Deep Learning Improves Antimicrobial Peptide Recognition

Supplementary Information

Daniel Veltri[1,2,*], Uday Kamath[3], and Amarda Shehu[4,5,6,*]

[1]Bioinformatics and Computational Biosciences Branch, Office of Cyber Infrastructure and Computational Biology, National Institute of Allergy and Infectious Diseases, U.S. National Institutes of Health, Rockville, MD, 20852, USA.
[2]Medical Science & Computing, LLC, 11300 Rockville Pike #1100, Rockville, MD, 20852, USA.
[3]Digital Reasoning, 1765 Greensboro Station Place #1200, McLean, VA, 22102, USA.
[4]Department of Computer Science, [5]Department of Bioengineering, George Mason University, Fairfax, VA, 22030, USA.
[6]School of Systems Biology, George Mason University, Manassas, VA, 20110, USA.
*To whom correspondence should be addressed.

1 Length distributions of AMP and non-AMP partitions

Sequence length distributions are shown for the training (top), evaluation (middle), and testing (bottom) partitions in Figure S1.

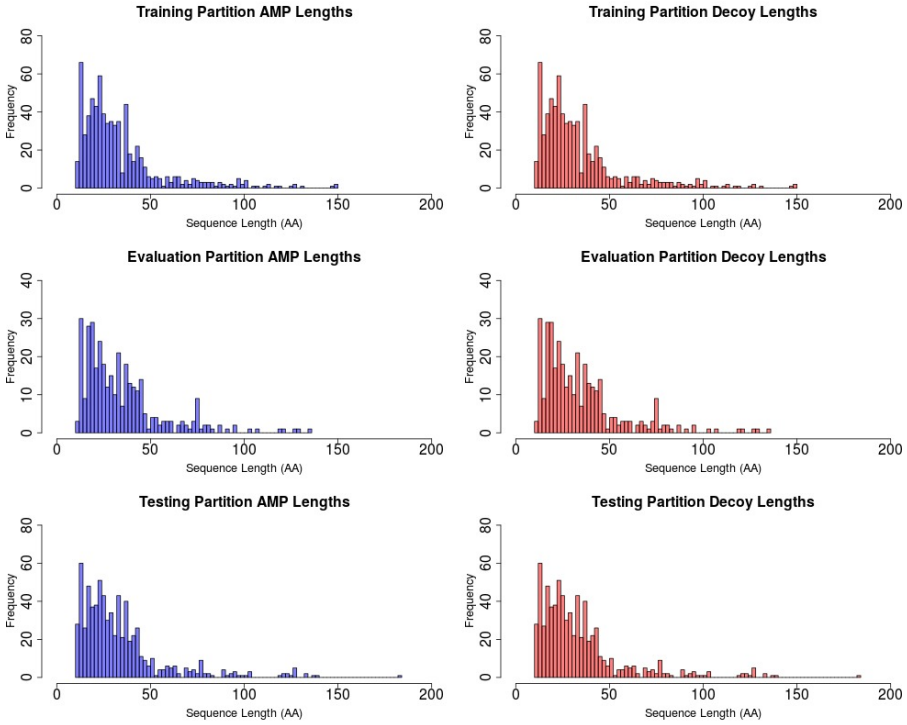

Figure S1: Sequence length distributions over AMPs are shown in blue (left), and over non-AMPs are shown in red (right).

2 Impact of Data Set Size on Model Performance

To assess the impact of data set size on our model performance, we use scikit-learn (vr.0.18.1) to construct learning curves (see learning_curve function page for details) using 10-fold CV for the entire data set. The curves are shown in Figure S2. The training (red) and testing (green) lines show that our model performance is not strongly dependent on data set size after roughly 700 observations are included (as the lines stay relatively flat from there on). Accordingly, this also suggests that simply adding more data to the model will not likely provide significant gains in recognition performance.

