Supporting information for: Drug repurposing using deep embeddings of gene expression profiles

Yoni Donner,[*,†] Stéphane Kazmierczak,[*,‡] and Kristen Fortney[*,‡]

†*Quantified Mind*

‡*BioAge Labs*

E-mail: yonidonner@gmail.com; stephane@bioagelabs.com; kristen@bioagelabs.com

Supporting Figures and Tables

ROC curves

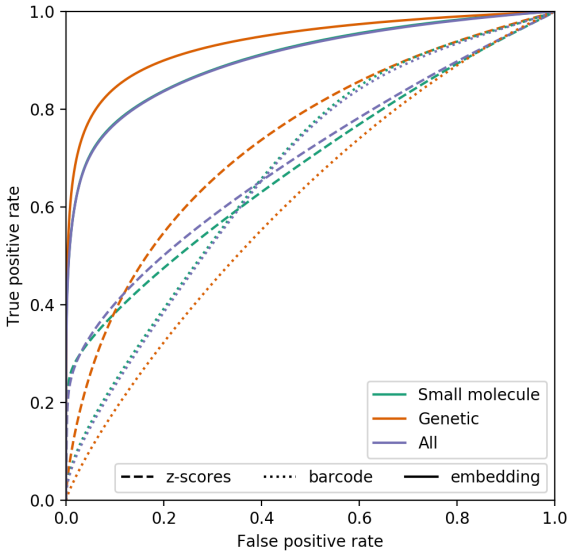


Figure S1: ROC curves of embedding and baselines on queries for profiles from the same perturbagen, by perturbagen group.

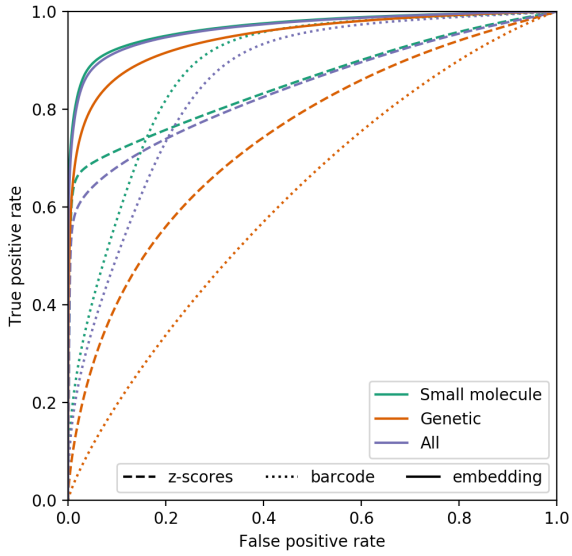


Figure S2: ROC curves of embedding and baselines on queries for profiles from the same set of biological replicates, by perturbagen group.

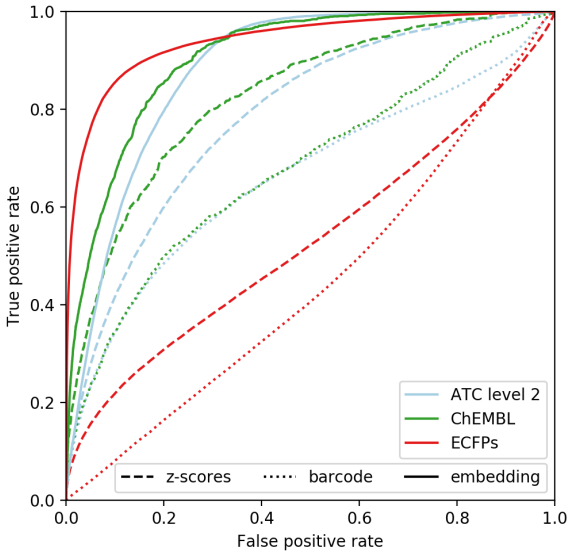


Figure S3: ROC curves for embedding and baselines on queries of similar therapeutic targets, protein targets and molecular structure.

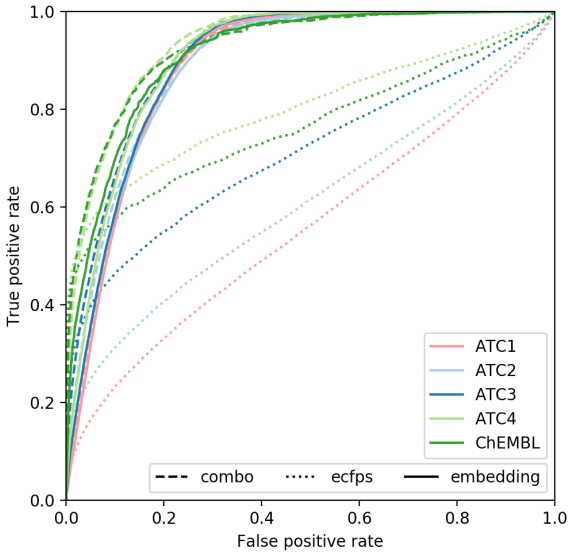


Figure S4: ROC curves for structural similarity (ECFPs Tanimoto coefficient), embedding and a combination of both on functional similarity queries based on ATC levels 1–4 and ChEMBL protein targets.

Results using Euclidean distance for z-scores

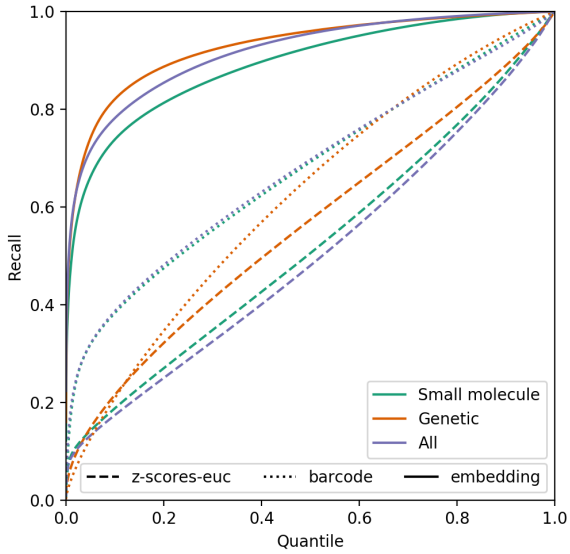


Figure S5: Recall by quantile for embedding and baselines on queries for profiles from the same perturbagen, by perturbagen group. Results for z-scores shown with Euclidean distance.

