state. We here present a novel model system constructed by merging the single-cell model, which consisted of intracellular biochemical reactions, together with a proper structure (the histological structure of the hepatic lobule) and emergent new properties of a higher-level order (zonal heterogeneity in ammonia metabolism). Using such a model, it is possible to study how nanoscopic and microscopic biological entities influence mesoscopic and macroscopic biological phenomena. Such approaches hold great promise for advancing our understanding of complicated multicellular tissues, organs, and the organism in a fitness landscape. This is an extremely useful and exclusive feature of systems biology.

Acknowledgments

This work was financially supported by a Grant-in-aid for the Leading Project for Biosimulation, a Grant-in-aid for the 21st Century Center of Excellence (COE) Program: Understanding and Control of Life's Function via Systems Biology, and a Grant-in-aid for Young Scientists B from the Ministry of Education, Culture, Sports, Science, and Technology of Japan; research funds from Yamagata Prefectural Government and Tsuruoka City; and the Inamori Foundation. Pacific Edit reviewed the manuscript prior to submission.

References

1. Achs, M. J., Anderson, J. H., & Garfinkel, D. (1971). Gluconeogenesis in rat liver cytosol. I. Computer analysis of experimental data. *Computers and Biomedical Research, an International Journal, 4*(1), 65–106.
2. Bachmann, C., & Colombo, J. P. (1981). Computer simulation of the urea cycle: Trials for an appropriate model. *Enzyme, 26*(5), 259–264.
3. Bachmann, C., Krahenbuhl, S., & Colombo, J. P. (1982). Purification and properties of acetyl-CoA:L-glutamate *n*-acetyltransferase from human liver. *The Biochemical Journal, 205*(1), 123–127.
4. Boon, L., Geerts, W. J., Jonker, A., Lamers, W. H., & Van Noorden, C. J. (1999). High protein diet induces pericentral glutamate dehydrogenase and ornithine aminotransferase to provide sufficient glutamate for pericentral detoxification of ammonia in rat liver lobules. *Histochemistry and Cell Biology, 111*(6), 445–452.
5. Christoffels, V. M., Sassi, H., Ruijter, J. M., Moorman, A. F., Grange, T., & Lamers, W. H. (1999). A mechanistic model for the development and maintenance of portocentral gradients in gene expression in the liver. *Hepatology, 29*(4), 1180–1192.
6. Crawford, J. M., & Blum, J. J. (1983). Quantitative analysis of flux along the gluconeogenic, glycolytic and pentose phosphate pathways under reducing conditions in hepatocytes isolated from fed rats. *The Biochemical Journal, 212*(3), 585–598.
7. Elliott, K. R., & Tipton, K. F. (1974). Kinetic studies of bovine liver carbamoyl phosphate synthetase. *The Biochemical Journal, 141*(3), 807–816.
8. Elliott, K. R., & Tipton, K. F. (1974). Product inhibition studies on bovine liver carbamoyl phosphate synthetase. *The Biochemical Journal, 141*(3), 817–824.
9. Gebhardt, R. (1992). Metabolic zonation of the liver: Regulation and implications for liver function. *Pharmacology & Therapeutics, 53*(3), 275–354.
10. Gebhardt, R., Gaunitz, F., & Mecke, D. (1994). Heterogeneous (positional) expression of hepatic glutamine synthetase: Features, regulation and implications for hepatocarcinogenesis. In G. Weber & C. E. Forrest Weber (Eds.), *Advances in enzyme regulation: Proceedings of the Twenty-Seventh Symposium on Regulation of Enzyme Activity and Synthesis in Normal and Neoplastic TI* (pp. 3427–3456). New York: Elsevier Science.
11. Gebhardt, R., & Mecke, D. (1983). Glutamate uptake by cultured rat hepatocytes is mediated by hormonally inducible, sodium-dependent transport systems. *FEBS Letters, 161*(2), 275–278.
12. Gupta, S., Rajvanshi, P., Sokhi, R. P., Vaidya, S., Irani, A. N., & Gorla, G. R. (1999). Position-specific gene expression in the liver lobule is directed by the microenvironment and not by the previous cell differentiation state. *The Journal of Biological Chemistry, 274*(4), 2157–2165.
13. Haussinger, D. (1989). Glutamine metabolism in the liver: Overview and current concepts. *Metabolism: Clinical and Experimental, 38*(8 Suppl. 1), 14–17.
14. Haussinger, D. (1990). Nitrogen metabolism in liver: Structural and functional organization and physiological relevance. *The Biochemical Journal, 267*(2), 281–290.

15. Haussinger, D. (1990). Organization of hepatic nitrogen metabolism and its relation to acid-base homeostasis. *Klinische Wochenschrift, 68*(22), 1096–1101.

16. Haussinger, D. (1992). Liver and systemic pH-regulation. *Zeitschrift für Gastroenterologie, 30*(2), 147–150.

17. Haussinger, D. (1997). Liver regulation of acid-base balance. *Mineral and Electrolyte Metabolism, 23*(3–6), 249–252.

18. Haussinger, D. (1998). Hepatic glutamine transport and metabolism. *Advances in Enzymology and Related Areas of Molecular Biology, 72*, 43–86.

19. Haussinger, D., Gerok, W., & Sies, H. (1983). Regulation of flux through glutaminase and glutamine synthetase in isolated perfused rat liver. *Biochimica et Biophysica Acta, 755*(2), 272–278.

20. Haussinger, D., Lamers, W. H., & Moorman, A. F. (1992). Hepatocyte heterogeneity in the metabolism of amino acids and ammonia. *Enzyme, 46*(1–3), 72–93.

21. Ideker, T., Galitski, T., & Hood, L. (2001). A new approach to decoding life: Systems biology. *Annual Review of Genomics and Human Genetics, 2*, 343–372.

22. Jungermann, K. (1986). Functional heterogeneity of periportal and perivenous hepatocytes. *Enzyme, 35*(3), 161–180.

23. Jungermann, K. (1995). Zonation of metabolism and gene expression in liver. *Histochemistry and Cell Biology, 103*(2), 81–91.

24. Jungermann, K., & Katz, N. (1989). Functional specialization of different hepatocyte populations. *Physiological Reviews, 69*(3), 708–764.

25. Jungermann, K., & Kietzmann, T. (1996). Zonation of parenchymal and nonparenchymal metabolism in liver. *Annual Review of Nutrition, 16*, 179–203.

26. Kitano, H. (2002). Systems biology: A brief overview. *Science, 295*(5560), 1662–1664.

27. Kohn, M. C. (1992). Propagation of information in metanet graph models. *Journal of Theoretical Biology, 154*(4), 505–517.

28. Kohn, M. C., Tohmaz, A. S., Giroux, K. J., Blumenthal, G. M., Feezor, M. D., & Millington, D. S. (2002). Robustness of metanet graph models: Predicting control of urea production in humans. *Bio Systems, 65*(1), 61–78.