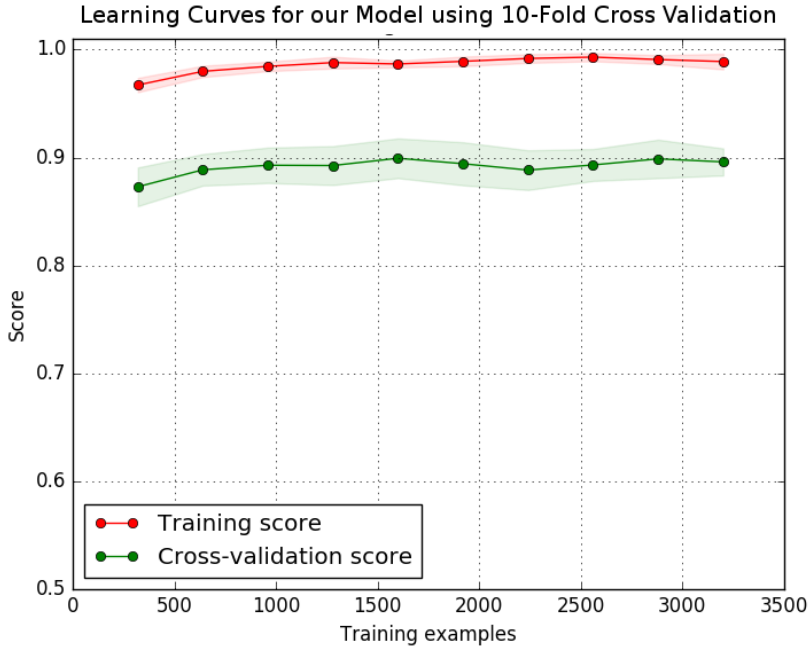

Figure S2: Learning curves using 10-fold CV for the entire data set are shown here. Training and testing lines are in red and green, respectively.

3 Impact of Balanced versus Unbalanced Testing Partitions on Model Performance

To see if our model performs differently on an unbalanced testing set, Table S1 shows evaluation results using 2262 additional decoy sequences selected by the same methodology described in the main article. Training is performed using the original balanced data set (712 AMPs, 712 Non-AMPs) while the new testing set (712 AMPs, 2974 Non-AMPs) has approximately a 1:4 AMP-to-decoy ratio.

Table S1: Model performance with unbalanced testing set

Model	Testing Set Version	No. AMPs	No. Non-AMPs	SENS(%)	SPEC(%)	ACC(%)	MCC
Our DNN	Balanced	712	712	86.95 (5.2)	94.54 (3.4)	90.75 (1.8)	0.8196 (0.03)
AntiBP2				87.91	90.80	89.37	0.7876
CAMP ANN				82.98	85.09	84.04	0.6809
CAMP DA				87.08	80.76	83.92	0.6797
CAMP RF				**92.70**	82.44	87.57	0.7554
CAMP SVM				88.90	79.92	84.41	0.6910
iAMP-2L				83.99	85.86	84.90	0.6983
iAMPpred				89.33	87.22	88.27	0.7656
Our DNN	Unbalanced	712	2974	86.22 (5.4)	**96.88** (2.2)	**93.49** (1.3)	**0.8463** (0.03)
AntiBP2				87.91	89.01	88.81	0.6903
CAMP ANN				82.98	84.73	84.40	0.5930
CAMP DA				87.08	83.25	83.99	0.6040
CAMP RF				**92.70**	85.47	86.87	0.6759
CAMP SVM				88.90	83.49	84.54	0.6208
iAMP-2L				86.24	85.71	85.81	0.6313
iAMPpred				89.33	88.20	88.42	0.6912

Column 1 in Table S1 lists the prediction method and other top-performing servers, while Column 2 lists the testing set version (balanced versus unbalanced). Columns 3 and 4 list the respective number of AMPs and Non-AMPs tested. Columns 5-8 show the classification performance in terms of SENS, SPEC, ACC, and MCC with the largest value in each column highlighted in bold. Our DNN model is evaluated in terms of 10-fold CV (SD listed in parentheses). We note that a new run of our (non-deterministic) model is shown for the balanced data set, thus numbers differ slightly from those in Table 3. The other methods are run using default settings for AntiBP2 (full sequence composition, SVM Threshold:0; note 371 of the unbalanced decoys are skipped due to server length restrictions) (Lata *et al.*, 2010), all CAMP predictors (Thomas *et al.*, 2009), iAMP-2L (Xiao *et al.*, 2013), and iAMPpred (Meher *et al.*, 2017). Overall, our method shows similar performance in both the balanced and unbalanced settings. Compared to results with the balanced set, ACC on the unbalanced set increases slightly for all methods except for AntiBP2 and CAMP RF. MCC increases on the unbalanced set for our method but decreases for the others.