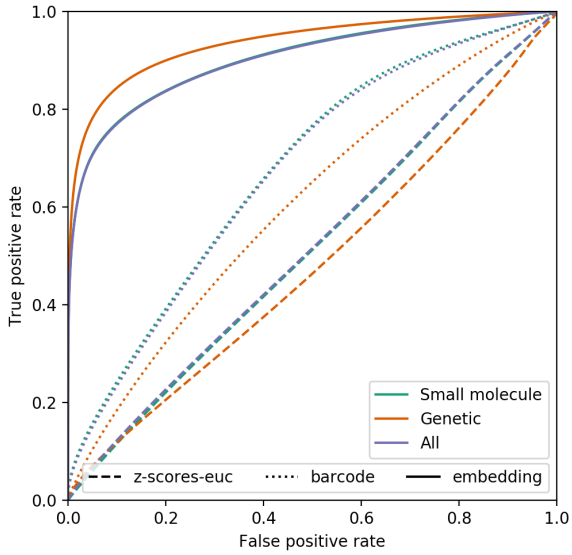


Figure S6: ROC curves for embedding and baselines on queries for profiles from the same set of biological replicates, by perturbagen group. Results for z-scores shown with Euclidean distance.

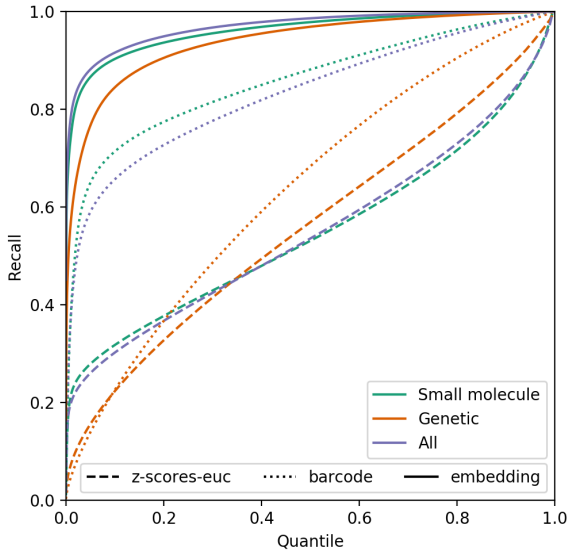


Figure S7: Recall by quantile for embedding and baselines on queries for profiles from the same set of biological replicates, by by perturbagen group. Results for z-scores shown with Euclidean distance.

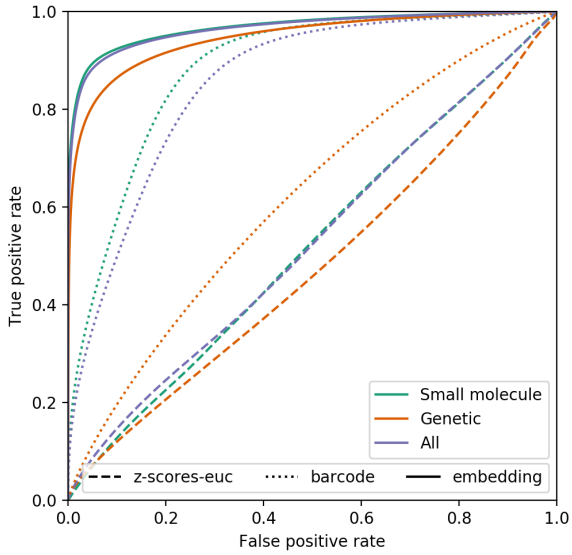
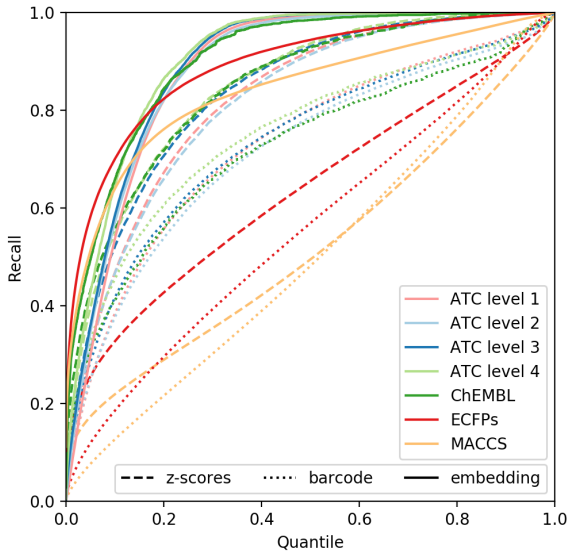


Figure S8: ROC curves for embedding and baselines on queries for profiles from the same set of biological replicates, by perturbagen group. Results for z-scores shown with Euclidean distance.

External evaluation results for all ATC levels and MACCS



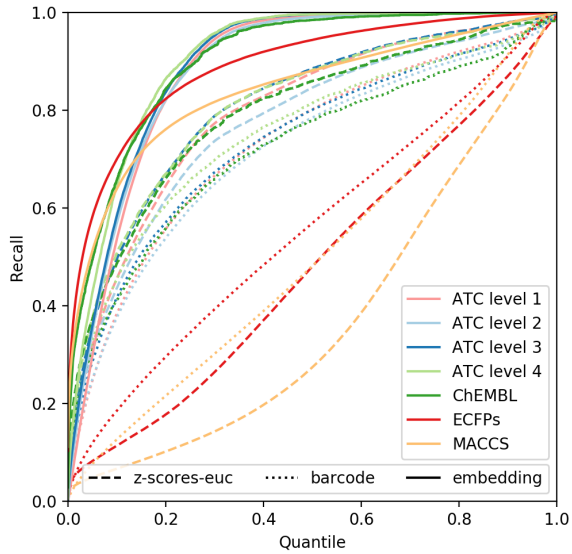


Figure S9: Recall by quantile for embedding and baselines on queries of similar therapeutic targets (ATC levels 1–4), protein targets (ChEMBL) and molecular structure (defined by ECFPs and MACCS). For z-scores, cosine (left) or Euclidean (right) distance was used.