29. Kuchel, P. W., Roberts, D. V., & Nichol, L. W. (1977). The simulation of the urea cycle: Correlation of effects due to inborn errors in the catalytic properties of the enzymes with clinical-biochemical observations. *The Australian Journal of Experimental Biology and Medical Science, 55*(3), 309–326.

30. Kuo, F. C., & Darnell, J. E., Jr. (1991). Evidence that interaction of hepatocytes with the collecting (hepatic) veins triggers position-specific transcription of the glutamine synthetase and ornithine aminotransferase genes in the mouse liver. *Molecular and Cellular Biology, 11*(12), 6050–6058.

31. Kuo, F. C., Hwu, W. L., Valle, D., & Darnell, J. E. (1991). Colocalization in pericentral hepatocytes in adult mice and similarity in developmental expression pattern of ornithine aminotransferase and glutamine synthetase mRNA. *Proceedings of the National Academy of Sciences of the United States of America, 88*(21), 9468–9472.

32. Low, S. Y., Salter, M., Knowles, R. G., Pogson, C. I., & Rennie, M. J. (1993). A quantitative analysis of the control of glutamine catabolism in rat liver cells. Use of selective inhibitors. *The Biochemical Journal, 295*(Pt. 2), 617–624.

33. Maly, I. P., & Sasse, D. (1991). Microquantitative analysis of the intra-acinar profiles of glutamate dehydrogenase in rat liver. *The Journal of Histochemistry and Cytochemistry, 39*(8), 1121–1124.

34. Maynard Smith, J. (1998). *Evolutionary genetics*, (2nd ed.) New York: Oxford University Press.

35. McGivan, J. D., & Bradford, N. M. (1983). Characteristics of the activation of glutaminase by ammonia in sonicated rat liver mitochondria. *Biochimica et Biophysica Acta, 759*(3), 296–302.

36. Notenboom, R. G., de Boer, P. A., Moorman, A. F., & Lamers, W. H. (1996). The establishment of the hepatic architecture is a prerequisite for the development of a lobular pattern of gene expression. *Development, 122*(1), 321–332.

37. Sasse, D., Spornitz, U. M., & Maly, I. P. (1992). Liver architecture. *Enzyme, 46*(1–3), 8–32.

38. Schneider, W., Siems, W., & Grune, T. (1990). Balancing of energy-consuming processes of rat hepatocytes. *Cell Biochemistry and Function*, *8*(4), 227–232.

39. Segel, I. H. (1993). *Enzyme kinetics: Behavior and analysis of rapid equilibrium and steady state enzyme systems.* New York: Wiley.

40. Seyama, S., Kuroda, Y., & Katunuma, N. (1972). Purification and comparison of glutamine synthetase from rat and chick livers. *Journal of Biochemistry*, *72*(4), 1017–1027.

41. Szweda, L. I., & Atkinson, D. E. (1989). Response of rat liver glutaminase to pH. Mediation by phosphate and ammonium ions. *The Journal of Biological Chemistry*, *264*(26), 15357–15360.

42. Tomita, M., Hashimoto, K., Takahashi, K., Shimizu, T. S., Matsuzaki, Y., Miyoshi, F., Saito, K., Tanida, S., Yugi, K., Venter, J. C., & Hutchison, C. A., III. (1999). E-CELL: Software environment for whole-cell simulation. *Bioinformatics*, *15*(1), 72–84.

43. Wagenaar, G. T., Chamuleau, R. A., de Haan, J. G., Maas, M. A., de Boer, P. A., Marx, F., Moorman, A. F., Frederiks, W. M., & Lamers, W. H. (1993). Experimental evidence that the physiological position of the liver within the circulation is not a major determinant of zonation of gene expression. *Hepatology*, *18*(5), 1144–1153.

44. Wagenaar, G. T., Chamuleau, R. A., Maas, M. A., de Bruin, K., Korfage, H. A., & Lamers, W. H. (1994). The physiological position of the liver in the circulation is not a major determinant of its functional capacity. *Hepatology*, *20*(6), 1532–1540.

45. Watford, M. (1993). Hepatic glutaminase expression: Relationship to kidney-type glutaminase and to the urea cycle. *The FASEB Journal*, *7*(15), 1468–1474.

Appendix I: Details of Mathematical Model

AI.1 Mathematical Models of Chemical Reactions and Transports

See Web Supplement, Table S3, for parameter values.

AI.1.1 Carbamoyl Phosphate Synthetase (EC. 6.3.4.16)

The enzyme catalyzes

$$2ATP + NH_4^+ + HCO_3^- \rightarrow 2AMP + 2Pi + CP$$

in mitochondria. The kinetic model was obtained from previous literature [7, 8]:

