2008 HHMI Investigator Competition: Eligibility Responses

First Name: William
Last Name: Noble

Doctoral Degree: Ph.D.

Primary Institution with academic appointment: University of Washington

Institution has full-time tenure-track faculty appointment: YES
Applicant is: Tenured

Academic Title: Associate Professor

First faculty appointment as an assistant professor or equivalent rank: 7/1999

Is lead investigator on one national peer-reviewed grant: YES
Grant Information: NIH
R01
EB007057
Machine learning analysis of tandem mass spectra
3/1/07--2/28/11

Has agreed with all of HHMI's conditions: YES

2008 HHMI Investigator Competition: Contact Info

First Name: William
Last Name: Noble
Middle Name: Stafford
Primary Phone: 206 543-8930 (Office)
Alternate Phone: 206 355-5596 (Cell)
E-mail Address: noble@gs.washington.edu
Advanced Degree: Ph.D.
Position Title: Associate Professor

Institutional Mailing Address:

University of Washington
Genome Sciences
Box 355065
Foege Building, S220B
1705 NE Pacific St.
Seattle, Washington 98195
United States

Research Focus: Machine learning techniques for application to problems in molecular biology
Research Keywords: Bioinformatics, Machine learning, Mass spectrometry, Motif discovery, Protein-protein interaction prediction, Gene function prediction, Analysis of heterogeneous data, Support vector machines,

Scientific Disciplines:

Primary: Computational biology
Secondary: N/A

BIOGRAPHICAL SKETCH

NAME	POSITION TITLE
William Stafford Noble (formerly William Noble Grundy)	Associate Professor

EDUCATION/TRAINING

INSTITUTION AND LOCATION	DEGREE	YEAR(S)	FIELD OF STUDY
Stanford University	BS	1991	Symbolic Systems
University of California, San Diego	MS	1996	Computer Science
University of California, San Diego	PhD	1998	Computer Science & Cognitive Science
University of California, Santa Cruz	Postdoc	1999	Computational Biology

Professional positions

2006– Associate Professor, Department of Genome Sciences, University of Washington

2006– Adjunct Associate Professor, Department of Computer Science and Engineering, University of Washington

2006– Adjunct Associate Professor of Medicine, University of Washington

2005–06 Adjunct Assistant Professor of Medicine, University of Washington

2002–06 Assistant Professor, Department of Genome Sciences, University of Washington

2002–06 Adjunct Assistant Professor, Department of Computer Science and Engineering, University of Washington

2000–02 Pharmaceutical Research and Manufacturers of America Foundation Faculty Development Award in Bioinformatics.

1999–02 Assistant Professor, Department of Computer Science, Columbia University, with joint appointment at the Columbia Genome Center.

Professional Activities, honors and awards

2001–05 Research Fellow, Alfred P. Sloan Foundation.

2001–06 National Science Foundation CAREER Award.

1998–99 Fellow, Alfred P. Sloan Foundation and U.S. Department of Energy Postdoctoral Fellowships in Computational Molecular Biology

1994–97 Fellow, National Defense Science and Engineering Graduate Fellowship Program.

1991 Phi Beta Kappa, Stanford University.

1987 David Starr Jordan Scholar, Stanford University.

1987 National Merit Scholar.

Program committee member, AAAI 1998, ISMB 2002–2007, BIOKDD 2002, KDD 2000, 2003, COLT 2003, RECOMB 2004, 2007, ICML 2004, ECCB 2005, GIW 2005-2007, RECOMB computational protemics satellite 2007, BIRD 2007.

Scientific consultant, Rigel Pharmaceuticals, Inc., South San Francisco, CA, 1999–2001.

Member, Scientific Advisory Board, X-Mine, Inc., Hayward, CA, 2000–2002.

Panelist, National Science Foundation review panel on Information Technology Research at the intersection of biology and informatics, April 18-19, 2001.

Panelist, National Institutes of Health Special Bioinformatics Study Section, March 12, 2003, June 30, 2004 and March 17–18, 2005.

Panelist, National Institues of Health Biodata Management and Analysis study section, January 29–30, 2007.

Member, Public Affairs and Policies Committee, International Society for Computational Biology, 2003–present.

Guest co-editor, Special issue on Machine Learning for Bioinformatics, *IEEE Transactions on Computational Biology and Bioinformatics*, 2004.

Member, Scientific Advisory Board, Bioinformatics of Mammalian Gene Expression project, Canada s Michael Smith Genome Centre, Vancouver, BC, Canada, 2004–present.

Editorial board member, *Journal of Bioinformatics and Computational Biology*, 2004–present.

Editorial board member, *IEEE Transactions on Computational Biology and Bioinformatics*, 2005–present.

Co-chair, Workshop on Computational Biology and the Analysis of Heterogeneous Data, Nineteenth Annual Conference on Neural Information Processing Systems, Whistler, BC, Dec. 9–10, 2005.
Member, Internal Advisory Board, Center for Functional Genomics and HCV-Associated Liver Disease, University of Washington, Seattle, WA, 2006–present.
Member, Scientific Advisory Board, National Center for Systems Biology, Institute for Systems Biology, Seattle, WA, 2006–present.
Panelist, National Cancer Institute special emphasis panel on "Advance proteomic platforms and computation science for the NCI clinical proteomic technologies initiative," June 26-27, 2006.
Co-chair, Workshop on Computational Biology, Twentieth Annual Conference on Neural Information Processing Systems, Whistler, BC, Dec. 8–9, 2006.
Area chair, Bioinformatics and Kernel Methods, Twenty-first Annual Conference on Neural Information Processing Systems, Whistler, BC, Dec 3–8, 2007.

Peer-reviewed publications

1. JH Dennis, H-Y Fan, SR Reynolds, G Yuan, J Meldrim, DJ Richter, DG Peterson, OJ Rando, **WS Noble** and RE Kingston. "Independent and complementary methods for large-scale structural analysis of mammalian chromatin." *Genome Research.* To appear.

2. I Melvin, E Ie, R Kuang, J Weston, **WS Noble** and C Leslie. "SVM-fold: a tool for discriminative multi-class protein fold and superfamily recognition." *BMC Bioinformatics.* To appear.

3. N Day, A Hemmaplardh, RE Thurman, JA Stamatoyannopoulos and **WS Noble**. "Unsupervised segmentation of continuous genomic data." *Bioinformatics.* To appear.

4. RE Thurman, N Day, **WS Noble** and JA Stamatoyannopoulos. "Identification of higher-order functional domains in the human ENCODE regions." *Genome Research.* To appear.

5. C Leslie, I Melvin, E Ie, J Weston and **WS Noble**. "Multi-class protein classification using adaptive codes." *Journal of Machine Learning Research.* To appear.

6. AA Klammer, X Yi, MJ MacCoss and **WS Noble**. "Peptide retention time prediction yields improved tandem mass spectrum identification for diverse chromatography conditions." *Analytical Chemistry.* To appear.

7. The ENCODE Project Consortium. "Identification and analysis of functional elements in 1% of the human genome by the ENCODE project." *Nature*. To appear.

8. H Peckham, RE Thurman, Y Fu, JA Stamatoyannopoulos, **WS Noble**, K Struhl and Z Weng. "Nucleosome positioning signals in genomic DNA." *Genome Research*. To appear.

9. S Asthana, **WS Noble**, G Kryukov, CE Grant, S Sunyaev and JA Stamatoyannopoulos. "Widely distributed non-coding selection in the human genome." *Proceedings of the National Academy of Science.* To appear.

10. S Gupta, JA Stamatoyannopoulos, TL Bailey and **WS Noble**. "Quantifying similarity between motifs." *Genome Biology.* 8:R24, 2007.

11. J Qiu, M Hue, A Ben-Hur, J-P Vert and **WS Noble**. "A structural alignment kernel for protein structures." *Bioinformatics.* 23(9):1090-1098, 2007.

12. A Klammer, X Yi, MJ MacCoss and **WS Noble**. "Peptide retention time prediction yields improved tandem mass spectrum identification for diverse chromatography conditions." *Proceedings of the International on Research in Computational Biology (RECOMB)*. April 21–25, 2007. pp. 459–472.

13. J-P Vert, R Thurman and **WS Noble**. "Kernels for gene regulatory regions." *Advances in Neural Information Processing Systems 19*. 2006. Acceptance rate: 27.4%.

14. J Weston, R Kuang, C Leslie and **WS Noble**. "Protein ranking by semi-supervised network propagation." *BMC Bioinformatics*. 7(Suppl 1):S10, 2006.

15. A Ben-Hur and **WS Noble**. “Choosing negative examples for the prediction of protein-protein interactions.” *BMC Bioinformatics*. 7(Suppl 1):S2, 2006.

16. DP Lewis, T Jebara and **WS Noble**. “Nonstationary kernel combination.” *Proceedings of the International Conference on Machine Learning*, June 25-29, 2006, Pittsburgh, PA. Acceptance rate: 20.0%.

17. T Mann and **WS Noble**. “Efficient identification of DNA binding partners in a sequence database.” *Bioinformatics (Proceedings of the Intelligent Systems for Molecular Biology Conference)*. 22:e359–3367, 2006. Acceptance rate: 16.6%.

18. PJ Sabo, MS Kuehn, R Thurman, C Grant, B Johnson, S Johnson, H Kao, M Yu, J Goldy, M Weaver, MA Singer, TA Richmond, MO Dorschner, P Navas, R Green, **WS Noble** and JA Stamatoyannopoulos. “Genome-scale mapping of DNaseI sensitivity *in vivo* using tiling DNA microarrays.” *Nature Methods.* 3(7):511–518, 2006.

19. BE Frewen, GE Merrihew, **WS Noble** and MJ MacCoss. “Analysis of peptide MS/MS spectra from large-scale proteomics experiments using spectrum libraries.” *Analytical Chemistry.* 78(16):5678–5684, 2006.

20. T Pramila, W Wu, **WS Noble** and LL Breeden. “The Forkhead transcription factor Hcm1 regulates chromosome segregation genes and fills the S phase gap in the transcriptional circuitry of the cell cycle.” *Genes and Development.* 20(16):2266–2278, 2006.

21. **WS Noble.** “What is a support vector machine?” *Nature Biotechnology.* 24(12):1565–1567, 2006.

22. T Mann, R Humbert, JA Stamatoyannopoulos and **WS Noble**. “Automated validation of polymerase chain reactions using amplicon melting curves.” *Journal of Bioinformatics and Computational Biology.* 22(14):350–358, 2006.

23. DP Lewis, T Jebara and **WS Noble**. “Support vector machine learning from heterogeneous data: an empirical analysis using protein sequence and structure.” *Bioinformatics.* 22(22):2753–2760, 2006.

24. M Tompa, N Li, TL Bailey, GM Church, B De Moor, E Eskin, AV Favorov, MC Frith, Y Fu, WJ Kent, VJ Makeev, AA Mironov, **WS Noble**, G Pavesi, G Pesole, M Régnier, N Simonis, S Sinha, G Thijs, J van Helden, M Vandenbogaert, Z Weng, C Workman, C Ye and Z Zhu. “Assessing computational tools for the discovery of transcription factor binding sites.” *Nature Biotechnology.* 23(1):137–144, 2005.

25. A Ben-Hur and **WS Noble**. “Kernel methods for predicting protein-protein interactions.” *Bioinformatics (Proceedings of the Intelligent Systems for Molecular Biology Conference).* 21(Suppl 1):i38–i46, 2005. Acceptance rate: 13%.

26. **WS Noble**, S Kuehn, R Thurman, R Humbert, JC Wallace, M Yu, M Hawrylycz and JA Stamatoyannopoulos. “Predicting the *in vivo* signature of human gene regulatory sequences.” *Bioinformatics (Proceedings of the Intelligent Systems for Molecular Biology Conference).* 21(Suppl 1):i338–i343, 2005. Acceptance rate: 13%.

27. **WS Noble**, R Kuang, C Leslie and J Weston. “Identifying remote protein homologs by network propagation.” *FEBS Journal.* 272(20):5119–5128, 2005.

28. J Weston, C Leslie, E Ie, D Zhou, A Eliseeff and **WS Noble.** “Semi-supervised protein classification using cluster kernels.” *Bioinformatics.* 21(15):3241–3247, 2005.

29. W Sheffler, E Upfal, J Sedivy and **WS Noble**. “A learned comparative expression measure for Affymetrix GeneChip DNA microarrays.” *Proceedings of the Computational Systems Bioinformatics Conference*, August 8-11, 2005, Stanford, CA. pp. 144-154. Acceptance rate: 12.2%.

30. T Mann, R Humbert, JA Stamatoyannopoulos and **WS Noble**. “Automated validation of polymerase chain reactions using amplicon melting curves.” *Proceedings of the Computational Systems Bioinformatics Conference*, August 8-11, 2005, Stanford, CA. pp. 377–385. Acceptance rate: 12.2%.

31. A Klammer, CW Wu, MJ MacCoss, **WS Noble**. “Peptide charge state determination for low-resolution tandem mass spectra.” *Proceedings of the Computational Systems Bioinformatics Conference*, August 8-11, 2005, Stanford, CA. pp. 175–185. Acceptance rate: 12.2%.

32. E Ie, J Weston, **WS Noble** and C Leslie. “Adaptive codes for multi-class protein classification.” *Proceedings of the International Conference on Machine Learning*, August 7-11, 2005, Bonn, Germany.

33. R Kuang, J Weston, **WS Noble** and C Leslie. “Motif-based protein ranking by network propagation.” *Bioinformatics.* 21(19):3711–3718, 2005.

34. JP Miller, RS Lo, A Ben-Hur, C Desmarais, I Stagljar, **WS Noble** and S Fields. “Large-scale identification of yeast integral membrane protein interactions.” *Proceedings of the National Academy of Science.* 102(34):12123–12128, 2005.

35. J Weston, A Elisseeff, D Zhou, CS Leslie and **WS Noble.** “Protein ranking: From local to global structure in the protein similarity network.” *Proceedings of the National Academy of Science*. 101(17):6559–6563, 2004.

36. C Leslie, E Eskin, A Cohen, J Weston and **WS Noble**. “Mismatch string kernels for discriminative protein classification.” *Bioinformatics.* 20(4):467–476, 2004.

37. P Pavlidis, I Wapinski and **WS Noble**. “Support vector machine classification on the web.” *Bioinformatics.* 20(4):586–587, 2004.

38. **WS Noble**. “Support vector machine applications in computational biology.” *Kernel Methods in Computational Biology*. B Schölkopf, K Tsuda and JP Vert, ed. MIT Press, 2004. pp. 71–92.

39. W Wu and **WS Noble**. “Genomic data visualization on the web.” *Bioinformatics.* 20(11):1804–1805, 2004.

40. K Tsuda and **WS Noble**. “Learning kernels from biological networks by maximizing entropy.” *Bioinformatics (Proceedings of the Intelligent Systems for Molecular Biology Conference).* 20(Suppl. 1):i326–i333, 2004. Acceptance rate: 10%.

41. J Weston, C Leslie, D Zhou and **WS Noble**. “Semi-supervised protein classification using cluster kernels.” *Advances in Neural Information Processing Systems 16*, 2004. pp. 595–602. Acceptance rate for spotlight presentation: 9.1%.

42. GRG Lanckriet, M Deng, N Cristianini, MI Jordan and **WS Noble**. “Kernel-based data fusion and its application to protein function prediction in yeast.” *Proceedings of the Pacific Symposium on Biocomputing*, January 3-8, 2004. pp. 300-311.

43. GRG Lanckriet, T De Bie, N Cristianini, MI Jordan and **WS Noble**. “A statistical framework for genomic data fusion.” *Bioinformatics.* 20(16):2626-2635, 2004.

44. H Lu, W Li, **WS Noble**, DG Payan and DC Anderson. “Riboproteomics of the hepatitis C virus internal ribosomal entry site.” *Journal of Proteome Research* 3(5):949–57, 2004.

45. E Feingold, PJ Good, . . . , **WS Noble**, . . . , FS Collins. “The ENCODE (ENCyclopedia Of DNA Elements) Project.” *Science.* 306:636–640, 2004.

46. P Pavlidis and **WS Noble**. “Matrix2png: A utility for visualizing matrix data.” *Bioinformatics.* 19(2):295-296, 2003.

47. NH Segal, P Pavlidis, **WS Noble**, CR Antonescu, A Viale, UV Wesley, K Busam, H Gallardo, D DeSantis, MF Brennan, C Cordon-Cardo, JD Wolchok and AN Houghton. “Classification of clear cell sarcoma as melanoma of soft parts by genomic profiling.” *Journal of Clinical Oncology.* 21:1775–1781, 2003.

48. DC Anderson, W Li, DG Payan and **WS Noble**. “A new algorithm for the evaluation of shotgun peptide sequencing in proteomics: support vector machine classification of peptide MS/MS spectra and SEQUEST scores” *Journal of Proteome Research.* 2(2):137–146, 2003.

49. NH Segal, P Pavlidis, CR Antonescu, RG Maki, **WS Noble**, JM Woodruff, JJ Lewis, MF Brennan, AN Houghton and C Cordon-Cardo. "Classification and subtype prediction of soft tissue sarcoma by functional genomics and support vector machine analysis." *American Journal of Pathology.* 169:691-700, 2003.

50. T Gururaja, W Li, **WS Noble**, DG Payan and DC Anderson. "Multiple functional categories of proteins identified in an *in vitro* cellular ubiquitin affinity extract using shotgun peptide sequencing." *Journal of Proteome Research.* 2:383–393, 2003.

51. P Pavlidis, Q Li and **WS Noble**. "The effect of replication on gene expression microarray experiments." *Bioinformatics.* 19(13):1620-1627, 2003.

52. J Qin, DP Lewis and **WS Noble**. "Kernel hierarchical clustering of microarray gene expression data." *Bioinformatics.* 19:2097-2014, 2003.

53. E Eskin, **WS Noble** and Y Singer. "Protein family classification using sparse Markov transducers." *Journal of Computational Biology*. 10(2):187–213, 2003.

54. L Liao and **WS Noble**. "Combining pairwise sequence similarity and support vector machines for detecting remote protein evolutionary and structural relationships." *Journal of Computational Biology.* 10(6):857–868, 2003.

55. C Leslie, E Eskin, J Weston and **WS Noble**. "Mismatch string kernels for SVM protein classification." *Advances in Neural Information Processing Systems 15*, 2003. pp. 1441–1448. Acceptance rate for oral presentation: 3.7%.

56. TL Bailey and **WS Noble**. "Searching for statistically significant regulatory modules." *Bioinformatics (Proceedings of the European Conference on Computational Biology).* 19(Suppl. 2):ii16–ii25, 2003. Acceptance rate: 22%.

57. SM Gomez, **WS Noble** and A Rzhetsky. "Learning to predict protein-protein interactions from protein sequences." *Bioinformatics (Proceedings of the Georgia Tech International Conference on Bioinformatics).* 19:1875–1881, 2003.

58. P Pavlidis, J Weston, J Cai and **WS Noble**. "Learning gene functional classifications from multiple data types." *Journal of Computational Biology*. 9(2):401-411, 2002.

59. C Leslie, E Eskin and **WS Noble**. "The spectrum kernel: An SVM-string kernel for protein classification." *Proceedings of the Pacific Symposium on Biocomputing*, January 2-7, 2002. pp. 564–575. Acceptance rate: 30%.

60. P Pavlidis, DP Lewis and **WS Noble**. "Exploring gene expression data with class scores." *Proceedings of the Pacific Symposium on Biocomputing*, January 2-7, 2002. pp. 474–485. Acceptance rate: 30%.

61. E Eskin, **WS Noble** and Y Singer. "Using substitution matrices to estimate probability distributions for biological sequences." *Journal of Computational Biology.* 9(6):775-791, 2002.

62. L Liao and **WS Noble**. "Combining pairwise sequence similarity and support vector machines for remote protein homology detection." *Proceedings of the Sixth International Conference on Computational Molecular Biology*, April 18-21, 2002. pp. 225–232. Acceptance rate: 30%.

63. B Schölkopf, J Weston, E Eskin, C Leslie and **WS Noble**. "A kernel approach for learning from almost orthogonal patterns." *Proceedings of the 13th European Conference on Machine Learning*, August 19-23, 2002. pp. 511-528.

64. P Pavlidis, TS Furey, M Liberto, D Haussler and **WN Grundy**. "Promoter region-based classification of genes." *Proceedings of the Pacific Symposium on Biocomputing*, January 3-7, 2001. pp. 151-163. Acceptance rate: 30%.

65. P Pavlidis, J Weston, J Cai and **WN Grundy**. "Gene functional classification from heterogeneous data." *Proceedings of the Fifth International Conference on Computational Molecular Biology*, April 21-24, 2001. pp. 242-248. Acceptance rate: 27%.

66. E Eskin, **WN Grundy** and Y Singer. "Using mixtures of common ancestors for estimating the probabilities of discrete events in biological sequences." *Proceedings of the Ninth International Conference on Intelligent Systems for Molecular Biology.* July 21-25, 2001. pp. 65-73. Acceptance rate: 21%.

67. P Pavlidis, C Tang and **WS Noble**. "Classification of genes using probabilistic models of microarray expression profiles." *Proceedings of BIOKDD 2001: Workshop on Data Mining in Bioinformatics.* August 26, 2001. pp. 15-21.

68. RA Muhle, P Pavlidis, **WN Grundy** and E Hirsch. "A high throughput study of gene expression in preterm labor using a subtractive microarray approach." *American Journal of Obstetrics and Gynecology.* 185(3):716-24, 2001.

69. P Pavlidis and **WS Noble**. "Analysis of strain and regional variation in gene expression in mouse brain." *Genome Biology.* 2(10): research0042.1-0042.15, 2001.

70. E Eskin, **WN Grundy** and Y Singer. "Protein family classification using sparse markov transducers." *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology*, August 20-23, 2000. pp. 134–145. Acceptance rate: 29%.

71. MPS Brown, **WN Grundy**, D Lin, N Cristianini, C Sugnet, TS Furey, M Ares, Jr. and D Haussler. "Knowledge-based analysis of microarray gene expression data by using support vector machines." *Proceedings of the National Academy of Science*. 97(1):262-267, 2000.

72. ME Baker, **WN Grundy** and CP Elkan. "A common ancestor for a subunit in the mitochondrial proton-translocating NADH:ubiquinone oxidoreductase (complex I) and short-chain dehydrogenases/reductases." *Cellular and Molecular Life Sciences*. 55(3):450-455, 1999.

73. **WN Grundy** and TL Bailey. "Family Pairwise Search with embedded motif models." *Bioinformatics*. 15(6):463-470, 1999.

74. **WN Grundy** and GJP Naylor. "Phylogenetic inference from conserved sites alignments." *Journal of Experimental Zoology*. 285(2):128-139, 1999.

75. TL Bailey and **WN Grundy**. "Classifying proteins by family using the product of correlated p-values." *Proceedings of the Third International Conference on Computational Molecular Biology*, April 11-14, 1999. pp. 10-14.

76. ME Baker, **WN Grundy** and CP Elkan. "Spinach CSP41, an mRNA-binding protein and ribonuclease, is homologous to nucleotide-sugar epimerases and hydroxysteroid dehydrogenases." *Biochemical and Biophysical Research Communications*. 248(2):250-254, 1998.

77. **WN Grundy**. "Homology detection via Family Pairwise Search." *Journal of Computational Biology*. 5(3):479-492, 1998.

78. **WN Grundy**. "Family-based homology detection via pairwise sequence comparison." *Proceedings of the Second Annual International Conference on Computational Molecular Biology*, March 22-25, 1998. pp. 94-100. Acceptance rate: 31%.

79. **WN Grundy**, TL Bailey, CP Elkan and ME Baker. "Meta-MEME: Motif-based hidden Markov models of protein families." *Computer Applications in the Biosciences*. 13(4):397-406, 1997.

80. **WN Grundy**, TL Bailey, CP Elkan and ME Baker. "Hidden Markov model analysis of motifs in steroid dehydrogenases and their homologs." *Biochemical and Biophysical Research Communications*. 231(3):760-766, 1997.

81. J Batali and **WN Grundy**. "Modeling the evolution of motivation." *Evolutionary Computation*. 4(3):235-270, 1997.

82. **WN Grundy**, TL Bailey and CP Elkan. "ParaMEME: a parallel implementation and a web interface for a DNA and protein motif discovery tool." *Computer Applications in the Biosciences*. 12(4):303-310, 1996.

C. Research support

Ongoing Research Support

R01 RR021692 (Noble) 8/1/05–6/30/09 1.8 calendar
NIH/NCRR $219,713
The MEME suite of motif-based sequence analysis tools.
This project supports, maintains and develops the MEME software suite of motif analysis software. This is a joint project with Tim Bailey at the University of Queensland, Australia. This proposal received a percentile ranking of 0.6%.
Role: PI

R01 EB007057 (Noble) 3/1/07–2/28/11 2.4 calendar
NIH $285,876
Machine learning analysis of tandem mass spectra
This project applies techniques and tools from the field of machine learning to the analysis of mass spectrometry data. The primary aim is to produce software that increases the sensitivity and specificity of protein identifications from complex mixtures.
Role: PI

R33 HG003070 (Noble) 9/1/04–8/31/07 1.8 calendar
NIH/NHGRI $385,569
Detecting relations among heterogeneous genomic datasets.
The long-term objective of this work is to provide a coherent computational framework for integrating and drawing inferences from a collection of genome-wide measurements. This project includes subcontracts at three other institutions.
Role: PI

U01 HG003161 (G Stamatoyannopoulos) 9/30/03-7/31/07 1.2 calendar
NIH/NHGRI $17,100 (subaward)
Identification of Functional DNA Elements by HSqPCR
This is an ENCODE project, aimed at detecting DNaseI hypersensitive sites *in vivo* using a high-throughput screen.
Role: Co-investigator

R01 GM071923 (J Stamatoyannopoulos) 9/1/04–8/31/09 1.2 calendar
NIH/NIGMS $93,665 (subaward)
Computational discovery of *cis*-regulatory sequences
This project uses quantitative chromatin profiling to identify *cis*-regulatory elements in a high-throughput fashion. This proposal received a percentile ranking of 2.2%.
Role: Co-investigator

P41 RR11823 (Davis) 9/1/04–8/31/11 1.2 calendar
NIH/NCRR $83,964 (subproject)
Comprehensive biology: Exploiting the yeast genome.
The mission of the YRC is to facilitate the identification and characterization of protein complexes in the yeast *Saccharomyces cerevisiae*.
Role: Co-investigator

R01 GM074257 (Leslie) 5/1/05–4/30/10 1.2 calendar
NIH/NCRR $31,004 (subcontract)
Recognizing protein folds with discriminative learning
This project develops discriminative methods for classifying proteins into structural families based upon their amino acid sequences.
Role: Co-investigator

P42 ES004696 (Checkoway) 5/1/06–3/31/11 0.48 calendar
NIH $7,100 (subaward)
Gene/Environment intteractions in Parkinson s disease
The major goal of this project is to investigate associations of Parkinson s disease risk with environmental factors.
Role: Co-investigator

Major Achievements

My research develops machine learning approaches and applies them to fundamental problems in biology, including recognizing remote protein homologies, inferring gene function and protein-protein interactions from heterogeneous data sets, predicting characteristics of local chromatin structure, and assigning peptides to tandem mass spectra.

- The RankProp algorithm exploits the global structure of the protein similarity network to identify remote protein homologies. This algorithm is very fast, dramatically outperforms the widely used PSI-BLAST algorithm and is in regular use via the UCSC Gene Sorter.
- The correct assignment of functional annotations to genes requires methods that consider diverse types of genomic and proteomic data. Our statistical framework combines semidefinite programming and the support vector machine (SVM) algorithm to solve this problem. We applied and extended these methods to assign gene function and to predict protein-protein interactions in yeast, mouse, worm and human.
- In the human genome, regions of local chromatin disruption can be identified by DNaseI hypersensitivity assays. We used the SVM to demonstrate the existence of DNA sequence patterns that correspond to these hypersensitive sites and to predict the locations of new sites. These predictions were subsequently validated by our collaborators via qPCR and Southern blot analyses.
- Machine learning dramatically improves the ability to correctly interpret tandem mass spectra. A semi-supervised learning method, the Percolator algorithm, more than doubles the number of spectra identified at a fixed false discovery rate, compared with state-of-the-art methods. Our collaborators now use Percolator on a daily basis, and we are preparing to disseminate the software more broadly.

Research statement

The trend in biology toward the development and application of high-throughput, genome- and proteome-wide assays necessitates an increased reliance upon computational techniques to organize and understand the results of biological experiments. Without appropriate computational tools, biologists cannot hope to fully understand, for example, a complete genome sequence or a library of microarray expression profiles. My research focuses on the development and application of methods for interpreting complex biological data sets. These methods may be used, for example, to uncover distant structural and functional relationships among protein sequences, to identify transcription factor binding site motifs, to classify cancerous tissues on the basis of microarray mRNA expression profiles, to predict properties of local chromatin structure from a given DNA sequence, and to accurately map tandem mass spectra to their corresponding peptides.

The goals of my research program are to develop and apply powerful new computational methods to gain insights into the molecular machinery of the cell. In selecting research areas to focus on, I am drawn to research problems in which I can solve fundamental problems in biology while also pushing the state of the art in machine learning.

Pattern recognition in diverse and heterogeneous genomic and proteomic data sets

Genome sciences is, in many ways, a data-driven enterprise because available technologies define the types of questions that we can ask. Each assay DNA sequencing, mRNA expression microarrays, the yeast two-hybrid screen provides one view of the molecular activity within the cell. An ongoing theme in my research is the integration of heterogeneous data sets, with the aim of providing a unified interpretation of the underlying phenomenon. We focus, in particular, on inferring gene function and on predicting protein-protein interactions. For example, to determine whether a given target pair of proteins interact, we take into account direct experimental evidence in the form of a yeast two-hybrid assay or tandem affinity purification followed by mass spectrometry. In addition, we consider as evidence the sequence similarity between the target pair of proteins and one or more pairs of proteins that are known to interact with one another, the similarity of the target proteins mRNA expression profiles or chip-ChIP expression profiles, and evidence of cellular colocalization. We have developed a statistical inference framework that considers all of these sources of evidence, taking into account dependencies among them and weighting each type of evidence according to its relevance and its trustworthiness.

Much of my research program relies on a class of methods, developed recently in machine learning, known as *kernel methods* [43]. An algorithm is a kernel method if it relies on a particular type of function (the *kernel function*) to define similarities between pairs of objects. For these algorithms, a data set of N objects can be sufficiently represented using an N-by-N matrix of kernel values. The kernel matrix thereby provides a mechanism for representing diverse data types using a common formalism.

In collaboration with a variety of research groups, we have demonstrated the broad applicability of kernel methods to problems in genomics and proteomics, focusing on a particular kernel method known as the support vector machine (SVM) [6]. The SVM is a kernel-based classification algorithm that boasts strong theoretical underpinnings [47] as well as state-of-the-art performance in a variety of bioinformatics applications [30]. We have shown that

- SVMs can successfully classify yeast genes into functional categories on the basis of microarray expression profiles [7] or motif patterns within promoter sequences [35, 48].
- SVMs can discriminate with high accuracy among subtypes of soft tissue sarcoma on the basis of microarray expression profiles [45, 44]. Our SVM classifier provided strong evidence for several previously described histological subtypes, and suggested that a subset of one controversial subtype exhibits a consistent genomic signature.
- A series of SVM-based methods can recognize protein folds and remote homologs [28, 26, 27, 49]. Our early work in this area set the baseline against which much subsequent work was compared, including many SVM-based classifiers that derive from our work [4, 23, 8, 11, 33, 34, 40, 42]. Our recent work continues to provide the best known performance on this task [19, 29].
- SVMs have been applied to a variety of applications within the field of tandem mass spectrometry, including re-ranking peptide-spectrum matches produced by a database search algorithm [1, 20] and discriminating between 2+ and 3+ charged spectra [21].
- SVMs can draw inferences from heterogeneous genomic and proteomic data sets. We first demonstrated how to infer gene function from a combination of microarray expression profiles and phylogenetic profiles [36], and we subsequently described a statistical framework for learning relative weights for each data set with respect to a given inference task [25, 24] (see figure, top). Recently, we used this framework to predict protein-protein interactions [5] and protein co-complex relationships [39] from heterogeneous data sets.

The SVM is now one of the most popular methods for the analysis of biological data sets: Pubmed includes 204 papers published within the last 12 months whose abstracts contain the phrase support vector machine, and 691 such papers in the last five years. *Nature Biotechnology* recently invited me to write a primer on SVMs [31]. My research bears considerable responsibility for the SVM s popularity, because I have repeatedly demonstrated the power and flexibility of this algorithm in new bioinformatics domains.

In the future, we will focus on methods that combine the inference power of the SVM with the probabilistic framework of Bayesian networks to infer gene function and protein-protein interactions. A Bayesian network is a formal graphical representation of a joint probability distribution over a collection of random variables. For example, we are using Bayesian network models to calibrate and combine the results of a large collection of SVMs with respect to the Gene Ontology (GO). The GO consists of three directed acyclic graphs, in which each node represents a particular term describing cellular localization, molecular function or biological process or pathway. In our hybrid model, each node in the GO is populated by a collection of SVMs, one per kernel. The probability model is responsible for calibrating the SVM outputs, weighting them with respect to one another, and ensuring that the predictions with respect to a particular gene respect the constraints of the GO network topology. Adding a Bayesian network to the SVM has several important advantages, including allowing a principled method for handling missing data, providing a complementary means of encoding prior knowledge, and providing a model that gives explanations for its predictions.

In addition to improving our analytical methods, we will expand our methods to handle new types of biological data, as well as to make predictions on functional elements other than protein-coding genes. We are developing new kernels to represent known and predicted protein structures, as well as a kernel derived from a dynamic Bayesian network that describes transmembrane protein topology. We will build predictive models that incorporate the diverse functional data being generated by the ENCODE consortium [12], of which my group is a member. The data sets include DNaseI hypersensitivity, methylation, origins of replication, replication timing, transcription factor binding sites, promoters, protein-coding genes, histone modifications, and phylogenetic conservation. A unique aspect of the ENCODE data is that they are collected from a common set of cell lines. Using these data, we will characterize protein function with respect to these cell lines. Furthermore, ENCODE aims to identify all functional genomic elements, including protein-coding as well as non-protein-coding genes, plus various classes of cis-regulatory elements. Hence, we will also expand our predictive models to assign functional labels and to identify interactions among other types of genomic functional elements.

The relationships among chromatin, primary DNA sequence and gene regulation

DNA in the nucleus of the cell is bound in a complex and dynamic molecular structure known as chromatin. Over the past several years, my research group has investigated the relationships among the primary DNA sequence, nucleosomes, *cis*-regulatory factors and higher-order chromatin structure. Initially, we focused on local disruptions of chromatin structure known as DNaseI hypersensitive sites (DHSs), because these sites are a prerequisite for any type of *cis*-regulatory activity, including enhancers, silencers, insulators, and boundary elements. We demonstrated that DHSs exhibit a distinct sequence signature, which can be used to predict with high accuracy hypersensitive locations in the human genome [32]. We used these signatures to predict novel hypersensitive sites, which were then validated via qPCR and Southern blot analysis by our collaborators in the lab of John Stamatoyannopoulos. More recently, we demonstrated that the converse phenomenon, well-positioned nucleosomes, can be predicted with high accuracy [37]. We also collaborated with several research groups in the development of high-throughput assays for interrogating local chromatin structure in the human genome [41, 10].