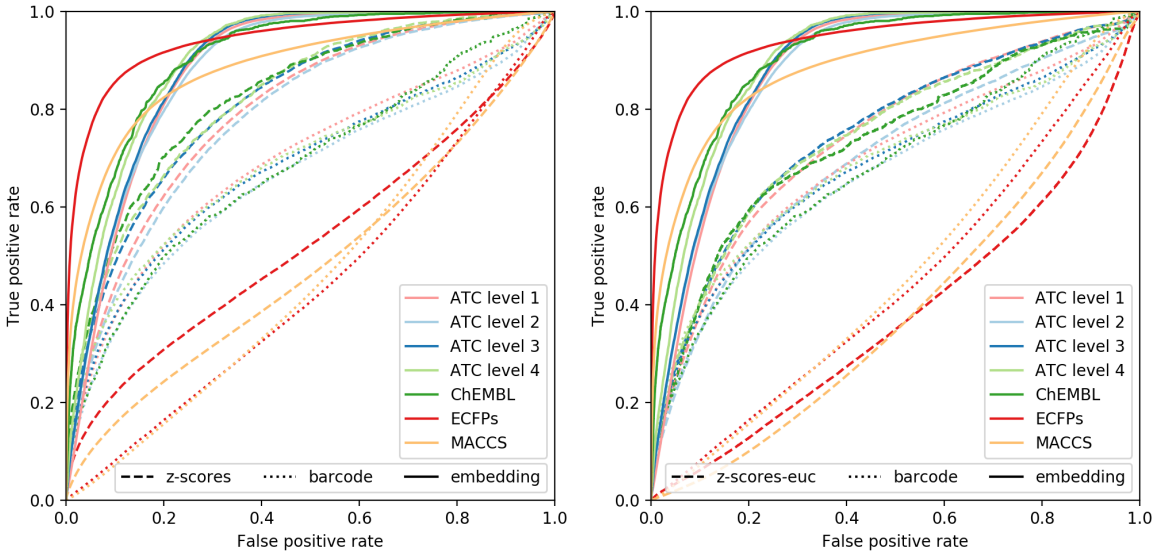


Figure S10: ROC curves for embedding and baselines on queries of similar therapeutic targets (ATC levels 1–4), protein targets (ChEMBL) and molecular structure (defined by ECFPs and MACCS). For z-scores, cosine (left) or Euclidean (right) distance was used.

Table S1: Performance of embedding and baselines on queries of similar therapeutic targets, protein targets and molecular structure. 5-class κ is not available for ECFPs and MACCS, because these targets do not define fixed classes that can be used to evaluate κ (Methods).

Query type	Method	Median	Top-0.001	Top-0.01	Top-0.1	AUC	MCC	κ (5-class
ATC level 1	z-scores	0.1126	0.015	0.092	0.468	0.795	0.421	0.174
	z-scores-euc	0.1201	0.019	0.105	0.456	0.740	0.330	0.013
	Barcode	0.1614	0.013	0.076	0.384	0.694	0.252	0.079
	Embedding	0.0914	0.006	0.064	0.536	0.881	0.568	0.257
ATC level 2	z-scores	0.1142	0.022	0.103	0.466	0.785	0.412	0.268
	z-scores-euc	0.1311	0.022	0.109	0.436	0.698	0.266	0.022
	Barcode	0.1712	0.018	0.083	0.376	0.662	0.208	0.120
	Embedding	0.0824	0.013	0.096	0.570	0.882	0.563	0.351
ATC level 3	z-scores	0.0917	0.034	0.132	0.521	0.813	0.438	0.204
	z-scores-euc	0.1017	0.032	0.136	0.496	0.749	0.333	-0.015
	Barcode	0.1456	0.029	0.104	0.417	0.683	0.228	0.097
	Embedding	0.0796	0.022	0.114	0.585	0.889	0.573	0.321
ATC level 4	z-scores	0.0814	0.052	0.165	0.545	0.820	0.446	0.426
	z-scores-euc	0.0971	0.047	0.158	0.508	0.744	0.325	0.034
	Barcode	0.1226	0.042	0.131	0.455	0.687	0.235	0.167
	Embedding	0.0656	0.036	0.163	0.648	0.901	0.573	0.627
ChEMBL target	z-scores	0.0733	0.101	0.218	0.558	0.826	0.438	0.484
	z-scores-euc	0.1062	0.086	0.195	0.487	0.733	0.290	0.246
	Barcode	0.1555	0.069	0.153	0.411	0.680	0.215	0.211
	Embedding	0.0513	0.090	0.288	0.658	0.906	0.552	0.620
Structure (ECFPs)	z-scores	0.2888	0.073	0.149	0.327	0.530	0.025	
	z-scores-euc	0.5183	0.021	0.048	0.114	0.380	-0.167	
	Barcode	0.4196	0.012	0.044	0.185	0.443	-0.100	
	Embedding	0.0296	0.181	0.360	0.700	0.940	0.753	
Structure (MACCS)	z-scores	0.5069	0.056	0.107	0.219	0.482	-0.046	
	z-scores-euc	0.6776	0.015	0.029	0.067	0.389	-0.155	
	Barcode	0.5195	0.006	0.025	0.127	0.463	-0.075	
	Embedding	0.0462	0.181	0.316	0.639	0.884	0.627	

External evaluation results by cell line, incubation time and dose

Table S2: Embedding performance on queries of similar therapeutic targets, protein targets and molecular structure by incubation time.

Query type	Incubation time	Median	Top-0.001	Top-0.01	Top-0.1	AUC	MCC
ATC level 1	6h	0.0901	0.007	0.058	0.539	0.870	0.556
	24h	0.0989	0.006	0.069	0.504	0.871	0.559
ATC level 2	6h	0.0818	0.015	0.086	0.572	0.868	0.544
	24h	0.0894	0.012	0.096	0.538	0.873	0.557
ATC level 3	6h	0.0764	0.026	0.115	0.591	0.879	0.560
	24h	0.0870	0.018	0.106	0.548	0.879	0.564
ATC level 4	6h	0.0602	0.040	0.163	0.666	0.896	0.561
	24h	0.0703	0.029	0.154	0.609	0.891	0.564
ChEMBL	6h	0.0481	0.105	0.282	0.657	0.900	0.543
	24h	0.0515	0.078	0.262	0.647	0.901	0.557
Structure (ECFPs)	6h	0.0410	0.101	0.295	0.662	0.927	0.711
	24h	0.0372	0.169	0.327	0.674	0.930	0.728
Structure (MACCS)	6h	0.0789	0.118	0.230	0.546	0.812	0.478
	24h	0.0485	0.162	0.306	0.630	0.879	0.616

Table S3: Embedding performance on queries of similar therapeutic targets, protein targets and molecular structure by dose.

