedy algorithm

size: the feature size after dimension reduction

ETF: an extreme tree forest

Output:

Feature subset after dimension reduction, and a trained ETF.

Algorithm process:

1: Use data complexity metric to calculate the overlap area (F_i) of each feature in D. // Refer to Eq. (1)

2: Sort all the features in D by ascending order according to F_i.

3: Divide D into R subsets evenly, $D_r \subset D$, $r \in [1,R]$.

4: Initialize Size=0.

5: **For** r in Range R

6: T=D\ D_r //(R-1) subsets as the training set

7: V=D_r //One subset as the validation set

8: ACC=Accuracy(ETF, T, V)

// Use T to train the extreme tree forest, then use V to verify its performance

9: Initialize BestSubset=∅, ACC_{max}= ACC

10: **While** T≠∅ **Do**

11: T = T[: , : (d-s)] // Training set removes the last s dimension

12: V= V[: , : (d-s)] // Validation set removes the last s dimension

13: ACC_{temp}= Accuracy(ETF, T, V)

14: **If** $ACC_{temp} > ACC_{max}$ **Then**

15: $ACC_{max}= ACC_{temp}$

16: BestSubset=T

17: **Else Break**

18: **End While**

19: size=size+|BestSubset| // |*| represents the dimension of the set

20: **End For**

21: size= size/R

22: **Return** D[: , : size] // Return the feature subset after dimension reduction

23: Using D[: , : size] to train a ETF as the unit of the deep forest.