4 Impact of Natural Termini and Length on Non-AMP Sequences

Table S2: Model performance using non-AMPs with original N- and C-termini

Testing Set Non-AMP Type	SENS(%)	SPEC(%)	ACC(%)	MCC
Original testing set non-AMP fragments	86.95 (5.2)	94.54 (3.4)	90.75 (1.8)	0.8196 (0.03)
Non-AMP fragments of same length but natural N- and C-termini	88.13 (3.7)	93.31 (3.4)	90.72 (2.2)	0.8166 (0.04)
Full-length non-AMP sequences	85.85 (5.3)	96.72 (2.2)	93.40 (1.3)	0.8444 (0.03)

Table S2 shows average 10-fold CV results (SD in parentheses) after testing to see if the proposed DNN model performs differently using non-AMPs containing natural N- and C-terminal AA, rather than the randomly-selected peptide fragments from our original data set. Column 1 in Table S2 lists the type of non-AMPs considered for both the training and testing models. Columns $2-5$ show performance results in terms of SENS, SPEC, ACC, and MCC, respectively. Row 2 lists results from the original non-AMP fragments (the balanced set in Table S1 above). Row 3 shows results using non-AMPs of the same length as the original non-AMPs; however, these are constructed using the original N- and C-termini AAs for each half of the peptide (those with an odd number of residues contain one additional N-terminal AA). Row 4 contains results using the original full-length non-AMPs downloaded from UniProt. With a difference in ACC less than 0.1%, there appears to be little difference to our DNN model between "natural" termini and randomly-selected ones for the non-AMP fragments. Results for the full-length non-AMPs suggest the model is able to slightly better discriminate AMPs ($\sim +3\%$ ACC) when longer non-AMPs are considered.

5 Comparison with Other Data Sets

Table S3: AMP recognition performance on other data sets

Method	Data Set	No. AMPs (Overlap)	No. Non-AMPs (Overlap)	ACC(%)	MCC
Our DNN	Lata et al. 2010	999 (75%)	999 (0%)	**92.95**	**0.860**
AntiBP2				91.64	0.831
CAMP ANN				81.03	0.624
CAMP DA				84.28	0.690
CAMP RF				87.09	0.752
CAMP SVM				86.69	0.739
iAMP-2L				86.34	0.735
iAMPpred				92.84	0.858
Our DNN	Fernandes et al. 2012	115 (62%)	116 (0%)	**90.93**	**0.827**
AntiBP2				85.30	0.706
CAMP ANN				77.06	0.553
CAMP DA				77.06	0.572
CAMP RF				79.65	0.640
CAMP SVM				77.06	0.584
iAMP-2L				87.90	0.759
iAMPpred				84.00	0.691
Our DNN	Xiao et al. 2013	Train Set: 878 (77%) Test Set: 920 (62%)	Train Set: 2368† (0.3%) Test Set: 920 (0%)	**97.42**	**0.949**
AntiBP2				89.10	0.781
CAMP ANN				80.00	0.610
CAMP DA				71.79	0.487
CAMP RF				65.27	0.396
CAMP SVM				67.77	0.429
iAMP-2L				92.23	0.845
iAMPpred				72.99	0.509

†37 sequences were removed from the original data set to remove duplicates or peptides containing fragments identical to known AMPs as in Veltri (2015).

Table S3 above shows classification performance on additional data sets for our DNN and seven other AMP classification servers, as listed in Column 1. The additional data sets used for evaluation come from Lata *et al.* (2010), Fernandes *et al.* (2012), and Xiao *et al.* (2013), as listed in Column 2. Columns 3 and 4 list the respective number of AMPs and non-AMPs in the set with the percentage of sequences found in our own data set given in parentheses. Columns 5 and 6 list classification performance in terms of ACC and MCC, respectively (highest value for each data set per column in bold). We note that our method was evaluated using 10-fold CV when a single data set was provided, and was trained and evaluated on respective training and testing partitions when available. Results for classifiers and data sets from the same source are listed as shown in their respective publications (i.e. iAMP-2L for Xiao *et al.* (2013) and AntiBP2 for Lata *et al.* (2010)).