Query type	Dose	Median	Top-0.001	Top-0.01	Top-0.1	AUC	MCC
ATC level 1	$< 1\mu M$	0.3688	0.001	0.017	0.177	0.584	0.109
	$(1\mu M, 10\mu M)$	0.0597	0.012	0.111	0.665	0.883	0.510
	$\geq 10\mu M$	0.1868	0.003	0.030	0.284	0.746	0.370
ATC level 2	$< 1\mu M$	0.3586	0.002	0.020	0.186	0.588	0.119
	$(1\mu M, 10\mu M)$	0.0604	0.016	0.124	0.662	0.886	0.519
	$\geq 10\mu M$	0.1660	0.008	0.054	0.334	0.751	0.360
ATC level 3	$< 1\mu M$	0.3414	0.003	0.024	0.202	0.595	0.128
	$(1\mu M, 10\mu M)$	0.0561	0.019	0.143	0.673	0.889	0.517
	$\geq 10\mu M$	0.1576	0.013	0.074	0.352	0.766	0.389
ATC level 4	$< 1\mu M$	0.2697	0.006	0.038	0.261	0.626	0.165
	$(1\mu M, 10\mu M)$	0.0433	0.038	0.210	0.696	0.893	0.523
	$\geq 10\mu M$	0.1247	0.018	0.110	0.428	0.799	0.435
ChEMBL	$< 1\mu M$	0.2110	0.053	0.110	0.399	0.686	0.237
	$(1\mu M, 10\mu M)$	0.0280	0.141	0.336	0.795	0.927	0.550
	$\geq 10\mu M$	0.1056	0.066	0.208	0.490	0.808	0.421
Structure (ECFPs)	$< 1\mu M$	0.0628	0.050	0.159	0.654	0.896	0.722
	$(1\mu M, 10\mu M)$	0.0344	0.191	0.347	0.674	0.958	0.784
	$\geq 10\mu M$	0.0203	0.172	0.407	0.702	0.853	0.536
Structure (MACCS)	$< 1\mu M$	0.0322	0.051	0.205	0.808	0.878	0.650
	$(1\mu M, 10\mu M)$	0.0528	0.217	0.333	0.600	0.915	0.679
	$\geq 10\mu M$	0.0640	0.119	0.288	0.550	0.743	0.340

Table S4: Embedding performance on queries of similar therapeutic targets and protein targets by cell line. A375: human malignant melanoma; A549: human non-small cell lung carcinoma; HA1E: human kidney epithelial immortalized; HT29: human colorectal adenocarcinoma; MCF7: human breast adenocarcinoma; PC3: human prostate adenocarcinoma; VCAP: human metastatic prostate cancer.

Query type	Cell line	Median	Top-0.001	Top-0.01	Top-0.1	AUC	MCC
ATC level 1	A375	0.1718	0.004	0.038	0.316	0.798	0.501
	A549	0.1192	0.008	0.064	0.444	0.838	0.540
	HA1E	0.3587	0.002	0.017	0.164	0.605	0.166
	HT29	0.1845	0.003	0.032	0.299	0.787	0.494
	MCF7	0.1454	0.003	0.036	0.367	0.816	0.514
	PC3	0.1571	0.004	0.036	0.339	0.805	0.498
	VCAP	0.1051	0.010	0.064	0.481	0.861	0.550
ATC level 2	A375	0.1692	0.008	0.057	0.335	0.790	0.478
	A549	0.1077	0.016	0.092	0.477	0.833	0.529
	HA1E	0.3544	0.004	0.033	0.188	0.598	0.146
	HT29	0.1834	0.005	0.040	0.315	0.776	0.469
	MCF7	0.1290	0.007	0.058	0.419	0.820	0.500
	PC3	0.1425	0.011	0.063	0.386	0.809	0.486
	VCAP	0.1021	0.016	0.082	0.493	0.855	0.542
ATC level 3	A375	0.1525	0.014	0.084	0.376	0.809	0.507
	A549	0.0998	0.023	0.126	0.501	0.846	0.545
	HA1E	0.3184	0.008	0.058	0.236	0.631	0.183
	HT29	0.1651	0.008	0.049	0.355	0.795	0.498
	MCF7	0.1263	0.009	0.066	0.425	0.825	0.521
	PC3	0.1384	0.019	0.081	0.399	0.818	0.508
	VCAP	0.0916	0.021	0.099	0.531	0.868	0.551
ATC level 4	A375	0.1313	0.023	0.120	0.432	0.822	0.503
	A549	0.0766	0.034	0.183	0.576	0.868	0.558
	HA1E	0.2617	0.013	0.092	0.317	0.668	0.231
	HT29	0.1432	0.015	0.075	0.409	0.809	0.501
	MCF7	0.1050	0.016	0.104	0.486	0.842	0.528
	PC3	0.1123	0.025	0.115	0.468	0.837	0.516
	VCAP	0.0729	0.027	0.124	0.604	0.882	0.553
ChEMBL	A375	0.0889	0.066	0.240	0.524	0.849	0.485
	A549	0.0351	0.119	0.322	0.670	0.895	0.540
	HA1E	0.1663	0.057	0.208	0.421	0.739	0.313
	HT29	0.1100	0.053	0.150	0.481	0.838	0.496
	MCF7	0.0923	0.060	0.172	0.520	0.852	0.512
	PC3	0.0945	0.050	0.203	0.510	0.838	0.479
	VCAP	0.0778	0.046	0.151	0.581	0.891	0.559

Table S5: Embedding performance on queries of similar molecular structure by cell line. A375: human malignant melanoma; A549: human non-small cell lung carcinoma; HA1E: human kidney epithelial immortalized; HT29: human colorectal adenocarcinoma; MCF7: human breast adenocarcinoma; PC3: human prostate adenocarcinoma; VCAP: human metastatic prostate cancer.