$$v_{CPS} = \frac{k_{cat,CPS}[CPS]}{\text{denominator}_{CPS}}$$

where

$$\begin{aligned}
\text{denominator}_{CPS} = 1 &+ \frac{K_{mATP_1,CPS} + K_{mATP_2,CPS}}{[ATP]} + \frac{K_{mHCO_3^-,CPS}}{[HCO_3^-]} + \frac{K_{mNAG,CPS}}{[NAG]} + \frac{K_{mNH_4^+,CPS}}{[NH_4^+]} \\
&+ \frac{K_{sATP_1,CPS}K_{m'HCO_3^-,CPS} + K_{sHCO_3^-,CPS}\left(K_{mATP_2,CPS} + K_{m'ATP_2,CPS}\right)}{[ATP]} + \frac{K_{sNAG,CPS}K_{mATP_1,CPS}}{[ATP][NAG]} \\
&+ \frac{K_{sMg^{2+},CPS}K_{mATP_1,CPS}}{[ATP][Mg^{2+}]} + \frac{K_{sATP_2,CPS}K_{mNH_4^+,CPS}}{[ATP][NH_4^+]} + \frac{K_{sNAG,CPS}K_{m'HCO_3^-,CPS}}{[NAG][HCO_3^-]} \\
&+ \frac{K_{sNAG,CPS}K_{sMg^{2+},CPS}K_{mATP_1,CPS}}{[ATP][Mg^{2+}][NAG]} \\
&+ \frac{K_{sATP_1,CPS}K_{sNAG,CPS}K_{m'HCO_3^-,CPS} + K_{sNAG,CPS}K_{sHCO_3^-,CPS}K_{m'ATP_2,CPS}}{[ATP][NAG][HCO_3^-]} \\
&+ \frac{K_{sATP_1,CPS}K_{sMg^{2+},CPS}K_{m'HCO_3^-,CPS}}{[ATP][Mg^{2+}][HCO_3^-]} + \frac{K_{sATP_2,CPS}K_{sHCO_3^-,CPS}K_{mNH_4^+,CPS}\left(K_{mNH_3,CPS} + K_{m'NH_4^+,CPS}\right)}{[ATP][HCO_3^-][NH_4^+]} \\
&+ \frac{K_{sATP_1,CPS}K_{sMg^{2+},CPS}K_{sNAG,CPS}K_{m'HCO_3^-,CPS}}{[ATP][Mg^{2+}][NAG][HCO_3^-]} + \frac{K_{sATP_2,CPS}K_{sNAG,CPS}K_{sHCO_3^-,CPS}K_{m'NH_4^+,CPS}}{[ATP][NAG][HCO_3^-][NH_4^+]} \\
&+ \frac{K_{sATP_1,CPS}K_{sHCO_3^-,CPS}K_{m'ATP_2,CPS}}{[ATP]^2[HCO_3^-]} + \frac{K_{sATP_1,CPS}K_{sNAG,CPS}K_{sHCO_3^-,CPS}K_{m'ATP_2,CPS}}{[ATP]^2[NAG][HCO_3^-]} \\
&+ \frac{K_{sATP_1,CPS}K_{sMg^{2+},CPS}K_{sHCO_3^-,CPS}K_{m'ATP_2,CPS}}{[ATP]^2[Mg^{2+}][HCO_3^-]} + \frac{K_{sATP_1,CPS}K_{sATP_2,CPS}K_{sHCO_3^-,CPS}K_{m'NH_4^+,CPS}}{[ATP]^2[HCO_3^-][NH_4^+]} \\
&+ \frac{K_{sATP_1,CPS}K_{sMg^{2+},CPS}K_{sNAG,CPS}K_{sHCO_3^-,CPS}K_{m'ATP_2,CPS}}{[ATP]^2[Mg^{2+}][NAG][HCO_3^-]} \\
&+ \frac{K_{sATP_1,CPS}K_{sATP_2,CPS}K_{sMg^{2+},CPS}K_{sHCO_3^-,CPS}K_{m'NH_4^+,CPS}}{[ATP]^2[Mg^{2+}][HCO_3^-][NH_4^+]} \\
&+ \frac{K_{sATP_1,CPS}K_{sATP_2,CPS}K_{sNAG,CPS}K_{sHCO_3^-,CPS}K_{m'NH_4^+,CPS}}{[ATP]^2[NAG][HCO_3^-][NH_4^+]} \\
&+ \frac{K_{sATP_1,CPS}K_{sATP_2,CPS}K_{sMg^{2+},CPS}K_{sNAG,CPS}K_{sHCO_3^-,CPS}K_{m'NH_4^+,CPS}}{[ATP]^2[Mg^{2+}][NAG][HCO_3^-][NH_4^+]}
\end{aligned}$$

A1.1.2 N-Acetyl Glutamate Synthetase (EC. 2.3.1.1)

The enzymes catalyze AcCoA + Glu → CoA + NAG.
The reaction mechanism is a nonreversible rapid equilibrium random bi-bi mechanism [3]:

$$v_{AGS} = \frac{\dfrac{k_{cat,AGS}[AGS][AcCoA][Glu]}{\left(1+\frac{K_{aArg,AGS}}{[Arg]}\right)}}{\text{denominator}_{AGS}}$$

where

$$\begin{aligned}\text{denominator}_{AGS} = {} & K_{iAcCoA,AGS}K_{mGlu,AGS}\left(1+\frac{[CoA]}{K_{iCoA,AGS}}\right)\left(1+\frac{[NAG]}{K_{iNAG,AGS}}\right) \\ & + K_{mGlu,AGS}\left(1+\frac{[NAG]}{K_{iNAG,AGS}}\right)[AcCoA] \\ & + K_{mAcCoA,AGS}\left(1+\frac{[CoA]}{K_{iCoA,AGS}}\right)[Glu] + [AcCoA][Glu]\end{aligned}$$

A1.1.3 Glutamine Synthetase (EC. 6.3.1.2)

The enzyme catalyzes ATP + Glu + NH_4^+ → AMP + Pi + Gln:

$$v_{GS} = \frac{k_{cat,GS}[GS][Glu][ATP][NH_4^+]}{\left(K_{mGlu,GS}+[Glu]\right)\left(K_{mATP,GS}+[ATP]\right)\left(K_{mNH_4^+,GS}+\left[NH_4^+\right]\right)}$$

A1.1.4 Phosphate-Dependent Glutaminase (EC. 3.5.1.2)

The enzyme catalyzes Gln + Pi → Glu + NH_4^+. It is activated by the product: ammonia [27]. Cooperativity of glutamine and Pi, which is an essential activator for phosphate-dependent glutaminase, were modeled by the Hill equation [39]:

$$v_{Glnase} = \frac{\dfrac{k_{cat,Glnase}[Glnase]}{1+\dfrac{K_{a,Glnase}}{\left[NH_4^+\right]}}[Gln]^{n_{Gln,Glnase}}}{[Gln]_{0.5}^{n_{Gln,Glnase}}\left(1+\dfrac{[Pi]_{0.5}^{n_{Pi,Glnase}}}{[Pi]^{n_{Pi,Glnase}}}\right)+[Gln]^{n_{Gln,Glnase}}\left(1+\dfrac{[Pi]_{0.5}^{n_{Pi,Glnase}}}{[Pi]^{n_{Pi,Glnase}}}\right)}$$

A1.1.5 Ornithine Carbamoyltransferase (EC. 2.1.3.3)

The enzyme catalyzes CP + Orn ↔ Pi + Cit. The reaction mechanism is an ordered bi-bi sequential mechanism [29]:

$$v_{OCT} = \frac{\left(k_{1,OCT}k_{3,OCT}k_{5,OCT}k_{7,OCT}[CP][Orn] - k_{2,OCT}k_{4,OCT}k_{6,OCT}k_{8,OCT}[Cit][Pi]\right)[OCT]}{\text{denominator}_{OCT}}$$

where

$$\begin{aligned}\text{denominator}_{OCT} = {} & k_{2,OCT}k_{7,OCT}\left(k_{4,OCT} + k_{5,OCT}\right) + k_{1,OCT}k_{7,OCT}\left(k_{4,OCT} + k_{5,OCT}\right)[CP] \\ & + k_{2,OCT}k_{8,OCT}\left(k_{4,OCT} + k_{5,OCT}\right)[Pi] + k_{3,OCT}k_{5,OCT}k_{7,OCT}[Orn] \\ & + k_{2,OCT}k_{4,OCT}k_{6,OCT}[Cit] + k_{1,OCT}k_{3,OCT}\left(k_{5,OCT} + k_{7,OCT}\right)[CP][Orn] \\ & + k_{6,OCT}k_{8,OCT}\left(k_{2,OCT} + k_{4,OCT}\right)[Pi][Cit] + k_{1,OCT}k_{4,OCT}k_{6,OCT}[CP][Cit] \\ & + k_{1,OCT}k_{3,OCT}k_{6,OCT}[CP][Orn][Cit] + k_{3,OCT}k_{5,OCT}k_{8,OCT}[Orn][Pi] \\ & + k_{3,OCT}k_{6,OCT}k_{8,OCT}[Orn][Pi][Cit]\end{aligned}$$