Table S4: Bayesian sign-rank test results for mean difference in AMP classifier performance ACC (region of practical equivalence $[-1,+1]\%$)

Classifier 1	Classifier 2	C1≪C2 (%)	C1=C2 (%)	C1≫C2 (%)
Our DNN	AntiBP2	0	7.701	92.299
Our DNN	CAMP ANN	0	0.228	99.772
Our DNN	CAMP DA	0	0.207	99.793
Our DNN	CAMP RF	0	0.226	99.774
Our DNN	CAMP SVM	0	0.232	99.768
Our DNN	iAMP-2L	0	0.248	99.752
Our DNN	iAMPpred	0	4.955	95.045
AntiBP2	CAMP ANN	0	0.245	99.755
AntiBP2	CAMP DA	0	0.250	99.750
AntiBP2	CAMP RF	0	1.088	98.912
AntiBP2	CAMP SVM	0	0.233	99.767
AntiBP2	iAMP-2L	33.482	5.594	60.924
AntiBP2	iAMPpred	0	32.750	67.250
CAMP ANN	CAMP DA	25.443	35.783	38.774
CAMP ANN	CAMP RF	68.380	0.483	31.137
CAMP ANN	CAMP SVM	25.416	36.050	38.534
CAMP ANN	iAMP-2L	95.036	4.964	0
CAMP ANN	iAMPpred	86.064	0.986	12.950
CAMP DA	CAMP RF	68.645	0.489	30.866
CAMP DA	CAMP SVM	27.139	45.875	26.986
CAMP DA	iAMP-2L	95.229	4.771	0
CAMP DA	iAMPpred	98.911	1.089	0
CAMP RF	CAMP SVM	17.731	19.597	62.672
CAMP RF	iAMP-2L	79.584	8.042	12.374
CAMP RF	iAMPpred	95.183	4.817	0
CAMP SVM	iAMP-2L	72.850	27.150	0
CAMP SVM	iAMPpred	99.796	0.204	0
iAMP-2L	iAMPpred	42.500	1.237	56.263

Results in Table S4 above show Bayesian sign-rank tests for pairs of AMP classifiers using the ACC results on our own (balanced) set listed in Table S1 plus the three data sets listed in Table S3. Tests are performed following the approach outlined in Benavoli *et al.* (2014, 2017) using their accompanying R code (https://github.com/BayesianTestsML/tutorial/; Accessed: Jan. 21, 2018) with 100000 Markov chain Monte Carlo samplings and default Dirichlet prior parameters ($s= 0.5$, $z_0 = 0$). We set the region of practical equivalence to $\pm 1\%$ - in other words we consider two classifiers "equivalent" if their mean difference in ACC is within $\pm 1\%$ of each other (Kruschke and Liddell, 2015). For each row in Table S4, the ACCs for the classifiers listed in Columns 1 and 2 are compared and three posterior probabilities calculated. Column 3 gives the probability that Classifier 1 performs worse than Classifier 2 (the mean difference in ACC between Classifier 1 and 2 is $< 1\%$). Column 4 gives the probability that the performance of both classifiers is essentially equal (the mean absolute difference in ACC between Classifier 1 and 2 is

$\leq 1\%$). Column 5 gives the probability that Classifier 1 performs better than Classifier 2 (the mean difference in ACC between Classifier 1 and 2 is $> 1\%$). Comparisons to our DNN method in Rows 2-8 show that the probability of our method performing better is $> 92\%$ in all cases. The highest probability of equal performance to our method is seen with AntiBP2 at 8%.