Our work on chromatin structure has been carried out within the context of the ENCODE consortium [12]. With the Stamatoyannopoulos lab, we led the analysis performed within the chromatin and replication subgroup of ENCODE, developing tools to integrate data on DNaseI sensitivity, replication timing, histone modifications, bulk RNA transcription, and regulatory factor binding region density. We also combined wavelet analyses and hidden Markov models [9] to simultaneously visualize and segment multiple genomic data sets at a variety of scales. Using these tools, our analyses led to the following conclusions, which are reported in the forthcoming ENCODE paper [13], as well as in a companion paper [46]:

- Chromatin accessibility and histone modification patterns are highly predictive of both the presence and activity of transcription start sites.
- Distal DNaseI hypersensitive sites have characteristic histone modification patterns that reliably distinguish them from promoters; some of these distal sites show marks consistent with insulator function.
- DNA replication timing is correlated with chromatin structure.
- Larger-scale relationships between chromatin accessibility and histone modifications are dominated by sub-regions in which higher average DNaseI sensitivity is accompanied by high levels of H3K4me2, H3K4me3 and H3ac modifications (see figure, bottom).
- At smaller scales (< 2 kb), DNaseI hypersensitive sites and peaks in histone acetylation are not strongly correlated. This conclusion is surprising given the widely accepted notion that histone acetylation has a central role in mediating chromatin accessibility by disrupting higher-order chromatin folding.
- Histone modifications, DNaseI sensitivity, replication, transcript density and protein factor binding are organized systematically across the genome into active domains, generally corresponding to domains with high levels of H3ac and RNA transcription, low levels of H3K27me3 marks, and early replication timing, and repressed domains with low H3ac and RNA, high H3K27me3, and late replication (see figure, center right)

In the future, my research in this area will follow three complementary threads. First, we will develop and apply algorithms for characterizing the motif composition of DHSs. My PhD research focused on algorithms for identifying and searching with protein and DNA sequence motifs [17, 15, 16, 3], and I have continued to work in this area, developing new statistical methods for searching for *cis*-regulatory modules [2] and for quantifying similarity between motifs [18]. We recently used our methods to identify a yeast transcription factor (Hcm1) that fills the S phase gap in the transcriptional circuitry of the cell [38]. We expect DHSs to be significantly enriched for transcription factor binding sites; therefore, we will search our growing library of DHSs, using known motifs as well as *de novo* motif discovery algorithms and taking into account the observed degree of evolutionary conservation. Using a new oligonucleotide tiling array assay, our collaborators have performed genome-wide assays in 12 different cell lines, and this data set will continue to grow. In any single tissue, only a small portion of observed DHSs are constitutively active. Hence, we are particularly interested in segregating the DHSs according to their tissue specificity, and according to the mRNA expression profiles of their proximal genes, thereby identifying motifs that are tissue- or condition-specific.

Second, we will develop methods that classify DHSs according to their function. DHSs may perform a variety of *cis*-regulatory roles, as enhancers, silencers, insulators, boundary elements, etc. We showed that the pattern of histone modifications around a given DHS can be used to predict, with high accuracy, whether the DHS is proximal to or distal from a transcription start site. This method has the potential to uncover previously unrecognized transcription start sites in the human genome. These results also suggest that, by exploiting complementary information such as patterns of evolutionary conservation, histone modifications and transcription factor binding data from chip-ChIP experiments, an automated classification algorithm such as the SVM will be able to further distinguish among functional classes of DHSs.

Third, we will continue to investigate the large-scale properties of chromatin structure. We will develop more sophisticated probabilistic models designed specifically to segment DNaseI sensitivity data, and we will further explore the patterns of local and regional correlation among sensitivity to cleavage by DNaseI or micrococcal nuclease and patterns of histone modifications.

Analysis of mass spectrometry data

Mass spectrometry promises to enable scientists to identify and quantify the entire complement of molecules that comprise a complex biological sample. In biomedicine, mass spectrometry is commonly used in a high-throughput fashion to identify proteins in a mixture. However, the primary bottleneck in this type of experiment is computational. Existing algorithms for interpreting mass spectra are slow and fail to identify a large proportion of the given spectra.

My research group focuses on the peptide identification problem. Given a fragmentation mass spectrum and the sequence of the proteome from which it was derived, the task is to identify the particular peptide that generated the observed spectrum. Initially, we showed how to use an SVM classifier to post-process the output of an existing database search algorithm, discriminating between correct and incorrect peptide-spectrum matches (PSMs) [1]. More recently, we demonstrated

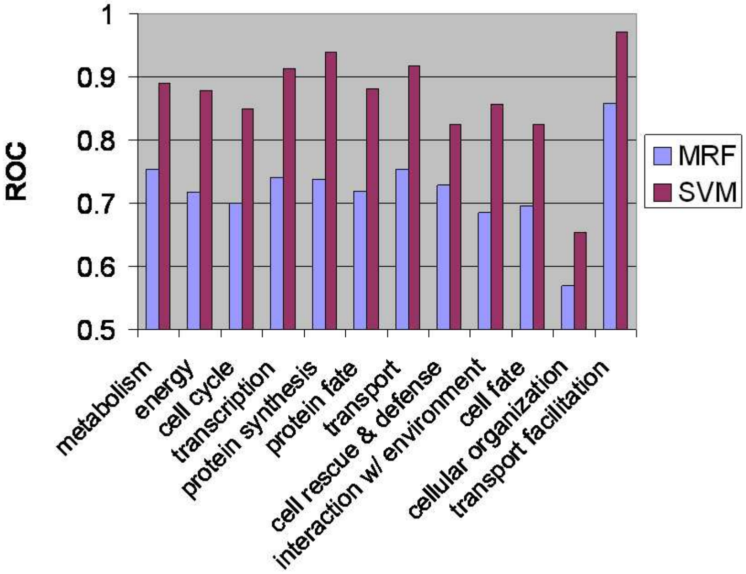
- how to speed up the database search procedure by predicting the charge state of an observed spectrum [21],
- how to search against a database of previously identified spectra [14], and
- how to exploit chromatographic retention time to eliminate a large proportion of false positive PSMs [22].

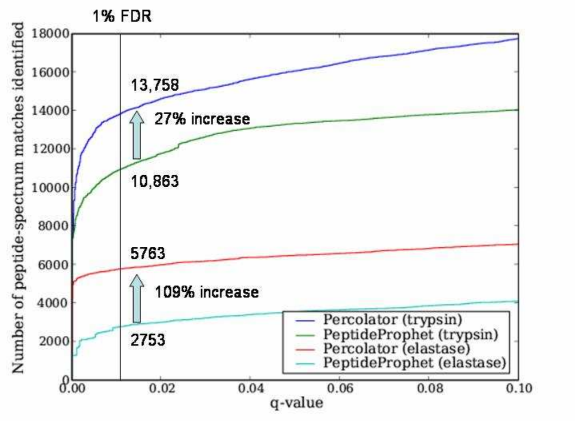
We maintain and support software packages that implement these algorithms, and these software are currently used by a variety of academic and commercial research groups.

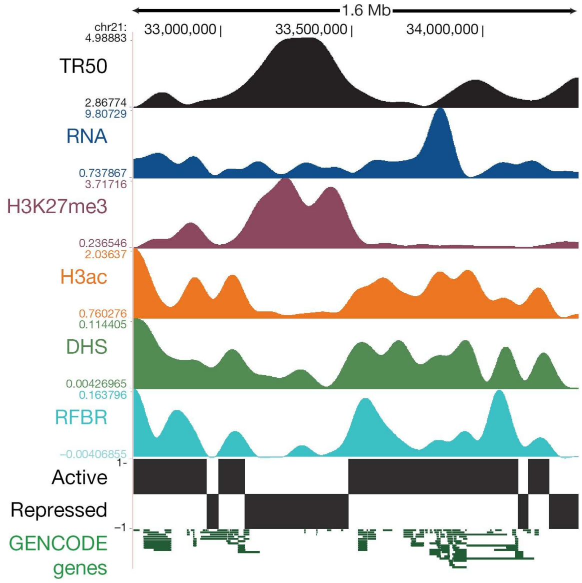
One significant challenge in applying machine learning to mass spectra is the variability of the data due to different types of samples (*e.g.*, soluble versus membrane proteins), enzyme specificity, modified versus unmodified peptides, mass spectrometer type, database size, instrument calibration, etc. Recently, we successfully addressed this problem by applying a technique known as semi-supervised learning to the classification of PSMs [20]. In semi-supervised learning, the training set consists of two subsets of examples, one subset with labels and one without. In this application, we search a given set of spectra against two databases, the real (target) database and a shuffled (decoy) version of the same database. PSMs against the decoy database can be confidently labeled as incorrect identifications, but PSMs against the target database are comprised of a mixture of correct and incorrect identifications. We designed an iterative, semi-supervised algorithm in which the inner loop is an SVM classifier. The algorithm, called Percolator, can be applied to any given mass spectrometry data set, learning model parameters that are appropriate for those data. Relative to a state-of-the-art fully supervised machine learning method, this semi-supervised approach more than doubles the number of correctly identified peptides for some data sets (see figure, center left). Because the manuscript describing Percolator is under review, we have distributed the software only to our collaborators, Michael MacCoss and Christine Wu. They use the software routinely in their labs, consistently obtaining large increases in peptide identifications.

We are currently developing a Bayesian network that models the fragmentation of the peptide within the mass spectrometer. This model captures a variety of known characteristics of the peptide fragmentation process, including for example, the tendency of the peptide backbone to cleave preferentially near some types of amino acids. The goal of this work is to improve our ability to predict, from the peptide sequence, the heights of individual peaks within the corresponding mass spectrum. The model parameters can provide insight into fragmentation biochemistry, and the model itself can be used to increase the accuracy of peptide identification algorithms and to pre-select the highest peaks in a spectrum for monitoring in a targeted proteomics experiment.

In the future, we to extend our models using techniques borrowed from the fields of speech recognition and natural language processing. Lattice models, which are used in these fields to represent the grammar of a natural language, can naturally encode the relationships among common peptides and proteins. Currently, a mass spectrometrist identifies proteins in a complex mixture by first mapping individual spectra to their respective peptides, and then inferring the protein identities from the collection of identified spectra. Lattice models will enable us to perform protein identification in a single, efficient procedure. Furthermore, well established methods exist for allowing minor violations of a pre-specified grammar. Using these techniques will allow us to perform *de novo* protein identification, in which the algorithm can identify proteins that are not members of the given protein database. This ability is critical to identify polymorphisms and post-translational modifications.


ROC
1
0.9
0.8
0.7
0.6
0.5
MRF
SVM
metabolism
energy
cell cycle
transcription
protein synthesis
protein fate
transport
cell rescue & defense
interaction w/ environment
cell fate
cellular organization
transport facilitation


1% FDR
Number of peptide-spectrum matches identified
18000
16000
14000
12000
10000
8000
6000
4000
2000
0
13,758
27% increase
10,863
5763
109% increase
2753
Percolator (trypsin)
PeptideProphet (trypsin)
Percolator (elastase)
PeptideProphet (elastase)
0.00
0.02
0.04
0.06
0.08
0.10
q-value


1.6 Mb
chr21:
33,000,000
33,500,000
34,000,000
4.98883
TR50
2.86774
9.80729
RNA
0.737867
3.71716
H3K27me3
0.236546
2.03637
H3ac
0.760276
0.114405
DHS
0.00426965
0.163796
RFBR
-0.00406855
1-
Active
Repressed
-1
GENCODE
genes

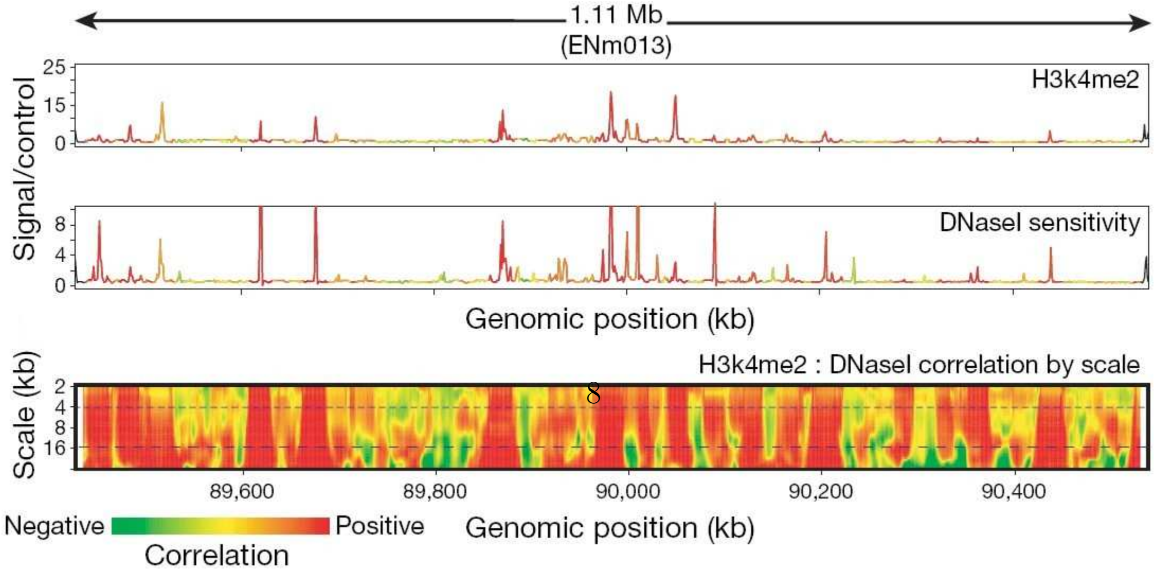

1.11 Mb
(ENm013)
Signal/control
25
15
0
H3k4me2
8
4
0
DNaseI sensitivity
Genomic position (kb)
H3k4me2 : DNaseI correlation by scale
Scale (kb)
2
4
8
16
89,600
89,800
90,000
90,200
90,400
Negative
Positive
Correlation
Genomic position (kb)

Figure captions

Top: Predicting yeast gene function from heterogeneous data. The height of each bar is proportional to the cross-validated receiver operating characteristic score for prediction of the given class of yeast genes. The figure compares the performance of a previously published Markov random field method (MRF) and our SVM-based method. In every case, the SVM significantly outperforms the MRF. [25].

Center left: Comparison of mass spectrum peptide identification methods. The figure plots the number of spectra identified, as a function of false discovery rate, for two data sets and two analysis methods. For typical data, digested with the standard enzyme trypsin, Percolator method improves the identification rate by 27% at a 1% false discovery rate. When we switch to a non-standard enzyme, elastase, Percolator yields more than twice as many identifications. [20]

Center right: Concordance of multiple data types for an illustrative ENCODE region (ENM005). The tracks labeled Active and Repressed are derived from a simultaneous HMM segmentation of eight data types: replication time (TR50), bulk RNA transcription (RNA), histone modifications H3K27me3 and H3ac, DHS density and regulatory factor binding region density (RFBR). [13]

Bottom: Wavelet correlations of histone marks and DNaseI sensitivity. The relationship between histone modification H3K4me2 and DNaseI sensitivity is shown for ENCODE region ENm013. The top two curves are colored with the strength of the local correlation at the 4-kb scale. Below, the same data are represented as a wavelet correlation. The y axis shows the differing scales decomposed by the wavelet analysis from large to small scale (in kb); the color at each point in the heatmap represents the level of correlation at the given scale, measured in a 20 kb window centered at the given position. [13]

References

[1] D. C. Anderson, W. Li, D. G. Payan, and W. S. Noble. A new algorithm for the evaluation of shotgun peptide sequencing in proteomics: support vector machine classification of peptide MS/MS spectra and SEQUEST scores. *Journal of Proteome Research*, 2(2):137 146, 2003.

[2] T. L. Bailey and W. S. Noble. Searching for statistically significant regulatory modules. *Bioinformatics*, 19(Suppl. 2):ii16 ii25, 2003.

[3] M. E. Baker, W. N. Grundy, and C. P. Elkan. Spinach CSP41, an mRNA-binding protein and ribonuclease, is homologous to nucleotide-sugar epimerases and hydroxysteroid dehydrogenases. *Biochemical and Biophysical Research Communications*, 248(2):250 254, 1998.

[4] A. Ben-Hur and D. Brutlag. Remote homology detection: a motif based approach. *Proceedings of the Eleventh International Conference on Intelligent Systems for Molecular Biology*, 19 suppl 1:i26 i33, 2003.

[5] A. Ben-Hur and W. S. Noble. Kernel methods for predicting protein-protein interactions. *Bioinformatics*, 21 suppl 1:i38 i46, 2005.

[6] B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In D. Haussler, editor, *5th Annual ACM Workshop on COLT*, pages 144 152, Pittsburgh, PA, 1992. ACM Press.

[7] M. P. S. Brown, W. N. Grundy, D. Lin, N. Cristianini, C. W. Sugnet, T. S. Furey, M. Ares, Jr., and D. Haussler. Knowledge-based analysis of microarray gene expression data using support vector machines. *Proceedings of the National Academy of Sciences of the United States of America*, 97(1):262 267, 2000.

[8] S. Busuttil, J. Abela, and G. J. Pace. Support vector machines with profile-based kernels for remote protein homology detection. *Genome Informatics*, 15(2):191 200, 2004.

[9] N. Day, A. Hemmaplardh, R. E. Thurman, J. A. Stamatoyannopoulos, and W. S. Noble. Unsupervised segmentation of continuous genomic data. *Bioinformatics*, 2007.

[10] J. H. Dennis, H. Fan, S. M. Reynolds, G. Yuan, J. C. Meldrim, D. J. Richter, D. G. Peterson, O. J. Rando, W. S. Noble, and R. E. Kingston. Independent and complementary methods for large-scale structural analysis of mammalian chromatin. *Genome Research*, 2007. To appear.

[11] Q. W. Dong, X. L. Wang, and L. Lin. Application of latent semantic analysis to protein remote homology detection. *Bioinformatics*, 22(3):285 290, 2006.

[12] ENCODE Consortium. The ENCODE (ENCyclopedia Of DNA Elements) project. *Science*, 306(5696):636 640, 2004.

[13] ENCODE Consortium. Identification and analysis of functional elements in 1% of the human genome by the ENCODE pilot project. *Nature*, 2007. To appear.

[14] B. E. Frewen, G. E. Merrihew, C. C. Wu, W. S. Noble, and M. J. MacCoss. Analysis of peptide MS/MS spectra from large-scale proteomics experiments using spectrum libraries. *Analytical Chemistry*, 78(5678 5684), 2006.

[15] W. N. Grundy. *A Bayesian Approach to Motif-based Protein Modeling.* PhD thesis, University of California, San Diego, La Jolla, CA, 1998.

[16] W. N. Grundy, T. L. Bailey, C. P. Elkan, and M. E. Baker. Hidden Markov model analysis of motifs in steroid dehydrogenases and their homologs. *Biochemical and Biophysical Research Communications*, 231(3):760 766, 1997.

[17] W. N. Grundy, T. L. Bailey, C. P. Elkan, and M. E. Baker. Meta-MEME: Motif-based hidden Markov models of protein families. *Computer Applications in the Biosciences*, 13(4):397 406, 1997.

[18] S. Gupta, J. A. Stamatoyannopoulos, T. L. Bailey, and W. S. Noble. Quantifying similarity between motifs. *Genome Biology*, 8:R24, 2007.

[19] E. Ie, J. Weston, W.S. Noble, and C. Leslie. Adaptive codes for multi-class protein classification. In *Proceedings of the International Conference on Machine Learning*, 2005.

[20] L. Kall, J. Canterbury, J. Weston, W. S. Noble, and M. J. MacCoss. A semi-supervised machine learning technique for peptide identification from shotgun proteomics datasets. Submitted, 2007.

[21] A. A. Klammer, C. C. Wu, M. J. MacCoss, and W. S. Noble. Peptide charge state determination for low-resolution tandem mass spectra. In *Proceedings of the Computational Systems Bioinformatics Conference*, pages 175 185, 2005.

[22] A. A. Klammer, X. Yi, M. J. MacCoss, and W. S. Noble. Peptide retention time prediction yields improved tandem mass spectrum identification for diverse chromatography conditions. Submitted, 2006.

[23] R. Kuang, E. Ie, K. Wang, K. Wang, M. Siddiqi, Y. Freund, and C. Leslie. Profile-based string kernels for remote homology detection and motif extraction. *Journal of Bioinformatics and Computational Biology*, 3(3):527 550, 2005.

[24] G. R. G. Lanckriet, T. De Bie, N. Cristianini, M. I. Jordan, and W. S. Noble. A statistical framework for genomic data fusion. *Bioinformatics*, 20(16):2626 2635, 2004.

[25] G. R. G. Lanckriet, M. Deng, N. Cristianini, M. I. Jordan, and W. S. Noble. Kernel-based data fusion and its application to protein function prediction in yeast. In R. B. Altman, A. K. Dunker, L. Hunter, T. A. Jung, and T. E. Klein, editors, *Proceedings of the Pacific Symposium on Biocomputing*, pages 300 311. World Scientific, 2004.

[26] C. Leslie, E. Eskin, and W. S. Noble. The spectrum kernel: A string kernel for SVM protein classification. In R. B. Altman, A. K. Dunker, L. Hunter, K. Lauderdale, and T. E. Klein, editors, *Proceedings of the Pacific Symposium on Biocomputing*, pages 564 575, New Jersey, 2002. World Scientific.

[27] C. Leslie, E. Eskin, J. Weston, and W. S. Noble. Mismatch string kernels for SVM protein classification. In Suzanna Becker, Sebastian Thrun, and Klaus Obermayer, editors, *Advances in Neural Information Processing Systems*, pages 1441 1448, Cambridge, MA, 2003. MIT Press.

[28] L. Liao and W. S. Noble. Combining pairwise sequence similarity and support vector machines for remote protein homology detection. In *Proceedings of the Sixth Annual International Conference on Computational Molecular Biology*, pages 225 232, Washington, DC, April 18 21 2002.

[29] I. Melvin, E. Ie, R. Kuang, J. Weston, W. S. Noble, and C. Leslie. SVM-fold: a tool for discriminative multi-class protein fold and superfamily recognition. *BMC Bioinformatics*, 2007. To appear.

[30] W. S. Noble. Support vector machine applications in computational biology. In B. Schoelkopf, K. Tsuda, and J.-P. Vert, editors, *Kernel methods in computational biology*, pages 71 92. MIT Press, Cambridge, MA, 2004.

[31] W. S. Noble. What is a support vector machine? *Nature Biotechnology*, 24(12):1565 1567, 2006.

[32] W. S. Noble, S. Kuehn, R. Thurman, R. Humbert, J. C. Wallace, M. Yu, M. Hawrylycz, and J. Stamatoayannopoulos. Predicting the *in vivo* signature of human gene regulatory sequences. *Bioinformatics*, 2005. To appear.

[33] H. Ogul and E. U. Mumcuoglu. SVM-based detection of distant protein structural relationships using pairwise probabilistic suffix trees. *Computational and Biological Chemistry*, 30(4):292 299, 2006.

[34] H. Ogul and E. U. Mumcuoglu. A discriminative method for remote homology detection based on n-peptide compositions with reduced amino acid alphabet. *Biosystems*, 87(1):75 81, 2007.

[35] P. Pavlidis, T. S. Furey, M. Liberto, D. Haussler, and W. N. Grundy. Promoter region-based classification of genes. In R. B. Altman, A. K. Dunker, L. Hunter, K. Lauderdale, and T. E. Klein, editors, *Pacific Symposium on Biocomputing 2001*, pages 151 163, Singapore, 2001. World Scientific.

[36] P. Pavlidis, J. Weston, J. Cai, and W. N. Grundy. Gene functional classification from heterogeneous data. In *Proceedings of the Fifth Annual International Conference on Computational Molecular Biology*, pages 242 248, 2001.

[37] H. E. Peckham, R. E. Thurman, Y. Fu, J. A. Stamatoyannopoulos, W. S. Noble, K. Struhl, and Z. Weng. Nucleosome positioning signals in genomic DNA. *Genome Research*, 2007. To appear.

[38] T. Pramila, W. Wu, W. S. Noble, and L. L. Breeden. The Forkhead transcription factor Hcm1 regulates chromosome segregation genes and fills the S phase gap in the transcriptional circuitry of the cell cycle. *Genes and Development*, 20(16):2266 2278, 2006.

[39] J. Qiu and W. S. Noble. Predicting co-complexed protein pairs from heterogeneous data. Submitted, 2007.

[40] H. Rangwala and G. Karypis. Profile based direct kernels for remote homology detection and fold recognition. *Bioinformatics*, 21:4239 4247, 2005.

[41] P. J. Sabo, R. Humbert, M. Hawrylycz, J. C. Wallace, M. O. Dorschner, M. McArthur, and J. A. Stamatoyannopoulos. Genome-wide identification of DNase1 hypersensitive sites using active chromatin sequence libraries. *Proceedings of the National Academy of Sciences of the United States of America*, 101(13):4537 4542, 2004.

[42] H. Saigo, J.-P. Vert, N. Ueda, and T. Akutsu. Protein homology detection using string alignment kernels. *Bioinformatics*, 20(11):1682 1689, 2004.

[43] B. Scholkopf, C. J. C. Burges, and A. J. Smola, editors. *Advances in Kernel Methods: Support Vector Learning*. MIT Press, Cambridge, MA, 1999.

[44] N. H. Segal, P. Pavlidis, C. R. Antonescu, R. G. Maki, W. S. Noble, J. M. Woodruff, J. J. Lewis, M. F. Brennan, A. N. Houghton, and C. Cordon-Cardo. Classification and subtype prediction of soft tissue sarcoma by functional genomics and support vector machine analysis. *American Journal of Pathology*, 2003. To appear.

[45] N. H. Segal, P. Pavlidis, W. S. Noble, C. R. Antonescu, A. Viale, U. V. Wesley, K. Busam, H. Gallardo, D. DeSantis, M. F. Brennan, C. Cordon-Cardo, J. D. Wolchok, and A. N. Houghton. Classification of clear cell sarcoma as melanoma of soft parts by genomic profiling. *Journal of Clinical Oncology*, 21:1775 1781, 2003.

[46] R. E. Thurman, N. Day, W. S. Noble, and J. A. Stamatoyannopoulos. Identification of higher-order functional domains in the human ENCODE regions. *Genome Research*, 2007. To appear.

[47] V. N. Vapnik. *Statistical Learning Theory*. Adaptive and learning systems for signal processing, communications, and control. Wiley, New York, 1998.

[48] J.-P. Vert, R. Thurman, and W. S. Noble. Kernels for gene regulatory regions. In Y. Weiss, B. Scholkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 1401 1408, Cambridge, MA, 2006. MIT Press.

[49] J. Weston, C. Leslie, D. Zhou, and W. S. Noble. Semi-supervised protein classification using cluster kernels. In *Advances in Neural Information Processing Systems 16*, pages 595 602, 2004.

WN Grundy, TL Bailey, CP Elkan and ME Baker. Meta-MEME: Motif-based hidden Markov models of protein families. *Computer Applications in the Biosciences.* 13(4):397-406, 1997.

This article describes an initial version of the Meta-MEME software toolkit, which was one of the primary products of my dissertation research. Meta-MEME builds motif-based models of DNA and protein sequences. These models can be used to produce multiple alignments of protein or DNA sequences, to search for remote protein homologs and to search for novel *cis*-regulatory modules (CRMs). In 1997, we did not have a collection of well-annotated CRMs that was large enough to validate our models; therefore, in this article, we focus on modeling proteins.

Meta-MEME uses hidden Markov models (HMMs), which became popular for computational biology applications in the mid-1990s. An HMM captures, within a probabilistically rigorous framework, the properties of a given collection of related DNA or protein sequences, including position-specific probabilities of mutations, insertions and deletions.

Relative to previous types of HMMs, Meta-MEME models are smaller, focusing on the pattern of biologically significant motifs that characterize a protein family. These motifs may be identified in any fashion. In this work, we use the MEME motif discovery algorithm. This paper demonstrates that building models with fewer parameters is beneficial, particularly when searching for remote protein homologies.

In addition to being technically innovative, Meta-MEME is practically useful. The software is described in textbooks (e.g., Figure 2 is reproduced in the popular textbook by David Mount), has been commercially licensed and continues to be widely used. I currently have NIH R01 funding (with Tim Bailey as co-PI) to continue to maintain and improve MEME and Meta-MEME.

DC Anderson, W Li, DG Payan and WS Noble. A new algorithm for the evaluation of shotgun peptide sequencing in proteomics: support vector machine classification of peptide MS/MS spectra and SEQUEST scores *Journal of Proteome Research.* 2(2):137 146, 2003.

In the analysis of complex biological samples using shotgun mass spectrometry, the primary computational challenge is to identify the peptide corresponding to each observed fragmentation spectrum. This identification is typically performed by searching the observed spectrum against a database of theoretical spectra and selecting the closest matching pair.

This paper describes one of the first attempts to use machine learning to improve the rate of successful protein identifications. A database search program such as SEQUEST or Mascot produces as output a collection of peptide-spectrum matches (PSMs), one per spectrum. The majority of these PSMs represent incorrect matches because the true peptide is not in the given database due to polymorphisms or post-translational modifications, the search procedure incorrectly identified the best-matching peptide, or the spectrum was generated by non-peptide contaminants or by a heterogeneous mixture of peptides. We train a classification algorithm known as a support vector machine (SVM) to discriminate between correct and incorrect PSMs. The input to the classifier is a vector of 13 scores, representing various characteristics of the PSM. To train and evaluate the method, we use cross-validation on a gold standard derived from a mixture of purified proteins. We measure our method s ability to rank correct PSMs above incorrect PSMs, and we show that the SVM dramatically outperforms two other methods the score function used by SEQUEST and a probabilistic score function called QScore.

To our knowledge, this is the second published application of machine learning to the analysis of shotgun mass spectrometry data. The first was published while our manuscript was in press, and describes a method (PeptideProphet) that is similar to ours but uses a different classification algorithm (linear discriminant analysis rather than the SVM) and a smaller feature space (four dimensions rather than 13). Both of these papers contine to be widely cited (ours was cited 23 times in 2006 2007), and the general idea of using machine learning methods for mass spectrometry analysis is now gaining in popularity. I recently received NIH R01 funding to continue this line of work.

GRG Lanckriet, T De Bie, N Cristianini, MI Jordan and WS Noble. A statistical framework for genomic data fusion. *Bioinformatics.* 20(16):2626-2635, 2004.

One of the core problems in bioinformatics is reconciling the various views of the cell that are provided by different types of high-throughput assays. In this paper, we describe a framework for representing and reasonining about heterogeneous data sets, and we demonstrate how the framework can be used to infer gene function from a data set consisting of protein sequences, hydrophobicity profiles, protein-protein interaction data, and gene expression data.

This work demonstrates how to use the support vector machine (SVM) algorithm to classify heterogeneous genomic data. The work also describes a method, using semidefinite programming, to simultaneously learn the parameters of the SVM classifier as well as the relative weights of the input kernels. Thus, for example, the method learns to assign a large weight to microarray gene expression data when classifying yeast ribosomal proteins, whereas a larger weight is assigned to the protein sequence when classifying membrane versus non-membrane proteins.

In a separate paper, we compared the performance of our method to that of a state-of-the-art Markov random field method. Across 12 diverse functional classes of yeast genes, our approach significantly outperforms the MRF. We have subsequently applied a variant of this SVM-based framework to the prediction of gene function in the mouse genome, and to the prediction of protein-protein interactions in yeast, worm and human.

J Weston, A Elisseeff, D Zhou, CS Leslie and WS Noble. Protein ranking: From local to global structure in the protein similarity network. *Proceedings of the National Academy of Science.* 101(17):6559 6563, 2004.

A strong analogy can be drawn between searching a database of proteins for homologs of a query protein and searching the World-Wide Web for web pages that are relevant to a given query word or phrase. The WWW can be represented as a network of pages connected by hyperlinks, and the protein database can be represented as a network of proteins connected by edges representing pairwise sequence similarity. The query is a single phrase or a query protein, and the output is a ranked list of web pages or target proteins.

The power of the Google web search engine derives in large part from its ranking algorithm, PageRank, which captures global properties of the WWW network topology.

In this paper, we describe the RankProp algorithm, which uses a similar insight in the context of protein database searching. Initially, we define a protein similarity network by using PSI-BLAST in an all-versus-all fashion. RankProp takes this network as input, with one node designated as the query. RankProp performs a diffusion operation on the network, pumping activation scores outward from the query and ranking the target proteins with respect to the amount of activation score that they receive during the diffusion. The algorithm produces rankings that improve dramatically with respect to the initial PSI-BLAST rankings, when evaluated using a gold standard derived from proteins of known structure. Although the all-versus-all PSI-BLAST computation is expensive, this operation can be performed once and the results stored. RankProp itself is efficient, requiring approximately one minute for a database of 100,000 proteins.

RankProp has been implemented and made available via the UCSC Gene Sorter, and we are currently developing a stand-alone RankProp web server

WS Noble, S Kuehn, R Thurman, R Humbert, JC Wallace, M Yu, M Hawrylycz and JA Stamatoyannopoulos. Predicting the *in vivo* signature of human gene regulatory sequences. *Bioinformatics (Proceedings of the Intelligent Systems for Molecular Biology Conference).* 21(Suppl 1):i338 i343, 2005.

In the nucleus of the living cell, DNA is packaged into a complex molecular structure known as chromatin. When regulatory proteins bind to the DNA strand, the chromatin experiences local disruptions, or openings. DNaseI is an endonuclease that cleaves DNA non-specifically and that can be used to identify these sites using Southern blot, qPCR or microarray-based assays. Canonically, a DNaseI hypersensitive site (DHS) is ∼250 bp in length, and DNaseI hypersensitivity is considered a prerequisite for any type of regulatory activity, including promoters, enhancers, silencers, boundary elements, etc.

In this work, we demonstrate that DNaseI hypersensitive sites can be reliably identified in the human genome using a purely computational assay. The method employs a supervised classification algorithm known as a support vector machine (SVM) to learn to discriminate between DHS and non-DHS sequences. In a cross-validated test, the accuracy of the resulting classifier is 85%. We also performed prospective validation on 146 sites with SVM probabilities > 80%. After qPCR and Southern blot analyses, we found 74% of the predicted sites to be hypersensitive, compared with 5.3% in a randomly selected sample of sites.

This work suggests that nucleosome positioning relative to hypersensitive sites is at least partially determined by the DNA sequence. In addition, the SVM classifier itself is useful in the search for regulatory motifs and to increase the throughput of targeted assays for DNaseI hypersensitive sites.

CABIOS

Vol. 13 no. 4 1997
Pages 397-406

Meta-MEME: Motif-based hidden Markov models of protein families

William N.Grundy[3], Timothy L.Bailey[2], Charles P.Elkan and Michael E.Baker[1]

Abstract

Motivation: *Modeling families of related biological sequences using Hidden Markov models (HMMs), although increasingly widespread, faces at least one major problem: because of the complexity of these mathematical models, they require a relatively large training set in order to accurately recognize a given family. For families in which there are few known sequences, a standard linear HMM contains too many parameters to be trained adequately.*

Results: *This work attempts to solve that problem by generating smaller HMMs which precisely model only the conserved regions of the family. These HMMs are constructed from motif models generated by the EM algorithm using the MEME software. Because motif-based HMMs have relatively few parameters, they can be trained using smaller data sets. Studies of short chain alcohol dehydrogenases and 4Fe-4S ferredoxins support the claim that motif-based HMMs exhibit increased sensitivity and selectivity in database searches, especially when training sets contain few sequences.*

Availability: *http://www.sdsc.edu/MEME*

Contact: *bgrundy@cs.ucsd.edu*

Introduction

A hidden Markov model describes a series of observations by a 'hidden' stochastic process. Although introduced relatively recently to computational molecular biology (Churchill, 1989), HMMs have been in use for speech recognition for many years (Baker, 1975). In speech recognition, the series of observations being modeled is a spoken utterance; in computational biology, the series of observations is a biological sequence. One immediately apparent difference between these two domains is the amount of available training data. Training sets for state-of-the-art speech recognition systems can contain many gigabytes of recorded speech; in contrast, families of related biological sequences usually consist of kilobytes or even hundreds of bytes of characters. Even for speech recognition systems, for which the training set size is relatively large, researchers attempt to simplify their models in order to reduce the number of trainable parameters (Woodland *et al.*, 1994). When modeling biological sequences, the need for smaller models is even more pronounced. This paper addresses that need by developing hidden Markov models which precisely model only the highly conserved regions of a family of sequences.

These motif-based HMMs consist primarily of motif models generated by MEME (Multiple EM for Motif Elicitation) (Bailey and Elkan, 1995a; Bailey and Elkan, 1995b). Meta-MEME is a software tool for combining MEME motif models within a standard linear HMM framework. Because Meta-MEME operates in an automated fashion, it is particularly useful for analyzing the increasingly large sequence databases becoming available.

In addition to being trainable from smaller data sets, motif-based HMMs are well suited for recognizing distant homologies. By modeling the spacer regions between motifs in a very simple way, these models selectively discard information from the training set about the contents of spacer regions. This discarding of information is beneficial for distantly related sequences, because distant homologs typically show conservation only in functionally or structurally important portions of their sequences. Meta-MEME focuses on these regions and does not attempt to model the less-conserved, intermediate regions in detail.