A1.1.6 Argininosuccinate Synthetase (EC. 6.3.4.5)

The enzyme catalyzes ATP + Cit + Asp ↔ AMP + Pi + ASA. The reaction mechanism is an ordered ter-ter mechanism [29]:

$$v_{ASS} = \frac{\left(k_{1,ASS}k_{3,ASS}k_{5,ASS}k_{7,ASS}k_{9,ASS}k_{11,ASS}[Cit][Asp][ATP] - k_{2,ASS}k_{4,ASS}k_{6,ASS}k_{8,ASS}k_{10,ASS}k_{12,ASS}[ASA][AMP][Pi]\right)}{\text{denominator}_{ASS}}[ASS]$$

where

$$
\begin{aligned}
\text{denominator}_{ASS} = {} & k_{2,ASS}k_{4,ASS}k_{9,ASS}k_{11,ASS}\left(k_{6,ASS}+k_{7,ASS}\right) \\
& +k_{1,ASS}k_{4,ASS}k_{6,ASS}k_{8,ASS}k_{11,ASS}[\mathrm{Cit}][\mathrm{Pi}] \\
& +k_{1,ASS}k_{4,ASS}k_{9,ASS}k_{11,ASS}\left(k_{6,ASS}+k_{7,ASS}\right)[\mathrm{Cit}] \\
& +k_{2,ASS}k_{5,ASS}k_{7,ASS}k_{9,ASS}k_{12,ASS}[\mathrm{Asp}][\mathrm{ASA}] \\
& +k_{2,ASS}k_{5,ASS}k_{7,ASS}k_{9,ASS}k_{11,ASS}[\mathrm{Asp}] \\
& +k_{1,ASS}k_{3,ASS}k_{6,ASS}k_{8,ASS}k_{11,ASS}[\mathrm{Cit}][\mathrm{ATP}][\mathrm{Pi}] \\
& +k_{1,ASS}k_{3,ASS}k_{9,ASS}k_{11,ASS}\left(k_{6,ASS}+k_{7,ASS}\right)[\mathrm{Cit}][\mathrm{ATP}] \\
& +k_{1,ASS}k_{4,ASS}k_{6,ASS}k_{8,ASS}k_{10,ASS}[\mathrm{Cit}][\mathrm{AMP}][\mathrm{Pi}] \\
& +k_{1,ASS}k_{5,ASS}k_{7,ASS}k_{9,ASS}k_{11,ASS}[\mathrm{Cit}][\mathrm{Asp}] \\
& +k_{3,ASS}k_{5,ASS}k_{7,ASS}k_{9,ASS}k_{12,ASS}[\mathrm{ATP}][\mathrm{Asp}][\mathrm{ASA}] \\
& +k_{3,ASS}k_{5,ASS}k_{7,ASS}k_{9,ASS}k_{11,ASS}[\mathrm{ATP}][\mathrm{Asp}] \\
& +k_{2,ASS}k_{5,ASS}k_{7,ASS}k_{9,ASS}k_{12,ASS}[\mathrm{AMP}][\mathrm{Asp}][\mathrm{ASA}] \\
& +k_{2,ASS}k_{4,ASS}k_{9,ASS}k_{12,ASS}\left(k_{6,ASS}+k_{7,ASS}\right)[\mathrm{ASA}] \\
& +k_{1,ASS}k_{3,ASS}k_{6,ASS}k_{8,ASS}k_{10,ASS}[\mathrm{Cit}][\mathrm{ATP}][\mathrm{AMP}][\mathrm{Pi}] \\
& +k_{1,ASS}k_{3,ASS}k_{5,ASS}\left(k_{7,ASS}k_{9,ASS}+k_{7,ASS}k_{11,ASS}+k_{9,ASS}k_{11,ASS}\right)[\mathrm{Cit}][\mathrm{ATP}][\mathrm{Asp}] \\
& +k_{1,ASS}k_{3,ASS}k_{5,ASS}k_{8,ASS}k_{11,ASS}[\mathrm{Cit}][\mathrm{Asp}][\mathrm{ATP}][\mathrm{Pi}] \\
& +k_{2,ASS}k_{4,ASS}k_{6,ASS}k_{8,ASS}k_{11,ASS}[\mathrm{Pi}] \\
& +k_{1,ASS}k_{3,ASS}k_{5,ASS}k_{7,ASS}k_{10,ASS}[\mathrm{Cit}][\mathrm{Asp}][\mathrm{ATP}][\mathrm{AMP}] \\
& +k_{2,ASS}k_{4,ASS}k_{6,ASS}k_{8,ASS}k_{10,ASS}[\mathrm{AMP}][\mathrm{Pi}] \\
& +k_{3,ASS}k_{5,ASS}k_{7,ASS}k_{10,ASS}k_{12,ASS}[\mathrm{Asp}][\mathrm{ATP}][\mathrm{ASA}][\mathrm{AMP}] \\
& +k_{2,ASS}k_{4,ASS}k_{6,ASS}k_{8,ASS}k_{12,ASS}[\mathrm{ASA}][\mathrm{Pi}]
\end{aligned}
$$

A1.1.7 Argininosuccinate Lyase (EC. 4.3.2.1)

The enzyme catalyzes ASA ↔ Fum + Arg. The reaction mechanism is an ordered uni-bi mechanism:

$$v_{\mathrm{ASL}} = \frac{\left(k_{1,\mathrm{ASL}}k_{3,\mathrm{ASL}}k_{5,\mathrm{ASL}}[\mathrm{ASA}] - k_{2,\mathrm{ASL}}k_{4,\mathrm{ASL}}k_{6,\mathrm{ASL}}[\mathrm{Fum}][\mathrm{Arg}]\right)[\mathrm{ASL}]}{\begin{aligned}&k_{5,\mathrm{ASL}}\left(k_{2,\mathrm{ASL}} + k_{3,\mathrm{ASL}}\right) + k_{1,\mathrm{ASL}}\left(k_{3,\mathrm{ASL}} + k_{5,\mathrm{ASL}}\right)[\mathrm{ASA}] + k_{2,\mathrm{ASL}}k_{4,\mathrm{ASL}}[\mathrm{Fum}]\\ &+ k_{6,\mathrm{ASL}}\left(k_{2,\mathrm{ASL}} + k_{3,\mathrm{ASL}}\right) + k_{4,\mathrm{ASL}}k_{6,\mathrm{ASL}}[\mathrm{Fum}][\mathrm{Arg}] + k_{1,\mathrm{ASL}}k_{4,\mathrm{ASL}}[\mathrm{ASA}][\mathrm{Fum}]\end{aligned}}$$

A1.1.8 Arginase (EC. 3.5.3.1)