Query type	Cell line	Median	Top-0.001	Top-0.01	Top-0.1	AUC	MCC
Structure (ECFPs)	A375	0.0108	0.280	0.491	0.827	0.965	0.827
	A549	0.0155	0.216	0.434	0.813	0.953	0.804
	HA1E	0.0224	0.192	0.387	0.733	0.882	0.627
	HT29	0.0079	0.288	0.536	0.840	0.955	0.795
	MCF7	0.0119	0.307	0.481	0.783	0.936	0.742
	PC3	0.0129	0.228	0.460	0.811	0.936	0.741
	VCAP	0.0986	0.095	0.192	0.503	0.815	0.482
Structure (MACCS)	A375	0.0272	0.208	0.374	0.750	0.917	0.690
	A549	0.0217	0.230	0.419	0.685	0.874	0.613
	HA1E	0.0468	0.127	0.277	0.644	0.838	0.552
	HT29	0.0120	0.261	0.480	0.726	0.890	0.624
	MCF7	0.0133	0.293	0.471	0.734	0.889	0.640
	PC3	0.0109	0.297	0.492	0.736	0.886	0.621
	VCAP	0.0662	0.078	0.233	0.578	0.914	0.672

Embeddings and MACCS

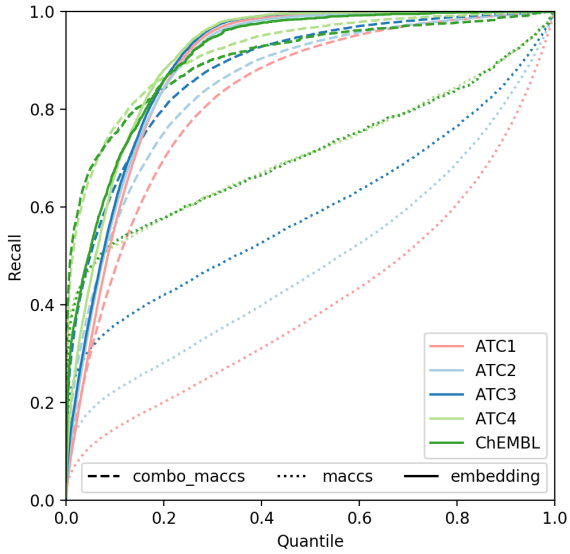


Figure S11: Recall by quantile for structural similarity (MACCS Tanimoto coefficient), embedding and a combination of both on functional similarity queries based on ATC levels 1–4 and ChEMBL protein targets.

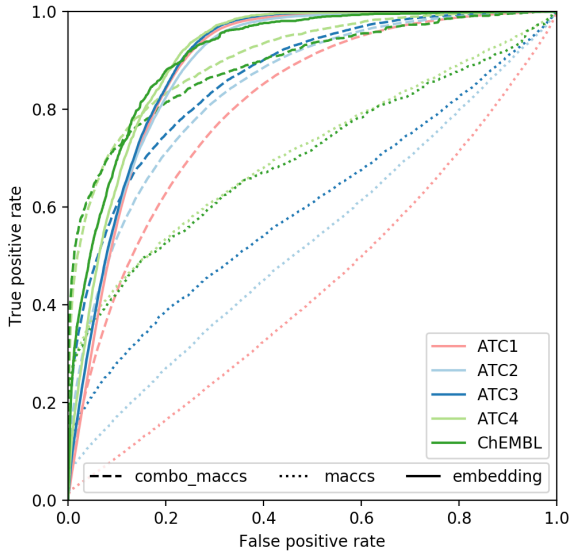


Figure S12: ROC curves for structural similarity (MACCS Tanimoto coefficient), embedding and a combination of both on functional similarity queries based on ATC levels 1–4 and ChEMBL protein targets.

Table S6: Performance of embedding and baselines on queries of similar therapeutic targets, protein targets and molecular structure.

Query type	Method	Median	Top-0.001	Top-0.01	Top-0.1	AUC
ATC level 1	Combo (MACCS)	0.1100	0.025	0.101	0.471	0.805
	Embedding	0.0857	0.014	0.089	0.560	0.889
	MACCS	0.6892	0.021	0.055	0.146	0.438
ATC level 2	Combo (MACCS)	0.0785	0.064	0.187	0.561	0.842
	Embedding	0.0773	0.022	0.121	0.590	0.889
	MACCS	0.5604	0.060	0.120	0.224	0.532
ATC level 3	Combo (MACCS)	0.0498	0.122	0.287	0.642	0.864
	Embedding	0.0746	0.033	0.144	0.605	0.896
	MACCS	0.3457	0.125	0.227	0.358	0.601
ATC level 4	Combo (MACCS)	0.0137	0.232	0.468	0.764	0.905
	Embedding	0.0604	0.049	0.197	0.665	0.908
	MACCS	0.0753	0.244	0.385	0.520	0.711
ChEMBL target	Combo (MACCS)	0.0092	0.239	0.507	0.750	0.890
	Embedding	0.0459	0.106	0.317	0.677	0.913
	MACCS	0.0691	0.204	0.368	0.528	0.701

Histograms of rank quantiles for structural similarity

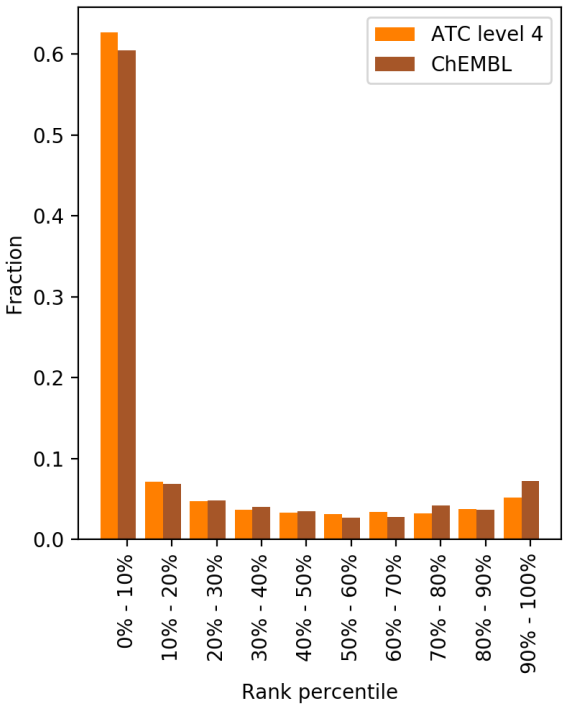


Figure S13: Histogram of rank quantiles for queries based on sets from ATC level 4 and ChEMBL, using ECFPs Tanimoto coefficient for similarity.