In many ways, Meta-MEME resembles the BLOCKS method for protein family classification (Henikoff and Henikoff, 1994b; Henikoff and Henikoff, 1996). The BLOCKMAKER program discovers highly conserved regions of protein families by combining motifs found by either the MOTIF algorithm (Smith *et al.*, 1990) or the Gibbs sampling algorithm (Lawrence *et al.*, 1993). Individual blocks may be represented as ungapped position-specific scoring matrices, similar to the motif models created by MEME. However, MEME is more likely than BLOCKMAKER to split a motif in two if any of the sequences contains an insertion or deletion, so MEME motifs tend to be shorter than BLOCKMAKER blocks. Since motifs (and blocks) are supposed to model ungapped regions, MEME generally produces more accurate models. The BLOCKS database (Blocks, 1996) contains, for each known protein family, an ordered set of blocks along with the minimum and maximum observed spacings between the blocks in the training set. The BLIMPS program (Henikoff *et al.*, 1995) searches this database using a single sequence

Department of Computer Science and Engineering and [1]Department of Medicine, University of California, San Diego, La Jolla, CA 92093, [2]San Diego Supercomputer Center, PO Box 85608, San Diego, CA 92186, USA

[3]To whom correspondence should be addressed

as a query, thus taking into account the order and spacing of blocks. Clearly, Meta-MEME and the BLOCKS method share many features. In general, however, a hidden Markov model approach is more attractive because of its well-founded underlying probabilistic theory.

Hidden Markov models

A hidden Markov model is a mathematical framework which models a series of observations based upon a hypothesized, underlying but hidden process. The model consists of a set of states and transitions between these states. Each state emits a signal based upon a set of emission probabilities and then stochastically transitions to some other state, based upon a set of transition probabilities. These two probability distributions, when combined with the initial state distribution, completely characterize an HMM.

A useful HMM tutorial was written by Rabiner (Rabiner, 1995), and more detailed information is available in (Rabiner and Juang, 1993). The tutorial describes three basic problems for HMMs: given an observation sequence and a model, how do we (1) efficiently compute the probability of the observation sequence, given the model, (2) choose a corresponding state sequence which is optimal in some meaningful sense (i.e., best 'explains' the observations), and (3) adjust the parameters of the model to maximize the probability of the sequence, given the model? In computational biology, an HMM models a family of related sequences. Thus, Rabiner's three problems correspond to (1) determining whether a given sequence belongs to the modeled family, (2) finding an alignment of the given sequence to the rest of the family, and (3) training the model based upon known members of the family.

Standard HMMs for molecular biology

Hidden Markov models were first applied to problems in molecular biology by (Churchill, 1989). (Krogh *et al.*, 1994) applied HMMs to protein modeling and brought widespread recognition to the approach. We refer to the linear HMMs described in that paper as 'standard HMMs'. The structure of these HMMs attempts to reflect the process of evolution.

The core of the standard model is a sequence of states, called 'match states', which represent the canonical sequence for this family. Each match state corresponds to one position in the canonical sequence. This series of states is similar to a profile (Gribskov *et al.*, 1990), since each state contains a frequency distribution across the entire alphabet. The probabilities that a given state emits each possible base are taken from this frequency distribution and are called the 'emission probabilities' for that state.

To model the process of evolution, two additional types of states—insert and delete states—are included in the HMM. One delete state lies in parallel with each match state and allows the match state to be skipped. Since delete states do not emit characters, aligning a sequence to a delete state corresponds to the sequence having a deletion at that position. Insert states with self-loops are juxtaposed between match states, allowing one or more bases to be inserted between two match states. These three series of states are connected as shown in Figure 1. The topology of the model is linear: once a state has been traversed, it cannot be entered a second time. Although this type of model may fail to accurately model genetic copying events, the enforced linearity allows for efficient training of the models.

Standard HMMs have been most successfully applied to the task of recognizing families of proteins containing a relatively large number of known sequences (Krogh *et al.*, 1994; Baldi *et al.*, 1994; Eddy, 1995). For families for which fewer known sequences are known, a standard HMM contains too many parameters to be trained to precision. A standard HMM of length n using an alphabet of size 20 contains 6 transition probabilities and 19 match state emission probabilities for each of n positions, as well as 19 insert state emission probabilities, yielding a total of $25n + 19$ trainable parameters. For a short sequence of length 100, such a model contains 2519 parameters. Many small families of biological sequences contain less than this number of characters in all known family members combined.

Small families such as these cannot effectively train a standard linear HMM because reliable training requires that the number of samples greatly exceeds the number of free

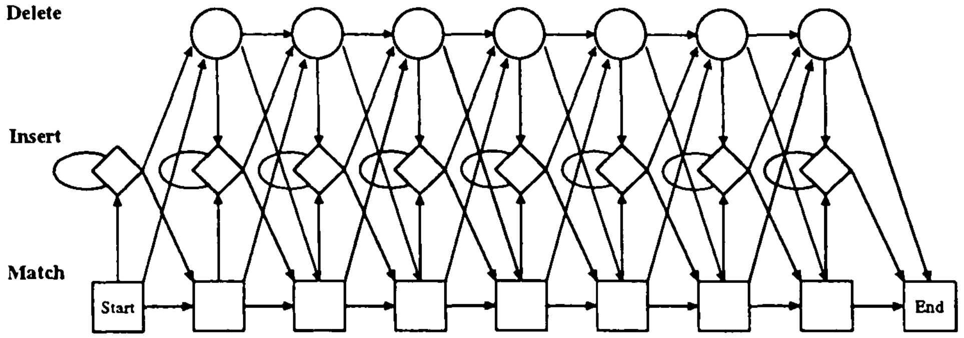


Fig. 1. Outline of the topology of a standard linear HMM. Emission probability distributions for match and insert states are not shown.

parameters. For example, (Krogh *et al.*, 1994) mention a lower limit of approximately 70 carefully selected training sequences in order to adequately model the globin family. A model based upon a smaller data set may overfit the data, modelling details specific to the training set but not to the larger protein family. In order to avoid overfitting, standard HMMs often rely upon a set of Bayesian prior probabilities (Brown *et al.*, 1995; Sjolander *et al.*, 1996). In this case, however, with a small training set and a large model, the trained model may depend upon the prior probabilities more than it reflects the training sequences. The only effective means of ensuring that the trained model reflects the characteristics of a particular protein family is to keep the number of model parameters small.

Searching using HMMs

Having constructed an HMM, the model can be applied to the task of recognizing a family of biological sequences in a sequence database. An ideal HMM would pick out all and only the members of the family from the rest of the database. This database search can be carried out using existing software. Two standard HMM packages are freely available, SAM (Hughey and Krogh, 1996; SAM, 1996) and HMMER (Eddy, 1995; HMMER, 1996). Although the SAM package allows for slightly more complicated models, HMMER is more appropriate for our needs because it includes a variety of searching algorithms.

The results of HMM searches may be compared using a modified form of the receiver operating characteristic (ROC), which we describe in more detail below. We have performed a series of such searches on two different families, using varying training set sizes. The data from these searches show that, for the data sets we investigated, motif-based HMMs perform as well as standard HMMs for large training set sizes and significantly outperform standard HMMs for smaller training sets.

Algorithm

Overview of the algorithm

Meta-MEME is a software tool for creating hidden Markov models which focus on highly conserved regions, called motifs. Because of their relatively small size, these motif-based HMMs address the problems caused by insufficient training data.

Meta-MEME currently uses motif models as generated by MEME, a tool which uses expectation-maximization to discover motifs in sets of DNA or protein sequences. Given such a set of sequences, MEME outputs one or more probabilistic models of motifs found in the data. The models consist of a frequency matrix and are therefore similar to a gapless profile. A parallelized version of MEME running on a supercomputer is available on the World-Wide Web (Grundy *et al.*, 1996; ParaMEME, 1996).

MEME motifs provide reliable indicators of family membership. If trained on a set of related sequences, MEME will build motif models of the most highly conserved regions in that data set. For related sequences, these highly conserved regions represent evidence of the sequences' shared evolutionary history. A candidate sequence which closely matches the other members of the family in motif regions is much more likely to be homologous than a candidate for which the match lies in a region of lower conservation. The motifs therefore provide a concise signature for the family. Because MEME can find such signatures, it is a powerful tool for recognizing families of proteins. Hidden Markov models provide a framework for combining MEME motifs into an even more accurate and precise recognition tool.

Meta-MEME extends the MEME software to build sequence-length models, rather than models of single motifs. Meta-MEME generates models by first finding a set of motif models and then combining these models within a linear HMM framework. The MAST software, as described below, is used to search a database, finding a schema representing the canonical order and spacing of motifs within the family.

The motif-based hidden Markov models constructed by Meta-MEME are a simplified form of the standard HMM (see Figure 2). The motifs themselves allow neither gaps nor insertions; thus, each motif is modeled by a sequence of match states, with transition probabilities of 1.0 between adjacent states.

The regions between motifs are not modeled very precisely, since the contents of these spacer regions are not highly conserved. Each spacer region is modeled using a single insert state. The transition probabilities into this state and on the state's self-loop are calculated such that the expected length of the emission from this state equals the length of the corresponding spacer region in the canonical motif occurrence schema. The insert state's emission probability distribution is set to a uniform distribution, but this distribution is ignored by the HMMER search tools described below. In effect, then, each spacer region is modeled by a single length parameter. A model of length n containing m motifs therefore contains $19n$ match state emission probabilities and $m + 1$ transition probabilities, for a total of $19n + m + 1$ trainable parameters. In practice, this number will be much smaller than the corresponding number for standard HMMs, since motif-based HMMs contain far fewer match states.

The length of the spacer region is not highly constrained by the model. An insert state gives an exponentially decaying distribution of spacer lengths. For spacers of any appreciable length, that distribution is very flat. Thus, the model should be fairly resilient to insertions or deletions within the spacer regions.

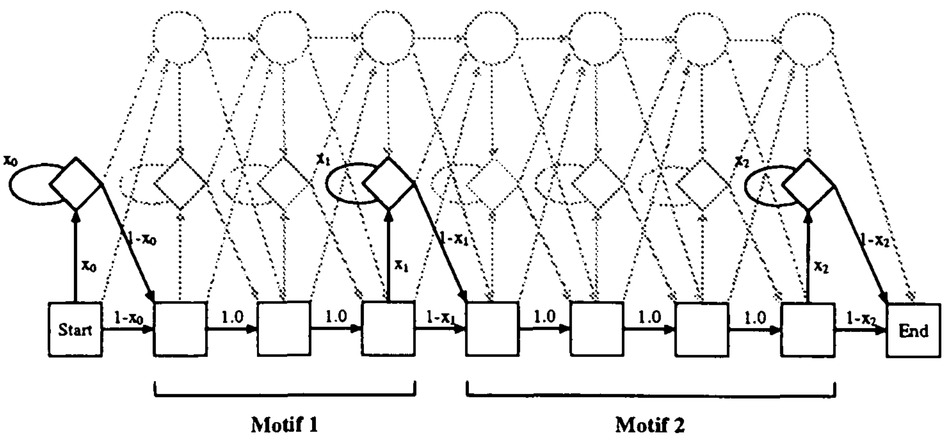


Fig. 2. A small motif-based HMM. Only the darker nodes and transitions are used in the model; the gray background nodes would appear in a standard HMM but are unreachable in this HMM. Note that this is a simplified example; real motifs generated by MEME are longer.

MEME parameters

One of Meta-MEME's primary goals is to operate in a completely unsupervised fashion. While it might be possible and even desirable in many cases to build expert human knowledge into the model of a particular family, the increasing quantity of sequence data available precludes such an approach in general. We have therefore run MEME using its default parameters, as specified on the ParaMEME web site. Specifically, we use the ZOOPS motif occurrence model, which stands for 'zero or one occurrence per sequence'. Note that, although the resulting model is tuned to find motifs which appear no more than once in each sequence, it may still find repeated motifs. We use Dirichlet mixtures for prior probabilities, modified by the megaprior heuristic (Bailey and Gribskov, 1996). The minimum width of a motif is specified as 12 (although the motifs returned may be shorter than this, due to a shortening heuristic in MEME), and the maximum width is 55.

Selecting motifs: a majority heuristic

In order for Meta-MEME to build multi-motif models from MEME output in an unsupervised way, the program must decide automatically how many motifs to use. To do so, Meta-MEME uses a simple heuristic. As MEME generates successive motifs for a data set, it first finds the highly significant motifs and then begins to model motifs which are conserved in only a subset of the given sequences. In effect, MEME finds motifs representing subfamilies of the given family. Since such subfamily motifs are not useful for characterizing the entire family, they should not be included in the Meta-MEME model. Models generated by Meta-MEME, therefore, only incorporate those motifs for which the motif occurs in the majority of the training sequences, up to a maximum of six motifs.

Finding the canonical motif occurrence schema

Once the motif models have been generated by MEME and selected according to the majority occurrence heuristic, they must be combined into a single model. In order to use the standard HMM framework, the motifs must be arranged in a linear fashion. Ideally, the order and spacing of motifs should reflect the canonical order and spacing of motifs in the family. The Motif Annotation and Search Tool (MAST) (Bailey and Gribskov, 1997) is part of the MEME software distribution (MEME, 1996). MAST searches a database for motif occurrences and assigns a score to each sequence based upon the sequence's most likely match to each of the given motifs. The sequences from the database with statistically significant matches to the given set of motifs are returned as part of the MAST output. For each such sequence, MAST produces a motif occurrence schema which shows the motif occurrences with p-values less than 0.0001, as well as the lengths of the spaces between occurrences. Meta-MEME searches this output for the highest-scoring sequence containing significant matches to each of the motifs selected for use in the HMM. The motif occurrence schema associated with this sequence is then used as the canonical schema.

Calculating spacer state transition probabilities

The transition probabilities for insert states between motifs must be calculated such that the expected spacer lengths correspond to the values in the canonical motif occurrence schema. Consider an HMM state for which the incoming transition probability is x, the outgoing transition probability is $1 - x$, and the probability of a self-loop is x. Let n be the number of times the node is visited. Then the expected number of visits, μ, to such a node is, by definition,

$$\mu = \sum_{n=0}^{\infty} n(1 - x)x^n \qquad (1)$$

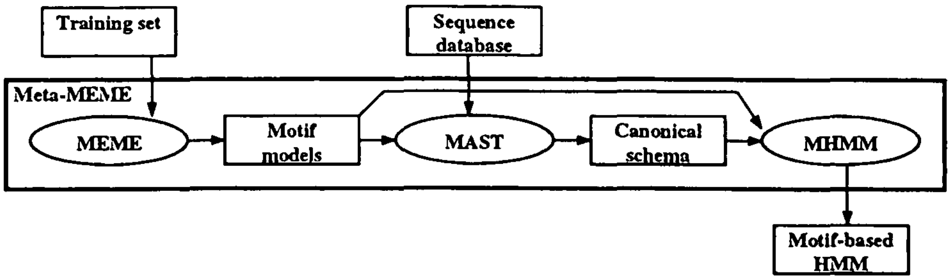


Fig. 3. A schematic diagram of Meta-MEME. The primary inputs are a set of sequences and a sequence database. The program produces a linear HMM of the given family in ASCII HMMER format.

At first there are two possibilities: visit the node with probability x, or skip it with probability $1 - x$. Skipping the node gives a spacer of length 0, while visiting it gives a spacer length 1 plus the expected remaining path length, ν. So we have

$$\mu = (1 - x)0 + x(1 + \nu) \qquad (2)$$

Because of the Markov property, regardless of the path length so far, if we reach this node again then the expected path length from it is simply μ. So we have

$$\mu = x(1 + \mu) \qquad (3)$$

Solving for x yields

$$x = \mu/(1 + \mu) \qquad (4)$$

This equation is used to calculate transition probabilities for spacer states.

A schematic diagram of Meta-MEME is shown in Figure 3. Given a set of motif models and the canonical sequences, the program *mhmm* calculates the appropriate spacer state transition probabilities and writes out a linear, motif-based HMM in HMMER format.

Results

Data sets

We first applied Meta-MEME to a group of dehydrogenases that includes mammalian 11β-hydroxysteroid and 17β-hydroxysteroid dehydrogenase and their homologs in the short chain alcohol dehydrogenase family. We chose this data set because it is large and phylogenetically diverse (Persson *et al.*, 1991; Baker, 1994; 1996), providing a good test of the sensitivity and selectivity of Meta-MEME on a protein family of biological interest.

The thirty-eight sequences used in the training set are listed in Appendix A. Pairwise alignments of almost all of these sequences are less than 30% identical after using gaps and insertions to maximize identities. Many sequences are less than 20% identical after use of gaps and insertions. These thirty-eight sequences represent a small portion of the approximately 650 known dehydrogenases in genpept release 95 (GenBank, 1996).

We also applied Meta-MEME to a set of 4Fe-4S ferredoxins. The family members are listed in Appendix B. These 159 sequences comprise all known 4Fe-4S ferredoxins in SWISSPROT release 33 (Bairoch, 1994). Family members were selected using PROSITE 13.1 (Bairoch, 1992). Ten additional members were added to the family, based upon ROC analysis and sequence comparisons. The SWISSPROT identifiers for all 159 sequences, as well as the justifications for including the ten additional sequences, are given in Appendix B. Nested training sets were selected at random from all 159 sequences, without regard to sequence similarity.

Creating standard linear HMMs

The standard linear HMMs used for comparison with Meta-MEME were constructed using the default settings of the HMMER program *hmmt*, version 1.8. The training algorithm begins with a uniform model with length equal to the average length of sequences in the training set. The model is trained via expectation-maximization, using a simulated annealing protocol to avoid local optima. The initial Boltzmann temperature is 5.0, with a temperature decrease of 5% at each iteration.

Smith/Waterman search

Numerous algorithms exist for searching a database using a hidden Markov model. HMMER offers four such programs, which vary in the way they match sequences against models. The first, *hmmsw*, performs a local Smith/Waterman search for matches of a partial sequence to a partial model; *hmms* matches a complete model against complete sequences; *hmmls* matches a complete model against one or more partial sequences; and *hmmfs* matches fragments of a model to multiple non-overlapping partial sequences. Informal experiments with these programs yielded consistently better results using *hmmsw*.

In the best case, a database search with an HMM would return sequence scores which ranked all of the family members above all of the non-family members. However, all of the HMMER programs suffered from intermediate-scoring sequence fragments. When a sequence fragment exists in the database, it will match only a portion of the model, giving a relatively low score. Then, even though the fragment is a member of the family, it may be ranked among the non-family members.

Because sequence fragments are a deficiency of the database rather than of the search method, and because many fragments are redundant with the whole sequences included in the database, we opted to filter such fragments from the database. Rather than use a fixed threshold for all models, we calculated from the canonical motif signature the minimum length of a sequence containing two motifs and two spacers. All sequences in the database shorter than this value are filtered out. The filtered database is then used for both the Meta-MEME search and the standard HMM search.

Comparing search results: ROC_{50}

We compare search results using a modified form of the receiver operating characteristic. The ROC curve plots true positives as a function of true negatives using a continuously varying decision threshold. The area under this curve, the ROC value, combines measures of a search's selectivity and sensitivity into a single value. Unfortunately, for large database searches, the number of negatives far exceeds the number of positives, so ROC values must be computed to a high degree of precision. A similar statistic, ROC_{50} (Gribskov and Robinson, 1996), provides a wider spread of values. ROC_{50} is the area under the ROC curve plotted until 50 false positives are found. This value has the advantages of being easier to compute, of requiring less storage space, and of corresponding to the typical biologist's willingness to sift through only approximately fifty false positives. ROC_{50} scores are normalized to range from 0.0 to 1.0, with 1.0 corresponding to the most sensitive and selective search.

Short-chain alcohol dehydrogenases

Figure 4(a) shows that Meta-MEME outperforms standard linear HMMs for most subsets of the dehydrogenase training set, with the most striking difference between the two methods appearing for smaller data sets. Each series in the figure represents the average of ten successions of training and testing runs, using randomly selected, nested subsets of the 38-sequence training set. Error bars represent standard error. For each subset of sequences, a standard and a motif-based HMM were built and were used to search genpept 95. Not only does Meta-MEME consistently score better than the standard linear HMMs, the motif-based HMMs appear to be more robust across different random subsets, as evidenced by the relative smoothness of the Meta-MEME curve.

Figure 5 shows an 'alignment' of four different motif-based HMMs, built from nested subsets of the dehydrogenase training set. These motifs illustrate the biological basis for the sensitivity of Meta-MEME. Motifs 1 and 2 are part of the nucleotide cofactor binding site (Branden and Tooze, 1991; Wierenga *et al.*, 1985; Wierenga *et al.*, 1986); motif 3 is part of the catalytic site. A protein sequence that had, for example, motifs 1 and 3 interchanged would not have the same 3D structure and could not function as a steroid dehydrogenase. By scoring protein similarity and dissimilarity on the basis of motif order and spacing, Meta-MEME effectively models spatial information in the 3D structure of the canonical dehydrogenase. This information differentiates homologs from unrelated proteins which contain isolated fragments resembling sequences in the training set. Comparison of protein 3D structures is the most sensitive method for determining homology (Chothia and Lesk, 1986). This explains Meta-MEME's excellent ability to recognize alcohol dehydrogenase homologs as seen in Figure 4(a).

The motifs discovered using smaller training sets correspond strongly to the original motifs found using the largest training set. In the figure, motifs are numbered consecutively according to the order in which they were discovered. Any

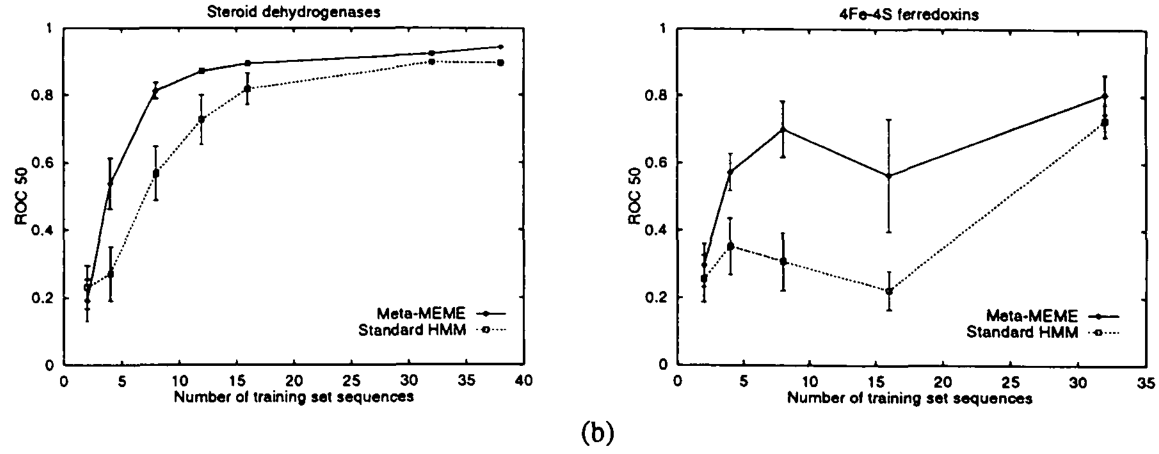


Fig. 4. Comparison of Meta-MEME and standard linear HMMs in recognizing (a) short chain alcohol dehydrogenases and (b) 4Fe-4S ferredoxins. Each point represents an average of ten separate runs, except for the ferredoxin runs using 16-sequence training sets, for which only three runs completed (see the discussion below). Error bars represent standard error.

38 sequences: 9-[2]-64-[1]-12-[6]-17-[4]-9-[3]-73

```
---------LVTGAASGIG----------------------------------------------------------
------VDVLVNNAG*------------EDWDRVIxVNLTGVF*-----------------GRIVNVSSVAG-----
----YSASKAAVxGLTRSLALELAPxGIRVNVVAPG-----------------------------------------
--------------------------------
```

16 sequences: 5-[2]-61-[1]-42-[4]-12-[3a]-5-[3b]-33-[5]-13

```
****-----LVTGASRGIG****------------------------------------------------------
-------DVLVNNAG****------------------------------------------GRIVNVSS--------
----YSASKAALxGLTRSLALE-----IRVNAVAPGFVxTDM---------------------------------FL
ASDEASYIT-------------**********
```

8 sequences: 11-[2]-65-[1]-64-[3a]-22-[3b]-26-[7]-28

```
-----------TGASSGIG----------------------------------------------------------
-------DVLVNNAG**------------------------------------------------------------
----YAASKAAL----------------------PGxIxTDM--------------------------IPIGRMGQP
EEIA---------------------------*
```

4 sequences: 13-[1]-18-[6]-37-[3a]-22-[3b]-41

```
***********************************************************************------
-------DALINNAG------------------VFHINVVGPIR---------------------------------
----YxMSKAAL----------------------PGWVxTDM-----------------------------------
------**************************
```

Fig. 5. Comparison of four motif-based HMMs built from a nested series of random subsets of the 38-sequence dehydrogenase training set. The canonical schema for each model is shown, with the lengths of spacers alternating with motif numbers in brackets. In the models, motifs are represented by their consensus sequence. Hyphens ('-') represent the expected length of spacers generated by insert nodes, and asterisks ('*') are gaps inserted into this diagram in order to align the models.

motif from one training set which overlaps with a motif from a previous training set is assigned the same number as the first. Using the largest training set, MEME finds five motifs which appear in more than half of the training set. The third of these motifs, however, is very long (32 residues); in subsequent analyses using smaller data sets, motif 3 gets split into two halves (marked 3a and 3b). Furthermore, motif 5, which was discarded because of the majority occurrence heuristic in the 38-sequence analysis, is found and included in the HMM based upon sixteen sequences. Motif 6 is lost when the training set is reduced from thirty-eight to sixteen sequences but is recovered when the training set size reaches 4 sequences. Motifs 4 and 5 are lost between sixteen and eight sequences, and motif 2 is lost when four sequences are used. Only one new motif (marked 7) is introduced in the smaller training sets; other candidates are discarded because of the majority occurrence heuristic.

The order and spacing of the motifs within the different models is also conserved. In all four models, the order of motifs is identical. Furthermore, spaces between motifs are consistent across the four models. In the figure, hyphens represent spacer states in the model, whereas asterisks represent 'gaps', which were inserted into the figure in order to align the motifs. Very few asterisks were required in order to generate a perfect alignment. Only the last model, based upon four training sequences, contains a significant missing portion.

The motif-based HMMs are considerably smaller than their standard HMM counterparts. For the dehydrogenase family, the average model from Meta-MEME contains 58 states; the standard models average 264 states. Assuming six motifs per model, the average Meta-MEME model therefore contains $(19 * 58) + 6 + 1 = 1109$ trainable parameters. The standard HMM, by contrast, averages $25 * 264 = 6600$ parameters. The standard model is therefore 6.0 times as large as the motif-based model.

4Fe-4S ferredoxins

A similar set of experiments was conducted using the 4Fe-4S ferredoxin data set. In addition to using a different, considerably smaller family, the ferredoxin searches were carried out on a different database, SWISSPROT 33 instead of genpept 95. Nonetheless, Meta-MEME again consistently outperforms the standard HMMs, as shown in Figure 4(b). The degree of separation between the two series is even greater than for the dehydrogenases. The standard HMMs of the ferredoxin family are on average 5.1 times as large as the average motif-based HMM.

Although Meta-MEME outperforms standard HMMs, both methods perform more poorly for ferredoxin data sets of size 16 than for smaller, 8- or 4-sequence data sets. This anomaly results from the interaction of two of the heuristics described above. For many of the 16-sequence data sets, the majority occurrence heuristic selected a relatively large number of motifs. Unfortunately, it was often impossible for MAST to locate a single sequence containing all of these motifs. Consequently, a canonical motif occurrence schema was found for only three of the runs. As a result, neither Meta-MEME nor HMMER completed the other runs, since the filtering of the database depends upon the canonical schema. This adverse interaction of heuristics only occurred with the ferredoxin data set and only with training sets of size 16. A variant of our heuristics would overcome this problem; however, our emphasis in this work is to demonstrate the general utility of motif-based HMMs. Rather than fine-tuning heuristics, future work will replace these heuristics by, for example, completely connecting the motifs and learning the occurrence schema from the given data.

Discussion

Results from Meta-MEME are encouraging. As expected, motif-based HMMs discriminate better than their standard linear counterparts for the two protein families we investigated, yet due to their small size, motif-based HMMs require fewer training sequences in order to be trained to precision. Furthermore, since HMM search algorithms are generally linear in the size of the model, motif-based HMMs can search a database 5–6 times faster than a standard model. By focusing its models on highly conserved regions of the training set, Meta-MEME effectively ignores noisy portions of the data, thereby allowing the software to recognize distant homologs. Finally, because Meta-MEME operates in an unsupervised fashion, the software is appropriate for the analysis of large databases, where domain-specific expert knowledge may not be available for every family.

Meta-MEME's performance may be affected by biases in the training set. In the experiments reported here, the dehydrogenase training set was hand-selected so as to fairly uniformly represent a particular protein family. However, in the ferredoxin experiments, randomly selected training sets containing several closely related sequences may have biased some of the trained ferredoxin models. These biases would explain the relatively large standard error bars in Figure 4(b). Such biases could have been reduced by first removing highly similar sequences using a program such as PURGE (Neuwald and Green, 1994). In addition to reducing training set bias, this approach reduces the amount of computation required during training. Several researchers have shown that weighting schemes, which attempt to compensate for bias in the training set by assigning weights to individual sequences, may significantly improve the performance of database searching algorithms (Henikoff and Henikoff, 1994a; Altschul *et al.*, 1989; Sibbald and Argos, 1990; Thompson *et al.*, 1994). (Eddy *et al.*, 1995) have developed a maximum discrimination training algorithm for hidden Markov models which addresses the same problem. Use of such methods may also provide a means of improving Meta-MEME's performance.

We hope to improve Meta-MEME's models in several ways. First, we will use them as initialization for standard HMM training. This method will allow the motif-based HMMs to be tuned more precisely tot the training set. Second, we plan to improve the modeling of spacer regions. A standard HMM insert state gives an exponential distribution of gap lengths, which is not biologically realistic. In order to model spacer lengths more realistically, we will include at each insert state an explicit probability distribution for its output length. In addition, we will investigate improved methods for choosing the number of motifs to include in each model.

Eventually, we hope that motif-based HMMs can address another problem faced by linear HMMs: their inability to adequately model sequence families containing large-scale copying of domains. The linearity of motif-based HMMs may be removed if the motif models are completely connected to one another. Because the total number of motifs is small, such a model may still be trained effectively. This generalized HMM will allow a sequence to possess occurrences of the motifs in any order. For each pair of motifs, the HMM will learn the probability of the second motif following the first motif directly. If, as is typical, one ordering of the motifs is most common, the trained HMM will assign a higher probability to a sequence that has the motifs in this order.

Acknowledgments

The authors would like to thank Michael Gribskov for providing a carefully curated 4Fe-4S ferredoxin data set, and the reviewers for useful comments and suggestions. William Grundy is funded by the National Defense Science and Engineering Grant Fellowship Program. Charles Elkan is funded by a Hellman Faculty Fellowship from UCSD. Paragon time was made available through a grant to NBCR (NIH P41 RR08605). Timothy L. Bailey was supported by the National Biomedical Computation Resource, an NIH/NCRR funded research resource (P41 RR-08605), and the NSF through cooperative agreement ASC-02827.

References

Altschul,S.F., Carroll,R.J. and Lipman,D.J. (1989) Weights for data related by a tree. *J. Mol. Biol.*, **207,** 647.

Bailey,T.L. and Elkan,C.P. (1995) Unsupervised learning of multiple motifs in biopolymers using EM. *Mach. Learn.*, **21,** 51.

Bailey,T.L. and Elkan,C.P. (1995) The value of prior knowledge in discovering motifs with MEME. In Rawlings,C. *et al.* (eds), *Proc. 3rd Int. Conf. Intel. Syst. Mol. Biol.*, pp. 21–29. AAAI Press.

Bailey,T.L. and Gribskov,M. (1996) The megaprior heuristic for discovering protein sequence patterns. In States,D.J., Agarwal,P., Gaasterland,T., Hunter,L. and Smith,R. (eds), *Proc. 4th Int. Conf. Intel. Syst. Mol. Biol.*, pp. 15–24. AAAI Press.

Bailey,T.L. and Gribskov,M. (1997) MAST—motif alignment and search tool. In preparation.

Bairoch,A. (1992) PROSITE: a dictionary of sites and patterns in proteins. *Nucl. Acids Res.*, **20,** 2013.

Bairoch,A. (1994) The SWISS-PROT protein sequence data bank: current status. *Nucl. Acids Res.*, **22,** 3578.

Baker,J.K. (1975) The dragon system—an overview. *IEEE Trans. Acoust. Speech Signal Process.*, **ASSP-23,** 24.

Baker,M.E. (1994) Sequence analysis of steroid and prostaglandin metabolizing enzymes: application to understanding catalysis. *Steroids*, **59,** 248.

Baker,M.E. (1996) Unusual evolution of mammalian 11β- and 17β-hydroxysteroid and retinol dehydrogenases. *Bioessays*, **18,** 63.

Baldi,P., Chauvin,Y., Hunkapiller,T. and McClure,M.A. (1994) Hidden Markov models of biological primary sequence information. *Proc. Natl. Acad. Sci. USA*, **91,** 1059.

Blocks WWW server. (1996) http://www.blocks.fhcrc.org/

Branden,C. and Tooze,J. (1991) *Introduction to Protein Structure*. Garland.

Brown,M., Hughey,R., Krogh,A., Mian,I., Sjolander,K. and Haussler,D. (1995) Using Dirichlet mixture priors to derive hidden Markov models for protein families. In Rawlings,C. *et al.* (eds), *Proc. 3rd Int. Conf. Intel. Syst. Mol. Biol.*, pp. 47–55. AAAI Press.

Chothia,C. and Lesk,A.M. (1986) The relation between the divergence of sequence and structure in proteins. *EMBO J.*, **5**, 823.

Churchill,G.A. (1989) Stochastic models for heterogeneous DNA sequences. *Bull. Math. Biol.*, **51**, 79.

Eddy,S.R., Mitchison,G. and Durbin,R. (1995) Maximum discrimination hidden Markov models of sequence consensus. *J. Computat. Biol.*, **2**, 9.

Eddy,S.R. (1995) Multiple alignment using hidden Markov models. In Rawlings,C. *et al.* (eds), *Proc. 3rd Int. Conf. Intel. Syst. Mol. Biol.*, pp. 114–120. AAAI Press.

GenBank overview (1996) http://www.ncbi.nlm.nih.gov/Web/Genbank/index.html

Gribskov,M. and Robinson,N.L. (1996) Use of receiver operating characteristic (ROC) analysis to evaluate sequence matching. *Comp. Chem.*, **20**, 25.

Gribskov,M., Lüthy,R. and Eisenberg, D. (1990) Profile analysis. *Meth. Enzymol.*, **183**, 146.

Grundy,W.N., Bailey,T.L. and Elkan,C.P. (1996) ParaMEME: A parallel implementation and a web interface for a DNA and protein motif discovery tool. *CABIOS*, **12**, 303.

Henikoff,S. and Henikoff,J.G. (1994) Position-based sequence weights. *J. Mol. Biol.*, **243**, 574.

Henikoff,S. and Henikoff,J.G. (1994) Protein family classification based on searching a database of blocks. *Genomics*, **19**, 97.

Henikoff,J.G. and Henikoff,S. (1996) Blocks database and its applications. *Meth. Enzymol.*, **266**, ???.

Henikoff,S., Henikoff,J.G., Alford,W.J. and Pietrokovski,S. (1995) Automated construction and graphical presentation of protein blocks from unaligned sequences. *Gene-COMBIS, Gene*, **163**, 17.

Eddy,S.R. group (1996) Dept. of Genetics, Washington University. http://genome.wustl.edu/eddy/hmm.html

Hughey,R. and Krogh,A. (1996) Hidden Markov models for sequence analysis: Extension and analysis of the basic method. *CABIOS*, **12**, 95.

Krogh,A., Brown,M., Mian,I., Sjolander,K. and Haussler,D. (1994) Hidden Markov models in computational biology: Applications to protein modeling. *J. Mol. Biol.*, **235**, 1501.

Lawrence,C.E., Altschul,S.F., Boguski,M.S., Liu,J.S., Neuwald,A.F. and Wootton,J.C. (1993) Detecting subtle sequence signals: A Gibbs sampling strategy for multiple alignment. *Science*, **262**, 208.

MEME ANSI C source code (1996) ftp://cs.ucsd.edu/pub/tbailey/meme

Neuwald,A.F. and Green,P. (1994) Detecting patterns in protein sequences. *J. Mol. Biol.*, **239**, 698.

MEME—multiple EM for motif elicitation (1996) http://www.sdsc.edu/MEME

Persson,B., Krook,M. and Jornvall,H. (1991) Characteristics of short chain alcohol dehydrogenases and related enzymes. *Euro. J. Biochem.*, **200**, 7.

Rabiner,L.R. and Juang,B. (1993) *Fundamentals of Speech Recognition*. Prentice Hall.