The enzyme catalyzes Arg → urea + Orn. The reaction is an irreversible process and inhibited by ornithine:

$$v_{\mathrm{Argase}} = \frac{k_{1,\mathrm{Argase}}k_{3,\mathrm{Argase}}k_{4,\mathrm{Argase}}[\mathrm{Arn}][\mathrm{Argase}]}{\begin{aligned}&k_{4,\mathrm{Argase}}\left(k_{2,\mathrm{Argase}} + k_{3,\mathrm{Argase}}\right) + k_{5,\mathrm{Argase}}\left(k_{2,\mathrm{Argase}} + k_{3,\mathrm{Argase}}\right)[\mathrm{Orn}]\\ &+ k_{1,\mathrm{Argase}}\left(k_{3,\mathrm{Argase}} + k_{4,\mathrm{Argase}}\right)[\mathrm{Arn}]\end{aligned}}$$

A1.1.9 MetaNet Model

OTL, GTL, GATL, OAT, GOT_m, GOT_c, GDH, GAT, and GAMT were modeled using MetaNet [28]. The reaction stoichiometries were defined as follows:

OTL: $\mathrm{Cit}_m + \mathrm{Orn}_c \leftrightarrow \mathrm{Cit}_c + \mathrm{Orn}_m$

GTL: $\mathrm{Glu}_c \leftrightarrow \mathrm{Glu}_m$

GATL: $\mathrm{Glu}_c + \mathrm{Asp}_m \leftrightarrow \mathrm{Glu}_m + \mathrm{Asp}_c$

OAT: $\mathrm{Orn} + \mathrm{AKG} \leftrightarrow \mathrm{Pyrroline\text{-}5\text{-}carboxylate} + \mathrm{Glu}$

GOT_m: $\mathrm{Glu} + \mathrm{OAA} \leftrightarrow \mathrm{Asp} + \mathrm{AKG}$

GOT_c: $\mathrm{Asp} + \mathrm{AKG} \leftrightarrow \mathrm{Glu} + \mathrm{OAA}$

GDH: $\mathrm{Glu} + \mathrm{NAD}^+ \leftrightarrow \mathrm{AKG} + \mathrm{NH}_4^+ + \mathrm{NADH}$

GAT: $\mathrm{Gly} + \mathrm{Arg} \leftrightarrow \mathrm{Orn} + \mathrm{GAA}$

GAMT: $\mathrm{SAM} + \mathrm{GAA} \rightarrow \mathrm{Cre}$

Although MetaNet is not guaranteed to accurately reproduce enzyme kinetics, it was used in our model with the expectation it would roughly estimate the rates of reactions. The velocities of the reactions were calculated as follows:

$$v_N = \frac{V_{\max,N}\left(1 - \frac{V'_{r,N}}{V'_{f,N}}\frac{\prod_{j_N}\left(\frac{c_{j_N}}{K'_{j_N,N}}\right)}{\prod_{i_N}\left(\frac{c_{i_N}}{K'_{i_N,N}}\right)}\right)}{1 + \sum_{i_N}\left(\frac{c_{i_N}}{K'_{i_N,N}}\right)^{n_{i_N,N}}\left(1 + \sum_{j_N}\left(\frac{c_{j_N}}{K'_{j_N,N}}\right)^{n_{j_N,N}}\right)\left(1 + \sum_{k'_N}\left(\frac{K'_{k_N,N}}{c_{k'_N}}\right)^{n_{k'_N,N}} + \sum_{l'_N}\left(\frac{c_{l'_N}}{K'_{l_N,N}}\right)^{n_{l'_N,N}}\right)}$$

where

$$K'_{s,x} = K_{s,x}\left(1 + \sum_{k_x}\left(\frac{K_{k_x,x}}{c_{k_x}}\right)^{n_{k_x,x}} + \sum_{h}\left(\frac{c_{h_x}}{K_{h_x,x}}\right)^{n_{h_x,x}}\right)$$

$K_{s,x}$ and $n_{s,x}$ are the binding constant and the *cooperativity index* (essentially a Hill exponent) of substance (or effector) s of enzyme x (equilibrium constant for dissociation of the enzyme-ligand complex), respectively, and $K'_{s,x}$ is the former's effective binding constant, which reflects the activities of the competitive activators k_x and the competitive inhibitors h_x; c_s is the concentration of substance s; and k'_x and h'_x are the noncompetitive activators and noncompetitive inhibitors of the reaction catalyzed by enzyme x, respectively [27].

A1.1.10 System N

Glutamine is transported into the cytoplasm by a sodium-dependent transport mechanism. This process is inhibited by histidine [32]:

$$v_{SysN} = V_{max,SysN}\left[\left(\frac{[Na^+]_e}{[Na^+]_e + K_{mNa,SysN}}\right)\left(\frac{[Glu]_e}{[Glu]_e + K_{mGlu,SysN}\left(1 + \frac{[His]_e}{K_{iHis,SysN}}\right)}\right) - \left(\frac{[Na^+]_c}{[Na^+]_c + K_{mNa,SysN}}\right)\left(\frac{[Glu]_c}{[Glu]_c + K_{mGlu,SysN}\left(1 + \frac{[His]_c}{K_{iHis,SysN}}\right)}\right)\right]$$

A1.1.11 System L

Glutamine is transported into the cytoplasm by a sodium-independent transport mechanism. This process is inhibited by tryptophan [32].

$$v_{SysL} = V_{max,SysL}\left(\frac{[Glu]_e}{[Glu]_e + K_{mGlu,SysL}\left(1 + \frac{[Trp]_e}{K_{iTrp,SysL}}\right)} - \frac{[Glu]_c}{[Glu]_c + K_{mGlu,SysL}\left(1 + \frac{[Trp]_c}{K_{iTrp,SysL}}\right)}\right)$$

A1.1.12 Ammonia Transport between Sinusoid and Cytoplasm

Ammonia transport between the sinusoid and cytoplasm was modeled based on the general mass action law:

$$v_{NH_4^+4\text{-}tp} = k_{NH_4^+4\text{-}tp}\left([NH_4^+]_c - [NH_4^+]_e\right)$$

A1.1.13 Transportation of Glutamine, Arginine, and Ammonia between Cytoplasm and Mitochondria

Transports of glutamine, arginine, and ammonia across the mitochondrial membrane were presumed to rapidly attain equilibrium:

$$K_{eq,x}\left([S]_c - v_x\right) = \left([S]_m + v_x\right)$$

A1.1.14 Urea Transport to Sinusoid

Excretion of urea in the sinusoidal space was modeled based on the general mass action law:

$$v_{\text{Urea-tp}} = k_{\text{Urea-tp}}\left([\text{urea}]_c - [\text{urea}]_e\right)$$

A1.1.15 Glutamate Transport between Sinusoid and Cytoplasm

Glutamate transport between the sinusoid and cytoplasm was modeled as Michaelis-Menten reversible kinetics:

$$v_{\text{Glu-tp}} = V_{m\text{F,Glu-tp}}\left(\frac{[\text{Glu}]_e}{[\text{Glu}]_e + K_{m\text{Glu,Glu-tp}}}\right) - V_{m\text{R,Glu-tp}}\left(\frac{[\text{Glu}]_c}{[\text{Glu}]_c + K_{m\text{Glu,Glu-tp}}}\right)$$

A1.1.16 Glutamate Flux from the Outside Pathways

Glutamate flux from the outside pathways of the model was represented by the difference between zero-order influx and efflux based on the general mass action law:

$$v_{\text{Glu-spp}} = J_{\text{Glu-spp}} - k_{\text{Glu-spp}}[\text{Glu}]_c$$

A1.1.17 Degradation of Metabolites

Degradation of *N*-acetyl glutamate, Pi, and CoA was modeled based on the general mass action law under the assumption of steady state:

$$v_{\text{deg},s} = k_{\text{deg},s}[s]$$

where s is a substance.