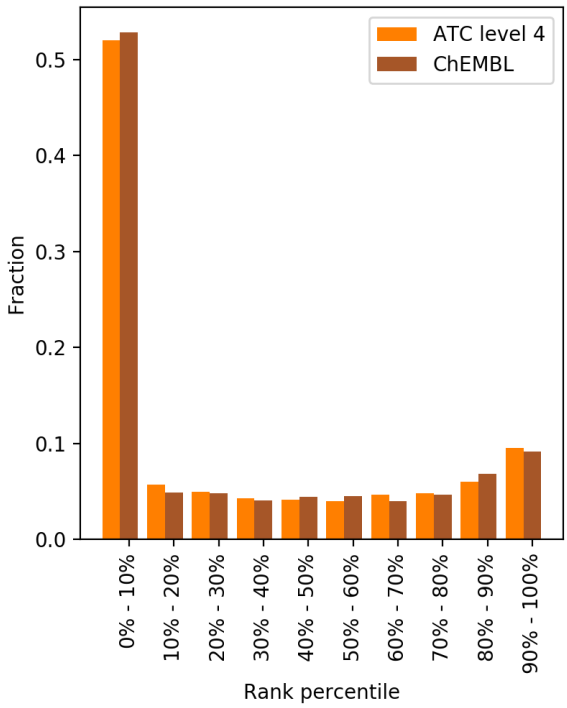


Figure S14: Histogram of rank quantiles for queries based on sets from ATC level 4 and ChEMBL, using MACCS Tanimoto coefficient for similarity.

Hyperparameter settings

In our method, several hyperparameters must be set. To obtain the main results reported in this manuscript, we did not optimize the hyperparameters, but instead focused on developing a robust network architecture that would not be very sensitive to hyperparameter choices. To achieve this robustness, we used a combination of several factors to achieve this robustness: 1. building blocks (SELU activations and densely connected architecture) that have proven effective at training very deep networks across many domains.[S1,S2] 2. data augmentation by injection of Gaussian noise into the standardized inputs, which diminishes the potential effects of overfitting; and 3. regularization by the margin parameter, which like augmentation allows us to train large networks while reducing concerns of overfitting.

In our main experiments, we set the hyperparameter values as follows. The margin parameter was set to 0.25 based on a computer vision study that used a very similar margin.[S3] The noise standard deviation was set to 0.3 as in Ladder Networks.[S4] Although there are many differences between our network and Ladder Networks, we did not initially have a reason to change this value initially. For embedding size, we noted that face recognition networks typically use an embedding of size 128.[S5,S6] In those applications the goal is typically binary verification or top-1 accuracy, whereas in our application, the focus is for the similarity in embedding space to generalize to similar but not identical perturbagens. We were concerned that a high dimensionality (e.g. 128) would lead to better separation but worse generalization, so we chose a smaller value, 32. In practice, the experiments described below revealed that there was almost no difference between embedding size of 32 and 128 and we could use either 32, 64 or 128 with nearly identical results. For network complexity, defined by the number of parameters, we note that DenseNet for ImageNet yielded good results at around 10-20M parameters.[S2] The training dataset we used is comparable in size to that of ImageNet, but it is not clear how to compare the augmentation methods used for ImageNet (for which images are the input, allowing the use of random crops) and our network (for which the augmentation involves adding noise to the input). To err on the side of caution, and emphasize generalization over separation in the metric space, we decreased the number of parameters by an order of magnitude to about 1M. Our experiments described below indicated that larger networks may perform only slightly better. For network depth, we observed based the training loss alone that our networks trained well at any depth that we tried. Because increasing the depth increases the training time, and the training loss changed very little with increasing depth, we chose a value of 64. Based on the experiments described below, this is much deeper than necessary, but it doesn't perform any worse. Ultimately, we chose a growth rate of 16 which put us in the desired range of parameters. The training parameters of batch size, learning rate and number of steps were determined empirically by examining the training loss during training.

We observed very good results with the parameter settings describe above and therefore used these values to obtain the results described in the main text. However, we also wanted to explore the effect of changing the hyperparameters to determine whether we could obtain even better results. The results of the onsuing experiments revealed that our method performs very well (i.e., with almost no change in outcome) over very wide ranges of values of all hyperparameters. In each experiment, we varied one hyperparameter, keeping the rest fixed. Although it is unlikely that optimizing a single hyperparameter at a time could lead to a global optimum, we adopted this approach due to its simplicity, and leave global hyperparameter optimization for future work.

For evaluation, rank distributions were aggregated over four splits to 80% training and 20% test perturbagens, and the summary statistics are the same as in the main paper.

First, we explored embedding sizes between 4 and 128 in powers of 2. The embedding size needs to be large enough to capture sufficient dimensions of functional variation, but if it is too large, it may lead to worse generalization. In particular, since the training goal is to separate the perturbagens to clusters, there may be an embedding size large enough to completely separate all the perturbagens in a space where distance is functionally meaningless. We found that performance increased rapidly up to an embedding size of 32, and changed very little beyond that (Table S7).

Table S7: Performance as a function of embedding size, keeping other hyperparameters fixed.

Embedding size	Median	Top-0.001	Top-0.01	Top-0.1	AUC
4	0.0580	0.189	0.304	0.579	0.837
8	0.0160	0.307	0.453	0.700	0.887
16	0.0117	0.325	0.485	0.726	0.899
32	0.0101	0.333	0.500	0.739	0.902
64	0.0102	0.320	0.499	0.745	0.904
128	0.0100	0.334	0.501	0.744	0.903

Keeping the embedding size at 32, we next explored the network depth. The number of parameters in the network is a function of the depth d and the growth rate k. The input size is 978 and each layer grows by k, so (ignoring the final embedding layer) the number

of parameters is $n(d,k) = \sum_{i=0}^{d-1} k\,(978 + ki) = 978kd + \frac{1}{2}k^2 d\,(d-1)$. To experiment with depth, we fixed the number of parameters N and for a given d chose the minimal $k > 0$ such that $n(d,k) \geq N$. With the number of parameters approximately fixed, performance improved with increasing depth up to 16 and beyond that point it remained almost constant up to 128 which is the deepest network that we have trained (Table S8). For the rest of the experiments in this section, we fixed the depth to 16. For the number of parameters, performance improved with the number of parameters even above 1M parameters, but the improvement beyond 400K parameters was small (Table S9). We chose 1M parameters, corresponding to a growth rate of $k = 45$.