Rabiner,L.R. (1995) A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE*, **77**, 257.

SAM: Sequence alignment and modeling system (1996) http://www.cse.ucsc.edu/research/compbio/sam.html

Sibbald,P.R. and Argos,P. (1990) Weighting aligned protein or nucleic acid sequences to correct for unequal representation. *J. Mol. Biol.*, **216**, 813.

Sjolander,K., Karplus,K., Brown,M., Hughey,R., Krogh,A., Mian,I.S. and Haussler,D. (1996) Dirichlet mixtures: A method for improving detection of weak but significant protein sequence homology. *Comput. Applic. Biosci.*, **12**, 327.

Smith,H.O., Annau,T.M. and Chandrasegaran,S. (1990) Finding sequence motifs in groups of functionally related proteins. *Proc. Natl. Acad. Sci. USA*, **87**, 826.

Thompson,J.D., Higgins,D.G. and Gibson,T.J. (1994) Improved sensitivity of profile searches through the use of sequence weights and gap excision. *CABIOS*, **10**, 19.

Wierenga,R.K., De Maeyer,M.C. and Hol,W.G.J. (1985) Interaction of pyrophosphate moieties with α-helices in dinucleotide binding proteins. *Biochemistry*, **24**, 1346.

Wierenga,R.K., Terpstra,P.P. and Hol,W.G.J. (1986) Prediction of the occurrence of the ADP-binding β-α-β-fold in proteins using an amino acid sequence fingerprint. *J. Mol. Biol.*, **187**, 101.

Woodland,P.C., Odell,J.J., Valtchev,V. and Young,S.J. (1994) Large vocabulary continuous speech recognition using HTK. In *IEEE Int. Conf. Acoustics, Speech and Signal Process*, vol. 2, pp. 125–128. IEEE.

Received on November 5, 1996; accepted on January 14, 1997

Appendix A. Short-chain alcohol dehydrogenases

2BHD_STREX	20-Beta-Hydroxysteroid Dehydrogenase
3BHD_COMTE	3-Beta-Hydroxysteroid Dehydrogenase
ACT3_STRCO	Putative Ketoacyl Reductase
ADH_DROME	Alcohol Dehydrogenase
AP27_MOUSE	Adipocyte P27 Protein (AP27)
BA72_EUBSP	7-Alpha-Hydroxysteroid Dehydrogenase
BDH_HUMAN	D-Beta-Hydroxybutyrate Dehydrogenase Precursor
BEND_ACICA	Cis-1,2-Dihydroxy-3,4-Cyclohexadiene-1-Carboxylate Dehydrogenase
BPHB_PSEPS	Biphenyl-2,3-Dihydro-2,3-Diol Dehydrogenase
BUDC_KLETE	Acetoin(Diacetyl) Reductase
CSGA_MYXXA	C-Factor
DHB2_HUMAN	Estradiol 17 Beta-Dehydrogenase 2
DHB3_HUMAN	Estradiol 17 Beta-Dehydrogenase 3
DHCA_HUMAN	Carbonyl Reductase (NADPH)
DHES_HUMAN	Estradiol 17 Beta-Dehydrogenase
DHGB_BACME	Glucose 1-Dehydrogenase B
DHII_HUMAN	Corticosteroid 11-Beta-Dehydrogenase
DHMA_FLAS1	N-Acylmannosamine 1-Dehydrogenase
ENTA_ECOLI	2,3-Dihydro-2,3-Dihydroxybenzoate Dehydrogenase
FABG_ECOLI	3-Oxoacyl-[Acyl-Carrier Protein] Reductase
FABI_ECOLI	Enoyl-[Acyl-Carrier-Protein] Reductase (NADH)
FIXR_BRAJA	Fixr Protein
FVT1_HUMAN	Follicular Variant Translocation Protein 1 Precursor (FVT-1)
GUTD_ECOLI	Sorbitol-6-Phosphate 2-Dehydrogenase
HDE_CANTR	Hydratase-Dehydrogenase-Epimerase (HDE)
HDHA_ECOLI	7-Alpha-Hydroxysteroid Dehydrogenase
HMTR_LEIMA	H Region Methotrexate Resistance Protein
LIGD_PSEPA	C Alpha-Dehydrogenase
MAS1_AGRRA	Agropine Synthesis Reductase
NODG_RHIME	Nodulation Protein G (Host-Specificity Of Nodulation Protein C)
PCR_PEA	Protochorophyllide Reductase Precursor
PGDH_HUMAN	15-Hydroxyprostaglandin Dehydrogenase (NAD(+))
PHBB_ZOORA	Acetoacetyl-Coa Reductase
RFBB_NEIGO	Dtdp-Glucose 4,6-Dehydratase
RIDH_KLEAE	Ribitol 2-Dehydrogenase
YINL_LISMO	Hypothetical 26.8 Kd Protein In Inla 5′region (ORFA)
YRTP_BACSU	Hypothetical 25.3 Kd Protein In Rtp 5′region (ORF238)
YURA_MYXXA	Hypothetical Protein In Uraa 5′region (Fragment)

SWISSPROT identifiers and descriptions for the 38 steroid dehydrogenase training set.

Appendix B. 4Fe-4S ferredoxins

FER1_AZOVI	FER2_RHOCA	FER2_RHORU	FER_MYCSM	FER_SACER
FER_STRGR	FER_PSEPU	FER_PSEST	FER_THETH	FER_CLOAC
FER_CLOBU	FER_CLOPA	FER_CLOPE	FER_CLOSP	FER_CLOST
FER_CLOTM	FER_CLOTS	FER_MEGEL	FER_PEPAS	FER1_RHORU
FER_BUTME	FER_CHLLT	FER1_CHLLI	FER2_CHLLI	FER_CHRVI
FER_METBA	FER_METTL	FER_THEAC	FER2_DESDN	FER3_DESAF
FER1_DESVM	FER_ENTHI	FERX_ANASP	FERN_AZOCH	FERV_AZOVI
FDXN_RHILT	FERN_RHIME	FERN_BRAJA	FER1_RHOCA	FER_ALIAC
FER_SULAC	FER1_RHOPA	FERN_AZOVI	FER3_ANAVA	FER3_PLEBO
FER3_RHOCA	FER_CLOTH	FER_DESGI	FER1_DESDN	FER2_DESVM
FER_THELI	FER_THEMA	FIXX_RHILP	FIXX_RHILE	FIXX_RHIME
FIXX_RHILT	PSAC_ANTSP	PSAC_CHLRE	PSAC_CUCSA	PSAC_EUGGR
PSAC_MAIZE	PSAC_MARPO	PSAC_PEA	PSAC_PINTH	PSAC_SPIOL
PSAC_TOBAC	PSAC_WHEAT	PSAC_CYAPA	PSAC_ANASP	PSAC_ANAVA
PSAC_FREDI	PSAC_SYNEN	PSAC_SYNP2	PSAC_SYNP6	PSAC_SYNY3
PSAX_SYNY3	DHSB_BACSU	DHSB_ECOLI	FRDB_ECOLI	FRDB_HAEIN
FRDB_PROVU	YFRA_PROVU	FRDB_WOLSU	FDHB_METFO	FRHG_METTH
FIXG_RHIME	RDXA_RHOSH	PHFL_DESVH	PHFL_DESVO	COOF_RHORU
DMSB_ECOLI	DMSB_HAEIN	YFFE_ECOLI	FDNH_ECOLI	FDOH_ECOLI
FDXH_HAEIN	FDHB_WOLSU	HMC2_DESVH	HMC6_DESVH	ASRA_SALTY
GLPC_ECOLI	GLPC_HAEIN	HYCB_ECOLI	HYCF_ECOLI	HYDN_ECOLI
PHSB_SALTY	PSRB_WOLSU	NRFC_ECOLI	NRFC_HAEIN	NAPF_ECOLI
NAPF_HAEIN	NAPG_ECOLI	NAPG_HAEIN	NAPH_ECOLI	NAPH_HAEIN
YGL5_BACST	YJES_ECOLI	YA43_HAEIN	DHSB_USTMA	DHSB_YEAST
DHSB_SCHPO	DHSB_HUMAN	DHSB_RAT	DHSB_DROME	DHSB_ARATH
MBHT_ECOLI	PHF1_CLOPA	ASRC_SALTY	NUIC_MAIZE	NUIC_MARPO
NUIC_ORYSA	NUIC_TOBAC	NUIC_WHEAT	NUIC_PLEBO	NUIC_SYNY3
NUIM_BOVIN	NUIM_RHOCA	NQO9_PARDE	NUOI_ECOLI	DCMA_METSO
YJJW_ECOLI	FER1_DESAF	FIXX_AZOCA	FIXX_BRAJA	ISP1-TRYBB
NARH_ECOLI	NARY_ECOLI	NIFJ_ANASP	NIFJ_KLEPN	YAAT_ECOLI
FER_METTE	PSAC_ODOSI	YEIA_ECOLI	FER_BACTH	FER_BACST
DHSB_CHOCR	DHSB_CYACA	NARH_BACSU	YWJF_BACSU	

SWISSPROT numbers for the 159 4Fe-4S ferredoxins.

Ten of the sequences above are not included in the PROSITE 13.1 listing for this family. DHSB_CHOCR, DHSB_CYACA, FER_METTE, and PSAC_ODOSI are included here based on homology to PROSITE annotated families in this group, and ROC analysis. ISP1_TRYBB, excluded from this group by PROSITE, appears to be closely related to NADH oxidoreductases in this group as shown by ROC and sequence comparisons (NQQ9, NUIM, NUOI, HYCF, NUIC). NARH_BACSU, NARH_ECOLI and NARY_ECOLI, while showing lower ROC, have excellent 4Fe-4S sequences highly similar to those in DMSB, PHSB, FDNH, HYCB, etc. YEIA_ECOLI is a possible type III ferredoxin and has a very strong ROC. YWJF_BACSU is included in the positives because of high ROC, significant similarity to glycerol-3-phosphate dehydrogenase subunits (GLPC) which are ferredoxins, and clear presence of two appropriate 4Fe-4S binding sequences.

research articles Journal of proteome research

A New Algorithm for the Evaluation of Shotgun Peptide Sequencing in Proteomics: Support Vector Machine Classification of Peptide MS/MS Spectra and SEQUEST Scores

D. C. Anderson,*,† Weiqun Li,† Donald G. Payan,† and William Stafford Noble‡

Rigel Incorporated, 240 East Grand Avenue, South San Francisco, California 94080 and Department of Genome Sciences, University of Washington, Seattle, Washington 98195

Received September 3, 2002

Shotgun tandem mass spectrometry-based peptide sequencing using programs such as SEQUEST allows high-throughput identification of peptides, which in turn allows the identification of corresponding proteins. We have applied a machine learning algorithm, called the support vector machine, to discriminate between correctly and incorrectly identified peptides using SEQUEST output. Each peptide was characterized by SEQUEST-calculated features such as delta Cn and Xcorr, measurements such as precursor ion current and mass, and additional calculated parameters such as the fraction of matched MS/MS peaks. The trained SVM classifier performed significantly better than previous cutoff-based methods at separating positive from negative peptides. Positive and negative peptides were more readily distinguished in training set data acquired on a QTOF, compared to an ion trap mass spectrometer. The use of 13 features, including four new parameters, significantly improved the separation between positive and negative peptides. Use of the support vector machine and these additional parameters resulted in a more accurate interpretation of peptide MS/MS spectra and is an important step toward automated interpretation of peptide tandem mass spectrometry data in proteomics.

Keywords: shotgun peptide sequencing • SEQUEST • support vector machine • machine learning • mass spectrometry • capillary LC/MS/MS • proteomics

Introduction

The separation and sequencing by capillary HPLC-tandem mass spectrometry of femtomole (or below) peptide levels is the basis for the high-throughput identification of proteins present in cell or tissue samples. The technique has broad applicability: applications include the identification of peptides binding individual MHC proteins of defined haplotype,[1] the identification of a peptide recognized by melanoma-specific human CTL cell lines,[2] the identification of individual protein complexes,[3−4] the large scale analysis of the yeast proteome,[5] the identification in yeast of interacting proteins for a large number of tagged protein baits,[6−7] the identification of proteins in urine,[8] and the definition of proteins of the nucleolus.[9]

The analysis of peptide collision-induced dissociation spectra to give information on a peptide's sequence was developed by Hunt and co-workers[10−14] and Biemann.[15] To identify proteins from mass spectrometry data, protein database searches initially used peptide fragments[16] or sequence tags,[17] and included sequenced genomes[18] and more sophisticated search techniques.[19−22] Yates and co-workers developed correlations of peptide tandem mass spectrometry data and sequences from protein databases,[23−25] incorporated these in the program SEQUEST, and coupled this software with capillary LC/MS/MS data and database searches to identify proteins[26] and protein complexes.[27] Due to its early implementation, availability and the widespread use of ion trap, triple quadrupole, and quadrupole time-of-flight mass spectrometers that generate compatible data, SEQUEST is one of the most commonly used programs.

The use of database search programs introduces questions about how to interpret their output. SEQUEST outputs for each spectrum one or more peptides from the given database whose theoretical spectra closely match the given spectrum. Associated with each match is a collection of statistics. Initially, the difference between normalized cross-correlation functions (delta Cn) for the first and second ranked results from a search of a relatively small database was used to indicate a correctly selected peptide sequence.[23,25] Additional criteria were subsequently added, including the cross-correlation score between the observed peptide fragment mass spectrum and the theoretically predicted one (Xcorr), followed by a manual examination of the MS/MS spectra.[27] More stringent criteria combined the use of Xcorr cutoffs, delta Cn, and the correspondence of peptide sequences with those expected for cleavage with the enzymes used for proteolysis.[5,28]

Recently, Moore et al. described a probabilistic algorithm called Qscore,[29] for evaluating SEQUEST database search results. In contrast to previous heuristic techniques, Qscore is

* To whom correspondence should be addressed. E-mail: dca0210@earthlink.net.

† Rigel Incorporated.

‡ Department of Genome Sciences, University of Washington.

10.1021/pr0255654 CCC: $25.00

PAGE EST: 9.9

Published on Web 00/00/0000

based upon a probability model which includes the expected number of matches from a given database, the effective database size, a correction for indistinguishable peptides, and a measurement of match quality. The algorithm performs well in distinguishing between true and false matches from SEQUEST outputs.

The approach described here addresses a similar problem using a different approach. Rather than building an algorithm by hand, we use a machine learning algorithm, called the support vector machine (SVM), to learn to distinguish between correctly and incorrectly identified peptides. The support vector machine (SVM)[30−32] is a supervised learning algorithm, useful for recognizing subtle patterns in complex data sets. The algorithm has been applied in domains as diverse as text categorization, image recognition, hand-written digit recognition[32] and in various bioinformatics domains, including protein remote homology detection,[33] protein fold recognition,[34] and microarray gene expression analysis.[35−36] The SVM is fundamentally a binary classifier: given two classes of data, the SVM learns to distinguish between them and to predict the classification of previously unseen examples. In the application described here, the algorithm is trained from a labeled collection of SEQUEST outputs, where the labels indicate whether the peptide represents a correct or incorrect identification. The SVM then learns to distinguish between peptides that were correctly and incorrectly identified by SEQUEST.

The SVM algorithm is surprisingly simple. It treats each training example as a point in a high-dimensional space and searches for a hyperplane that separates the positive from the negative examples. As such, the SVM is closely related to the perceptron algorithm,[37] with three important differences. First, motivated by statistical learning theory,[31] the SVM searches for a hyperplane that separates the two classes with the largest margin; the SVM finds a hyperplane that maximizes the minimum perpendicular distance to any training example. Choosing the maximum margin hyperplane reduces the chance that the SVM will overfit the training data. Second, for data sets that are not separable by a simple hyperplane, the SVM uses a mathematical trick, known as the kernel trick, to operate implicitly in a higher-dimensional space. By increasing the dimensionality of the space in which the points reside, the SVM can learn complex decision boundaries between the two given classes. Finally, for data sets that contain some mislabeled examples, the SVM incorporates a soft margin. The SVM may find a decision boundary that nearly, but not perfectly, separates the two given classes. A few outlier examples are allowed to fall on the wrong side of the decision boundary.

Here, we use tryptic digests of mixtures of known protein standards, purified proteins, or of a variety of affinity extracts by specific antibodies or other binding proteins, to generate LC/MS/MS data using ion trap or quadrupole time-of-flight (QTOF) mass spectrometers. Peptides are classified as positive examples (derived from proteins known or expected to be in the samples) or negative examples (peptides not expected to be in the samples). Each example in the training set is characterized by a vector of features, including observed data (peptide mass, precursor ion intensity) and SEQUEST-calculated statistics (such as the parameters Xcorr, delta Cn, Sp, and RSp). These labeled vectors are then used to train an SVM to distinguish between positive and negative examples.

Our experiments show that the trained SVM, when tested on its ability to classify previously unseen examples, exhibits high sensitivity and specificity. We illustrate the learning procedure using two differently sized databases, as well as using data generated on ion trap and QTOF mass spectrometers. The SVM yields significantly fewer false positive and false negative peptides than any of the cutoffs previously proposed and gives a cleaner separation of positive and negative peptides than Qscore-based single peptide analysis. The trained SVM is an accurate, high-throughput technique for the examination of SEQUEST results, which will enable the processing of large amounts of data generated from examinations of complex mixtures of proteins.

Experimental Section

Peptide Samples. Tryptic digest test mixtures containing 500 pmol of reduced, iodoacetic acid-alkylated hen egg white lysozyme, horse myoglobin, and horse cytochrome *c*, bovine serum albumin, and bovine carbonic anhydrase were purchased from Michrom Bioresources (Auburn CA). These standards were mixed so that individual test samples contained from 5 to 80 fmol of each of the five proteins, with 2-fold differences in each concentration. Affinity extracts of cultured human Jurkat cells were prepared using antibodies specific for individual antigens, and were carried out as described.[38] Individual baits or antigens and the source of the antibodies used for the affinity extractions included heat shock protein 90 (MA3−010 antibody, Affinity BioReagents, Golden CO), RbAp48 (13D10 antibody, Upstate Biotechnology, Lake Placid, NY), the synthetic C-terminal p21$^{cip1/waf1}$ peptide biotin-GSGSGSGSGSKRRQTSMTDFYHSKRRLIFS-acid, the fusion protein glutathione S-transferase-S5a (AFFINITI Research Products Ltd., Exeter, U.K.), and green fluorescent protein (rabbit polyclonal antibody, Molecular Probes, Eugene OR).

Mass Spectrometry and Database Searches. Ion trap mass spectrometry utilized a Finnigan LCQ (ThermoFinnigan, San Jose CA) and an LC Packings (San Francisco CA) Ultimate capillary hplc and custom packed 75 micron internal diameter capillary C18 reversed phase columns for sample injection and chromatography, as described.[38] Quadrupole time-of-flight mass spectrometry was carried out using a Micromass QTOF-1 mass spectrometer coupled to an LC Packings capillary hplc as above. Peptides were eluted using a 1% acetonitrile/min. gradient. Database searches utilized either the nonredundant human protein database of March 15, 2002, or the nonredundant protein database of March 6, 2002. Both were downloaded from the National Center for Biotechnology Information (http://www.ncbi.nlm.nih.gov). Proteins from the human immunodeficiency virus were removed from the nonredundant human protein database before use. The version of SEQUEST (ThermoFinnigan, San Jose, CA) used for database searches was SEQUEST 2.0 that was distributed with Sequest Browser. Tryptic cleavages at only lys or arg and up to two missed internal cleavage sites in a peptide were allowed. The maximal allowed uncertainty in the precursor ion mass was 1.5 m/z. SEQUEST searches allowed optional met oxidation and cys carboxamidomethylation because cysteines were derivatized in this fashion after protein thermal denaturation and reduction. Peptides with masses from 700 to 3500 m/z and precursor charge states of +1, +2, and +3 were allowed. A few peptides analyzed on the QTOF-1 were present as +4 ions and were left in the appropriate positive or negative category. For spectra collected on the LCQ, the minimum total ion current required for precursor ion fragmentation was 1.0×10^5, the minimum number of ions was 25, and IonQuest filtering was turned off. Single precursor ion scans from 350 to 1800 m/z were followed by 6 MS/MS scans from 50 to twice the precursor ion *m*/*z*, up to a limit of 1800 Da. Dynamic exclusion was turned on for a duration of 1 min. A collision energy on the LCQ of 30 was used for all fragment ion spectra. For the QTOF-1, a precursor charge-dependent and peptide mass-dependent collision energy was used, ranging from 16 to 55 ev for +1 ions of 388−2000 Da, 22−65 ev for +2 ions of 400−2000 Da, 16−50 ev for +3 ions between 435 and 2000 Da, and 19−36 ev for +4 ions between 547 and 2000 Da. For database searches using nonhuman protein test samples, sequences for the nonhuman proteins were added to the nonredundant human protein database.

Positive and Negative Peptides. Positive peptides were selected by several criteria. One was tryptic peptides from five known proteins in tryptic digest standards. A second was peptides from proteins expected to be present in affinity extracts because they are derived from the antibody or affinity reagent used in the extraction, from the known antigen for the antibody, from known interacting partners of the antigen, or are autolytic fragments of trypsin. A third category includes peptides from extracted proteins thought to be present due to the identification of at least two peptides from that protein with SEQUEST scores that meet stringent criteria.[5,28] This includes common contaminating proteins such as myosin, heat shock proteins, defined cytokeratins, and may include proteins not previously demonstrated to interact with a particular bait. Tryptic digests from isolated protein standards were injected at different levels between 5 and 1000 fmol to include SEQUEST scores from peptides with strong as well as weak signals.

Negative peptides were selected from tryptic digests of known protein standards, in which these peptides were assigned by SEQUEST to proteins other than the injected protein or its human homolog. A second category of negative peptides included peptides selected from lower scoring peptide matches (i.e., from incorrect proteins) to MS/MS data from peptides from a known standard protein.

Construction of Training Sets. Training sets were constructed using data collected and analyzed under three different conditions: data collected using an ion trap mass spectrometer and analyzed using the nonredundant human and full nonredundant databases, and data collected on a QTOF mass spectrometer analyzed using the nonredundant human database. All three sets included nine experimentally measured and SEQUEST-calculated parameters:[23−25] MS/MS spectrum total ion current, peptide charge, peptide precursor ion mass, the difference in observed and theoretical precursor ion masses for the best-fit peptide, the SEQUEST variables Xcorr (cross-correlation score of the observed to the theoretical MS/MS spectrum for a peptide sequence), delta Cn (the magnitude of the difference in normalized cross-correlation parameter values for the first and second hits found by SEQUEST), Sp (the preliminary score for a peptide after correlation analysis to the predicted fragment ion values), RSp (the final correlation score rank), and the percent of predicted y and b ions matched in the MS/MS spectrum.

A training set representing ion trap data and a SEQUEST nonredundant human database search was constructed containing 696 positive peptides, including 338 unique peptides representing 47 different proteins. Multiple copies of some individual peptides were obtained from independent runs using from 5 fmol to 1 pmol of individual standard proteins, resulting in peptides with a large dynamic range in signal-to-noise. There were a total of 465 negative peptides, of which, 435 were unique; 30 negatives were generated using peptides that were second or lower choices below a top ranked positive peptide. Initial support vector machine calculations incorrectly assigned negative labels to a number of positive peptides. Upon examination, in a number of cases the top ranked peptide from a SEQUEST database search, for a given precursor ion and MS/MS spectrum, was instead from a different protein. SEQUEST had selected a lower ranked peptide from the protein of interest and incorrectly listed it as being top ranked. As a result of this round of SVM calculations, all gi or accession numbers for positive peptides were verified as corresponding to the protein identified, and a number of positive peptides with relatively low scores were individually blasted against the nonredundant database to check the identity of their source protein.

A second training set representing ion trap data and a SEQUEST search using the full nonredundant database was constructed. It contained 497 positive peptides assigned to 280 unique sequences from 33 different proteins. It also contained 479 negative peptides assigned to 460 different peptide sequences; 67 negatives were generated using peptides that were second or lower choices below a top ranked postive peptide. This database had approximately 8 times as many sequences as the nonredundant human protein database, and may be useful for finding protein homologues from other organisms when the human protein sequence is not in a database (for analyses using human cells) or for analysis of nonhuman samples. The most significant outliers from initial SVM analyses were examined to uncover errors in SEQUEST-peptide sequence assignment or errors in data handling.

A third training set representing QTOF data and a SEQUEST search using the nonredundant human database was also constructed. It contained 1017 positive and 532 negative peptides analyzed on a quadrupole time-of-flight mass spectrometer. This training set was created for comparison with the two previous training sets since data for these peptides was collected on a different instrument. The positive peptides were derived from 45 different proteins, and represented 493 unique sequences. The negatives contained MS/MS spectra assigned to 335 different sequences. Seventy additional negative peptides were derived by selecting lower choices than the top ranked peptide, when the top ranked peptide was correctly assigned to a known protein from a standard peptide map. As before, initial support vector machine analysis was used to uncover mistakes in data entry or incorrect assignments of sequences to proteins by SEQUEST, by analysis of individual false positives and false negatives.

Four New Parameters used to Evaluate SEQUEST Output. The basic parameters used to evaluate SEQUEST output included experimentally measured or calculated parameters such as precursor ion mass, precursor ion current, or peptide charge. They also included those calculated using SEQUEST: mass difference between observed and calculated precursor ions for the best fit sequence, Xcorr, delta Cn, Sp, RSp, and % y and b ions matched. Four additional parameters were measured or calculated. These included a count of the number of peaks in the MS/MS spectrum, and the fraction of these peaks matched by predicted peptide fragments. An MS/MS peak for ion trap data was defined as having over 10^3 counts, and for QTOF-1 data as having over 1 count. In a noisy MS/MS spectrum, the fraction of matched peaks should be low, for both positive and negative peptides. In other MS/MS spectra it should be lower for negative than for positive peptides. A third parameter is the fraction of the MS/MS spectrum total

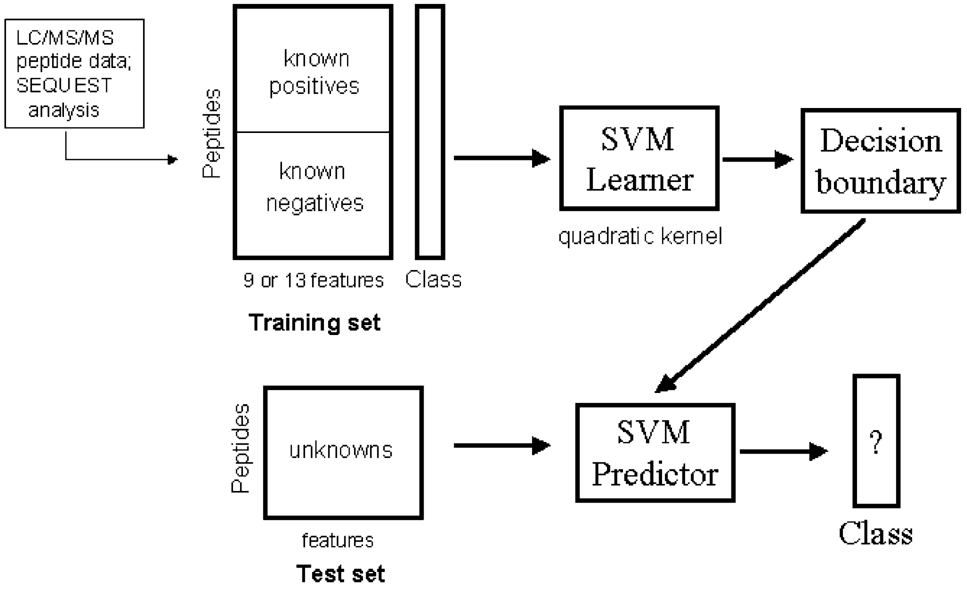


Figure 1. The support vector machine learns to recognize high-quality peptide matches. The figure illustrates how an SVM learns to discriminate between true and false peptide matches (listed as positives and negatives). Peptide data is obtained from LC/MS/MS experiments analyzed by SEQUEST. A training set consists of a collection of individual peptide matches, each characterized by a vector of statistics (as described in the text) and a binary classification (true or false match) provided by manual inspection of the training set. The SVM learning algorithm finds a decision boundary that separates the true matches from the false matches. This decision boundary can then be used by the SVM prediction algorithm to determine the classification of previously unseen peptides. The prediction produced by the SVM is a binary classification, along with a discriminant value that can be used to estimate the SVM's confidence in its prediction. Analysis of training sets using single or pairwise feature analysis can indicate which individual or pairwise features contribute the most to separation of positive and negative peptides in 9- or 13-feature space. Comparison of training sets obtained using different mass spectrometers or databases estimates the contribution of these variables to the separation of positive and negative peptides, and thus to accurate peptide and protein identification.

ion current that is in matched peptide fragments. For a good match, this fraction should be high, and for a poor match it should be low. A fourth parameter is the sequence similarity between the top peptide choice and second ranked choice. When delta Cn is low, this parameter is intended to mark these peptides for further examination. When the value of this parameter is close to 1 (high sequence similarity) and other scores are good, the individual peptides (and consequently proteins) identified are examined to see if they are similar. If so, then the identification may be useful. If the sequences are different, a unique peptide/protein is not defined by the combined scores.

Support Vector Machine Calculations. The SEQUEST output data is summarized in a (number of peptides) by (9 or 13 parameter) matrix, in which each row contains a vector consisting of the SEQUEST output parameters associated with a particular protein. These data are then normalized in two ways. First, to give equal importance to each of the features, the columns of the matrix are normalized by dividing each entry by the column sum. This operation ensures that the total for each column is 1.0. Second, each 9- or 13-element vector is converted to unit length by dividing each vector component by the Euclidean length of the vector. This operation projects the data onto a unit sphere in the 9- or 13-dimensional space defined by the data. Note that this latter normalization can be performed in the feature space, by defining a normalized kernel K′ in terms of the original kernel K

$$K'(X,Y) = \frac{K(X,Y)}{\sqrt{(K(X,X)\ K(Y,Y))}}$$

The kernelized normalization has the advantage of implicitly operating in the higher-dimensional kernel space.

Support vector machines are trained using a simple optimization algorithm[33]. A software implementation in ANSI C is freely available at http://microarray.cpmc.columbia.edu/gist. The output of the SVM optimization is a set of weights, one per peptide in the training set. The magnitude of each weight reflects the importance of that peptide in defining the separating hyperplane found by the optimization: peptides with zero weights are correctly classified and far from the hyperplane; peptides with small weights are correctly classified and close to the hyperplane; peptides with large weights are incorrectly classified by the hyperplane, as described next. The SVM weights, together with the original training set, can be used to predict the classification of a previously unseen peptide vector.

In most classification tasks, the positive and negative class labels assigned to the training set are not 100% correct. Therefore, the SVM employs a soft margin, which allows some of the training examples to fall on the "wrong" side of the separating hyperplane. An SVM soft margin may be implemented in several ways. We employ a 2-norm soft margin, which charges each misclassified example with a penalty term that increases quadratically according to the example's perpendicular distance from the hyperplane. To account for differences in the number of positive and negative examples, errors in the positive class (for which we have fewer examples) are charged more heavily than examples in the negative class. The asymmetric 2-norm soft margin is implemented by adding a constant to the diagonal entries in the kernel matrix.[32] The diagonal factor added to $K(X,X)$ is 0.2* (n_X/n), where n_X is the number of training examples in the same class as example X, and n is the total number of training examples.[35]

To test the generalization performance of the algorithm, the SVM is trained and tested using leave-one-out cross-validation.

Table 1. Analysis of Training Sets Using Different Methods[a]

method		SVM-9 analysis		SVM-13 analysis	
training set	positive, negative peptides	false positives, negatives	ROC scores	false positives, negatives	ROC scores
ion trap, NR human	696, 465	48, 117	0.929	44, 100	0.950
ion trap, full NR	497, 479	62, 81	0.920	53, 70	0.939
QTOF, NR human	1017, 532	27, 81	0.981	18, 51	0.988

[a] The training sets used either the nonredundant human database (NR human) or the full nonredundant database.

In this paradigm, a single example is removed from the matrix, and the SVM is trained on the remaining examples. The resulting classifier is applied to the held-out example, and the predicted classification is compared to the true classification. The held-out example is counted as a true positive, false positive, true negative, or false negative, depending upon the agreement between the true and predicted class. This leave-one-out procedure is repeated for every example in the data set.

Evaluation of Results. A straightforward method for evaluating the quality of the predictions made by the SVM is to compare the classifications assigned by the SVM to the classifications assigned a priori. Disagreements between the two are counted either as false positives or false negatives. Prediction quality can be measured more precisely using the receiver operating characteristic (ROC) curve. Rather than depending upon a particular classification threshold, the ROC curve integrates information about the complete ranking of examples created by the SVM. The ROC curve plots, for varying classification thresholds, the number of true positives as a function of the number of false positives. The area under this curve, normalized to range from 0 to 1, is called the ROC score. A perfect classifier will rank all of the positive examples above negative examples, yielding an ROC score of 1. A random classifier will produce an approximately diagonal curve, yielding a score close to 0.5.

Results and Discussion

SVM Provides Good Discrimination Performance on Three Different Data Sets. Support vector machine calculations were run on all three datasets, and the results compared (Table 1). For the dataset derived from ion trap mass spectrometry and a SEQUEST search of the nonredundant human protein database, there were 48 false positives, 117 false negatives, 579 true positives, 417 true negative peptides, and a ROC score of 0.929. Of the initial training set peptides, 14.2% were false positives or negatives. For the dataset derived from ion trap mass spectrometry and a SEQUEST search of the full nonredundant human protein database, there were 62 false positives, 81 false negatives, and a ROC score of 0.920. Of these peptides, 14.7% were false positives or negatives. For QTOF mass spectrometry data, searched using the nonredundant human database, calculations found 27 false positive and 81 false negative peptides. The ROC score for this analysis was 0.981; 7.0% of these peptides were false positives or negatives.

ROC plots for the 3 datasets examined with 9 parameters are shown in Figure 2A. Use of the full nonredundant protein database, containing approximately 8-fold more sequences, still allows a good separation between positive and negative peptides, but the ROC scores are slightly lower than for the smaller nonredundant human database. Using the same nonredundant human database for comparison, data collected on a quadrupole time-of-flight mass spectrometer is more readily separated by the SVM into positives and negatives than data collected on this ion trap.

To understand the errors made by the SVM, we looked in detail at each of the false positives and negatives. Many of the errors made by the SVM correspond to examples with noisy MS/MS spectra or poor fragementation of precursor ions. For the ion trap nonredundant human protein database training set, 7 of the 25 top false positive peptides had noisy MS/MS spectra and another 5 had poor fragmentation, with much of the ion current in a few major peaks. Nine of the top 25 false negatives had noisy MS/MS spectra, whereas 13 had poor fragmentation. For the ion trap-full nonredundant protein database training set, 6 of the top 22 false positives had noisy MS/MS spectra and an additional spectrum had poor fragmentation of the precursor ion. Seven of the top 24 false negatives had noisy MS/MS spectra, and 7 had poor precursor ion fragmentation. For the QTOF data, 4 of the top 20 false positive peptides had low signal-to-noise MS/MS spectra and an additional 4 fragmented poorly. Eight of the top 23 false negatives fragmented poorly, and 4 had noisy MS/MS spectra. A lower information content could make it difficult to match the correct peptide sequence for peptides with poor MS/MS fragmentation or noisy MS/MS spectra.

For each of the three training sets, some of the false positives or false negatives that did not have noisy MS/MS spectra, or poorly fragmenting precursor ions, matched the predicted MS/MS spectrum from the best-fit peptide fairly well. It is possible that some of the false positives were contaminants of the known proteins used as standards, and thus were true positives. Some of the false negatives had poor SEQUEST scores, and the SVM had trouble recognizing them as positive peptides. Overall the these peptides seem to represent a core of peptides that are currently difficult to correctly assign with the parameters used.