A1.1.18 Ornithine Inflow from Other Reactions

To hold the steady state, ornithine inflow from other reactions was presumed to be equal to the flux of ornithine aminotransferase, v_{OAT}.

A1.2 Mathematical Model of Metabolite Flows in Sinusoid

Flows of ammonia, glutamine, glutamate, and urea from the nth sinusoidal compartment to the n+1th compartment, v_{e,s_n}, were modeled based on the general mass action law:

$$v_{e,s_n} = k_e[s_n]_e$$

where s_n represents a substance in the nth compartment of the sinusoid.

A1.3 Mathematical Model of Gene Expression of Carbamoyl Phosphate Synthetase, Glutamine Synthetase, and Ornithine Aminotransferase in Hepatic Lobule

To describe the regulated gene expression of three enzymes—carbamoyl phosphate synthetase, glutamine synthetase, and ornithine aminotransferase—along the porto-central axis, we adopted the

mechanistic model proposed by Christoffels et al. [5]. The model is based on simple receptor-ligand kinetics, and the parameters are fitted by experimental values. $[F_x^*]$ is the concentration of the active transcription factor F of enzyme x, and assumed as follows [5]:

Carbamoyl phosphate synthetase: $[F_{CPS}^*] = 0.2 - 0.01X$

Glutamine synthetase and ornithine aminotransferase: $[F_{GS}^*] = [F_{OAT}^*] = 0.1X$

where X is the radius of the hepatic lobule: $X = 0$ corresponds to the portal tracts, and $X = 10$ corresponds to the central vein. Thus, X was defined as follows in our model:

$$X = 10 \times \frac{n}{\text{total number of sinusoidal compartments}}$$

where n is the number of a compartment among the eight compartments, $n = 1$ corresponds to the compartment adjacent to the portal tracts, and $n = 8$ corresponds to the compartment adjacent to the central vein. The total number of sinusoidal compartments is eight in our model.

$R_{GX,x}$ is the relative rate of transcription, assumed to correspond to the transcription rate in our model. $R_{GX,x}$ is calculated using the fractional saturation $Y_{GX,x}$, the dissociation constant $K_{GX,x}$, and the Hill coefficient $n_{GX,x}$ as follows [5]:

$$Y_{GX,x} = \frac{[F_x^*]^{n_{GX,x}}}{[F_x^*]^{n_{GX,x}} + K_{GX,x}^{n_{GX,x}}}$$

$$R_{GX,x} = R_{max,GX,x} Y_{GX,x}$$

Carbamoyl phosphate synthetase was fitted with high-affinity ($Y_{GX,CPS,h}$) and low-affinity ($Y_{GX,CPS,l}$) units as follow [5]:

$$R_{GX,CPS} = R_{max,GX,CPS}\left(Y_{GX,CPS,h} + Y_{GX,CPS,l}\right)$$

A1.4 Varying the Uncertain Parameters

The rate constants for glutamate supply from other pathways (the glutamate transport system and the sinusoidal flow model) were uncertain. Therefore we prepared 60 model instances for each type by varying these rate constant values under a steady-state assumption.

Figures 4 and 5 presented the results under the conditions in Table 3 as a representative of the 60 model instances in each gene expression pattern; after 50,000 s from the start of simulation, with the value 3E−5 M s^{-1} for the glutamate influx from pathways outside of the model, the ratio of $V_{mF,Glu\text{-}tp}$ and $V_{mR,Glu\text{-}tp}$ were set to 4.15 in the glutamate transport system, and $k_e = 1.0$ in the sinusoidal flow model.

Appendix 2: Abbreviations

CPS, carbamoyl phosphate synthetase; GS, glutamine synthetase; OAT, ornithine aminotransferase; AGS, *N*-acetyl glutamate synthetase; Glnase, phosphate-dependent glutaminase; OCT, ornithine carbamoyltransferase; ASS, argininosuccinate synthetase; ASL, argininosuccinate lyase; Argase, arginase;

Table 3. Variation parameters.

Regulation of gene expression

1. Not incorporated (N model)

2. Incorporated GS, CPS, and OAT gradients (GCO model)

3. Incorporated only GS gradients (G model)

4. Incorporated GS and CPS gradients (GC model)

5. Incorporated OAT gradients (O model)

6. Incorporated GS and OAT gradients (GO model)

Glutamate transporter

1. $V_{mF,Glu\text{-}tp}$: $V_{mR,Glu\text{-}tp} = 4.15$ ($V_{mF,Glu\text{-}tp} = 1.0629E{-}2$ M s^{-1}, $V_{mR,Glu\text{-}tp} = 2.5611E{-}3$ M s^{-1})

2. $V_{mF,Glu\text{-}tp}$: $V_{mR,Glu\text{-}tp} = 4.5$ ($V_{mF,Glu\text{-}tp} = 1.2573E{-}3$ M s^{-1}, $V_{mR,Glu\text{-}tp} = 2.7940E{-}4$ M s^{-1})

3. $V_{mF,Glu\text{-}tp}$: $V_{mR,Glu\text{-}tp} = 5.0$ ($V_{mF,Glu\text{-}tp} = 6.1467E{-}4$ M s^{-1}, $V_{mR,Glu\text{-}tp} = 1.2293E{-}4$ M s^{-1})

4. $V_{mF,Glu\text{-}tp}$: $V_{mR,Glu\text{-}tp} = 7.0$ ($V_{mF,Glu\text{-}tp} = 2.6560E{-}4$ M s^{-1}, $V_{mR,Glu\text{-}tp} = 3.7943E{-}5$ M s^{-1})

Glutamate Flux from Outside Pathways

1. $J_{Glu\text{-}spp} = 3E{-}5$ M s^{-1}, $k_{Glu\text{-}spp} = 7.0866E{-}2$ s^{-1}

2. $J_{Glu\text{-}spp} = 6E{-}5$ M s^{-1}, $k_{Glu\text{-}spp} = 8.2539E{-}2$ s^{-1}

3. $J_{Glu\text{-}spp} = 8E{-}5$ M s^{-1}, $k_{Glu\text{-}spp} = 9.0321E{-}2$ s^{-1}

Substance Flow in Sinusoid

1. $k_e = 0.5$

2. $k_e = 0.8$

3. $k_e = 1.0$

4. $k_e = 1.2$

5. $k_e = 1.6$

GOT, glutamate:oxaloacetate; GDH, glutamate dehydrogenase; GAT, arginine:glycine amidinotransferase; GAMT, guanidinoacetate methyltransferase; OTL, ornithine-citrulline translocase; GTL, glutamate translocase; GATL, glutamate-aspartate translocase; $NH4_4^+$-tp, ammonia transporter; Glu-tp, glutamate transporter; Gln-tp, glutamine transporter in mitochondrial membrane; Urea-tp, urea transporter. The entity abbreviation may be used with an index that represents the location of the entity. The indices *c*, *m*, and *s* indicate the cytoplasm, mitochondria, and sinusoid, respectively.