6 Misclassified AMPs

Table S5: AMPs classified by production model as false negatives

APD Identifier	Sequence
AP00399	HVDKKVADKVLLLKQLRIMRLLTRL
AP00612	AAEFPDFYDSEEQMGPHQEAEDEKDRADQRVLTEEEKKELENLAAMDLELQKIAEKFSQR
AP00749	EADEPLWLYKGDNIERAPTTADHPILPSIIDDVKLDPNRRYA
AP00787	GWRLLLKKAEVKTVGKLALKHYL
AP00812	FAEPLPSEEEGESYSKEPPEMEKRYGGFM
AP01234	FSKYERQKDKRPYSERKNQYTGPQFLYPPERIPPQKVIKWNEEGLPIYEIPGEGGHAEPAAA
AP01283	MRKEFHNVLSSGQLLADKRPARDYNRK
AP01339	FLSFPTTKTYFPHFDLSHGSAQVKGHGAK
AP01343	TESYFVFSVGM
AP01372	SKCKCSRKGPKIRYSDVKKLEMKPKYPHCEEKMVIITTKSVSRYRGQEHCLHPKLQSTKRFIKWYNAWNEKRRVYEE
AP01522	TYMPVEEGEYIVNISYADQPKKNSPFTAKKQPGPKVDLSGVKAYGPG
AP01624	HAEHKVKIGVEQKYGQFPQGTEVTYTCSGNYFLM
AP01918	IGVIKLSLCEEERNADEEKRRDDPDEMDVEVEKR
AP01919	FTLKKSQLLLFFLGTINFSLCEEERNAEEERRDYPEEKDVEVEKR
AP01974	YGQSTHAVIYAQGYTYSSDWR
AP02030	MQIFVKTLTGKTITLEVEPSDTIENVKAKIQDKEGIPPDQQRLIFAGKQLEDGRTLSDYNIQKESTLHLVLRLR
AP02053	GLSQGVEPDIGQTYFEESRINQD
AP02081	FPMFKRGRCLCIGPGVKAVKVADIEKASIMYPSNNCDKIEVIITLKENKGQRCLNPKSKQARLIIKKVERKNF
AP02169	AKISGPEETSELPEVVSEERVPATATEPMADLRHGVTREPISPASKDSLRDKFKEKLDKWFHRPNLLSKRD
AP02193	YSKSLPLSVLNP
AP02352	YPGPQAKEDSEGPSQGPASREK
AP02353	LPVNSPMNKGDTEVMKCIVEVISDTLSKPSPMPVSKECFETLRGDERILSILRHQNLLKELQDLALQGAKERTHQQ
AP02405	GGYKNFYGSALRKGFYEGEAGRAIRR
AP02407	SDYLNNNPLFPRYDIGNVELSTAYRSFANQKAPGRLNQNWALTADYTYR
AP02533	SDKPDVKEVESFDKSKLKKVETQEKNPLPTKETIEQEKKG
AP02712	MNSSSVLFVCILGACSVWTVHGRNLKVNDDDQEGAELDISVEARKLPGLCWVCKWSLNKVKKLLGRNTTAESVKEKLMRVCNEI GLLKSLCKKFVKGHLGELIEELTTSDDVRTICVNLKACKPKELSELDFESDEDAHTEMNDLLFE
AP02767	APKGVQGPNG
AP02791	ARTKQTARKSTGGKAPRKQLAT
AP02804	MSGRGKGGKVKGKSKSRSSRAGLQFPVGRIHRLLRKGNYAERVGAGAPVYLAAVMEYLAAEVLELAGNAARDNKKTRIVPRHLQ LAIRNDEELNKLLSGVTIAQGGVLPNIQAVLLPKKTEKK
AP02809	MARTKQTARKSTGGKAPRKQLATKAARKSAPATGGVKKPHRYRPGTVALREIRRYQKSTELLIRKLPFQRLVREIAQDFKTDLRFQS SAVMALQEACEAYLVGLFEDTNLCAIHAKRVTIMPKDIQLARRIRGERA
AP02813	ILELAGNAARDNKKTRIIPRHLQL
AP02879	LTAEDKKLIQQAWEKAASHQEEFGAEALTRMFTTYPQTKTY
AP02886	LLNQELLLNPTHQIYPV
AP02895	SMATPHVAGAAALILSKHPTWTNAQVRDRLESTATYLGNSFYYGK