Using Four Additional Parameters Improves the SVM's Performance. Based upon the initial analyses described above, we computed four additional parameters that we hypothesized would help the SVM recognize noisy or otherwise difficult examples. These parameters were tested in the analysis of the three training sets. The use of the number of peaks in an MS/MS spectrum, and the fraction of those peaks matched by fragments predicted from the best-fit database peptide sequence, was intended as an additional measure of the goodness-of-fit of a peptide sequence to the data. The fraction of the total ion current in the MS/MS spectrum matched by predicted peptide fragments was intended as an additional measure for the goodness-of-fit of a peptide sequence to the data, and to weight the fit by the intensity of the matched fragments. The sequence similarity between the top sequence and second choice sequence was intended to allow discrimination, for peptides with low delta Cn values, between dissimilar

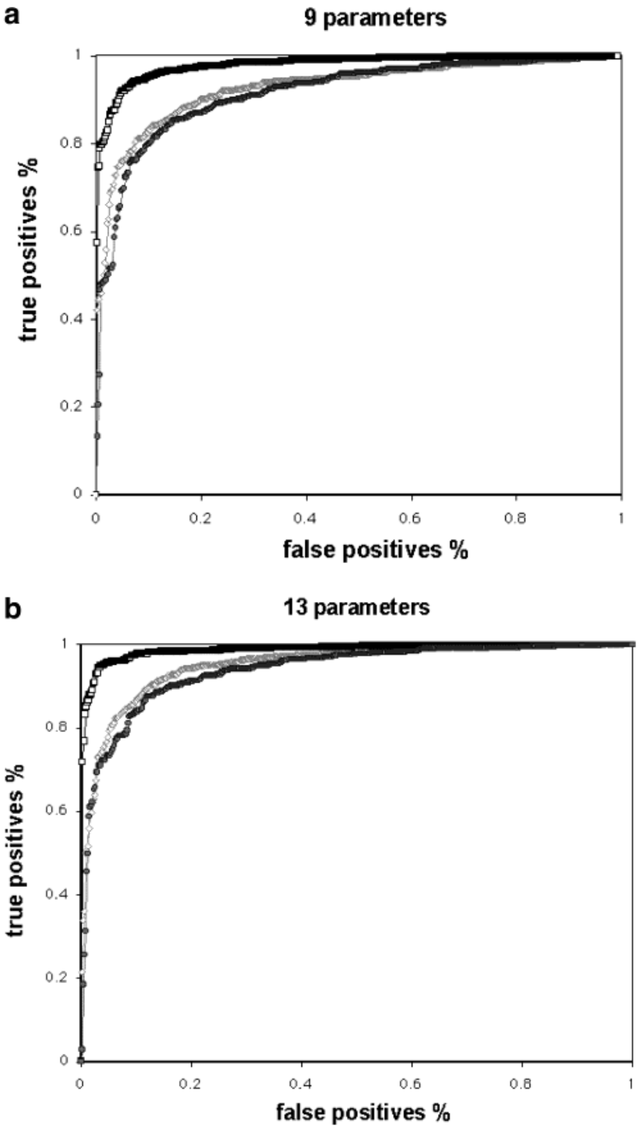


Figure 2. ROC plots of three different training sets used in SVM calculations. A. ROC plot of training sets containing nine parameters. The normalized true positives are plotted against the normalized false positives for each training set. The QTOF-nonredundant human database set is represented in open black squares, the ion trap-nonredundant human database set in light gray, and the ion trap-nonredundant human database set in darker gray. The QTOF training set has the fewest false positives relative to true positives of any set; the ion trap-full nonredundant database set, which has about 8 times as many entries as the ion trap-nonredundant human set, has the most false positives relative to true positives of any set. Thus, the SVM has the most success separating true from false positives with the QTOF dataset, and less success with ion trap data using either the full nonredundant or nonredundant human databases. **B.** ROC plot of training sets containing 13 parameters. The QTOF-nonredundant human database set (open black squares) has the fewest false positives relative to true positives, the ion trap-nonredundant human database (light gray) is intermediate in this respect, and the ion trap-full nonredundant database (darker gray) has the most false positives relative to true positives of any set. Again the SVM has the most success separating true from false positives with the QTOF dataset.

sequences almost equally well-matched to the data, and very similar sequences matched to the data. In the former case the top ranked sequence is not useful, whereas in the latter case, it may be useful if the matched peptides are from similar proteins.

Training sets were constructed as above for positive and negative peptides associated now with 13 parameters, the original 9 and the four additional parameters described above (Table 1). For the ion trap-nonredundant human protein database training set with SVM calculations, there were 44 false positives, 100 false negatives, and the ROC score was improved to 0.950. This represents a loss of 4 false positives and 17 false negatives compared to the 9-parameter dataset. 12.4% of the peptides were false positives or negatives. The addition of these parameters thus improved the overall performance of the SVM calcuations.

For SVM calculations on the ion trap full nonredundant protein database training set, use of the additional 4 parameters resulted in a reduction of false positives to 53 and the false negatives to 70. The ROC score was now 0.939; 12.6% of the peptides were false positives or negatives. Thus, for this training set, the use of the additional parameters also increased the separation between the positive and negative peptides.

For SVM calculations on the QTOF−full nonredundant protein database training set, the total false positive peptides decreased from 26 to 18, and the false negative peptides decreased from 81 to 51. The ROC score for this analysis was 0.988; now, only 4.5% of the training set peptides were found to be false positives or negatives. Thus the best separation of positives and negatives for any training set was obtained using QTOF-collected data and 13 parameter analysis. The QTOF data was collected without internal calibration of each run, and SEQUEST searches utilized a 1.5 Da window. Thus, the higher mass accuracy available with internal calibration or more advanced instruments may further improve the separation of these positive and negative peptides. The average mass deviation between observed and best-fit peptides for the positive peptides for QTOF data was 0.40 ± 0.25 Da, compared to an average mass deviation for ion trap positive peptides of 0.52 ± 0.38 Da. Thus, the uncalibrated QTOF data as used here appears to have a slightly higher mass accuracy than ion trap data.

For all three datasets, ROC scores increase with the use of the 4 additional parameters beyond the original 9 parameters. ROC curves for the 3 datasets examined with 13 parameters are shown in Figure 2B. For the full nonredundant protein database, containing ca. 8-fold more sequences, there is still a good separation between positive and negative peptides, but the ROC scores are slightly lower than for the smaller nonredundant human database. For the same database, data collected on the QTOF mass spectrometer is more readily separated by the SVM into positives and negatives than data collected on an ion trap. The QTOF ROC scores are noticeably higher for both 9- and 13-parameter training sets.

For the parameter representing the fraction of MS/MS peaks matched by predicted peptide fragments, this value was slightly higher in the ion trap training sets for positive peptides (0.499 ± 0.120 and 0.512 ± 0.113 for the NR human and full NR database sets) than for negative peptides (0.410 ± 0.098 and 0.438 ± 0.090 respectively). The difference was more pronounced for QTOF-1 training set data: 0.632 ± 0.120 for positive peptides, 0.352 ± 0.139 for negative peptides. For the parameter representing the average fraction of MS/MS total ion current matched by predicted peptide fragments, its value was slightly higher for ion trap positive peptides (0.646 ± 0.163

and 0.656 ± 0.156 for the NR human and full NR datasets) than for negative peptides (0.468 ± 0.153 and 0.520 ± 0.141 respectively). The difference was more pronounced for QTOF-1 data (0.750 ± 0.112 and 0.392 ± 0.182 for positive and negative peptides). This suggests that the QTOF-1 data may be less noisy than ion trap data, which is consistent with an examination of the MS/MS spectra.

Fisher Scores can be used to Understand What Features are Providing the Most Information. Although the support vector machine generally produces very accurate predictions, this accuracy comes at the price of reduced explanatory power. Unlike a decision tree classifier, the SVM does not explicitly select a few features that are most relevant to the classification task at hand. However, we can use a related technique to analyze the correlations between each feature and the classification labels associated with each peptide. The Fisher criterion score (FCS)[39] is a simple metric that is closely related to the Student's t-test. The score was developed in the context of linear discriminant analysis, which is closely related to the SVM methodology. The FCS has been used previously for feature selection in conjunction with the SVM classification of microarray data[36]. For a given pair of distributions A and B, with means A_m and B_m and standard deviations σ_A and σ_B, the FCS is defined as

$$\frac{(A_m - B_m)^2}{\sigma_A + \sigma_B}$$

Here, A and B correspond to the distributions of a given feature (say, Xcorr) within the positive and negative training sets, respectively. A high FCS indicates that the distribution of Xcorr scores associated with positively labeled peptides is markedly different from the Xcorr scores associated with negatively labeled peptides. We can compute the FCS for each feature in our data set, and rank the features to determine which ones are providing the most information to the SVM.

Unfortunately, SVM results are particularly difficult to explain because the SVM can operate in a higher-order feature space defined by the kernel function. In general, it is not possible to compute Fisher criterion scores of the features in this high-dimensional space. Indeed, for some functions, such as the radial basis function, the feature space is of infinite dimension. However, for a relatively simple kernel function, such as the quadratic polynomial kernel used here, we can explicitly calculate the higher-order features and then compute FCS's for each one.

On the basis of FCS analysis, the most predictive single feature (Table 2) for all three 9- and 13-parameter training sets was delta Cn,[23] the difference between the normalized cross-correlation parameters of the first and second ranked peptides. Xcorr, the raw correlation score of the top peptide sequence with the observed MS/MS spectrum, was the second most predictive single feature for all but two training sets. Threshold values of both of these parameters have been used previously to separate positive from negative peptides.[5,25,27,28] RSp, the ranking of the preliminary raw score, Sp, the preliminary score of the top peptide, and % ion match, the percent of predicted y and b ions for a given sequence that were matched in the experimental MS/MS spectrum, were also predictive. Two of the new parameters, fraction of matched MS/MS TIC and fraction of matched MS/MS peaks, were among the most highly predictive features, particularly for QTOF data. The least predictive features were delta mass, the difference between the observed and predicted masses for individual peptides, and the precursor ion current for individual peptides. The difference between observed and predicted precursor ion masses may not be predictive since this difference is already restricted when selecting peptides for SEQUEST analysis.

Table 2. Contribution of Single Features to the Separation of Positive and Negative Peptides as Reflected by Their Fisher Criterion Scores

mass spectrometer	ion trap	ion trap	QTOF-1
database:	NR human	NR full	NR human
features	9 or 13 parameters		
delta Cn	1.401	1.018	2.861
Xcorr	0.935	0.477	2.444
Sp	0.714	0.604	1.158
MH	0.000	0.000	0.704
charge	0.118	0.102	0.488
RSp	0.273	0.447	0.313
% ion match	0.607	0.447	0.079
dM	0.000	0.014	0.024
TIC	0.016	0.011	0.008
features	13 parameters		
fraction matched MSMS TIC	0.632	0.422	2.804
fraction matched MSMS peaks	0.335	0.260	2.314
peak count	0.062	0.018	0.209
seq similarity	0.060	0.130	0.115

Some Pairs of Features are More Informative than Either Feature Alone. Combinations of individual features were also analyzed for their utility separating positive from negative peptides. Table 3 shows the results of a Fisher criterion score analysis of the different data sets using pairwise features. Only discriminant scores of 1.0 or above for at least one training set were included for illustration purposes. When compared to the analysis using single features, the analysis of pairs of features shows that correlations (or perhaps anti-correlations) among some pairs of features can be much more informative. The combination of fraction matched MS/MS TIC and delta Cn receives an FCS of 4.74, much higher than the scores assigned to either feature alone. The relatively high ranking of pairwise scores explains why the quadratic kernel function yields good SVM classification performance.

The most highly predictive combinations included the fraction of matched MS/MS ion current and the fraction of matched MS/MS peaks (7 combinations each). Other highly predictive combinations included delta Cn or Xcorr with other features. For each of these combinations the predictive value was higher with data acquired on the QTOF-1. This illustrates the ability of the SVM to learn the predictive value of combinations of features that might not be obvious a priori. The mass difference between the observed precursor ion mass and calculated mass of the best-fit peptide, which was poorly predictive when analyzed alone (Table 2), was also poorly predictive in combination with other parameters (data not shown). Thus, not all parameters were highly predictive alone or in combination with other parameters. As a result of the utility of numerous pairwise feature combinations all combinations of features were included in the analysis. Individual variables that are highly predictive when analyzed in a pairwise fashion may be relatively independent variables.

The enhanced performance of the SVM with QTOF data compared to ion trap data appears to be due to better predictiveness of a number of parameters, including precursor ion charge measurement. This value was significantly more predictive for the separation of positives from negatives in

Table 3. Pairwise Contributions of Individual Feature Fisher Scores to the Separation of Positive and Negative Peptides

mass spectrometer:		ion trap	ion trap	QTOF-1
database:		NR human	NR full	NR human
feature 1	feature 2	9 or 13 parameters		
delta Cn	Xcorr	1.51	1.15	3.60
	charge	0.980	0.809	3.56
	MH	1.05	0.877	3.12
	% ion match	1.76	1.43	2.81
	SP	1.43	1.18	2.80
Xcorr	charge	0.208	0.068	1.94
	MH	0.366	0.162	1.89
	Sp	0.956	0.698	1.88
	%ion match	1.37	0.846	1.83
Sp	MH	0.743	0.593	1.92
	charge	0.502	0.402	1.77
%ion match	MH	1.53	1.13	2.09
	charge	0.402	0.322	1.25
	Sp	0.959	0.775	1.12
		13 parameters		
fraction matched MSMS peaks	delta Cn	1.48	1.21	4.23
	Xcorr	1.08	0.661	3.38
	Sp	0.998	0.811	2.38
	%ion match	0.998	0.811	2.38
	MH	0.114	0.087	1.67
	charge	0.014	0.007	1.53
	peak count	0.328	0.148	1.18
fraction matched MSMS TIC	delta Cn	1.68	1.34	4.74
	Xcorr	1.33	0.843	3.82
	fraction matched MSMS peaks	0.556	0.394	2.82
	Sp	1.17	0.931	2.58
	MH	0.327	0.229	2.10
	charge	0.108	0.057	1.90
	% ion match	1.08	0.700	1.47
peak count	delta Cn	1.01	0.861	1.42

QTOF data than for ion trap data (Table 2). The precursor charge was also highly predictive in pairwise feature analysis of QTOF data when combined with five other parameters (Table 3). One factor in this predictiveness might be the asymmetrical distribution of +1 charged precursors: 39 were included as part of the training set positive peptides, whereas 311 were included in the negative peptides. As discussed in the Methods section, positive and negative peptides were not selected on this basis. Thus, observation of a +1 precursor ion is significantly more likely for a negative than positive peptide. Other parameters—such as the MS/MS spectrum peak count, the fraction of matched MS/MS peaks, and the fraction of matched MS/MS total ion current—were also significantly more predictive that for ion trap data, either alone (Table 2) or in combination with other parameters in pairwise scoring (Table 3). An enhanced signal-to-noise ratio for these data may also be valuable for the separation of positives and negatives.

One explanation for the difference in performance for the QTOF versus ion trap datasets might be the larger size of the QTOF training set. A subset of the QTOF data including 497 positive and 479 negative peptides, the same size as the smaller of the two ion trap datasets, was examined by the SVM using 13 parameters, and the ROC score computed. The test results contained 20 false positives and 23 false negatives, and a ROC score of 0.989. This compares well with the ROC score for analysis of the full sized QTOF dataset using 13 parameters (0.988). This suggests that the quality of the QTOF data, rather than the larger number of examples in the dataset, explains the improved performance compared to the ion trap-based results.

Table 4. Analysis of Training Sets Using Different Methods

method:[a]	1	2	3	SVM-9	SVM-13
training set:	false positives, false negatives				
ion trap, NR human	115, 142	133, 187	132, 369	48, 117	44, 100
ion trap, full NR	87, 142	305, 55	180, 251	62, 81	53, 70
QTOF, NR human	108, 81	126, 86	57, 285	27, 81	18, 51

[a] The cutoffs used for these comparative analyses are taken from Eng et al.[23] and Yates et al.[25] for method 1, from Link et al.[27] for method 2, and from Washburn et al.[5] and Gygi et al.[28] for method 3; the SVM analyses used both 9 and 13 parameters. False positives and negatives for methods 1−3 were calculated as the number of negative and positive peptides missed by the cutoffs, respectively.

The SVM Provides Better Performance than other Techniques. Comparison of SVM Results with Previous Analyses of SEQUEST Results Based on Thresholds. The results of the SVM analysis of the above training sets can be compared with approximations of previous methods, employing different cutoffs for delta Cn and/or Xcorr, used to evaluate SEQUEST-generated matches between peptide data and database sequences (Table 4). One simple method, used before protein sequence databases became large, involved selection of peptides as positives with delta Cn values larger than 0.1[23,25]. Using a criterion of minimizing false positives (defined here as negative peptides missed using the defined cutoffs) and false negatives (defined as positive peptides missed), this was the best performing cutoff of the 3 sets of cutoffs examined. A second method[27] included selection, as positives, of +1 peptides with Xcorr values larger than 1.5, selection of +2 and +3 peptides with Xcorr values larger than 2, and several other criteria including manual examination. Use of these cutoffs alone resulted in intermediate performance among the 3 sets of cutoffs. A more stringent method[5,28] included retention of tryptic peptides with Xcorr values above 1.9, 2.2, and 3.75 for +1, +2, and +3 peptides, a delta Cn of 0.1 or greater, and tryptic ends, followed by manual confirmation of the sequence match to the MS/MS spectrum under some circumstances. The cutoffs from this method resulted in the highest sum of false positives and false negatives for the 3 methods considered, although it gave lower levels of false positives than some of the other sets of cutoffs. The SVM results using both 9 and 13 parameters gave a significantly lower sum of false positives and false negatives than these sets of cutoffs.

Comparison of SVM Results with Qscore Results. Training set peptides analyzed using SVM calculations were also analyzed using the Qscore algorithm[29]. Qscore is a program that evaluates the quality of protein identifications from SEQUEST results using probabilistic scoring. The program requires at least two peptides for a protein identification, thus for comparison purposes with individual peptides contained in the nonredundant human database-ion trap training set, we modified the Qscore program to allow the display of calculated scores for single peptides. Qscore is not a binary classifier, thus true and false positives and negatives were not calculated. In Figure 3, the ROC curve for Qscore analysis of the ion trap-nonredundant human dataset is compared with ROC curves generated using SVM calculations. For both the 9- and the 13-parameter SVM results, the ROC curves are shifted to the upper left, indicating that for a fixed percent of false positives, there are significantly more true positive peptides from the SVM analysis. Although Qscore does not attempt to identify proteins with fewer than two peptides, these results suggest that a similar use of SVM peptides, combined with careful examination for mistakes of

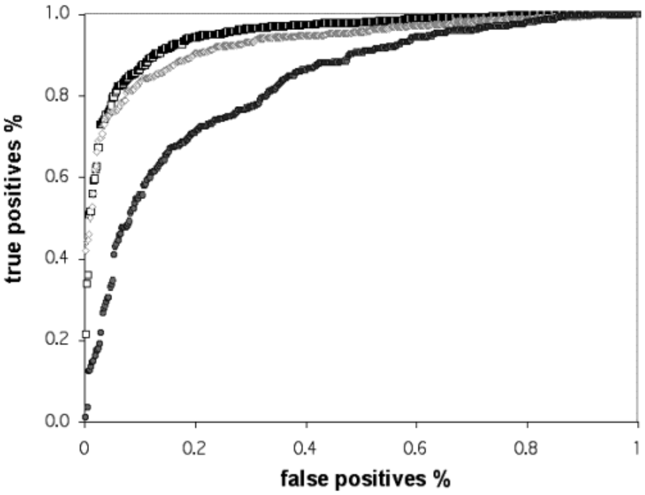


Figure 3. Comparison of Qscore with SVM analyses of a peptide training set. A ROC plot was used to compare SVM and Qscore analysis of an ion-trap nonredundant human database training set using either 9 parameters (light gray diamonds) or 13 parameters (top curve, open black boxes). Qscore was modified to calculate values for single peptides rather than requiring two peptides for an analysis, and these scores were used for the comparison. Both SVM analyses gave a higher number of true positives for a fixed number of false positives than the modified Qscore analysis (lowest curve, filled circles).

outliers from initial SVM analysis of SEQUEST data, might provide higher quality protein identifications.

Recently, Keller et al.[40] described an algorithm similar to the methods described here. They use discriminant function analysis, which is closely related to the SVM algorithm, to discriminate between true and false peptides. They also employ the expectation maximization algorithm to fit a curve that converts their predictions into probabilities. However, the method incorporates only four SEQUEST scores plus the number of tryptic peptide termini present in the matched peptides. Ions of different charge (+2 and +3) are analyzed separately, using ion trap peptide data. Xcorr', delta Cn, and ln RSp contribute to most of the discrimination between positive and negative peptides. Our data includes more parameters, and +1, +2, and +3 ions are included in one analysis. For our training sets, we find that using more parameters significantly improves the discrimination between positive and negative peptides.

Comments on Results. The support vector machine is a binary classifier, and is thus useful for making decisions about membership of analyzed entities in either of two classes. Here, we have defined the two classes as peptides correctly or incorrectly matching SEQUEST-assigned sequences. Additional applications using mass spectrometry data might include binary decisions between classes such as presence and absence of an early stage disease such as cancer[41]. Similar decision making could be applied to de novo sequenced peptides if there was sufficient information describing the fit of a de novo sequence to a peptide, and if the problem was constructed as to whether or not the de novo sequence was correct. This would likely involve other algorithms than SEQUEST, which relies mainly on pattern matching between predicted and observed MS/MS spectra.

On the basis of our experience and on the training set data examined, there are several categories of incorrectly predicted peptides. First, we initially encountered false positives based on the SEQUEST selection of peptides, matched to a given precursor ion and its MS/MS spectrum, that were not the top ranked peptides. These, and incorrectly labeled peptides, were removed after manual examination of results from initial rounds of SVM analysis. Second, the analysis of some of the tryptic maps of individual "pure" proteins indicated that there were other proteins present with more than one high-scoring peptide. Examples of negative peptides were not taken from these samples. They were instead substituted with samples of at least 97% protein purity, which were limited to injections of no more than 100 fmol of peptides. The presumed levels of impurity should thus be below the routine limit of detection for our ion trap or QTOF mass spectrometers (ca. 10 fmol) as currently configured. Nonetheless, it is possible that some of the proteins assigned as negatives might represent impurities present in the sample.

We have not been able to completely separate positives from negatives in any of the training sets examined, for data acquired on either mass spectrometer. Some of the reasons discussed below may help explain this observation. First, the training sets included the lowest scoring available positive peptides, which were often among multiple peptides correctly identifying a known protein. A number of false positive sequences with high SEQUEST scores, for example peptides selected as second choices for known positive peptides, were also included. Similar examples have been reported when using reversed-sequence databases as controls.[29] For the ion trap-non redundant human database-searched training set, there were 124 positive peptides with delta Cn values below 0.1. For +1, +2, and +3 ions there were 4, 33, and 75 additional peptides that did not meet the most stringent Xcorr cutoffs (method 3) in Table 4. There were 108 negative peptides with delta Cn values of 0.1 or above, and 14, 74, and 0 additional +1, +2, and +3 *negative* peptides with Xcorr values *above* those used for cutoffs in method 3 of Table 4. These were thus challenging training sets.

Second, a number of false positives and negatives were assigned to peptides with noisy MS/MS spectra, or with poor fragmentation in these spectra. In both cases, the information content necessary for correct sequencing will be compromised, and it is expected that accurate sequence assignments will be difficult. Of the 22 poorly fragmenting positive peptides incorrectly assigned as negatives, all but one contained an internal residue (pro, his, arg) thought to cause uneven peptide fragmentation,[42,43] and 14 contained more than one of these internal residues. It is not clear that even manual examination of these peptide MS/MS spectra will lead to a correct sequence match. A tentative identification of proteins based on these questionable peptides will require additional experiments, or additional matching peptides of higher quality, for verification. A computational indication of ambiguously identified peptides, indicated by the computed distance from the 9-parameter or 13-parameter hyperplane, should select any peptide so positioned for further scrutiny.

More generally, incorrect sequence assignments may also occur if the correct sequence is not in the database examined. For human protein sequences 80% of novel gene predictions from drafts of the Ensembl and Celera datasets occur in only one of these datasets,[44] thus an accurate and complete human protein sequence database is not yet available. Other incorrect assignments may be due to modifications to individual amino acids not incorporated into the sequences searched, or to incorrect assignment of the precursor ion charge when a lower mass accuracy instrument is used and the ratio of MS to MS/MS scans is low. The best resulting sequence will then be incorrect.

Conclusions

Using appropriate training sets, our approach allows an automated computational first-pass analysis of SEQUEST data on individual peptides. This should allow a higher throughput analysis of shotgun peptide sequencing results. For tandem mass spectrometry data, SVM analysis of experimentally obtained parameters, SEQUEST-calculated statistics, and additional parameters allows a better match between these data and peptide sequences than previous methods, using our training sets. The use of four new parameters tested here contributed significantly to the separation of positive and negative peptides. A good but not complete separation between positive and negative peptides was obtained for ion trap data using two different databases. A significantly better separation was obtained for uncalibrated QTOF MS/MS data. Using SVM calculations, the contributions of the parameters to the separation were individually examined. The parameters delta Cn, Xcorr, Sp, the fraction of the MS/MS spectrum ion current matched by peptide fragments, and the fraction of the total number of MS/MS spectrum peaks matched by peptide fragments contributed significantly to the separation of positive and negative peptides. Each training set is customized to the mass spectrometer used to collect data and the database examined. Protein identifications from these peptides will then be based on the number of individual peptides identifying a particular protein, and the distance of each peptide from the hyperplane separating positives and negatives in the appropriate training set. The reproducibility and uniqueness of the identification will also be important[38] for correct protein identifications. Manual examination of spectra of peptides with poor or ambiguous SVM-calculated scores should identify noisy or poorly fragmenting spectra that may compromise peptide identification.

References

(1) Hunt , D.; Michel, H.; Dickinson, T.; Shabanowitz, J.; Cox, A.; Sakaguchi, K.; Appella, E.; Grey, H.; Sette, A. *Science* **1992**, *256*, 1817−20.

(2) Cox, A.; Skipper, J.; Chen, Y.; Henderson, R.; Darrow, T.; Shabanowitz, J.; Engelhard, V.; Hunt, D.; Slingluff, Jr. C. *Science* **1994**, *264*, 716−719.

(3) Neubauer, G.; Gottschalk, A.; Fabrizio, P.; Seraphin, B.; Luhrmann, R.; Mann M. *Proc. Natl. Acad. Sci. U. S. A.* **1997**, *94*, 385−90.

(4) Rout, M.; Aitchison, J.; Suprapto, A.; Hjertaas, K.; Zhao, Y.; Chait B. *J. Cell Biol.* **2000**, *148*, 635−51.

(5) Washburn, M.; Wolters, D.; Yates III, J. *Nature Biotechnology* **2002**, *19*, 242−247.

(6) Gavin, A.; Bosche, M.; Krause, R.; Grandi, P.; Marzioch, M.; Bauer, A.; Schultz, J.; Rick, J.; Michon, A.; Cruciat, C.; Remor, M.; Hofert, C.; Schelder, M.; Brajenovic, M.; Ruffner, H.; Merino, A.; Klein, K.; Hudak, M.; Dickson, D.; Rudi, T.; Gnau, V.; Bauch, A.; Bastuck, S.; Huhse, B.; Leutwein, C.; Heurtier, M.; Copley, R.; Edelmann, A.; Querfurth, E.; Rybin, V.; Drewes, G.; Raida, M.; Bouwmeester, T.; Bork, P.; Seraphin, B.; Kuster, B.; Neubauer, G.; Superti-Furga, G. *Nature* **2002**, *415*, 141−7.

(7) Ho, Y.; Gruhler, A.; Heilbut, A.; Bader, G.; Moore, L.; Adams, S.; Millar, A.; Taylor, P.; Bennett, K.; Boutilier, K.; Yang, L.; Wolting, C.; Donaldson, I.; Schandorff, S.; Shewnarane, J.; Vo, M.; Taggart, J.; Goudreault, M.; Muskat, B.; Alfarano, C.; Dewar, D.; Lin, Z.; Michalickova, K.; Willems, A.; Sassi, H.; Nielsen, P.; Rasmussen, K.; Andersen, J.; Johansen, L.; Hansen, L.; Jespersen, H.; Podtelejnikov, A.; Nielsen, E.; Crawford, J.; Poulsen, V.; Sorensen, B.; Matthiesen, J.; Hendrickson, R.; Gleeson, F.; Pawson, T.; Moran, M.; Durocher, D.; Mann, M.; Hogue, C.; Figeys, D.; Tyers, M. *Nature* **2002**, *415*, 180−3.

(8) Spahr, S.; Davis, M.; McGinley, M.; Robinson, R.; Bures, E.; Beierle, J.; Mort, J.; Courchesne, P.; Chen, K.; Wahl, R.; Yu, W.,; Luethy, R.; Patterson, S. *Proteomics* **2001**, *1*, 93−107.

(9) Andersen, J.; Lyon, C., Fox, A.; Leung, A.; Lam, Y.; Steen, H.; Mann, M.; Lamond, A. *Current Biol.* **2002**, *12*, 1−11.

(10) Hunt, D.; Bone, W.; Shabanowitz, J.; Rhodes, J.; Ballard, J. *Anal. Chem.* **1981**, *53*, 1704−1706.

(11) Hunt D.; Buko A.; Ballard, J.; Shabanowitz, J.; Giordani A. *Biomed. Mass Spectrom.* **1981**, *8*, 397−408.

(12) Hunt D.; Shabanowitz , J.; Yates, J.; MeIver, R.; Hunter, R.; Syka, J.; Amy, J. *Anal. Chem.* **1985**, *57*, 2728−33.

(13) Hunt, D.; Yates, J.; Shabanowitz, J.; Winston, S.; Hauer, C. *Proc. Natl. Acad. Sci. U. S. A.* **1986**, *83*, 6233−7.

(14) Hunt, D.; Zhu, N.; Shabanowitz, J. *Rapid Commun. Mass Spectrom.* **1989**, *3*, 122−4.

(15) Biemann, K. *Biomed. Environ. Mass Spectrom.* **1988**, *16*, 99.

(16) Henzel, W.; Billeci, T.; Stults, J.; Wong, S.; Grimley, C.; Watanabe, C. *Proc. Natl. Acad. Sci. U. S. A.* **1993**, *90*, 5011−5.

(17) Mann, M.; Wilm, M. *Anal. Chem.* **1994**, *66*, 4390−9.

(18) Shevchenko, A.; Jensen, O.; Podtelejnikov, A.; Sagliocco, F.; Wilm, M.; Vorm, O.; Mortensen, P.; Shevchenko, A.; Boucherie, H.; Mann M. *Proc. Natl. Acad. Sci. U. S. A.* **1996**, *10*, 14 440−5.

(19) Qin, J.; Fenyo, D.; Zhao, Y.; Hall, W.; Chao, D.; Wilson, C.; Young, R.; Chait, B. *Anal. Chem.* **1997**, *69*, 3995−4001.

(20) Zhang, W.; Chait, B. *Anal. Chem.* **2000**, *72*, 2482−9.

(21) Clauser, K.; Baker P.; Burlingame, A. *Anal. Chem.* **1999**, *71*, 2871−82.

(22) Perkins, D.; Pappin, D.; Creasy, D.; Cottrell, J. *Electrophoresis* **1999**, *20*, 3551−67.

(23) Eng, J.; McCormack, A.; Yates III, J. *J. Am. Soc. Mass Spectrom.* **1994**, *5*, 976−989.

(24) Yates, J.; Eng, J.; McCormack, A.; Schieltz, D. *Anal. Chem.* **1995**, *15*, 1426−36.

(25) Yates, J.; Eng J.; McCormack, A. *Anal. Chem.* **1995**, *15*, 3202−10.

(26) McCormack, A.; Schieltz, D.; Goode, B.; Yang, S.; Barnes, G.; Drubin, D.; Yates, J. *Anal. Chem.* **1997**, *69*, 767−76.

(27) Link, A.; Eng, J.; Schieltz, D.; Carmack, E.; Mize, G.; Morris, D.; Garvik, B.; Yates, J. *Nat. Biotechnol.* **1999**, *17*, 676−82.

(28) Gygi, S.; Rist, B.; Griffin, T.; Eng, J.; Aebersold, R. *J. Proteome Res.* **2002**.

(29) Moore, R.; Young, M.; Lee, T. *J. Am. Soc. Mass Spectrom.* **2002**, *13*, 378−386.

(30) Boser, B.; Guyon, I.; Vapnik, V. In: *5th Annual ACM Workshop on COLT*; Haussler, D., Ed.; Pittsburgh, 1992; pp 144−152.

(31) Vapnik, V. *Statistical Learning Theory*; John Wiley and Sons: New York, 1998.

(32) Cristianini, N.; Shawe-Taylor, J. *An Introduction to Support Vector Machines*; Cambridge University Press: Cambridge, 2000.

(33) Jaakkola, T.; Diekhans, M.; Haussler, D. *Proc. Int. Conf. Intell. Syst. Mol. Biol.* **1999**, 149−58.

(34) Ding, C.; Dubchak, L. *Bioinformatics* **2001**, *17*, 349−58.

(35) Brown, M.; Grundy, W.; Lin, D.; Cristianini, N.; Sugnet, C.; Furey, T.; Ares Jr., M.; Haussler, D. *Proc. Nat. Acad. Sci.* **2000**, *97*, 262−267.

(36) Furey T.; Cristianini, N.; Duffy, N.; Bednarski D.; Schummer, M.; Haussler, D. *Bioinformatics* **2000**, *16*, 906−14.

(37) Rosenblatt, F. *Psychol. Rev.* **1959**, *65*, 386-408.

(38) Gururaja, T.; Li, W.; Bernstein, J.; Payan, D.; Anderson, D. *J. Proteome Res.* **2002**, *1*, 253−261.

(39) Keller, A.; Nesvizhskii, A.; Kolker, E.; Aebersold, R. *Anal. Chem.* **2002**, *74*, 5383−5392.

(40) Duda, R.; Hart, P. *Pattern Classification and Scene Analysis*; New York: John Wiley and Sons: 1973.

(41) Petricoin III, E.; Ardekani, A.; Hitt, B. et al. *Lancet* **2002**, *359*, 572−577.

(42) Pappayanopoulos, I. *Mass Spectrom. Rev.* **1995**, *14*, 49.

(43) Willard, B.; Kinter, M. *J. Am. Soc. Mass Spectrom.* **2001**, *12*, 1262−1271.

(44) Hogenesch, J.; Ching, K.; Batalov, S.; Su, A.; Walker, J.; Zhou, Y.; Kay, S.; Schultz, P.; Cooke, M. *Cell* **2001**, *106*, 413−415.

PR0255654

BIOINFORMATICS

Vol. 20 no. 16 2004, pages 2626–2635
doi:10.1093/bioinformatics/bth294

A statistical framework for genomic data fusion

Gert R. G. Lanckriet[1], Tijl De Bie[3], Nello Cristianini[4],
Michael I. Jordan[2] and William Stafford Noble[5,*]

[1]Department of Electrical Engineering and Computer Science, [2]Division of Computer Science, Department of Statistics, University of California, Berkeley 94720, USA, [3]Department of Electrical Engineering, ESAT-SCD, Katholieke Universiteit Leuven 3001, Belgium, [4]Department of Statistics, University of California, Davis 95618, USA and [5]Department of Genome Sciences, University of Washington, Seattle 98195, USA

Received on January 29, 2004; revised on April 7, 2004; accepted on April 23, 2004
Advance Access publication May 6, 2004

ABSTRACT

Motivation: During the past decade, the new focus on genomics has highlighted a particular challenge: to integrate the different views of the genome that are provided by various types of experimental data.

Results: This paper describes a computational framework for integrating and drawing inferences from a collection of genome-wide measurements. Each dataset is represented via a kernel function, which defines generalized similarity relationships between pairs of entities, such as genes or proteins. The kernel representation is both flexible and efficient, and can be applied to many different types of data. Furthermore, kernel functions derived from different types of data can be combined in a straightforward fashion. Recent advances in the theory of kernel methods have provided efficient algorithms to perform such combinations in a way that minimizes a statistical loss function. These methods exploit semidefinite programming techniques to reduce the problem of finding optimizing kernel combinations to a convex optimization problem. Computational experiments performed using yeast genome-wide datasets, including amino acid sequences, hydropathy profiles, gene expression data and known protein–protein interactions, demonstrate the utility of this approach. A statistical learning algorithm trained from all of these data to recognize particular classes of proteins—membrane proteins and ribosomal proteins—performs significantly better than the same algorithm trained on any single type of data.

Availability: Supplementary data at http://noble.gs.washington.edu/proj/sdp-svm

Contact: noble@gs.washington.edu

INTRODUCTION

The recent availability of multiple types of genome-wide data provides biologists with complementary views of a single genome and highlights the need for algorithms capable of unifying these views. In yeast, for example for a given gene we typically know the protein it encodes, that protein's similarity to other proteins, its hydrophobicity profile, the mRNA expression levels associated with the given gene under hundreds of experimental conditions, the occurrences of known or inferred transcription factor binding sites in the upstream region of that gene and the identities of many of the proteins that interact with the given gene's protein product. Each of these distinct data types provides one view of the molecular machinery of the cell. In the near future, research in bioinformatics will focus more and more heavily on methods of data fusion.