REVIEW

Informatics for peptide retention properties in proteomic LC-MS

Kosaku Shinoda[1, 2], Masahiro Sugimoto[1, 3], Masaru Tomita[1, 2] and Yasushi Ishihama[1, 4]

[1] Institute for Advanced Biosciences, Keio University, Tsuruoka, Yamagata, Japan
[2] Human Metabolome Technologies, Tsuruoka, Yamagata, Japan
[3] Bioinformatics Department, Mitsubishi Space Software, Amagasaki, Hyogo, Japan
[4] PRESTO, Japan Science and Technology Agency, Tokyo, Japan

Retention times in HPLC yield valuable information for the identification of various analytes and the prediction of peptide retention is useful for the identification of peptides/proteins in LC-MS-based proteomics. Informatics methods such as artificial neural networks and support vector machines capable of solving nonlinear problems made possible the accurate modeling of quantitative structure-retention relationships of peptides (including large polymers) up to 5 kDa to which classical linear models cannot be applied, as well as the proteome-wide prediction of peptide retention. Proteome-wide retention prediction and accurate mass-information facilitate the identification of peptides in complex proteomic samples. In this review, we address recent developments in solid informatics methods and their application to peptide-retention properties in 'bottom-up' shotgun proteomics. We also describe future prospects for the standardization and application of retention times.

Received: July 13, 2007
Revised: October 30, 2007
Accepted: November 1, 2007

Keywords:
Bioinformatics / Liquid chromatography-tandem mass spectrometry / Neural networks / Peptide / QSRR

1 Introduction

Liquid chromatography-mass spectrometry (LC-MS) is a powerful tool for the separation and identification of peptides in proteomics studies. While several methods and software tools are available for identifying peptides/proteins from mass spectra, the high complexity of a digested proteome (containing thousands or even millions of detectable peptides) and the vastly larger number of possible peptide sequences render accurate peptide/protein identification challenging. Consequently, proteome coverage remains limited. As the chromatographic retention times of peptides depend on their amino acid sequences, their retention times (http://iupac.org/goldbook/R05364.pdf) complement the information provided by MS and enhance their identification. Efforts to predict the chromatographic behavior of peptides span the last 50 years. In 1951, Knight [1] and Pardee [2] showed that in paper chromatography, synthetic peptide retardation factors could be predicted with some accuracy. More recently, the prediction of peptide retention times in RP [3–5] and normal-phase LC [6, 7] was reported. Most of these works used the so-called "retention coefficient" approach, which is based on the summation of empirically determined amino acid residue retention coefficients. The assumption that the chromatographic behavior of peptides is linearly dependent on their amino acid composition holds up fairly well for small peptides (up to 15–20 residues), but is

Correspondence: Dr. Yasushi Ishihama, Institute for Advanced Biosciences, Keio University, Tsuruoka, Yamagata 997-0017, Japan
E-mail: y-ishi@ttck.keio.ac.jp
Fax: +81-235-29-0536

Abbreviations: ANN, artificial neural networks; **GA**, genetic algorithms; **NET**, normalized elution time; **SMLR**, stepwise multiple linear regressions; **SVM**, support vector machines; **SVR**, support vector regressions; **QSRR**, quantitative structure-retention relationships

inadequate for proteomic applications, e.g. those that involve tryptic peptides, where the practical upper limit can exceed 50 amino acid residues [8, 24]. Furthermore, with the retention coefficient approach, isomeric peptides are predicted to elute at the same time, which, in fact, is not the case [9–11]. Another prediction method is based on machine learning methods such as artificial neural networks (ANN) and support vector machines (SVM). Machine-learning techniques capable of solving nonlinear problems [12-23] have been used to model the quantitative structure-retention relationships (QSRR) of various analytes in liquid chromatography [12, 21, 23]. In 2003, Petritis et al. [8] introduced an ANN-based method for predicting peptide retention times that was originally based on amino acid composition. Later they extended it to include partial amino acid sequence information [24]. Shinoda et al. [25] combined ANN and stepwise multiple linear regressions (SMLR) to predict peptide-retention times based on selected amino acid descriptors with statistically significant effects on LC retentions. Liu et al. [26] applied an SVM to develop predictive models between the retention factor (log k, http://iupac.org/goldbook/R05359.pdf) and seven peptide molecular constitutional and topological descriptors. These reports confirmed the usefulness of machine learning in peptide-retention predictions especially for longer peptides and several papers applied these techniques to peptide/protein identifications. However, machine learning involves both use and abuse in each step of model development, performance assessment, and application.

This article reviews the strategies, current progress, and underlying difficulties involved in the application of machine-learning methods to the prediction of peptide retentions and examines their application to peptide identification in proteomic studies.

2 Descriptors for peptides

Improving the peptide-retention time prediction in HPLC requires an understanding of the various factors affecting peptide retention behaviors. These factors have been thoroughly investigated [24], and it is now widely accepted that the retention behavior of peptides in HPLC is governed by (i) the amino acid composition [3–5], (ii) the peptide length (or mass) [3, 27, 28], and (iii) sequence-dependent effects [29–40] that can be further divided into nearest-neighbor and conformation effects, where the former are defined as amino acid sequence-dependent but independent of peptide conformation [40]. Krokhin et al. [41] applied separate retention coefficients for amino acids at the N terminus of the peptide in addition to the peptide length, further improving the retention-coefficient model. Using SVM, Liu et al. [26] adopted seven peptide molecular constitutional and topological descriptors (i.e. number of single bonds, number of rings, etc.) to predict the retention factors (log k). Kaliszan and coworkers [42, 43] used QSRR to predict peptide-retention times. Descriptors to derive the necessary QSRR included the logarithm of (i) the sum of the retention times of the amino acids that make up the peptides, (ii) the van der Waals volume of the peptide, and (iii) the peptide-calculated 1-octanol-water partition coefficient. Makrodimitris et al. [44] applied a mesoscopic simulation using Langevin dipoles on a lattice with calculated solute partial charges to estimate the free energies of the adsorption of peptides in RP chromatography. Their method is efficient and yields quantitative predictions of retention orders of peptides covering a wide range of structures. Petritis et al. [24] investigated several peptide descriptors such as peptide length, sequence, hydrophobicity/hydrophobic moment, and nearest-neighbor amino acid, as well as peptide-predicted structural configurations (i.e. helix, sheet, coil). They developed several ANN models with various combinations of these descriptors and empirically assessed the significance of tested descriptors. They found that ANN with a 1052-24-1 architecture, whose input layer consists of encoded peptide sequence information (21 * 25 * 2, amino acids, maximum length, and C/N termini, respectively), peptide length, and hydrophobic moments, yielded the best prediction accuracy.