Table S8: Performance as a function of network depth for a fixed number of trainable parameters, and other hyperparameters fixed.

Network depth	Median	Top-0.001	Top-0.01	Top-0.1	AUC
1	0.0186	0.313	0.449	0.677	0.885
2	0.0105	0.343	0.496	0.734	0.905
4	0.0100	0.344	0.501	0.740	0.905
8	0.0099	0.341	0.501	0.740	0.906
16	0.0099	0.344	0.502	0.741	0.908
32	0.0101	0.340	0.500	0.740	0.906
64	0.0101	0.330	0.500	0.739	0.904
128	0.0098	0.332	0.502	0.739	0.904

Table S9: Performance as a function of the number of trainable parameters changes, keeping other hyperparameters fixed.

Parameters	Median	Top-0.001	Top-0.01	Top-0.1	AUC
100000	0.0231	0.291	0.428	0.652	0.869
200000	0.0170	0.313	0.451	0.688	0.885
400000	0.0130	0.327	0.475	0.719	0.900
600000	0.0112	0.338	0.490	0.732	0.904
800000	0.0104	0.343	0.497	0.737	0.903
1200000	0.0100	0.338	0.501	0.741	0.904

Finally we varied the maximum value of the margin and the level of noise added to the input. Training with no margin led to much worse results, but at a maximum value between 0.1 and 0.5 the performance did not vary much (Table S10). This result is consistent with

the published results that led us to choose this value in the first place.[S3] Similarly, we found that adding no noise at all decreased the performance significantly, but performance was stable and optimal in the range between 0.3 and 0.6 (Table S11).

Table S10: Performance as a function of the maximum value of the margin parameter, keeping other hyperparameters fixed.

Max m	Median	Top-0.001	Top-0.01	Top-0.1	AUC
0.0	0.0199	0.282	0.447	0.634	0.761
0.1	0.0112	0.335	0.490	0.721	0.895
0.2	0.0104	0.334	0.497	0.733	0.903
0.3	0.0102	0.341	0.498	0.743	0.906
0.4	0.0116	0.339	0.486	0.738	0.907
0.5	0.0142	0.327	0.467	0.727	0.905

Table S11: Performance as a function of the standard deviation of the added noise changes, keeping other hyperparameters fixed.

Noise σ	Median	Top-0.001	Top-0.01	Top-0.1	AUC
0.0	0.0170	0.316	0.457	0.668	0.854
0.1	0.0139	0.328	0.471	0.697	0.878
0.2	0.0112	0.329	0.490	0.726	0.895
0.3	0.0100	0.340	0.500	0.743	0.906
0.4	0.0101	0.340	0.500	0.747	0.911
0.5	0.0102	0.348	0.498	0.749	0.914
0.6	0.0109	0.343	0.492	0.742	0.911
0.7	0.0120	0.339	0.483	0.732	0.907

In summary, our exploration of the effects of hyperparameter settings revealed that it was possible to further improve performance using slightly more optimal settings, however, performance was stable across a wide range of values and the improvements were relatively small. For this reason, we did not perform extensive hyperparameter tuning in this work.

Holdout similarity

In our main experiments, the similarity between two perturbagens is the dot product between the average embeddings of samples corresponding to these perturbagens. The neural

network is trained once on the entire data, and the trained network is used to compute the sample embeddings. Because functional labels are not used during training, there is no concern of overfitting; however, training and evaluating the network on the same samples could potentially lead to worse results when evaluating functional similarity. For example, if two different perturbagens in the data have nearly identical functional outcomes, their embeddings should be nearly identical, but the training objective will push their embeddings apart, resulting in an underestimation of their similarity. By contrast, if at least one of the perturbagens is not included in the training set, the similarity would not be underestimated and could be very close to 1. However, because underestimating the similarity affects all perturbagen pairs, it may not negatively impact the ranking of perturbagens.

To determine whether evaluating the embeddings directly on the training set negatively impacts the ranking of perturbagens by similarity, we also computed similarity between perturbagens using a holdout method. For this purpose, the entire set of perturbagens was split 41 times to a training set and test set. The splits were generated such that each pair of perturbagens appeared at least once in the same test set. The neural network was trained 41 times, once on each training set, and the embeddings for the corresponding test set samples were computed. The similarity between two perturbagens was computed as the average similarity between them over all test sets in which both appeared. Within each test set, perturbagen similarity was computed as the average dot product of sample embeddings, as before. Finally, we evaluated the ranking summary statistics and AUC as we did for the rest of the experiments, using ATC levels 1–4 and ChEMBL as queries.

The results obtained using the holdout similarity method were similar to those obtained using the direct training method, with a small difference in favor of the latter (Table S12 and Figures S15 and S16). We also compared the similarity ranks for each perturbagen using Kendall's tau rank distance (Figure S17). 90% of the distances were below 0.32, with a mean distance of 0.2615. In practice, similarity in embedding space is used to find perturbagens that are functionally similar to a query perturbagen, so only the top ranks are important.

Table S12: Comparison of two methods of computing perturbagen embeddings: direct training and holdout. The comparison was performed on ATC levels 1–4 and ChEMBL protein targets, similar to Figure 3d.