Different data sources are likely to contain different and thus partly independent information about the task at hand. Combining those complementary pieces of information can be expected to enhance the total information about the problem at hand. One problem with this approach, however, is that genomic data come in a wide variety of data formats: expression data are expressed as vectors or time series; protein sequence data as strings from a 20-symbol alphabet; gene sequences are strings from a different (4-symbol) alphabet; protein–protein interactions are best expressed as graphs and so on.

This paper presents a computational and statistical framework for integrating heterogeneous descriptions of the same set of genes. The approach relies on the use of kernel-based statistical learning methods that have already proven to be very useful tools in bioinformatics (Noble, 2004). These methods represent the data by means of a kernel function, which defines similarities between pairs of genes, proteins and so on. Such similarities can be quite complex relations, implicitly capturing aspects of the underlying biological machinery. One reason for the success of kernel methods is that the kernel function takes relationships that are implicit in the data and makes them explicit, so that it is easier to detect patterns. Each kernel function thus extracts a specific type of information from a given dataset, thereby providing a partial description or view of the data. Our goal is to find a kernel that best represents all the information available for a given statistical learning task. Given many partial descriptions of the data, we

[*]To whom correspondence should be addressed at: Health Sciences Center, Box 357730, 1705 NE Pacific Street, Seattle, WA 98195, USA.

solve the mathematical problem of combining them using a convex optimization method known as semidefinite programming (SDP) (Nesterov and Nemirovsky, 1994; Vandenberghe and Boyd, 1996). This SDP-based approach (Lanckriet *et al.*, 2004) yields a general methodology for combining many partial descriptions of data that is statistically sound, as well as computationally efficient and robust.

In order to demonstrate the feasibility of these methods, we apply them to the recognition of two important groups of proteins in yeast—ribosomal proteins and membrane proteins. The ribosome is a universal protein complex that is responsible for the translation of mRNA into the corresponding amino acid sequence via the universal genetic code. The structure of the ribosome has been solved (Schluenzen *et al.*, 2000; Harms *et al.*, 2001), although the precise roles of many auxiliary factors are not completely understood. Proteins that participate in the ribosome share similar sequence features and correlated mRNA expression patterns (Brown *et al.*, 2000).

Membrane proteins are proteins that anchor in one of the various membranes in the cell, including the plasma, ER, golgi, peroxisomal, vacuolar, cellular and mitochondrial inner and outer membranes. Many membrane proteins serve important communicative functions between cellular compartments and between the inside and the outside of the cell (Alberts *et al.*, 1998). Classifying a protein as a membrane protein or not based on protein sequence is non-trivial and has been the subject of much previous research (Engleman *et al.*, 1986; Krogh *et al.*, 2001; Chen and Rost, 2002). This is a typical statistical learning problem in which a single type of feature derived from the protein sequence cannot provide the full story.

For both of these protein classes, we demonstrate that incorporating knowledge derived from the amino acid sequences, gene expression data and known protein–protein interactions significantly improves classification performance relative to our method trained on any single type of data. The SDP-based approach also performs better than a classifier trained using a naive, unweighted combination of kernels, and the method continues to perform well in the presence of artificially induced experimental noise.

We begin by outlining the main ideas of the kernel approach to pattern analysis, providing examples of kernels defined on yeast genome-wide datasets. We then describe how these kernels can be integrated using SDP to provide a unified description. Finally, we describe a series of computational experiments that demonstrate the validity and power of the kernel approach to data fusion for recognition of ribosomal and membrane proteins in yeast.

KERNEL METHODS

Kernel methods work by embedding data items (corresponding to genes, proteins, and so on) into a vector space, $\mathcal{F}$, called a feature space (Cristianini and Shawe-Taylor, 2000; Schölkopf and Smola, 2002; Wahba, 1990; Vapnik, 1998, 1999). A key characteristic of kernel methods is that the embedding in feature space is generally defined implicitly, by specifying an inner product for the feature space. Thus, for a pair of data items, x_1 and x_2, denoting their embeddings as $\Phi(x_1)$ and $\Phi(x_2)$, respectively, we specify the inner product of the embedded data, $\langle \Phi(x_1), \Phi(x_2) \rangle$, via a kernel function $K(x_1, x_2)$. Any symmetric, positive semidefinite function is a valid kernel function, corresponding to an inner product in some feature space. Note that if all we require is inner products, then we neither need to have an explicit representation of the mapping Φ nor even need to know the nature of the feature space. It suffices to be able to evaluate the kernel function.

Evaluating the kernel on all pairs of data points yields a symmetric, positive semidefinite matrix known as the kernel matrix or the Gram matrix. Intuitively, a kernel matrix can be regarded as a matrix of generalized similarity measures among the data points. The first stage of processing in a kernel method is to reduce the data by computing this matrix.

The reduction to a kernel matrix reflects the fact that kernel methods are generally based on linear statistical procedures in feature space. In particular, the classification algorithm that we use in this paper—known as a support vector machine (SVM) (Boser *et al.*, 1992)—forms a linear discriminant boundary in feature space. Consider a dataset consisting of n pairs (x_i, y_i), where x_i is the i-th data item (e.g. a protein sequence) and $y_i \in \{-1, 1\}$ is a label (e.g. membrane or non-membrane). Compute the $n \times n$ kernel matrix whose (i, j)-th entry is $K(x_i, x_j)$. Given this matrix, and given the labels y_i, we can throw away the original data; the problem of fitting the SVM to data reduces to an optimization procedure that is based entirely on the kernel matrix and the labels.

Different kernel functions correspond to different embeddings of the data and thus can be viewed as capturing different notions of similarity. For example, in a space derived from amino acid sequences, two genes that are close to one another will have protein products with very similar amino acid sequences. This amino acid space would be quite different from a space derived from microarray gene expression measurements, in which closeness would indicate similarity of the expression profiles of the genes. In general, a single type of data can be mapped into many different feature spaces. The choice of feature space is made implicitly via the choice of kernel function.

For the tasks of ribosomal and membrane protein classification we experiment with seven kernel matrices derived from three different types of data: four from the primary protein sequence, two from protein–protein interaction data, and one from mRNA expression data. These are summarized in Table 1.

Protein sequence

Smith–Waterman, BLAST and Pfam HMM kernels A homolog of a membrane protein is likely to be located in

Table 1. Kernel functions

Kernel	Data	Similarity measure
K_{SW}	protein sequences	Smith-Waterman
K_{B}	protein sequences	BLAST
K_{Pfam}	protein sequences	Pfam HMM
K_{FFT}	hydropathy profile	FFT
K_{LI}	protein interactions	linear kernel
K_{D}	protein interactions	diffusion kernel
K_{E}	gene expression	radial basis kernel
K_{RND}	random numbers	linear kernel

The table lists the seven kernels used to compare proteins, the data on which they are defined, and the method for computing similarities. The final kernel, K_{RND}, is included as a control. All kernel matrices, along with the data from which they were generated, are available at noble.gs.washington.edu/proj/sdp-svm.

the membrane, and similarly for the ribosome. Therefore, we define three kernel matrices based upon standard homology detection methods. The first two sequence-based kernel matrices (K_{SW} and K_{B}) are generated using the BLAST (Altschul *et al.*, 1990) and Smith–Waterman (SW) (Smith and Waterman, 1981) pairwise sequence comparison algorithms, as described previously (Liao and Noble, 2002). Both algorithms use gap opening and extension penalties of 11 and 1, and the BLOSUM 62 matrix. As matrices of BLAST or Smith–Waterman scores are not necessarily positive semidefinite, we represent each protein as a vector of scores against all other proteins. Defining the similarity between proteins as the inner product between the score vectors (the so-called empirical kernel map, Tsuda 1999) leads to valid kernel matrices, one for the BLAST score and one for the SW score. Note that including in the comparison set proteins with unknown labels allows the kernel to exploit this unlabeled data. The third kernel matrix (K_{Pfam}) is a generalization of the previous pairwise comparison-based matrices in which the pairwise comparison scores are replaced by expectation values derived from hidden Markov models (HMMs) in the Pfam database (Sonnhammer *et al.*, 1997).

Fast Fourier Transform (FFT) kernel The fourth sequence-based kernel matrix (K_{FFT}) is specific to the membrane protein recognition task. This kernel directly incorporates information about hydrophobicity patterns, which are known to be useful in identifying membrane proteins. Generally, each membrane protein passes through the membrane several times. The transmembrane regions of the amino acid sequence are typically hydrophobic, whereas the non-membrane portions are hydrophilic. This specific hydrophobicity profile of the protein allows it to anchor itself in the cell membrane. Because the hydrophobicity profile of a membrane protein is critical to its function, this profile is better conserved in evolution than the specific amino acid sequence. Therefore, classical methods for determining whether a protein $\mathbf{p}_i$ (consisting of $|\mathbf{p}_i|$ amino acids) spans a membrane (Chen and Rost, 2002), depend upon its hydropathy profile $h(\mathbf{p}_i) \in \mathbb{R}^{|\mathbf{p}_i|}$: a vector containing the hydrophobicities of the amino acids along the protein (Engleman *et al.*, 1986; Black and Mould, 1991; Hopp and Woods, 1981). The FFT kernel uses hydropathy profiles generated from the Kyte–Doolittle index (Kyte and Doolittle, 1982). This kernel compares the frequency content of the hydropathy profiles of the two proteins. First, the hydropathy profiles are pre-filtered with a low-pass filter to reduce noise:

$$h_f(\mathbf{p}_i) = f \otimes h(\mathbf{p}_i),$$

where $f = \frac{1}{4}(1\ 2\ 1)$ is the impulse response of the filter and $\otimes$ denotes convolution with that filter. After pre-filtering the hydropathy profiles (and if necessary appending zeros to make them equal in length—a commonly used technique not altering the frequency content), their frequency contents are computed with the FFT algorithm:

$$H_f(\mathbf{p}_i) = \mathrm{FFT}[h_f(\mathbf{p}_i)].$$

The FFT kernel between proteins $\mathbf{p}_i$ and $\mathbf{p}_j$ is then obtained by applying a Gaussian kernel function to the frequency contents of their hydropathy profiles:

$$K_{\mathrm{FFT}}(\mathbf{p}_i, \mathbf{p}_j) = \exp[-\|H_f(\mathbf{p}_i) - H_f(\mathbf{p}_j)\|^2/2\sigma]$$

with width $\sigma = 10$. This kernel detects periodicities in the hydropathy profile, a feature that is relevant to the identification of membrane proteins and complementary to the previous, homology-based kernels.

Protein interactions: linear and diffusion kernels For the recognition of ribosomal proteins, protein–protein interactions are clearly informative, since all ribosomal proteins interact with other ribosomal proteins. For membrane protein recognition, we expect information about protein–protein interactions to be informative for two reasons. First, hydrophobic molecules or regions of molecules are probably more likely to interact with each other than with hydrophilic molecules or regions. Second, transmembrane proteins are often involved in signaling pathways, and therefore, different membrane proteins are likely to interact with a similar class of molecules upstream and downstream in these pathways (e.g. hormones upstream or kinases downstream). The two protein interaction kernels are generated using medium- and high-confidence interactions from a database of known interactions (von Mering *et al.*, 2002). These interactions can be represented as an interaction matrix, in which rows and columns correspond to proteins, and binary entries indicate whether the two proteins interact.

The first interaction kernel matrix (K_{LI}) is comprised of linear interactions, i.e. inner products of rows and columns from the centered, binary interaction matrix. The more similar the interaction pattern (corresponding to a row or column from the interaction matrix) for a pair of proteins, the larger the inner product will be.

An alternative way to represent the same interaction data is to consider the proteins as nodes in a large graph. In this graph, two proteins are linked when they interact and otherwise not. Kondor and Lafferty (2002) propose a general method for establishing similarities between the nodes of a graph, based on a random walk on the graph. This method efficiently accounts for all possible paths connecting two nodes, and for the lengths of those paths. Nodes that are connected by shorter paths or by many paths are considered more similar. The resulting diffusion kernel generates the second interaction kernel matrix (K_{D}).

An appealing characteristic of the diffusion kernel is its ability, like the empirical kernel map, to exploit unlabeled data. In order to compute the diffusion kernel, a graph is constructed using all known protein–protein interactions, including interactions involving proteins whose subcellular locations are unknown. Therefore, the diffusion process includes interactions involving unlabeled proteins, even though the kernel matrix only contains entries for labeled proteins. This allows two labeled proteins to be considered close to one another if they both interact with an unlabeled protein.

Gene expression: radial basis kernel Finally, we also include a kernel constructed entirely from microarray gene expression measurements. A collection of 441 distinct experiments was downloaded from the Stanford Microarray Database (genome-www.stanford.edu/microarray). This data provides us with a 441-element expression vector characterizing each gene. A Gaussian kernel matrix (K_{E}) is computed from these vectors by applying a Gaussian kernel function with width $\sigma = 100$ to each pair of 441-element vectors, characterizing a pair of genes. Gene expression data is expected to be useful for recognizing ribosomal proteins, since their expression signatures are known to be highly correlated with one another. We do not expect that gene expression will be particularly useful for the membrane classification task. We do not need to eliminate the kernel a priori, however; as explained in the following section, our method is able to provide an a posteriori measure of how useful a data source is relative to the other sources of data.

KERNEL METHODS FOR DATA FUSION

Each of the kernel functions described above produces, for the yeast genome, a square matrix in which each entry encodes a particular notion of similarity of one yeast protein to another. Implicitly, each matrix also defines an embedding of the proteins in a feature space. Thus, the kernel representation casts heterogeneous data—variable-length amino acid strings, real-valued gene expression data, and a graph of protein–protein interactions—into the common format of kernel matrices.

The kernel formalism also allows these various matrices to combine. Basic algebraic operations such as addition, multiplication and exponentiation preserve the key property of positive semidefiniteness, and thus allow a simple but powerful algebra of kernels (Berg *et al*., 1984). For example, given two kernel functions K_1 and K_2, inducing the embeddings $\Phi_1(x)$ and $\Phi_2(x)$, respectively, it is possible to define the kernel $K = K_1 + K_2$, inducing the embedding $\Phi(x) = [\Phi_1(x), \Phi_2(x)]$. Of even greater interest, we can consider parameterized combinations of kernels. In particular, given a set of kernels $\mathcal{K} = \{K_1, K_2, \ldots, K_m\}$, we can form the linear combination

$$K = \sum_{i=1}^{m} \mu_i K_i, \tag{1}$$

where the weights are constrained to be non-negative to assure positive semidefiniteness: $\mu_i \geq 0; i = 1, \ldots, m$. We consider this kind of kernel combination in this paper.

As we have discussed, fitting a kernel-based statistical classifier (such as the SVM) to data involves solving an optimization problem based on the kernel matrix and the labels. In particular, the SVM finds a linear discriminant in feature space that has maximal distance (‘margin’) between the members of the positive and negative classes. The algorithm for finding this optimal linear discriminant involves solving an optimization problem known as a quadratic program, a particular form of convex optimization problem for which efficient solutions are known (Nesterov and Nemirovsky, 1994).

The specific form of SVM that we use in this paper is the 1-norm soft margin support vector machine (Boser *et al*., 1992; Schölkopf and Smola, 2002). An SVM forms a linear discriminant boundary in the feature space $\mathcal{F}$: $f(x) = \mathbf{w}^{\mathrm{T}}\Phi(x) + b$, where $\mathbf{w} \in \mathcal{F}$ and $b \in \mathbb{R}$. Given a labeled sample $S_n = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, a 1-norm soft margin SVM optimizes with respect to $\mathbf{w}$ and b so as to maximize the distance (‘margin’) between the positive and negative class, allowing misclassifications (therefore ‘soft margin’):

$$\begin{aligned} &\min_{\mathbf{w},b,\xi} \ \mathbf{w}^{\mathrm{T}}\mathbf{w} + C\sum_{i=1}^{n} \xi_i \\ &\quad \text{subject to } y_i[\mathbf{w}^{\mathrm{T}}\Phi(x_i) + b] \geq 1 - \xi_i n \\ &\qquad\qquad \xi_i \geq 0, \ \ i = 1, \ldots, n \end{aligned} \tag{2}$$

where C is a regularization parameter, trading off error against margin. By considering the dual problem corresponding to Equation (2), one can prove (Schölkopf and Smola, 2002) that the weight vector can be expressed as $\mathbf{w} = \sum_{i=1}^{n} \alpha_i \Phi(x_i)$, where the support values α_i are solutions of the following dual quadratic program (QP):

$$\begin{aligned} &\max_{\alpha} \ 2\alpha^{\mathrm{T}}\mathbf{e} - \alpha^{\mathrm{T}}\mathrm{diag}(\mathbf{y})K\,\mathrm{diag}(\mathbf{y})\alpha \\ &\quad \text{subject to } 0 \leq \alpha \leq C, \ \ \alpha^{\mathrm{T}}\mathbf{y} = 0, \end{aligned} \tag{3}$$

where $\mathbf{y} = (y_1, \ y_2, \ldots, y_n)^{\mathrm{T}}$ and diag($\mathbf{y}$) is a diagonal matrix with entries given by the elements of $\mathbf{y}$. An unlabeled data item x_{new} can subsequently be classified by computing the linear function

$$f(x_{\text{new}}) = \mathbf{w}^{\mathrm{T}}\Phi(x_{\text{new}}) + b = \sum_{i=1}^{n} \alpha_i K(x_i, x_{\text{new}}) + b.$$

If $f(x_{\text{new}})$ is positive, then we classify x_{new} as belonging to class $+1$; otherwise, we classify x_{new} as belonging to class -1.

In Lanckriet *et al*. (2004), we show that for a fixed trace of K, the classification performance is bounded by a function of the optimum achieved in Equation (3): the smaller, the better the guaranteed performance. Thus, whereas in the standard SVM formulation K is a given kernel matrix, we can in fact learn an optimal kernel matrix by parameterizing K and minimizing Equation (3) with respect to these kernel parameters. More concretely, we consider the parameterization in Equation (1) with additional trace and positive semidefiniteness constraints. Plugging this into Equation (3) and minimizing with respect to μ_i gives:

$$\begin{aligned} \min_{\mu_i} \max_{\alpha} \; & 2\alpha^{\mathrm{T}}\mathbf{e} - \alpha^{\mathrm{T}}\mathrm{diag}(\mathbf{y})\left(\sum_{i=1}^{m} \mu_i K_i\right)\mathrm{diag}(\mathbf{y})\alpha \\ \text{subject to } & 0 \le \alpha \le C, \quad \alpha^{\mathrm{T}}\mathbf{y} = 0, \\ & \mathrm{trace}\left(\sum_{i=1}^{m} \mu_i K_i\right) = c, \\ & \sum_{i=1}^{m} \mu_i K_i \succeq 0, \end{aligned}$$

where c is a constant. Again considering the Lagrangian dual problem, we can show that this problem of finding optimal μ_i and α_i reduces to a convex optimization problem known as a semidefinite program (SDP):

$$\begin{aligned} \min_{\mu_i, t, \lambda, \gamma \ge 0} \; & t \\ \text{subject to } & \mathrm{trace}\left(\sum_{i=1}^{m} \mu_i K_i\right) = c, \\ & \sum_{i=1}^{m} \mu_i K_i \succeq 0, \\ & \begin{pmatrix} Y(\mu) & \mathbf{e} + \gamma + \lambda\mathbf{y} \\ (\mathbf{e} + \gamma + \lambda\mathbf{y})^{\mathrm{T}} & t - 2C\delta^{\mathrm{T}}\mathbf{e} \end{pmatrix} \succeq 0, \end{aligned} \tag{4}$$

where we let $Y(\mu) = \mathrm{diag}(\mathbf{y})(\sum_{i=1}^{m} \mu_i K_i)\mathrm{diag}(\mathbf{y})$. SDP can be viewed as a generalization of linear programming, where scalar linear inequality constraints are replaced by more general linear matrix inequalities (LMIs): $F(\mathbf{x}) \succeq 0$, meaning that the matrix F has to be in the cone of positive semidefinite matrices, as a function of the decision variables $\mathbf{x}$. Note that the first LMI constraint in Equation (4), $K = \sum_{i=1}^{m} \mu_i K_i \succeq 0$, emerges very naturally because the optimal kernel matrix must indeed come from the cone of positive semidefinite matrices. Linear programs and semidefinite programs are both instances of convex optimization problems, and both can be solved via efficient interior-point algorithms (Vandenberghe and Boyd, 1996).

In this paper, the weights μ_i are constrained to be non-negative and the K_i are positive semidefinite and normalized ($[K_i]_{jj} = 1$) by construction; thus $K \succeq 0$ is automatically satisfied. In that case, we can show that the SDP in Equation (4) reduces to a quadratically constrained quadratic program (*QCQP*), which is a special case of SDP that can be solved more efficiently:

$$\begin{aligned} \max_{\alpha, t} \; & 2\alpha^{\mathrm{T}}\mathbf{e} - ct \\ \text{subject to } & t \ge \frac{1}{n}\alpha^{\mathrm{T}}\mathrm{diag}(\mathbf{y})K_i\mathrm{diag}(\mathbf{y})\alpha, \\ & \alpha^{\mathrm{T}}\mathbf{y} = 0, \\ & 0 \le \alpha \le C, \end{aligned} \tag{5}$$

for $i = 1, \ldots, m$. Thus, by solving a QCQP, we are capable of finding an adaptive combination of kernel matrices—and thus an adaptive combination of heterogeneous information sources—that solves our classification problem. The output of our procedure is a set of weights μ_i and a discriminant function based on these weights. We obtain a classification decision that merges information encoded in the various kernel matrices, and we obtain weights μ_i that reflect the relative importance of these information sources.

EXPERIMENTAL DESIGN

In order to test our kernel-based approach in the setting of yeast protein classification, we use as a gold standard the annotations provided by the MIPS Comprehensive Yeast Genome Database (CYGD) (Mewes *et al*., 2000). The CYGD assigns 1125 yeast proteins to particular complexes, of which 138 participate in the ribosome. The remaining approximately 5000 yeast proteins are unlabeled. Similarly, CYGD assigns subcellular locations to 2318 yeast proteins, of which 497 belong to various membrane protein classes, leaving ~4000 yeast proteins with uncertain location.

The primary input to the classification algorithm is a collection of kernel matrices from Table 1. For membrane protein classification, for comparison with the SDP/SVM learning algorithm, we consider several classical biological methods that are commonly used to determine whether a Kyte–Doolittle plot corresponds to a membrane protein, as well as a state-of-the-art technique using HMMs to predict transmembrane helices in proteins (Krogh *et al*., 2001). The first method relies on the observation that the average hydrophobicity of membrane proteins tends to be higher than that of non-membrane proteins, because the transmembrane regions are more hydrophobic. We therefore define f_1 as the average

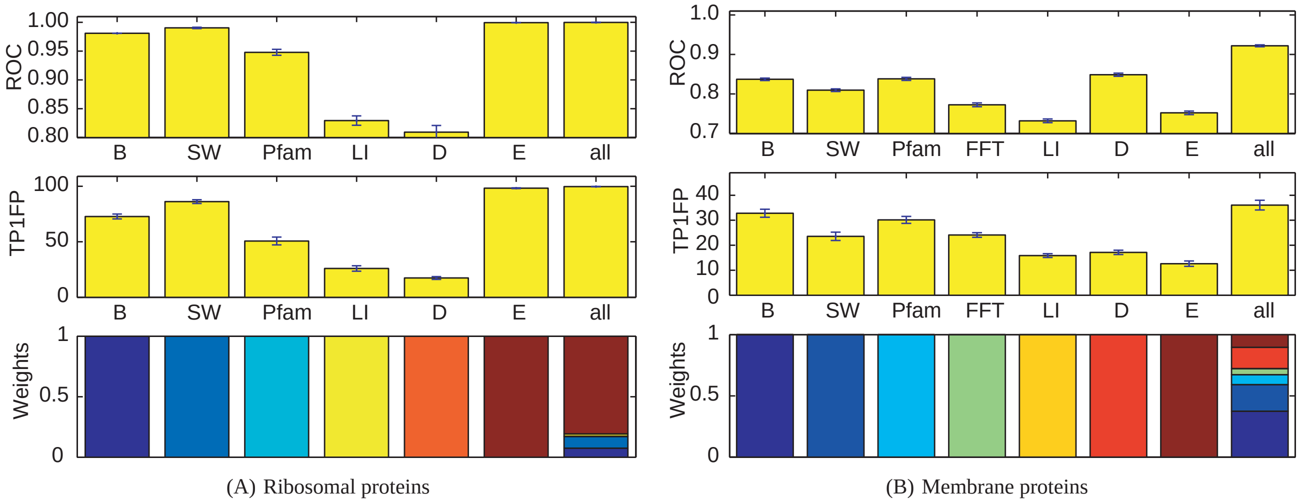


Fig. 1. Combining datasets yields better classification performance. The height of the bars in the upper two plots are proportional to the ROC score (top) and the percentage of true positives at one percent false positives (middle), for the SDP/SVM method using the given kernel. Error bars indicate standard error across 30 random train/test splits. In the lower plots, the heights of the colored bars indicate the relative weights of the different kernel matrices in the optimal linear combination. These results in tabular form, along with percent accuracy measurements, are given in the online supplement.

hydrophobicity, normalized by the length of the protein. We will compare the classification performance of our statistical learning algorithm with this metric.

However, clearly, f_1 is too simplistic. For example, protein regions that are not transmembrane only induce noise in f_1. Therefore, an alternative metric filters the hydrophobicity plot with a low-pass filter and then computes the number, the height and the width of those peaks above a certain threshold (Chen and Rost, 2002). The filter is intended to smooth out periodic effects. We implement two such filters, choosing values for the filter order and the threshold based on Chen and Rost (2002). In particular, we define f_2 as the area under the 7th-order low-pass filtered Kyte–Doolittle plot and above a threshold value 2, normalized by the length of the protein. Similarly, f_3 is the corresponding area using a 20th-order filter and a threshold of 1.6.

Finally, the transmembrane HMM (TMHMM) Web server (www.cbs.dtu.dk/services/TMHMM) is used to make predictions for each protein. In Krogh *et al.* (2001), transmembrane proteins are identified by TMHMM using three different metrics: the expected number of amino acids in transmembrane helices, the number of transmembrane helices predicted by the N-best algorithm, and the expected number of transmembrane helices. Only the first two of these metrics are provided in the TMHMM output. Accordingly, we produce two lists of proteins, ranked by the number of predicted transmembrane helices (T_{PH}) and by the expected number of residues in transmembrane helices (T_{ENR}).

Each algorithm's performance is measured by randomly splitting the data (without stratifying) into a training and test set in a ratio of 80/20. We report the receiver operating characteristic (ROC) score, which is the area under a curve that plots true positive rate as a function of false positive rate for differing classification thresholds (Hanley and McNeil, 1982; Gribskov and Robinson, 1996). The ROC score measures the overall quality of the ranking induced by the classifier, rather than the quality of a single point in that ranking. An ROC score of 0.5 corresponds to random guessing, and an ROC score of 1.0 implies that the algorithm succeeded in putting all of the positive examples before all of the negatives. In addition, we select the point on the ROC curve that yields a 1% false positive rate, and we report the rate of true positives at this point (TP1FP). Each experiment is repeated 30 times with different random splits in order to estimate the variance of the performance values.

RESULTS

We performed computational experiments that study the performance of the SDP/SVM approach as a function of the number of data sources, compare the approach to a simpler approach using an unweighted combination of kernels, study the robustness of the method to the presence of noise, and for membrane protein classification, compare the performance of the method to classical biological methods and state-of-the-art techniques for membrane protein classification.

Ribosomal protein classification

Figure 1A shows the results of training an SVM to recognize the cytoplasmic ribosomal proteins, using various kernel functions. Very good recognition performance can be achieved using several types of data individually: the Smith–Waterman kernel yields an ROC of 0.9903 and a TP1FP of 86.23%,

Table 2. Classification performance on the cytoplasmic ribosomal class, in the presence of noise or improper weighting

K_{SW}	K_{PF}	K_{LI}	K_{B}	K_{D}	$K_{\mathrm{R1,\ldots,R6}}$	$K_{\mathrm{R7,\ldots,R12}}$	TP1FP	ROC
5.08	0.31	0.22	0.39	0.00	–	–	88.21 ± 1.73%	0.9933 ± 0.0011
5.07	0.31	0.22	0.39	0.00	0.01	–	88.19 ± 1.60%	0.9932 ± 0.0011
5.06	0.30	0.22	0.38	0.01	0.02	0.01	88.08 ± 1.65%	0.9932 ± 0.0010
1.00	1.00	1.00	1.00	1.00	–	–	75.20 ± 2.38%	0.9906 ± 0.0012
1.00	1.00	1.00	1.00	1.00	1.00	–	59.66 ± 3.03%	0.9791 ± 0.0017
1.00	1.00	1.00	1.00	1.00	1.00	1.00	42.87 ± 2.59%	0.9620 ± 0.0027

The table lists the percentage of true positives at 1% false positives (TP1FP) and the ROC score for several combinations of kernels. The first three lines of results were obtained using SDP-SVM, and the last three lines by setting the weights uniformly. Columns 1 through 5 report the average weights for the potentially informative kernels (averaged over the training/test splits), column 6 contains the average weight for a first set of 6 random kernels (averaged over the 6 kernels and the training/test splits) and column 7 similarly for an additional set of 6 random kernels. Each random kernel was generated by computing inner products on randomly generated 400-element vectors, in which each vector component was sampled independently from a standard normal distribution. In the table, an en-rule indicates that the corresponding kernel is not considered in the combination.

and the gene expression kernel yields corresponding values of 0.9995 and 98.31%. However, combining all six kernels using SDP provides still better performance (ROC of 0.9998 and TP1FP of 99.71%). These differences, though small, are statistically significant according to a Bonferroni corrected Wilcoxon signed rank test.

For this task, the SDP approach performs no better than the naive approach of combining all six kernel matrices in an unweighted fashion. Note, however, that the SDP solution also provides an additional explanatory result, in the form of the weights assigned to the kernels. These weights are illustrated in Figure 1A and suggest that, as expected, the cytoplasmic ribosomal proteins are best defined by their expression profiles and, secondarily, by their sequences. An additional benefit offered by SDP over the naive approach is its robustness in the presence of noise. In order to illustrate this effect, we omit the expression kernel from the combination and add six kernels generated from Gaussian noise ($K_{R1,\ldots,R6}$). This set of kernels degrades the performance of the naive combination, but has no effect on the SDP/SVM. With six additional random kernels ($K_{R7,\ldots,R12}$) the benefit of optimizing the weights is even more apparent (Table 2 and the online supplement).

Among the 30 train/test splits, seven proteins are consistently mislabeled by SDP/SVM (see online supplement). These include one, YLR406C (RPL31B), that was previously misclassified as non-ribosomal in an SVM-based study using a smaller microarray expression dataset (Brown *et al.*, 2000). In order to better understand the seven false negatives, we separated out the kernel-specific components of the SVM discriminant score. In nearly every case, the component corresponding to the gene expression kernel is the only one that is negative (data not shown). In other words, these seven proteins show atypical expression profiles, relative to the rest of the ribosome, which explains their misclassification by the SVM. Visual inspection of the expression matrix (online supplement) verifies these differences.

Finally, the trained SVM was applied to the set of approximately 5000 proteins that are not annotated in CYGD as participating in any protein complex. Among these, the SVM predicts that 14 belong in the cytoplasmic ribosomal class (see online supplement). However, nine of these predictions correspond to questionable ORFs, each of which lies directly opposite a gene that encodes a ribosomal protein. In these cases, the microarray expression data for the questionable ORFs undoubtedly reflect the strong pattern of expression from the corresponding ribosomal genes. Among the remaining five proteins, two (YNL119W and YKL056C) were predicted to be ribosomal proteins in a previous SVM-based study (Brown *et al.*, 2000). YKL056C is particularly interesting: it is a highly conserved, ubiqitous protein homologous to the mammalian translationally controlled tumor protein (Gross *et al.*, 1989) and to human IgE-dependent histamine-releasing factor.

Membrane protein classification

The results of the first membrane protein classification experiment are summarized in Figure 1(B). The plot illustrates that SDP/SVM learns significantly better from the heterogeneous data than from any single data type. The mean ROC score using all seven kernel matrices (0.9219 ± 0.0024) is significantly higher than the best ROC score using only one matrix (0.8487 ± 0.0039 using the diffusion kernel). This improvement corresponds to a change in TP1FP of 18.91%, from 17.15 to 36.06% and a change in test set accuracy of 7.36%, from 81.30 to 88.66%.

As expected, the sequence-based kernels yield good individual performance. The value of these kernels is evidenced by their corresponding ROC scores and by the relatively large weights assigned to the sequence-based kernels by the SDP. These weights are as follows: $\mu_{\mathrm{B}} = 2.62$, $\mu_{\mathrm{SW}} = 1.52$, $\mu_{\mathrm{Pfam}} = 0.57$, $\mu_{\mathrm{FFT}} = 0.35$, $\mu_{\mathrm{LI}} = 0.01$, $\mu_{\mathrm{D}} = 1.21$ and $\mu_{\mathrm{E}} = 0.73$[1]. Thus, two of the three kernel matrices that receive weights >1 are derived from the amino acid sequence.

[1] For ease of interpretation, we scale the weights such that their sum is equal to the number m of kernel matrices.

Table 3. Classification performance on the membrane proteins, in the presence of noise or improper weighting

K_B	K_{SW}	K_D	K_E	K_{R1}	K_{R2}	K_{R3}	K_{R4}	TP1FP (%)	ROC
1.81	1.05	0.73	0.42	–	–	–	–	35.71 ± 2.13	0.9196 ± 0.0023
3.30	1.98	1.31	0.79	0.08	0.17	0.21	0.17	34.14 ± 2.09	0.9145 ± 0.0026
1.00	1.00	1.00	1.00	–	–	–	–	33.87 ± 2.20	0.9180 ± 0.0026
1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	26.24 ± 1.39	0.8627 ± 0.0033

The table lists the percentage true positives at 1% false positives (TP1FP) and the ROC score for several combinations of kernels. The first two lines of results were obtained using SDP–SVM, and the last two lines were obtained using a uniform kernel weighting. Columns 1 through 8 report the average weights for the respective kernels (averaged over the training/test splits). A en-rule indicates that the corresponding kernel is not considered in the combination.

The results also show that the interaction-based diffusion kernel is more informative than the expression kernel. The diffusion kernel yields an individual ROC score which is significantly higher than the expression kernel, and the SDP also assigns a larger weight to the diffusion kernel (1.21) than to the expression kernel (0.73). Accordingly, removing the diffusion kernel reduces the percentage true positives at one percent false positives from 36.06 to 34.52%, whereas removing the expression kernel has a smaller effect, leading to a TP1FP of 35.88%. Further description of the results obtained when various subsets of kernels are used is provided in the online supplement.

In order to test the robustness of our approach, we performed a second experiment using four real kernels—K_B, K_{SW}, K_D and K_E—and four Gaussian noise kernels $K_{R1,...,R4}$. Using all eight kernels, SDP assigns values to the random kernels weights that are close to zero. Therefore, the overall performance, as measured by TP1FP or ROC score, remains virtually unchanged. In contrast, the performance of the uniformly weighted kernel combination, which was previously competitive with the SDP combination, degrades significantly in the presence of noise, from TP1FP of 33.87% down to 26.24%. Thus, the SDP approach provides a kind of insurance against the inclusion of noisy or irrelevant kernels (Table 3).

We also compared the membrane protein classification performance of the SDP/SVM method with that of several other techniques for membrane protein classification. The ROC and TP1FP for these methods are listed in Table 4. The results indicate that using learning in this context dramatically improves the results relative to the simple hydropathy profile approach. The SDP/SVM method also improves, though to a lesser degree, upon the performance of the state-of-the-art TMHMM model. However, the comparison to TMHMM is somewhat problematic, for several reasons. First, TMHMM is provided as a pre-trained model. As such, a cross-validated comparison with the SDP/SVM is not possible. In particular, some members of the cross-validation test sets were almost certainly used in training TMHMM, making its performance estimate too optimistic. On the other hand, TMHMM aims to predict membrane protein topology across many different genomes, rather than in a yeast-specific fashion. Despite these difficulties, the results in Table 4 are interesting because they suggest that an approach that exploits multiple genome-wide datasets may provide better membrane protein recognition performance than a sequence-specific approach.