7 K-means Analysis and DNN-reduced Alphabet

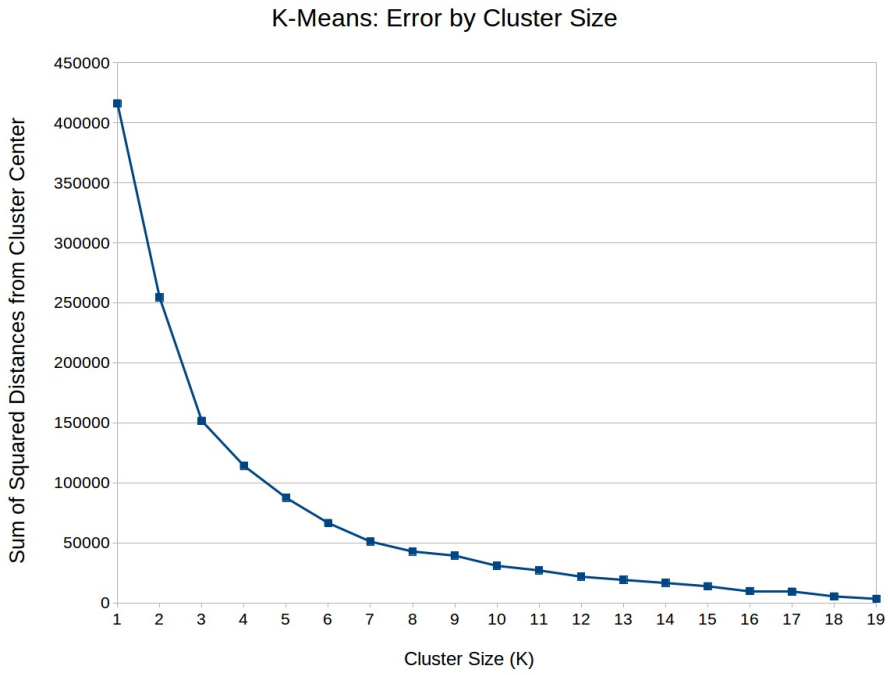

Figure S3: Sum of squared distances from k-means cluster centers are shown using various k $(1-19)$ for the 20 naturally-occurring AAs. The bend or "elbow" (Thorndike, 1953) at $k = 8$ is selected as the cluster size before restoring the padding character 'X' as its own cluster. Accordingly, 9 clusters are used to build the DNN-reduced alphabet. The representative character for each cluster is listed in Table S6.

Table S6: DNN-reduced AA alphabet for AMPs

Original Letter	**Mapping Letter**
X	X
ED	E
QST	Q
NH	N
PYRK	P
VAF	V
LMI	L
GW	G
C	C

Table S7: DNN-reduced AA alphabet performance various sizes of k

k	ACC(%)	MCC
3	77.94 ($\pm$0.5)	0.5688 ($\pm$0.01)
5	87.09 ($\pm$1.1)	0.7450 ($\pm$0.02)
9	89.57 ($\pm$0.9)	0.7938 ($\pm$0.02)
15	90.54 ($\pm$1.0)	0.8137 ($\pm$0.02)

Table S7 shows average DNN performance values on the testing set using reduced AA alphabets constructed using various sizes of k, as described in Section 2.6 in the main article. Column 1 shows the value of k, while Columns 2 and 3 respectively show the average ACC and MCC over 100 trials with SD shown in parentheses. We note results in Row 4 are the same as Row 11 in Table 2 from Section 3.3 in the main article. As the value of k increases, both metrics increase in an approximately logarithmic fashion. The k-means clustering error for the values in Column 1 can be seen above in Figure S3.

8 Data set Availability

Data sets are available in FASTA format from the AMP Scanner web server: http://www.ampscanner.com.

References

Benavoli, A., Corani, G., Mangili, F., Zaffalon, M., and Ruggeri, F. (2014). A bayesian wilcoxon signed-rank test based on the dirichlet process. In *International Conference on Machine Learning*, pages 1026–1034.

Benavoli, A., Corani, G., Demšar, J., and Zaffalon, M. (2017). Time for a change: a tutorial for comparing multiple classifiers through bayesian analysis. *Journal of Machine Learning Research*, **18**(77), 1–36.

Fernandes, F. C., Rigden, D. J., and Franco, O. L. (2012). Prediction of antimicrobial peptides based on the adaptive neuro-fuzzy inference system application. *Peptide Science*, **98**(4), 280–287.

Kruschke, J. K. and Liddell, T. M. (2015). The bayesian new statistics: two historical trends converge. *SSRN Electronic Journal*.

Lata, S., Mishra, N. K., and Raghava, G. P. (2010). AntiBP2: improved version of antibacterial peptide prediction. *BMC Bioinformatics*, **11**(Suppl 1), S1–S19.

Meher, P. K., Sahu, T. K., Saini, V., and Rao, A. R. (2017). Predicting antimicrobial peptides with improved accuracy by incorporating the compositional, physico-chemical and structural features into Chou's general PseAAC. *Scientific Reports*, **7**(42362).

Thomas, S., Karnik, S., Barai, R. S., Jayaraman, V. K., and Thomas, S. I. (2009). CAMP: a useful resource for research on antimicrobial peptides. *Nucl. Acids Res.*, **38**(Suppl 1), D774–D780.

Thorndike, R. L. (1953). Who belongs in the family? *Psychometrika*, **18**(4), 267–276.

Veltri, D. (2015). *A Computatioanl and Statistical Framework for Screening Novel Antimicrobial Peptides*. PhD dissertation, George Mason University.

Xiao, X., Wang, P., Lin, W.-Z., Jia, J.-H., and Chou, K.-C. (2013). iAMP-2L: A two-level multi-label classifier for identifying antimicrobial peptides and their functional types. *Analytical biochemistry*.