Sets	Method	Median	Top-0.001	Top-0.01	Top-0.1	AUC
ATC level 1	Direct training	0.0915	0.006	0.064	0.536	0.881
	Holdout	0.1094	0.005	0.056	0.464	0.869
ATC level 2	Direct training	0.0824	0.013	0.096	0.570	0.882
	Holdout	0.1003	0.012	0.084	0.499	0.870
ATC level 3	Direct training	0.0797	0.022	0.114	0.585	0.889
	Holdout	0.0963	0.018	0.101	0.513	0.875
ATC level 4	Direct training	0.0656	0.036	0.163	0.648	0.901
	Holdout	0.0783	0.027	0.147	0.575	0.888
ChEMBL target	Direct training	0.0513	0.090	0.288	0.658	0.906
	Holdout	0.0614	0.067	0.264	0.615	0.901

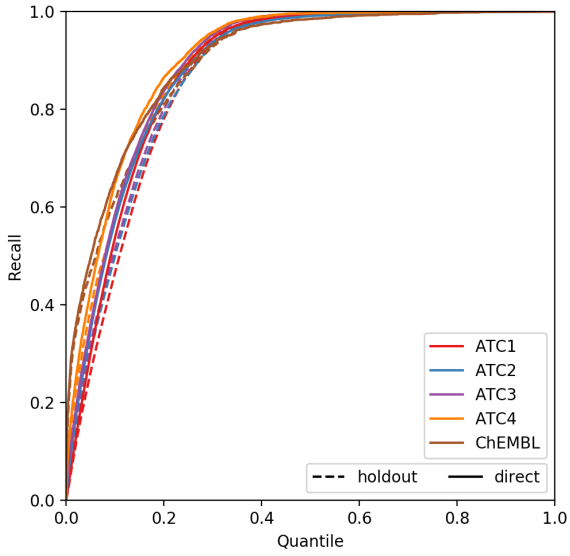


Figure S15: Plot of quantile vs recall for the direct training and holdout methods of computing embeddings.

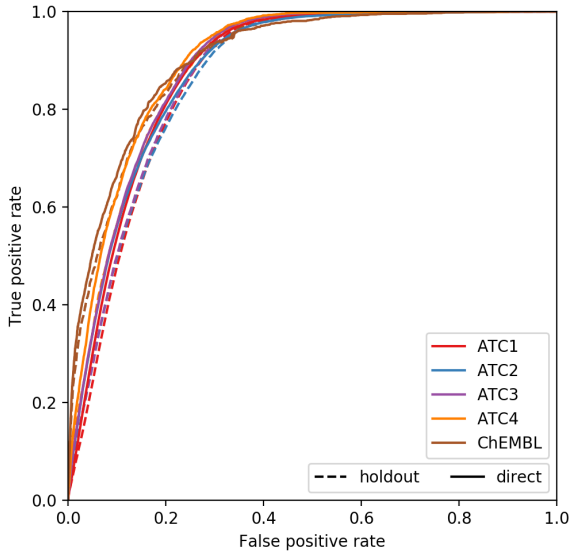


Figure S16: ROC plot of the direct training and holdout methods of computing embeddings.

We also visualized the joint distribution of similarity quantiles over all pairs of perturbagens using the direct training method and the holdout method (Figure S18). For the top 10% similarities, the correspondence was particularly accurate.

Because the two methods led to similar rankings, especially for the most similar perturbagen pairs, we chose to use the direct training method in our experiments. The difference, which was very small, was in favor of the direct training method. Moreover, because only one network is trained, the direct training method is much simpler to implement and much faster to train.

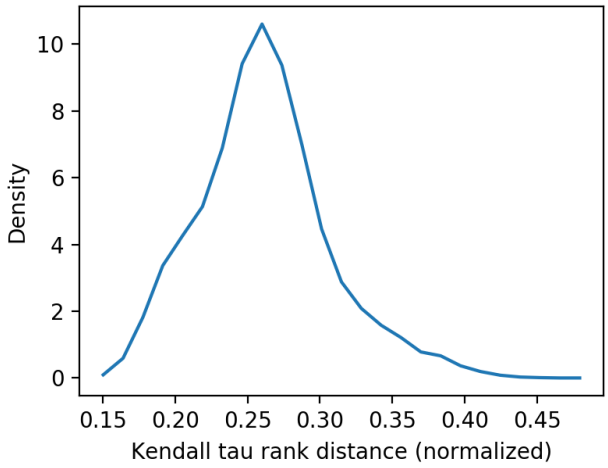


Figure S17: Estimated density of normalized Kendall distances between direct training and holdout similarities.

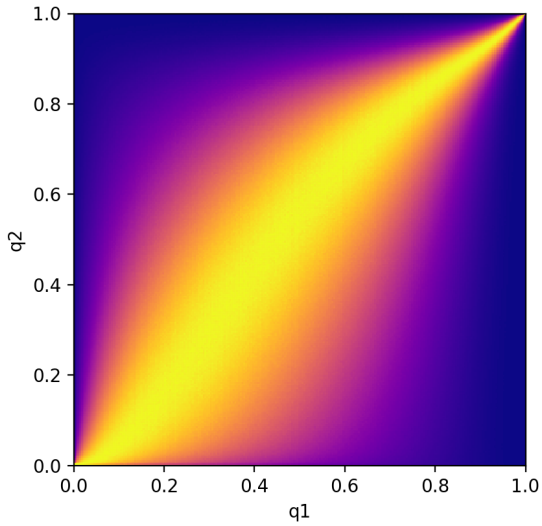


Figure S18: Heatmap histogram of quantiles of similarity using the holdout method q_2 for each quantile of similarity using the direct training method q_1.

References

(S1) Klambauer, G.; Unterthiner, T.; Mayr, A.; Hochreiter, S. Self-Normalizing Neural Networks. *CoRR* **2017**, *abs/1706.02515*.

(S2) Huang, G.; Liu, Z.; Weinberger, K. Q. Densely Connected Convolutional Networks. *CoRR* **2016**, *abs/1608.06993*.

(S3) Liang, X.; Wang, X.; Lei, Z.; Liao, S.; Li, S. Soft-Margin Softmax for Deep Classification. **2017**, 413–421.

(S4) Rasmus, A.; Valpola, H.; Honkala, M.; Berglund, M.; Raiko, T. Semi-Supervised Learning with Ladder Network. *CoRR* **2015**, *abs/1507.02672*.

(S5) Schroff, F.; Kalenichenko, D.; Philbin, J. FaceNet: A Unified Embedding for Face Recognition and Clustering. *CoRR* **2015**, *abs/1503.03832*.

(S6) Parkhi, O. M.; Vedaldi, A.; Zisserman, A. Deep Face Recognition. British Machine Vision Conference. 2015.