Table 4. Comparison of membrane protein recognition methods

Method	ROC	TP1FP (%)
f_1	0.7345	16.70
f_2	0.7504	13.48
f_3	0.7879	21.93
T_{PH}	0.7362	30.02
T_{ENR}	0.8018	31.38
SDP/SVM	0.9219	36.06

Each row in the table corresponds to one of the membrane protein recognition methods described in the text: three methods that apply filters directly to the hydrophobicity profile, two methods based upon the TMHMM model, and the SDP/SVM approach. For each method, the ROC and TP1FP are reported.

DISCUSSION

We have described a general method for combining heterogeneous genome-wide datasets in the setting of kernel-based statistical learning algorithms, and we have demonstrated an application of this method to the problems of classifying yeast ribosomal and membrane proteins. The performance of the resulting SDP/SVM algorithm improves upon the SVM trained on any single dataset or trained using a naive combination of kernels. Moreover, the SDP/SVM algorithm's performance consistently improves as additional genome-wide datasets are added to the kernel representation and is robust in the presence of noise.

Vert and Kanehisa (2003) have presented a kernel-based approach to data fusion that is complementary to that presented here. In their approach, canonical correlation analysis (CCA) is used to select features from the space defined by a second kernel, and can be generalized to operate with more than two kernels. Thus, whereas the SDP approach combines different sources into a joint representation, kernel CCA separates components of a single kernel matrix, identifying the most relevant ones.

Semidefinite programming is viewed as a tractable instance of general convex programming, because it is known to be solvable in polynomial time, whereas general convex programs need not be (Nesterov and Nemirovsky, 1994). In practice, however, there are important computational issues that must be faced in any implementation. In particular, our application requires the formation and manipulation of $n \times n$ kernel matrices. For genome-scale data, such matrices are large, and naive implementation can create serious demands on memory resources. However, kernel matrices often have special properties that can be exploited by more sophisticated implementations. In particular, it is possible to prove that certain kernels necessarily lead to low-rank kernel matrices, and indeed low-rank matrices are also often encountered in practice (Williams and Seeger, 2000). Methods such as incomplete Cholesky decomposition can be used to find low-rank approximations of such matrices, without even forming the full kernel matrix, and these methods have been used successfully in implementations of other kernel methods (Bach and Jordan, 2002; Fine and Scheinberg, 2001). Time complexity is another concern. The worst-case complexity of the SDP in Equation (4) is $O(n^{4.5})$ (Lanckriet *et al.*, 2004), although it can be solved in $O(n^3)$, as a QCQP, under reasonable assumptions. In practice, however, this complexity bound is not necessarily reached by any given class of problem, and indeed time complexity has been less of a concern than space complexity in our work far. Moreover, the low-rank approximation tools may also provide some help with regards to time complexity. Nonetheless, running time issues are a concern for deployment of our approach with higher eukaryotic genomes, and new implementational strategies may be needed.

Kernel-based statistical learning methods have a number of general virtues as tools for biological data analysis. First, the kernel framework accommodates not only the vectorial and matrix data that are familiar in classical statistical analysis, but also more exotic data types such as strings, trees, graphs and text. The ability to handle such data is clearly essential in the biological domain. Second, kernels provide significant opportunities for the incorporation of more specific biological knowledge, as we have seen with the FFT kernel and the Pfam kernel. Third, the growing suite of kernel-based data analysis algorithms require only that data be reduced to a kernel matrix; this creates opportunities for standardization. Finally, as we have shown here, the reduction of heterogeneous data types to the common format of kernel matrices allows the development of general tools for combining multiple data types. Kernel matrices are required only to respect the constraint of positive semidefiniteness, and thus the powerful technique of semidefinite programming can be exploited to derive general procedures for combining data of heterogeneous format and origin.

We thus envision the development of general libraries of kernel matrices for biological data, such as those that we have provided at noble.gs.washington.edu/proj/sdp-svm, that summarize the statistically-relevant features of primary data, encapsulate biological knowledge, and serve as inputs to a wide variety of subsequent data analyses. Indeed, given the appropriate kernel matrices, the methods that we have described here are applicable to problems such as the prediction of protein metabolic, regulatory and other functional classes, the prediction of protein subcellular locations, and the prediction of protein-protein interactions.

Finally, while we have focused on the binary classification problem in the current paper, there are many possible extensions of our work to other statistical learning problems. One notable example is the problem of transduction, in which the classifier is told a priori the identity of the points that are in the test set (but not their labels). This approach can deliver superior predictive performance (Vapnik, 1998), and would seem particularly appropriate in gene or protein classification problems, where the entities to be classified are often known a priori.

ACKNOWLEDGEMENTS

W.S.N. is supported by a Sloan Foundation Research Fellowship and by National Science Foundation grants BDI-0078523 and ISI-0093302. M.I.J. and G.L. acknowledge support from ONR MURI N00014-00-1-0637 and NSF grant IIS-9988642. T.D.B. is a Research Assistant with the Fund for Scientific Research, Flanders (F.W.O.–Vlaanderen).

REFERENCES

Alberts,B., Bray,D., Johnson,A., Lewis,J., Raff,M., Roberts,K. and Walter,P. (1998) *Essential Cell Biology: An Introduction to the Molecular Biology of the Cell*. Garland Science Publishing, London, UK.

Altschul,S.F., Gish,W., Miller,W., Myers,E.W. and Lipman,D.J. (1990) Basic local alignment search tool. *J. Mol. Biol.* **215**, 403–410.

Bach,F. and Jordan,M.I. (2002) Kernel independent component analysis. *J. Mach. Learning Res.*, **3**, 1–48.

Berg,C., Christensen,J. and Ressel,P. (1984) *Harmonic Analysis on Semigroups: Theory of Positive Definite and Related Functions*. Springer, New York.

Black,S.D. and Mould,D.R. (1991) Development of hydrophobicity parameters to analyze proteins which bear post- or cotranslational modifications. *Anal. Biochem.*, **193**, 72–82.

Boser,B.E., Guyon,I. and Vapnik,V. (1992) A training algorithm for optimal margin classifiers. *Computational Learing Theory*, ACM Press, NY, pp. 144–152.

Brown,M.P.S., Grundy,W.N., Lin,D., Cristianini,N., Sugnet,C., Furey,T.S., Ares,J.M. and Haussler,D. (2000) Knowledge-based analysis of microarray gene expression data using support vector machines. *Proc. Natl Acad. Sci. USA*, **97**, 262–267.

Chen,C. and Rost,B. (2002) State-of-the-art in membrane protein prediction. *Appl. Bioinform.*, **1**, 21–35.

Cristianini,N. and Shawe-Taylor,J. (2000) *An Introduction to Support Vector Machines*. Cambridge University Press, Cambridge, UK.

Engleman,D.M., Steitz,T.A. and Goldman,A. (1986) Identifying nonpolar transbilayer helices in amino acid sequences of membrane proteins. *Ann. Rev. Biophys. Biophys. Chem.*, **15**, 321–353.

Fine,S. and Scheinberg,K. (2001) Efficient SVM training using low-rank kernel representations. *J. Mach. Learning Res.*, **2**, 243–264.

Gribskov,M. and Robinson,N.L. (1996) Use of receiver operating characteristic (ROC) analysis to evaluate sequence matching. *Comput. Chem.*, **20**, 25–33.

Gross,G., Gaestel,M., Bohm,H. and Bielka,H. (1989) cDNA sequence coding for a translationally controlled human tumor protein. *Nucleic Acids Res.*, **17**, 8367.

Hanley,J.A. and McNeil,B.J. (1982) The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, **143**, 29–36.

Harms,J., Schluenzen,F., Zarivach,R., Bashan,A., Gat,S., Agmon,I., Bartels,H., Franceschi,F. and Yonath,A. (2001) High resolution structure of the large ribosomal subunit from a meshophilic eubacterium. *Cell*, **107**, 679–688.

Hopp,T.P. and Woods,K.R. (1981) Prediction of protein antigenic determinants from amino acid sequences. *Proc. Natl Acad. Sci., USA*, **78**, 3824–3828.

Kondor,R.I. and Lafferty,J. (2002) Diffusion kernels on graphs and other discrete input spaces. In Sammut,C. and Hoffmann,A. (eds), *Proceedings of the International Conference on Machine Learning*. Morgan Kaufmann, San Francisco, CA, pp. 315–322.

Krogh,A., Larsson,B., von Heijne,G. and Sonnhammer,E.L.L. (2001) Predicting transmembrane protein topology with a hidden Markov model: Application to complete genomes. *J. Mol. Biol.*, **305**, 567–580.

Kyte,J. and Doolittle,R.F. (1982) A simple method for displaying the hydropathic character of a protein. *J. Mol. Biol.*, **157**, 105–132.

Lanckriet,G.R.G., Cristianini,N., Bartlett,P., El Ghaoui,L. and Jordan,M.I. (2004) Learning the kernel matrix with semidefinite programming. *J. Mach. Learning Res.*, **5**, 27–72.

Liao,L. and Noble,W.S. (2002) Combining pairwise sequence similarity and support vector machines for remote protein homology detection. *Proceedings of the Sixth Annual International Conference on Computational Molecular Biology*, pp. 225–232.

Mewes,H.W., Frishman,D., Gruber,C., Geier,B., Haase,D., Kaps,A., Lemcke,K., Mannhaupt,G., Pfeiffer,F., Schüller,C., Stocker,S. and Weil,B. (2000) MIPS: a database for genomes and protein sequences. *Nucleic Acids Res.*, **28**, 37–40.

Nesterov,Y. and Nemirovsky,A. (1994) *Interior Point Polynomial Methods in Convex Programming: Theory and Applications*. SIAM, Philadelphia, PA.

Noble,W.S. (2004) Support vector machine applications in computational biology. In Schoekkopf,B., Tsuda,K. and Vert,J.-P. (eds), *Kernel Methods in Computational Biology*, MIT Press, Cambridge, MA, pp. 71–92.

Schluenzen, F., Tocilj,A., Zarivach,R., Harms,J., Gluehmann,M., Janell,D. Bashan,A., Bartels,H., Agmon,I., Franceschi,F. and Yonath,A. (2000) Structure of functionally activated small ribosomal subunit at 3.3 Å resolution. *Cell*, **102**, 615–623.

Schölkopf,B. and Smola,A. (2002) *Learning with Kernels*. MIT Press, Cambridge, MA.

Smith,T.F. and Waterman,M.S. (1981) Identification of common molecular subsequences. *J. Mol. Biol.*, **147**, 195–197.

Sonnhammer,E., Eddy,S. and Durbin,R. (1997) Pfam: a comprehensive database of protein domain families based on seed alignments. *Proteins*, **28**, 405–420.

Tsuda,K. (1999) Support vector classification with asymmetric kernel function. In Verleysen,M. (ed.), *Proceedings ESANN*, pp. 183–188.

Vandenberghe,L. and Boyd,S. (1996) Semidefinite programming. *SIAM Rev.*, **38**, 49–95.

Vapnik,V.N. (1998) *Statistical Learning Theory*. Wiley-Interscience.

Vapnik,V.N. (1999) *The Nature of Statistical Learning Theory*, 2nd edn. Springer-Verlag, New York.

Vert,J.-P. and Kanehisa,M. (2003) Graph-driven features extraction from microarray data using diffusion kernels and kernel CCA. In Becker,S., Thrun,S. and Obermayer,K. (eds), *Advances in Neural Information Processing Systems 15*, MIT Press, Cambridge, MA, pp. 1425–1432.

von Mering,C., Krause,R., Snel,B., Cornell,M., Olivier,S.G., Fields,S. and Bork,P. (2002) Comparative assessment of large-scale datasets of protein–protein interactions. *Nature*, **417**, 399–403.

Wahba,G. (1990) *Spline Models for Observational Data*. SIAM, Philadelphia.

Williams,C.K.I. and Seeger,M. (2000) Effect of the input density distribution on kernel-based classifiers. In Langley,P. (ed.), *Proceedings of Seventeenth International Conference on Machine Learning (ICML 2000)*. Morgan Kaufmann, San Francisco, CA.

PNAS
Proceedings of the National Academy of Sciences of the United States of America www.pnas.org

Protein ranking: From local to global structure in the protein similarity network

Jason Weston, Andre Elisseeff, Dengyong Zhou, Christina S. Leslie, and William Stafford Noble

PNAS published online Apr 15, 2004;
doi:10.1073/pnas.0308067101

This information is current as of May 2007.

Notes:

Protein ranking: From local to global structure in the protein similarity network

Jason Weston[†‡], Andre Elisseeff[‡], Dengyong Zhou[‡], Christina S. Leslie[§], and William Stafford Noble[¶‖]

[†]NEC Laboratories America, 4 Independence Way, Princeton, NJ 08540; [‡]Max Planck Institute for Biological Cybernetics, Spemannstrasse 38, 72076 Tübingen, Germany; [§]Department of Computer Science, Columbia University, 1214 Amsterdam Avenue, MC 0401, New York, NY 10027; and [¶]Department of Genome Sciences, University of Washington, Health Sciences Center, P.O. Box 357730, Seattle, WA 98195

Edited by Michael S. Waterman, University of Southern California, Los Angeles, CA, and approved March 16, 2004 (received for review December 4, 2003)

Biologists regularly search databases of DNA or protein sequences for evolutionary or functional relationships to a given query sequence. We describe a ranking algorithm that exploits the entire network structure of similarity relationships among proteins in a sequence database by performing a diffusion operation on a precomputed, weighted network. The resulting ranking algorithm, evaluated by using a human-curated database of protein structures, is efficient and provides significantly better rankings than a local network search algorithm such as PSI-BLAST.

Pairwise sequence comparison is the most widely used application of bioinformatics. Subtle sequence similarities frequently imply structural, functional, and evolutionary relationships among protein and DNA sequences. Consequently, essentially every molecular biologist working today has searched an online database of biosequences. This search process is analogous to searching the World Wide Web with a search engine such as Google: the user enters a query (a biological sequence or a word or phrase) into a web form. The search engine then compares the query with each entry in a database, and returns to the user a ranked list, with the most relevant or most similar database entry at the top of the list.

The World Wide Web consists of a network of documents connected to one another by means of hypertext links. A database of protein sequences can also be usefully represented as a network, in which edges may represent functional, structural, or sequence similarity. Two protein sequences are considered similar if they contain subsequences that share more similar amino acids than would be expected to occur by chance. We refer to the network of sequence similarities as a protein similarity network.

Early algorithms for detecting sequence similarities did not exploit the structure of the protein similarity network at all, but focused instead on accurately defining the individual edges of the network (1–3). Subsequent work used statistical models based on multiple alignments to model the local structure of the network (4, 5) and to perform local search through the protein similarity network by using short paths (6), average- or single-linkage scoring of inbound edges (7, 8), and iterative model-based search (9, 10). The popular PSI-BLAST (11) algorithm falls into the latter category: PSI-BLAST builds an alignment-based statistical model of a local region of the protein similarity network and then iteratively collects additional sequences from the database to be added to the alignment.

The critical innovation that led to the success of the Google search engine is its ability to exploit global structure by inferring it from the local hyperlink structure of the Web. Google's PAGERANK algorithm (12) models the behavior of a random web surfer, who clicks on successive links at random and also periodically jumps to a random page. The web pages are ranked according to the probability distribution of the resulting random walk. Empirical results show that PAGERANK is superior to the naive, local ranking method, in which pages are simply ranked according to the number of inbound hyperlinks.

We demonstrate that a similar advantage can be gained by including information about global network structure in a protein sequence database search algorithm. In contrast to iterative protein database search methods such as PSI-BLAST, which compute the local structure of the protein similarity network on the fly, the RANKPROP algorithm begins from a precomputed protein similarity network, defined on the entire protein database. Querying the database consists of adding the query sequence to the protein similarity network and then propagating link information outward from the query sequence. After propagation, database proteins are ranked according to the amount of link information they received from the query. This algorithm ranks the data with respect to the intrinsic cluster structure (13, 14) of the network. We evaluate the RANKPROP output by using a 3D-structure-based gold standard, measuring the extent to which known homologs occur above nonhomologs in the ranked list. Our experiments suggest that RANKPROP's ranking is superior to the ranking induced by the direct links in the original network.

The protein similarity network represents the degree of similarity between proteins by assigning weights to each edge. The degree of similarity between two sequences is commonly summarized in an E value, which is the expected number of times that this degree of sequence similarity would occur in a random database of the given size. By using a weighting scheme that is a function of the E value, an edge connecting two similar sequences is given a large weight, and *vice versa*.

To accommodate edge weights, the RANKPROP algorithm adopts recently described diffusion techniques (15) from the field of machine learning, which are closely related to the spreading activation networks of experimental psychology (16, 17). RANKPROP takes as input a weighted network on the data, with one node of the network designated as the query. In the protein ranking problem, the edges of the network are defined by using PSI-BLAST. The query is assigned a score, and this score is continually pumped to the remaining points by means of the weighted network. During the diffusion process, a protein P pumps to its neighbors at time t the linear combination of scores that P received from its neighbors at time $t - 1$, weighted by the strengths of the edges between them. The diffusion process continues until convergence, and the points are ranked according to the scores they receive. The RANKPROP algorithm is described formally in Fig. 1. This algorithm provably converges, and an exact closed form solution can be found (see *Supporting Information*, which is published on the PNAS web site).

Methods

We tested the quality of the protein rankings produced by RANKPROP, using the human-annotated SCOP database of protein 3D structural domains as a gold standard (18). SCOP has

This paper was submitted directly (Track II) to the PNAS office.

Abbreviation: ROC, receiver operating characteristic.

[‖]To whom correspondence should be addressed. E-mail: noble@gs.washington.edu.

```
Initialization: $y_1(0) = 1$; $y_i(0) = 0$
for $t = 0, 1, 2, \ldots$ do
    for $i = 2$ to $m$ do
        $y_i(t+1) \leftarrow K_{1i} + \alpha \Sigma_{j=2}^{m} K_{ji} y_j(t)$
    end for
until convergence
Termination: Let $y_i^*$ denote the limit of the sequence $\{y_i(t)\}$. Then $y_i^*$ is the ranking score of the $i^{th}$ point (largest ranked first).
```

Fig. 1. The RANKPROP algorithm. Given a set of objects (in this case, proteins) $X = x_{1}, \ldots, x_{m}$, let x_1 be the query and $x_{2}, \ldots, x_m$ be the database (targets) we would like to rank. Let K be the matrix of object–object similarities, i.e., K_{ij} gives a similarity score between x_i and x_j, with K normalized so that $\Sigma_{j=2}^{m} K_{ji} = 1$ for all i. For computational efficiency, we set $K_{1i} = K_{i1}$ for all i, so that we can compute weights involving the query using a single execution of PSI-BLAST. Let $y_{i,}$ $i = 2, \ldots, m$, be the initial ranking ''score'' of a target. In practice, for efficiency, the algorithm is terminated after a fixed number I of iterations, and $y_i(I)$ is used as an approximation of y_i^*. The parameter $\alpha \in [0,1]$ is set *a priori* by the user. For $\alpha = 0$, no global structure is found, and the algorithm's output is just the ranking according to the original distance metric. These experiments use $\alpha = 0.95$, looking for clear cluster structure in the data.

been used as a gold standard in many previous studies (19–21). Sequences were extracted from version 1.59 of the database, purged by using the web site http://astral.berkeley.edu so that no pair of sequences share more than 95% identity. For the purposes of selecting the RANKPROP parameter σ, the resulting collection of 7,329 SCOP domains was split into two portions: 379 superfamilies (4,071 proteins) for training and 332 (2,899 proteins) for testing. Note that training and testing sequences never come from the same superfamily. The SCOP database is organized hierarchically into classes, folds, superfamilies, and families. For the purposes of this experiment, two domains that come from the same superfamily are assumed to be homologous, and two domains from different folds are assumed to be unrelated. For pairs of proteins in the same fold but different superfamilies, their relationship is uncertain, and so these pairs are not used in evaluating the algorithm.

Three protein similarity networks were computed by using the BLAST and PSI-BLAST (version 2.2.2) algorithms. Two networks were defined by applying BLAST and PSI-BLAST to a database comprised only of the 7,329 SCOP domains. An additional network was created by applying PSI-BLAST to a larger database that also included all 101,602 proteins from SWISS-PROT (version 40). In each case, the programs were run by using the default parameters, including the BLOSUM 62 matrix, but with an E value threshold for reporting results of 10,000. PSI-BLAST was allowed to run a maximum of six iterations, which previous work indicates is sufficient for good performance (21), using the default E value threshold for inclusion in the model of 0.005. Each of these networks induces a ranking with respect to each query sequence.

Finally, we applied RANKPROP to the larger PSI-BLAST protein similarity network. In the network K used by RANKPROP, the weight K_{ij} associated with a directed edge from protein i to protein j is $exp(-S_j(i)/\sigma)$, where $S_j(i)$ is the E value assigned to protein i given query j. The value of $\sigma = 100$ is chosen by using the training set (see supporting information). For efficiency, the number of outgoing edges from each node is capped at 1000, unless the number of target sequences with E values <0.05 exceeds 1000. For each query, RANKPROP runs for 20 iterations, which brings the algorithm close to convergence (see supporting information).

We measure the performance of a protein database search algorithm by using a modified version of the receiver operating characteristic (ROC) score (22). The ROC score is the area under a curve that plots false-positive rate versus true-positive rate for various classification thresholds. The ROC score thus measures, for a single query, the quality of the entire ranking produced by the algorithm. In practice, only the top of this ranking is important. Therefore, we compute the ROC_{50} score (23), which is the area under the ROC curve up to the first 50 false-positives. A value of 1 implies that the algorithm successfully assigns all of the true relationships higher scores than the false relationships. For a random ranking of these data, the expected ROC_{50} score is close to 0 because most of the sequences are not related to the query.

Results

The experimental results, summarized in Fig. 2, show the relative improvements offered by the various algorithms. Even when using the small SCOP database, the PSI-BLAST protein similarity network improves significantly upon the network created using the simpler BLAST algorithm: PSI-BLAST yields better performance than BLAST for 51.3% of the test queries, and worse performance for only 8.2% of the queries. PSI-BLAST benefits from the availability of a larger sequence database: increasing the database size by adding the SWISS-PROT database yields an additional improvement of the same magnitude (50.9% and 11.4%, respectively). Finally, running RANKPROP on the larger protein similarity network defined by PSI-BLAST yields improved rankings for 55.3% of the queries, and decreases performance on only 9.7%. All of these differences are statistically significant at $P = 0.01$ according to a Wilcoxon signed-rank test. A comparison of PSI-BLAST and RANKPROP ROC scores by query is shown in Fig. 3, and a diagram illustrating how RANKPROP successfully re-ranks homologs of a single query is shown in Fig. 4.

Note that there is some obvious structure in Figs. 2 and 3. The steep slope in the RANKPROP plot (Fig. 2) at around 0.9 ROC_{50} corresponds to queries mostly from the largest superfamily in the database, the immunoglobulins with 623 proteins. These queries are also visible as a cluster at around (0.9, 0.7) in Fig. 3. RANKPROP's improved rankings for these queries suggests that the algorithm successfully exploits cluster structure in the protein similarity network.

RANKPROP is not misled by the presence of multidomain proteins in the database. Previous network-based protein similarity detection algorithms explicitly deal with multidomain proteins. For example, the INTERMEDIATE SEQUENCE SEARCH algorithm (6) includes a step that extracts the region of the target sequence that matched the query and then recalculates the statistical significance of that region with respect to the target sequences. This step prevents the algorithm from inferring a false relationship between protein domains A and B through an intermediate protein containing both A and B. RANKPROP delivers excellent performance, even when the database contains ≈100,000 full-length proteins, many of which contain more than one domain. Furthermore, Fig. 3 shows that RANKPROP generally performs better than PSI-BLAST, even when the SCOP query domain lies on the same protein as another domain in the test set. A closer investigation (see supporting information) reveals that RANKPROP does indeed rank these transitive domains higher than would be expected by chance. However, in general, as long as the query sequence is connected to many other proteins, then the true relationships will be mutually reinforcing during network propagation.

A well known problem with PSI-BLAST is the occasional case in which it mistakenly pulls in a false-positive match during an early iteration. This false-positive may then pull in more false-positives

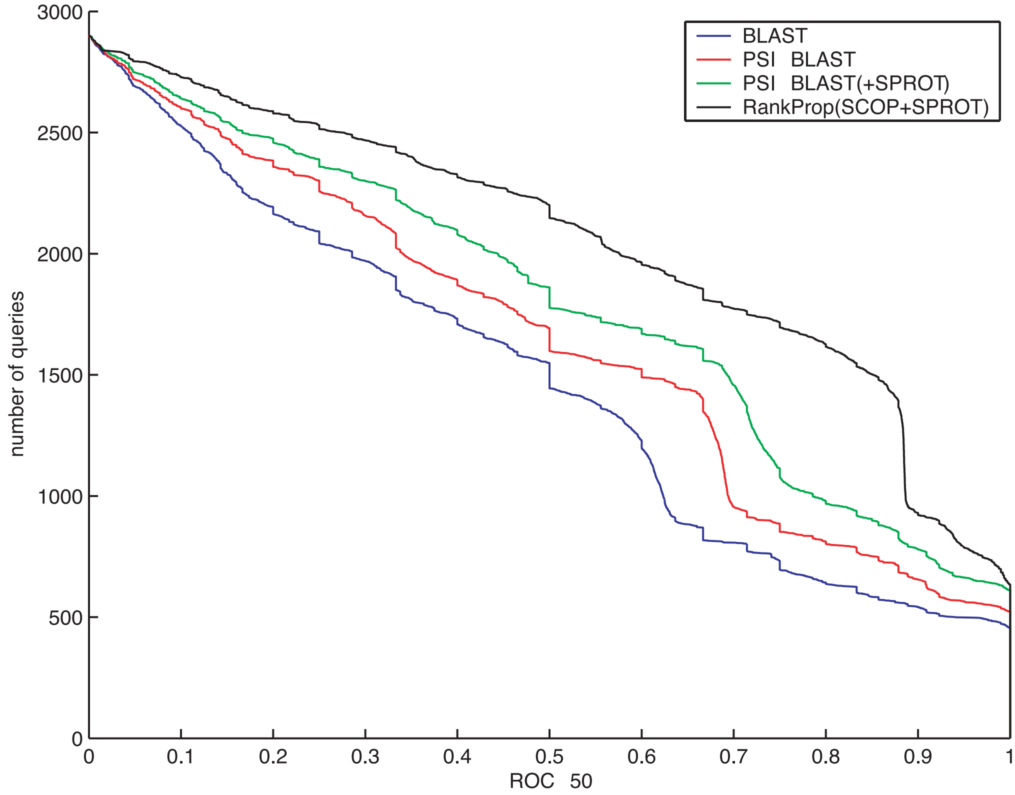


Fig. 2. Relative performance of protein ranking algorithms. The graph plots the total number of test set SCOP queries for which a given method exceeds an ROC_{50} score threshold. ROC_{50} is the area under a curve that plots true-positive rate as a function of false-positive rate, up to the 50th false-positive. In the plot, the lower three series correspond to the three protein similarity networks described in the text; the upper series is created by running RANKPROP on the larger PSI-BLAST network. For these data, the mean ROC_{50} for the four methods are 0.506 (BLAST), 0.566 [PSI-BLAST (SCOP)], 0.618 [PSI-BLAST (SCOP plus SPROT)], and 0.707 (RANKPROP).

in subsequent iterations, leading to corrupted results. Among the test set queries, there are 139 queries for which the PSI-BLAST ROC_{50} score is worse than the corresponding BLAST score, indicating that iteration hurt the performance of the algorithm. For these queries, RANKPROP outperforms BLAST in 106 cases, despite using as input a protein similarity network defined by PSI-BLAST. Furthermore, the degree of improvement produced by RANKPROP relative to BLAST is often large, with a difference in ROC_{50} >0.1 for 71 of the 106 queries (see supporting information).

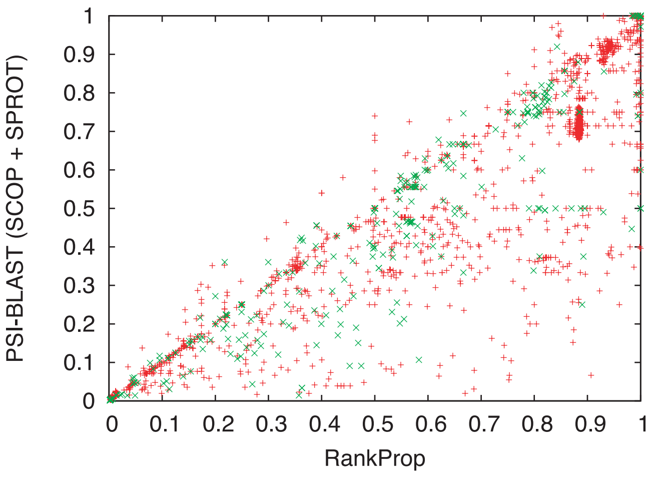


Fig. 3. Scatter plot of ROC_{50} scores for PSI-BLAST versus RANKPROP. The plot contains 2,899 points, corresponding to all queries in the test set. Green points correspond to query domains that lie on the same protein with another domain in the test set. All other queries are red.

Among the 282 queries for which PSI-BLAST produces a better ranking than RANKPROP, most of the differences in ROC are small. There are, however, 20 queries for which PSI-BLAST produces an ROC_{50} that is >0.1 greater than RANKPROP's ROC_{50}, and one query for which the difference is >0.2 (see supporting information). Some of these queries belong to SCOP class 3 (α-β proteins), which contains a number of homologous Rossmann folds. In these cases, the first false-positives may in fact be true-positives. For the other queries, RANKPROP's difficulty likely arises from overpropagation through the protein similarity network. Lowering the parameter α could potentially fix this problem, because as $\alpha \rightarrow 0$, we obtain the same ranking as PSI-BLAST.

Finally, the results indicate that RANKPROP does not spoil good initial rankings. Indeed, there is only one query for which PSI-BLAST produces an ROC_{50} score of 1 (a perfect ranking) and RANKPROP produces a score worse than 0.98. This query is the C-terminal fragment of DNA topoisomerase II, with an ROC_{50} of 0.93. Conversely, there are 30 queries for which PSI-BLAST has an ROC_{50} <0.93 and RANKPROP produces a perfect ranking.

To better understand the source of RANKPROP's improvement relative to the underlying PSI-BLAST protein similarity network, we performed an additional round of experiments using two variants of the RANKPROP algorithm. Each algorithmic variant restricts RANKPROP to a subset of the protein similarity network. In the first variant, RANKPROP sees only the local network structure: the target sequences that are linked directly to the query, plus the pairwise relationships among those sequences.

GENETICS

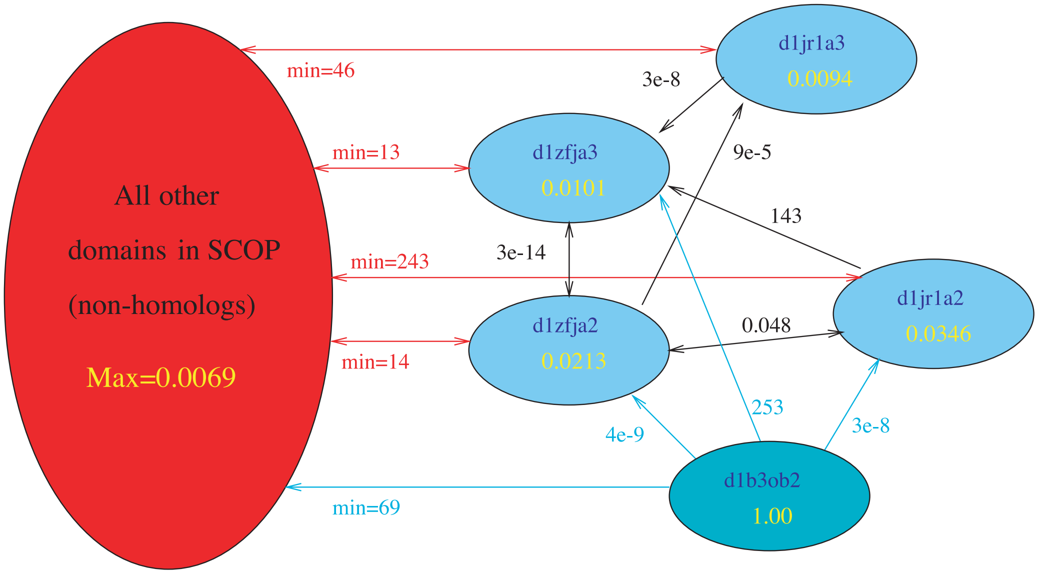


Fig. 4. Visualization of part of the similarity network. Shown is a small part of the protein similarity network, where d1b30b2 is the query, and the domains are represented by light blue nodes are its homologs. The large red node represents all other domains. The cyan-colored edges from the query to other nodes are labeled with weights equal to the PSI-BLAST *E* value, given d1b30b2 as the query. The rest of the edges indicate the similarity network which is formed of PSI-BLAST *E* values, as described in the text. Black edges are between homologs, and red edges are between all nonhomologs and a single homolog, with the minimum *E* value across all nonhomologs given as the weight of the edge. No edge is drawn if PSI-BLAST did not assign an *E* value. PSI-BLAST only correctly identifies two homologs, d1zfja2 and d1jr1a2. Although d1zfja3 is assigned an *E* value (of 253), this assignment is larger than three of the nonhomologs in SCOP. The yellow scores inside the nodes are the RANKPROP activation levels (y_i values). In this case, RANKPROP places all of the homologs at the top of the ranked list. This assignment occurs because there are very low *E* value paths (by traversing edges with not more than an *E* value of 9e-5) between the query and all homologs, whereas even the ''nearest'' nonhomologs are sufficiently far away (never closer than an *E* value of 13 to a homolog). Note that d1jr1a2 is assigned a higher score by RANKPROP than d1zfja2, even though the *E* values assigned by PSI-BLAST (although similar) indicate the opposite. This result is because d1zfja2 has much lower weighted edges to nonhomologs, and thus receives more of the nonhomologs' activation level (which are close to 0). Overall, the RANKPROP ranking gave an ROC$_{50}$ score of 1, whereas PSI-BLAST gave an ROC$_{50}$ score of 0.78 on this query.

This network of local relationships yields RANKPROP performance almost identical to PSI-BLAST (see supporting information). The second variant includes nonlocal edges but eliminates all weak edges, with *E* values $>$0.005. In contrast with the previous variant, this version of the algorithm performs only slightly worse than RANKPROP trained using the entire network. This result indicates that the improvement of RANKPROP over PSI-BLAST results primarily from RANKPROP's ability to learn from nonlocal network structure, and that the weak links in the network are of secondary importance. Data sets and FASTA files are available from the web site of J.W., which can be accessed at www.kyb.tuebingen.mpg.de/bs/people/weston/rankprot/supplement.html.

Discussion

RANKPROP is efficient enough to employ the algorithm as part of a web-based search engine. The precomputation of the PSI-BLAST protein similarity network is clearly computationally expensive; however, this operation can be performed in advance offline. Computing the ranking with respect to a given query requires first running PSI-BLAST with the query sequence (unless it is already in the network), and then propagating scores from the query through the network. In the experiments reported here, the propagation (20 iterations of RANKPROP) took on average 73 seconds to compute using a Linux machine with an Advanced Micro Devices (Sunnyvale, CA) MP 2200+ processor. BLAST and PSI-BLAST take ≈21 and 331 sec per query respectively on the same database (SCOP plus SPROT). The propagation time scales linearly in the number of edges in the network. The propagation time could be improved by removing weak edges from the protein similarity network [at a relatively small cost in accuracy (see supporting information)], by running the propagation in parallel, and by reducing the number of iterations. Finally, the initial query PSI-BLAST computation may be replaced with BLAST at a relatively small cost in accuracy (see supporting information), resulting in a query procedure that is faster than running a single PSI-BLAST query on the entire database.

The experiments described here were performed by using a single set of PSI-BLAST parameters. These parameters were previously selected by means of extensive empirical optimization using the SCOP database as a gold standard and ROC$_n$ scores as the performance metric (17). However, even if better PSI-BLAST parameters were available, the resulting improved *E* values would likely lead to a similar improvement in the performance of the RANKPROP algorithm.

The results reported here are given in terms of the ROC$_{50}$ performance measure. One might argue that a stricter (or looser) threshold might be more appropriate, depending on the cost associated with false-positives. Further experiments (see supporting information) show that RANKPROP continues to significantly outperform PSI-BLAST even for relatively small values of the ROC threshold (ROC$_5$ or ROC$_{10}$). At the most strict threshold, ROC$_1$ (which is equivalent to the percentage of positive examples appearing before the first negative example in the ranked output), the difference between the two algorithms is no longer statistically significant. However, by using the ROC$_1$ measure, RANKPROP performs better on smaller superfamilies using a small σ, and *vice versa*. Therefore, a simple modification to the algorithm, in which the value of σ depends on the number of strong matches to the query sequence, once again yields strong performance relative to PSI-BLAST. In future work, we plan to investigate more thoroughly algorithms that choose σ dynamically based on the local density of the protein similarity network.

A valuable component of the PSI-BLAST algorithm is its method for estimating statistical confidence, in the form of *E* values. Currently, RANKPROP does not produce *E* values; however, approximate E values may be derivable by means of interpolation and smoothing of the PSI-BLAST E values with respect to the RANKPROP ranking. Alternatively, it may be possible to fit a probability distribution to the output scores (24). This fitting will be the subject of future research.

The primary outcome of this work is not the RANKPROP algorithm *per se*, but the observation that exploiting the entire structure of the protein similarity network can lead to significantly improved recognition of pairwise protein sequence similarities. RANKPROP provides an efficient, powerful means of learning from the protein similarity network; however, other network-based algorithms may also yield similar improvements relative to the ranking induced by the underlying protein similarity network. Furthermore, this observation is applicable to a wide range of problem domains, including image and text ranking, as well as protein or gene ranking using different (or multiple) types of biological data.

This work is supported by National Science Foundation Awards EIA-0312706 and DBI-0078523, National Institutes of Health Grant LM07276–02, and an Award in Informatics from the Pharmaceutical Research and Manufacturers of America Foundation (to C.S.L.). W.S.N. is an Alfred P. Sloan Research Fellow.

1. Smith, T. & Waterman, M. (1981) *J. Mol. Biol.* **147,** 195–197.
2. Pearson, W. R. (1985) *Methods Enzymol.* **183,** 63–98.
3. Altschul, S. F., Gish, W., Miller, W., Myers, E. W. & Lipman, D. J. (1990) *J. Mol. Biol.* **215,** 403–410.
4. Gribskov, M., Lüthy, R. & Eisenberg, D. (1990) *Methods Enzymol.* **183,** 146–159.
5. Krogh, A., Brown, M., Mian, I., Sjolander, K. & Haussler, D. (1994) *J. Mol. Biol.* **235,** 1501–1531.
6. Park, J., Teichmann, S. A., Hubbard, T. & Chothia, C. (1997) *J. Mol. Biol.* **273,** 1–6.
7. Grundy, W. N. (1998) *Proceedings of the Second International Conference on Computational Molecular Biology*, eds. Istrail, S., Pevzner, P. & Waterman, M. (ACM, New York), pp. 94–100.
8. Yona, G., Linial, N. & Linial, M. (1999) *Proteins Struct. Funct. Genet.* **37,** 360–678.
9. Tatusov, R. L., Altschul, S. F. & Koonin, E. V. (1994) *Proc. Natl. Acad. Sci. USA* **91,** 12091–12095.
10. Hughey, R. & Krogh, A. (1996) *Comput. Appl. Biosci.* **12,** 95–107.
11. Altschul, S. F., Madden, T. L., Schaffer, A. A., Zhang, J., Zhang, Z., Miller, W. & Lipman, D. J. (1997) *Nucleic Acids Res.* **25,** 3389–3402.
12. Brin, S. & Page, L. (1998) *Comput. Networks ISDN Syst.* **30,** 107–117.
13. Roweis, S. T. & Saul, L. K. (2000) *Science* **290,** 2323–2326.
14. Tenenbaum, J. B., de Silva, V. & Langford, J. C. (2000) *Science* **290,** 2319–2323.
15. Zhu, X., Ghahramani, Z. & Lafferty, J. (2003) in *Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003)*, eds. Fawcett, T. & Mishra, N. (AAAI Press, Menlo Park, CA), pp. 329–336.
16. Anderson, J. R. (1983) *The Architecture of Cognition* (Harvard Univ. Press, Cambridge, MA).
17. Shrager, J., Hogg, T. & Huberman, B. A. (1987) *Science* **236,** 1092–1094.
18. Murzin, A. G., Brenner, S. E., Hubbard, T. & Chothia, C. (1995) *J. Mol. Biol.* **247,** 536–540.
19. Park, J., Karplus, K., Barrett, C., Hughey, R., Haussler, D., Hubbard, T. & Chothia, C. (1998) *J. Mol. Biol.* **284,** 1201–1210.
20. Jaakkola, T., Diekhans, M. & Haussler, D. (1999) *Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology*, eds. Lengauer, T., Schneider, R., Bork, B., Brutlag, D., Glasgow, J., Mewes, H.-W. & Zimmer, R. (AAAI Press, Menlo Park, CA), pp. 149–158.
21. Schäffer, A. A., Aravind, L., Madden, T. L., Shavirin, S., Spouge, J. L., Wolf, Y. I., Koonin, E. V. & Altschul, S. F. (2001) *Nucleic Acids Res.* **29,** 2994–3005.
22. Hanley, J. A & McNeil, B. J. (1982) *Radiology (Easton, Pa.)* **143,** 29–36.
23. Gribskov, M & Robinson, N. L. (1996) *Comput. Chem.* **20,** 25–33.
24. Platt, J. C. (1999) in *Advances in Large Margin Classifiers*, eds. Smola, A., Bartlett, P., Schölkopf, B. & Schuurmans, D. (MIT Press, Cambridge, MA), pp. 61–74.

BIOINFORMATICS

Vol. 21 Suppl. 1 2005, pages i338–i343
doi:10.1093/bioinformatics/bti1047

Predicting the in vivo *signature of human gene regulatory sequences*

William Stafford Noble[1,*], *Scott Kuehn*[2], *Robert Thurman*[2], *Man Yu*[2] *and John Stamatoyannopoulos*[3]

[1]Department of Genome Sciences and Department of Computer Science and Engineering, [2]Division of Medical Genetics, University of Washington, Seattle, WA, USA and [3]Department of Molecular Biology, Regulome, 2211 Elliott Avenue, Suite 600, Seattle, WA 98121, USA

Received on January 15, 2005; accepted on March 27, 2005

ABSTRACT

Motivation: In the living cell nucleus, genomic DNA is packaged into chromatin. DNA sequences that regulate transcription and other chromosomal processes are associated with local disruptions, or 'openings', in chromatin structure caused by the cooperative action of regulatory proteins. Such perturbations are extremely specific for *cis*-regulatory elements and occur over short stretches of DNA (typically ∼250 bp). They can be detected experimentally as DNaseI hypersensitive sites (HSs) *in vivo*, though the process is extremely laborious and costly. The ability to discriminate DNaseI HSs computationally would have a major impact on the annotation and utilization of the human genome.

Results: We found that a supervised pattern recognition algorithm, trained using a set of 280 DNaseI HS and 737 non-HS control sequences from erythroid cells, was capable of *de novo* prediction of HSs across the human genome with surprisingly high accuracy determined by prospective *in vivo* validation. Systematic application of this computational approach will greatly facilitate the discovery and analysis of functional non-coding elements in the human and other complex genomes.

Availability: Supplementary data is available at noble.gs.washington.edu/proj/hs

Contact: noble@gs.washington.edu; jstam@regulome.com

1 INTRODUCTION

The vast majority of gene regulatory sequences in the human and other complex genomes remain undiscovered. In the living cell nucleus, DNA is packaged into chromatin fibers by non-specific association with the histone proteins that make up the nucleosome. Binding of activating proteins to regulatory DNA sequences requires cooperativity between the regulatory factors in order to displace a nucleosome, which in turn disrupts the local architecture of chromatin. This fundamental feature of eukaryotic *cis*-regulatory sequences was recognized nearly 25 years ago (Wu, 1980; Gross and Garrard, 1988), when it was discovered that such sequences were hypersensitive to cutting by the non-specific endonuclease DNaseI *in vivo*.

DNaseI hypersensitive sites (HSs) have since proven to be extremely reliable and generic markers of *cis*-regulatory sequences. Mapping of DNaseI HSs is a gold-standard approach for discovering functional non-coding elements involved in gene regulation and has underpinned the discovery of most experimentally established distal *cis*-acting elements in the human genome. In most cases, identification of functional elements marked by HSs significantly preceded the assignment of a specific functional role (enhancer, insulator, etc.) to those elements (Gross and Garrard, 1988; Li *et al*., 2002).

Comprehensive identification of DNaseI HSs in the human genome would be expected to disclose the location of all known classes of *cis*-regulatory sequences, including promoters, enhancers, silencers, insulators, boundary elements and locus control regions. Computational methods for the identification of the DNaseI HSs would therefore be expected to accelerate dramatically the functional annotation of the human genome.

Traditional approaches to computational prediction of *cis*-regulatory sequences in complex genomes have focused on identification and combinatorial analysis of short sequence motifs (presumed to represent regulatory factor binding sites) derived from examples of known sites, analysis of upstream regions of co-regulated genes (Sinha and Tompa, 2002; Berman *et al*., 2002), analysis of phylogenetic data or combinations thereof (Prakash *et al*., 2004). Unfortunately, the performance of even the most advanced algorithms is poor (Tompa *et al*., 2005), and the described methods generally lack biological validation, particularly in the context of the human genome. Even in the case of extensively characterized loci, such as the α- and β-globin domains, computational motif-based approaches have proven to be of little value for the discovery or annotation of HSs.

[*] To whom correspondence should be addressed.

The core sequences giving rise to HSs *in vivo* are anticipated to contain complex features that facilitate recognition by specific sets of regulatory factors interacting cooperatively over relatively short distances (150–250 bp) (Felsenfeld, 1996; Stamatoyannopoulos *et al*., 1995). However, it is not clear a priori whether recognition of such features is computationally tractable.

Conventional molecular approaches to the visualization of HSs have relied on an indirect method (Wu, 1980), and subsequent experimental localization of the core 150–250 bp activating sequences is extremely laborious (Lowrey *et al*., 1992; Talbot *et al*., 1990). As a result, relatively few HSs identified with traditional methods have been localized definitively to specific sequence elements, precluding systematic computational analyses. Recently, however, novel methods for large-scale sequence-specific discovery of DNaseI HSs have been described (Sabo *et al*., 2004; Dorschner *et al*., 2004), providing the basis for the recovery of larger numbers of DNaseI HSs sequences that can be utilized in computational models.

In this paper, we demonstrate that a sequence-based classification algorithm can learn to recognize DNaseI HSs with high accuracy. To train the algorithm, we take advantage of a collection of 280 validated erythroid HS sequences from throughout the human genome. We also use a set of 737 confirmed non-HS sequences of equivalent length. We employ a support vector machine (SVM) classifier, which learns by example to discriminate between two given classes of data (Vapnik, 1998; Cristianini and Shawe-Taylor, 2000). In a cross-validated test, the SVM achieves an accuracy of 85.24 ± 5.03% in predicting HSs. Furthermore, we perform a prospective *in vivo* experimental validation of the SVM predictions on previously untested regions of the human genome, using the assay described by Sabo *et al*. (2004) and Dorschner *et al*. (2004). Among HS predictions to which the SVM assigns probabilities >80%, 79.4% prove to be HSs on experimental validation in two hematopoietic cell types.

2 METHODS

2.1 Data

For training and cross-validation of the SVM, we use 280 validated erythroid HS sequences from throughout the human genome. These enabling sequences emerged from the recent description of a novel methodology for the identification of HSs via cloning based on their *in vivo* activity in K562 erythroid cells (Sabo *et al*., 2004). We also collected 737 sequences from around the genome (distributed proportionally among the autosomes and X chromosome but excepting the Y chromosome) that were non-hypersensitive when tested in the same cell type. Both K562 HS and non-HS sequences were similar in size (mean length 242.1 versus 242.8 bp, respectively). The complete dataset is available at noble.gs.washington.edu/proj/hs

We designed primers using Primer3 (Rozen and Skaletsky, 2000) with the following parameters: target amplimer size = 250 bp ± 50 bases; primer T_m (melting temperature) optimal = 60 ± 2°C; %GC = 50% optimal, range 40–80%; length = 24 bp optimal, range 19–27 bp; poly X maximum = 4.

We cultured erythroid cells (K562, ATCC) under standard conditions [37°C, 5% CO_2 in air, RPMI 1640 plus 10% FBS (Invitrogen, Carlsbad, CA, USA)]. We harvested the cultures at a density of 5×10^5 cells/ml. We performed DNaseI digestions following a standard protocol (Reitman *et al*., 1993). DNA was subsequently purified using the Puregene system (Gentra Systems, Minneapolis, MN, USA).

2.2 Support vector machine

We use the freely available Gist SVM implementation (Pavlidis *et al*., 2004). For each SVM optimization, we use the default parameters: a linear kernel function and a 2-norm soft margin with asymmetric penalties assigned to the positive and negative classes. Experiments with higher-order kernel functions and different soft margin settings yielded only very small changes in performance (data not shown).

The output of the SVM is a unit-free discriminant score; however, this score can be converted into a more useful probability by performing a sigmoid curve fit (Platt, 1999). This approach involves holding out a portion of the training set from the SVM optimization and fitting the sigmoid parameters using the discriminants from the held-out data. A probability score of 50% corresponds approximately to the hyperplane identified by the SVM, and increasing or decreasing probabilities are reflective (non-linearly) of increasing distance from the hyperplane (in positive or negative directions). The Gist software implements this curve fitting procedure.

2.3 Performance measure

We measure the overall quality of an SVM classifier using a receiver operating characteristic (ROC) curve (Hanley and McNeil, 1982). The trained SVM receives as input a list of candidate HS sequences and produces as output a ranked list of these sequences, with the confidently predicted HSs at the top of the list. Setting a threshold anywhere in this ranked list produces a particular rate of true and false positives with respect to that threshold. The ROC curve plots true positive rate as a function of false positive rate as the threshold varies from the top to the bottom of the ranked list. The ROC score is the area under this curve. A classifier that correctly places all of the HSs at the top of its ranked list would receive a ROC score of 1, whereas a random ranking would receive a score of ∼0.5.

3 RESULTS

The SVM algorithm learns to separate a set of labeled training data by placing the data in a high-dimensional space (a *feature* *space*) and discovering in that space a hyperplane that separates the two classes. Predicting the label of a new, unlabeled data point simply involves determining on which side of the hyperplane that point lies. SVMs boast powerful theoretical underpinnings (Vapnik, 1998) and wide applicability because of their use of kernel functions to represent data. The kernel function defines similarities between pairs of data points and allows the SVM to operate in an implicit vector space even for non-vector data, such as teer, graphs and strings. In computational biology, SVMs have been applied to a wide variety of problems (Noble, 2004), including the classification of several types of DNA sequence elements: translation start sites (Zien *et al*., 2000) and splice sites (Zhang *et al*., 2003).

Before the SVM classification of HS and non-HS sequences, we need to embed the sequences into a vector space. In this work, this embedding is accomplished by using the spectrum kernel (Leslie *et al*., 2002). We hypothesize that the difference between HS and non-HS sequences can be well characterized in terms of the presence of various short, motif-like sequence features. The spectrum kernel exhaustively enumerates all such features (‘k-mers’) of a given length (k) and represents each sequence as the frequency with which each k-mer appears in the sequence. For example, the sequence ‘ACGT’ contains three distinct 2mers (‘AC,’ ‘CG’ and ‘GT’). The $k = 2$ spectrum kernel representation of this sequence is a 16-element vector (one entry for each possible dinucleotide), with 0.33 for the three k-mers listed above and 0 for all other entries. In general, we do not expect the k-mers to be strand-specific, so reverse complements are collapsed into a single feature. Thus, for $k = 2$, there are only 10 distinct dinucleotides. In the experiments reported here, we concatenate the feature vectors for $k = 1, \ldots, 6$. Thus, the feature vector representation of a sequence contains $2 + 10 + \ldots = 2772$ entries.

3.1 Cross-validation

We first tested the pattern recognition performance of the SVM via 10-fold cross-validation on the collection of 1017 (280 + 737) sequences. This test involves randomly dividing the sequence set into 10 equal-sized subsets, and then repeatedly training on 90% subsets of the data and testing the SVM’s generalization performance on the held-out 10%. For this data set, the mean area under the ROC curve across 10-fold cross-validation was 0.842 ± 0.021, indicative of excellent performance (Fig. 1). At the classification threshold selected by the SVM, the mean accuracy was $85.24 \pm 5.03\%$.

We hypothesize that the DNaseI sequences that the SVM fails to identify during cross-validation represent a distinct, hard-to-identify subclass of DNaseI HSs. To test this hypothesis, we collected a set of 83 HSs that were incorrectly classified as non-HS during cross-validation. Removing the 83-member false-negative (FN) class from the training set and then retraining and cross-validating a new SVM (using the same 737 non-HS sequences) produced an ROC of 0.970 ± 0.0045. Conversely, a second SVM trained to discriminate between the 83 FN sequences and the remaining 934 sequences achieved an ROC of 0.635 ± 0.026. This result signifies a weaker classifier, though one which performs substantially better than chance ($p < 0.0000017$). Thus, learning accurately to recognize this smaller and potentially more diverse class of HSs may require a larger training set or a different collection of sequence features.

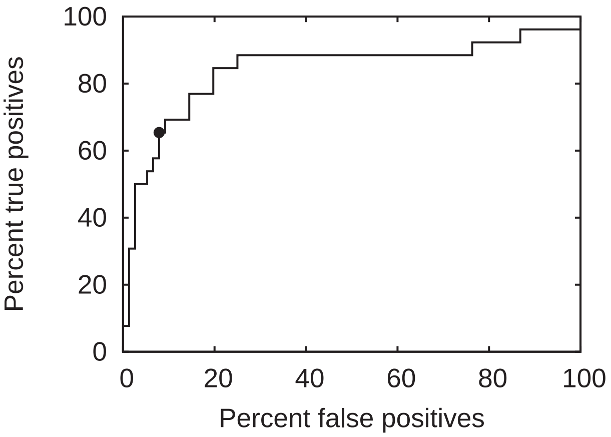


Fig. 1. Receiver operating characteristic curve for SVM discrimination of DNaseI HS versus non-HS sequences. The ROC was computed by training an SVM on a randomly selected 90% subset of a dataset comprising 280 HS and 737 non-HS sequences, followed by testing on the held-out 10%. The area under this particular curve is 0.84059, indicative of excellent performance. The dot marks the location of the decision boundary selected by the SVM. At this threshold, the SVM correctly identifies 17 HSs and 70 non-HSs, and makes 6 false positive and 9 false negative predictions.

3.2 Prospective experimental validation

Next we tested the ability of an SVM trained over a random 90% subset of the combined 1017 K562 HS and Non-HS examples to predict the *in vivo* DNaseI HS status in K562 cells of 60 000 non-repetitive sequences (as identified by the RepeatMasker track on the UCSC Genome Browser) with mean length 225 bp selected from throughout the human genome. The expected prevalence of HSs in this set of sequences is higher than random background but $<$10% (Sabo *et al*., 2004).

From the resulting SVM probabilities, we randomly selected for further testing sequences with assigned high probability ($>$80%; $n = 146$) and low probability ($<$20%; $n = 43$). Each sequence was tested for DNaseI hypersensitivity in K562 erythroid cells using a previously validated real-time quantitative PCR assay designed to discriminate DNaseI HSs with $>$95% confidence (Sabo *et al*., 2004). We found 108/146 of the high probability predictions to be DNaseI HSs when tested in K562 cells, yielding a positive-predictive value (PPV) for the SVM of 73.9% (Fig. 2). Testing of low probability predictions in the same cell type revealed that 39/43 were correctly

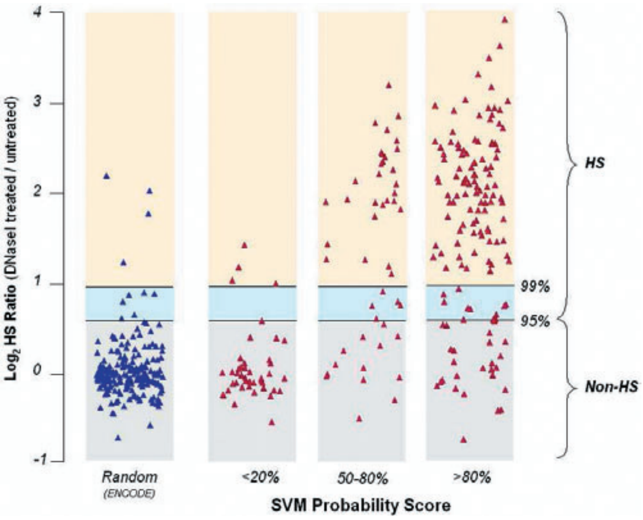


Fig. 2. DNaseI hypersensitivity testing of SVM predictions and control sequences in K562 cells. The y-axis plots the $\log_2$ of the DNaseI sensitivity ratio (copies remaining in DNaseI-untreated sample/DNaseI-treated sample) assayed by real-time quantitative PCR. Results from SVM predictions are stratified into low (<20%), intermediate (50–80%), and high (>80%) SVM-assigned probability groups. Means of six replicate measurements for each PCR amplified sequence ('amplicon') corresponding to an SVM prediction are shown with triangles. Results are classified as non-HS (gray shaded boxes) or HS [blue (95% confidence) and orange (99% confidence) shaded boxes] on the basis of quantitative DNaseI hypersensitivity measurements obtained with real-time PCR (Sabo *et al*., 2004; McArthur *et al*., 2001; Dorschner *et al*., 2004) using a validated model for K562 cells described by Sabo *et al*. (2004). Results from 186 randomly selected amplicons from the ENCODE regions (ENCODE Consortium, 2004) are also shown. The proportion of HS-positives in the random set (5.3%) is higher than expected for the genome at large, given the considerably higher gene and functional element density of the ENCODE regions. (Notably, HSs from the random set coincided with known or predicted regulatory sequences, including HS4 from the β-globin LCR and several promoters and CpG islands.)

classified as non-HSs, for a negative-predictive value (NPV) of 90.7%. We also examined 49 intermediate probability (50–80%) predictions, and found 33 (67.3%) to be positive. The cumulative PPV for all predictions with probability >50% was 70.6%. These results demonstrate the ability of the SVM to identify DNaseI HSs *in vivo* with high accuracy.

The high proportion of true-positive predictions within a single cell type suggests further that the elements identified by the SVM might represent a class of HSs that are active in many tissues or are even constitutive. Additionally, because some HSs are expected to be tissue or lineage-restricted, a proportion of predictions that yielded negative results in erythroid cells might prove to be HS in another tissue type. To address this possibility, we tested a subset ($n = 93$) of sequences with assigned probability >50% in another hematopoietic cell type, B-lymphoblastoid cells (EBV-transformed primary lymphoblast line GM0990, Coriell). Of 65 SVM-predicted HSs that were DNaseI hypersensitive in K562 cells, 58 (89.2%) were also HSs in lymphoblastoid cells. An exemplary SVM-predicted HS of this type lying upstream of the NF1 tumor suppressor gene is illustrated in Figure 3. Conversely, we found 8/28 (28.6%) sequences that tested negative in K562 cells were HS-positive in lymphoid cells. These results indicate that the overall PPV estimate for the SVM based on testing only in K562 cells represents a minimum value. More extensive testing in additional tissue types might reveal further SVM HS predictions to be correct.

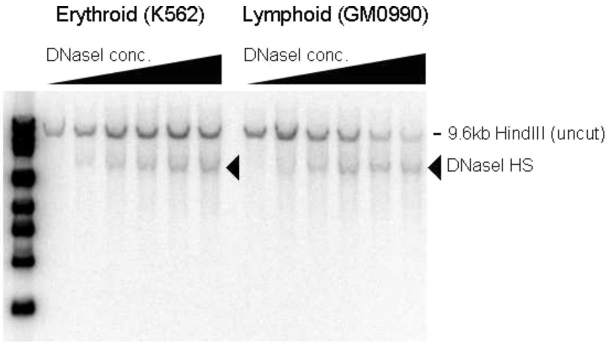


Fig. 3. Conventional DNaseI HS analysis of SVM predictions. To confirm further that SVM predictions correspond to classical DNaseI HSs, we selected positive predictions for conventional DNaseI HS assays employing the indirect end-label Southern blotting technique (Lowrey *et al*., 1992). Shown are exemplary results from an SVM prediction 400 bp upstream of the Nf1 tumor suppressor gene on chromosome 17 that coincides with a classical DNaseI HS in both erythroid (K562) and lymphoid (GM0990) cells. For each tissue type, lanes represent increasing (left to right) DNaseI treatment intensity (0, 1, 2, 4, 8 and 16 U DNaseI). A radiolabeled probe is targeted to the 5′ end of a 9.6 kb HindIII fragment encompassing the Nf1 transcriptional start site and upstream and downstream flanking sequences. As DNaseI concentration increases, the 9.6 kb parental band is cleaved specifically at the hypersensitive site, releasing the marked sub-band.

3.3 Genome-wide prediction

We then considered how frequently SVM-predicted sequences occur in the human genome. We first partitioned the human genome sequence (assembly hg16 = NCBI 34) into non-overlapping 225 bp segments and identified 4 217 066 segments lacking repetitive sequences. Next, we selected and scored all segments and applied a sigmoid fit to derive probabilities from the SVM discriminant scores. The SVM predicted 36 581 (0.89%) genomic segments to be HSs at a probability threshold of 50%; 19 429 (0.47%) had probability scores >80%. At a cumulative minimum PPV level of 70.6% for DNaseI HSs *in vivo*, these results suggest that the human genome contains >26 500 functional non-coding elements of the class predicted by the SVM. Analysis of the distribution of SVM predictions in relation to genes revealed strong clustering around annotated transcriptional start sites; however, 65% of predictions were located >5 kb distant from the nearest 5′ start site.

3.4 Feature analysis

In order to perform its classification, the SVM simultaneously exploits a large collection of simple k-mer sequence features. This collection does not correspond to traditional motifs, but encompasses them in the context of a rich feature space, which implicitly allows for mismatching and complex dependencies between sequence positions by combining many short k-mer features.

In order to gain some insight into this complex feature space representation, we analyzed the set of 83 improperly classified (FN) HSs from the initial training set for the presence of simple sequence features that distinguished them from the correctly recognized class. We observed that the CG dinucleotide frequency was significantly lower (1.3%) in the FN class than in the 197 correctly discerned HSs (6.8%), and that the AT dinucleotide frequency was also skewed, but to a lesser degree (6.2% versus 2.8%, respectively).

To examine whether the SVM had exploited these disparities in producing its initial classifications, we computed the Pearson correlation between the SVM discriminants and each of the 2772 sequence features. This analysis revealed that, during the initial training, the SVM had highlighted CG dinucleotides as the most important simple sequence feature, with a correlation of 0.916. Previous observations stemming from specific genes have suggested that certain CpG-rich sequences play a role in maintaining open chromatin structures (Tazi and Bird, 1990); however, the generality of this observation was unknown. A posteriori analysis of the 36 581 human genomic predictions revealed a sharply lower correlation (0.679), indicating that the SVM was integrating a complex array of additional features in performing predictions. Given the overlap between CpG islands and functionally important genomic locales, significant overlap between the SVM predictions and this feature is expected. However, 34% of the 36 581 predictions lie outside CpG islands, as defined by the CpG island track on the UCSC Genome Browser. Moreover, where overlap occurs, only a small fraction (13%) of the CpG sequence is highlighted by the SVM, suggesting that it is recognizing the functional core of these nebulously defined elements.

3.5 Enrichment in CTCF sites

Although most classes of regulatory sequences bind to a variety of regulatory proteins, insulator and chromatin domain boundary elements invariably contain recognition sites for the protein CTCF. Insulator and boundary elements organize the human genome by partitioning functional gene domains (Bell *et al*., 2001). These elements typically give rise to prominent DNaseI HSs that are manifest across a wide range of tissue types. We therefore hypothesized that CTCF sites should be significantly enriched in high versus low probability SVM predictions. We searched sets of sequences selected from the top 25% and bottom 25% of the SVM probability range for occurrences of the canonical CTCF binding motif CCGCNNGGNGGCAG. This search discovered 3462 CTCF sites that received positive log-odds scores in the top 25% set and only 335 such sites in the bottom 25% set. Using a more stringent log-odds threshold of 2, we found 548 CTCF sites in the top 25% and 29 sites in the bottom 25% set. Among the top 25% set, 3 CTCF sites perfectly match the consensus and 57 more match with a single mismatch. No sites match this well in the bottom 25%. The dramatic enrichment of CTCF sites in high probability SVM predictions suggests that a prominent subset of SVM-predicted HSs function *in vivo* as insulator or domain boundary elements.

4 DISCUSSION

Identification of DNaseI HSs is a gold-standard methodology for the identification of vertebrate *cis*-regulatory sequences and has facilitated the discovery of the vast majority of validated human *cis*-regulatory elements residing outside of core promoters. Although novel molecular approaches for large-scale mapping of DNaseI HSs have recently been described (Dorschner *et al*., 2004; Sabo *et al*., 2004), comprehensive annotation of human DNaseI HSs—even in the context of a single tissue—remains distant and will require substantial resources. In contrast, computational tools provide the basis for rapid coverage of the entire genome.

A priori, prediction of DNaseI HSs is expected to be an extremely challenging computational problem. The fact that it has proven tractable for a subclass of these elements is therefore quite surprising. Given the relatively modest size of the training sets employed here, the accuracy of the approach will probably improve with expanded numbers of examples. Although not every HS necessarily encodes a classical *cis*-regulatory element, most HSs do. It is therefore notable that the current level of predictive accuracy (PPV 70%) is substantially higher than that described for any computationally based methodology for identification of *cis*-regulatory sequences. Nor is attainment of 100% accuracy a requirement, given the potential for coupling of computational predictions to a platform for high-throughput biological validation, such as the high-throughput real-time PCR assay employed here for prospective examination of SVM annotations. Iterative application of the training-and-testing paradigm with additional HS sequences should enable generation of more powerful, accurate and diverse classifiers.

Although described and validated in the context of a single tissue (human erythroid cells), the approach described here is broadly applicable. Extension of this paradigm to other tissue types should enable recognition of additional classes of HSs and, thereby, delineation of large numbers of novel elements expected to play central roles in the transcriptional regulation of human genes. Because DNaseI HSs are a fundamental property of *cis*-regulatory sequences from a wide variety of organisms, the approach described here should be widely extensible to other vertebrate genomes, and to higher eukaryotic genomes generally.

In summary, our results demonstrate the feasibility of accurate, large-scale computational prediction of the *in vivo* signature of human *cis*-regulatory sequences and provide a powerful new tool for the annotation of complex genomes.

REFERENCES

Bell,A.C., West,A.G. and Felsenfeld,G. (2001) Insulators and boundaries: versatile regulatory elements in the eukaryotic genome. *Science*, **291**, 447–450.

Berman,B.P., Nibu,Y., Pfeifer,B.D., Tomancak,P., Celniker,S.E., Levine,M., Rubin,G.M. and Eisen,M.B. (2002) Exploiting transcription factor binding site clustering to identify cis-regulatory modules involved in pattern formation in the *Drosophila* genome. *Proc. Natl Acad. Sci. USA*, **99**, 757–762.

Cristianini,N. and Shawe-Taylor,J. (2000) *An Introduction to Support Vector Machines*. Cambridge University Press, Cambridge, UK.

Dorschner,M.O. *et al*. (2004) High-throughput localization of functional elements by quantitative chromatin profiling. *Nat. Meth.*, **1**, 219–225.

ENCODE Consortium (2004) The ENCODE (ENCyclopedia Of DNA Elements) project. *Science*, **306**, 636–640.

Felsenfeld,G. (1996) Chromatin unfolds. *Cell*, **86**, 13–19.

Gross,D.S. and Garrard,W.T. (1988) Nuclease hypersensitive sites in chromatin. *Annu. Rev. Biochem.*, **57**, 159–197.

Hanley,J.A. and McNeil,B.J. (1982) The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, **143**, 29–36.

Leslie,C., Eskin,E. and Noble,W.S. (2002) The spectrum kernel: a string kernel for SVM protein classification. In Altman,R.B., Dunker,A.K., Hunter,L., Lauderdale,K. and Klein,T.E. (eds), *Proceedings of the Pacific Symposium on Biocomputing*, World Scientific, New Jersey, pp. 564–575.

Li,Q., Peterson,K.R., Fang,X. and Stamatoyannopoulos,G. (2002) Locus control regions. *Blood*, **100**, 3077–3086.

Lowrey,C.H., Bodine,D.M. and Nienhuis,A.W. (1992) Mechanism of DNase I hypersensitive site formation within the human globin locus control region. *Proc. Natl Acad. Sci. USA*, **89**, 1143–1147.

McArthur,M., Gerum,S. and Stamatoyannopoulos,G. (2001) Quantification of DNaseI-sensitivity by real-time PCR: quantitative analysis of DNaseI-hypersensitivity of the mouse β-globin LCR. *J. Molec. Biol.*, **313**, 27–34.

Noble,W.S. (2004) Support vector machine applications in computational biology. In Schölkopf,B., Tsuda,K. and Vert, J.-P. (eds) *Kernel Methods in Computational Biology*, MIT Press, Cambridge, MA, pp. 71–92.

Pavlidis,P., Wapinski,I. and Noble,W.S. (2004) Support vector machine classification on the web. *Bioinformatics*, **20**, 586–587.

Platt,J.C. (1999) Probabilities for support vector machines. In Smola,A., Bartlett,P., Schölkopf,B. and Schuurmans,D. (eds), *Advances in Large Margin Classifiers*, MIT Press, Cambridge, MA, pp. 61–74.

Prakash,A., Blanchette,M., Sinha,S. and Tompa,M. (2004) Motif discovery in heterogeneous sequence data. In *Proceedings of the Pacific Symposium on Biocomputing*, Vol. 9, pp. 348–359.

Reitman,M., Lee,E., Westphal,H. and Felsenfeld,G. (1993) An enhancer/locus control region is not sufficient to open chromatin. *Molec. Cell. Biol.*, **13**, 3990–3998.

Rozen,S. and Skaletsky,H. (2000) Primer3 on the WWW for general users and for biologist programmers. In Krawetz,S. and Misener,S. (eds), *Bioinformatics Methods and Protocols*, Methods in Molecular Biology, Human Press, Totowa, NJ: pp. 365–386.

Sabo,P.J., Hawrylycz,M., Wallace,J.C., Humbert,R., Yu,M., Shafer,A., Kawamoto,J., Hall,R., Mack,J., Dorschner,M.O., McArthur,M. and Stamatoyannopoulos,J.A. (2004) Discovery of functional noncoding elements by digital analysis of chromatin structure. *Proc. Natl Acad. Sci. USA*, **101**, 16837–16842.

Sabo,P.J., Humbert,R., Hawrylycz,M., Wallace,J.C., Dorschner,M.O., McArthur,M. and Stamatoyannopoulos,J.A. (2004) Genome-wide identification of DNase1 hypersensitive sites using active chromatin sequence libraries. *Proc. Natl Acad. Sci. USA*, **101**, 4537–4542.

Sinha,S. and Tompa,M. (2002) Discovery of novel transcription factor binding sites by statistical overrepresentation. *Nucleic Acids Res.*, **30**, 5549–5560.

Stamatoyannopoulos,J.A., Goodwin,A., Joyce,T. and Lowrey,C.H. (1995) NF-E2 and GATA binding motifs are required for the formation of DNase I hypersensitive site 4 of the human beta-globin locus control region. *EMBO J.*, **14**, 106–116.

Talbot,D., Philipsen,S., Fraser,P. and Grosveld,F. (1990) Detailed analysis of the site 3 region of the human beta-globin dominant control region. *EMBO J.*, **9**, 2169–2177.

Tazi,J. and Bird,A. (1990) Alternative chromatin structure at CpG islands. *Cell*, **60**, 909–920.

Tompa,M. *et al.* (2005) Assessing computational tools for the discovery of transcription factor binding sites. *Nat. Biotechnol.*, **23**, 137–144.

Vapnik,V.N. (1998) Adaptive and learning systems for signal processing, communications, and control. In *Statistical Learning Theory*. Wiley, New York.

Wu,C. (1980) The 5′ ends of Drosophila heat shock genes in chromatin are hypersensitive to DNase I. *Nature*, **286**, 854–860.

Zhang,X.H.-F., Heller,K.A., Hefter,I., Leslie,C.S. and Chasin,L.A. (2003) Sequence information for the splicing of human pre-mRNA identified by support vector machine classification. *Genome Res.*, **13**, 2637–2650.

Zien,A., Rätch,G., Mika,S., Schölkopf,B., Lengauer,T. and Müller,K.-R. (2000) Engineering support vector machine kernels that recognize translation initiation sites. *Bioinformatics*, **16**, 799–807.