As outlined above, a number of descriptors have been introduced to represent a peptide; most reported studies typically use only a portion of these descriptors. In other omics applications such as DNA microarray, the selection of a proper subset of descriptors is useful for improving the performance of machine learning methods [45–50]. Moreover, the indiscriminate use of existing descriptors, particularly of overlapping and redundant descriptors, may introduce over-fitting for a particular subset of observable data and deteriorate the versatility of the method. Therefore, there is a need to explore varied combinations of descriptors and to select more optimal sets of descriptors for more cases. This process should not require manual efforts by experts with a deep understanding but rather, it should make use of automatic feature selection methods. For example, Shinoda et al. [25] utilized SMLR to select 16 significant descriptors from 20 amino acids to develop ANN while Tham et al. [23] applied genetic algorithms (GA) to select molecular descriptors of retention times in RP-HPLC. Efforts have also been directed at improving the efficiency and speed of feature selection methods [51] that will facilitate their more extensive application. Thus, it may be necessary to introduce new descriptors for models that have been described by overlapping and redundant descriptors.

3 Machine-learning methods

Below we describe the concepts and characteristics of representative machine-learning methods including ANN, SVM, and GA. Freely and commercially available solutions of these methods have been reported by Berrueta et al. [52] and are listed in Table 1. The characteristics of each method are summarized in Table 2.

Table 1. Websites that contain downloadable codes of machine learning methods

	URL	License	Platform
ANN			
Libneural	http://ieee.uow.edu.au/~daniel/software/libneural/	Free (LGPL)	UNIX (GNU/Linux, FreeBSD, NetBSD or OpenBSD) and Cygwin
FANN	http://leenissen.dk/fann/	Free (LGPL)	UNIX/Windows
Weka	http://www.cs.waikato.ac.nz/ml/weka/	Free (GPL)	Cross-platform (Java)
NeuralWorks Predict	http://www.neuralware.com/products.jsp	Commercial	Windows/UNIX
NeuroShell Predictor	http://www.mbaware.com/neurpred.html	Commercial	Windows 95-XP
BrainMaker	http://www.calsci.com/	Commercial	Windows XP, 2000 and Me
JMP	http://www.jmp.com/software/jmp.shtml	Commercial	Windows/MacOS X/Linux
SVM			
SVM light	http://svmlight.joachims.org/	Free for non-commercial use	PowerPC Mac/UNIX/Windows
LIBSVM	http://www.csie.ntu.edu.tw/~cjlin/libsvm/	Free (the modified BSD license)	Windows/UNIX
mySVM	http://www-ai.cs.uni-dortmund.de/SOFTWARE/MYSVM/	Free for non-commercial use	Windows/UNIX
BSVM	http://www.csie.ntu.edu.tw/~cjlin/bsvm/	Free for non-commercial use	Windows/UNIX
Weka	http://www.cs.waikato.ac.nz/ml/weka/	Free (GPL)	Cross-platform (Java)
MATLAB SVM Toolbox	http://theoval.sys.uea.ac.uk/svm/toolbox/	GPL (Matlab is Commercial)	Windows/UNIX/Mac OS X
GA			
AI::Genetic (CPAN module)	http://search.cpan.org	Free (GPL)	UNIX (Solaris, Linux, FreeBSD, NetBSD or OpenBSD) and Cygwin
GAlib	http://lancet.mit.edu/ga/	Free (GPL)	UNIX (Linux, MacOSX, SGI, Sun etc)/Windows/Mac OS
genalgy	http://hobbiton.thisside.net/genetic/	Free	Python code
JGAP	http://jgap.sourceforge.net/	Free (LPL or MPL)	Cross-platform (Java)
Weka	http://www.cs.waikato.ac.nz/ml/weka/	Free (GPL)	Cross-platform (Java)
Genetic Algorithm and Direct Search Toolbox Matlab	http://www.mathworks.com/products/gads/	Commercial	Windows/UNIX/Mac OS X

Table 2. Brief comparisons of machine learning methods described in this review

	Accuracy for nonlinear problems	Model interpretability	Preferable dataset size	Generalization ability	Possibility of over-fitting
MLR	–	++	$> X^{a)}+1$	Low	None
ANN	++	+	$> X*5.0^{b)}$	High	High
SVM (SVR)	++	–	$> X*2.0^{c)}$	High	Low

a) X is the number of variable.
b) Rough requirement. The preferable size strongly depends on the number of hidden nodes.
c) Rough requirement. The preferable size depends on the kernel function.

3.1 Artificial neural networks

An ANN is a generic designation for connectionist-approach-based data modeling tools inspired by the biological nervous system. ANN can be used to detect underlying relationships between inputs and outputs or to find patterns in data. Compared to classical statistical methods, ANN-based approaches offer advantages that include a capacity to self-learn and to model complex data without the need for a detailed understanding of the underlying phenomena.

Among various types of ANN, a multi-layer perceptron is the most common algorithm; it has been widely used for peptide retention predictions [8, 24, 25, 53, 54]. It is composed of a large number of neurons, nodes, or processing elements organized into a sequence of layers. As shown in Fig. 1, nodes in any layer can be fully or partially connected to nodes of a succeeding layer; each hidden or output node receives signals in parallel. The input signal to a node is modulated by a weight (w) along each link between nodes. The net input to a node is thus a function of all signals to the node and all of its associated weights. For example, the net input for a node j is given by:

$$net_j = \sum_i w_{ji} O_i \quad (1)$$

where i represents nodes in the previous layer, w_{ji} is the weight associated with the connection from node i to node j, and O_i is the output of node i. The process of adapting the weights to an optimal set of values is called "training" the neural network. For this, several training algorithms are available; the back-propagation (backwards propagation of errors) algorithm illustrated in Fig. 2 is the most popular [55]. The net inputs were transferred to the neuron using a transfer function. Several transfer functions are available, satisfying a requirement of differentiability set by the back-propagation algorithm. The most popular is the logistic function given by:

$$O_j = \frac{1}{1 + e^{-net_j}} \quad (2)$$

Overall, the structure of a multi-layer perceptron contains at least three layers, *i.e.* an input layer with one node for each variable in a data vector and an output layer consisting of one node for each variable to be investigated. Additionally, one or more hidden layers can be added between the input and output. Funahashi [56] previously demonstrated that a single

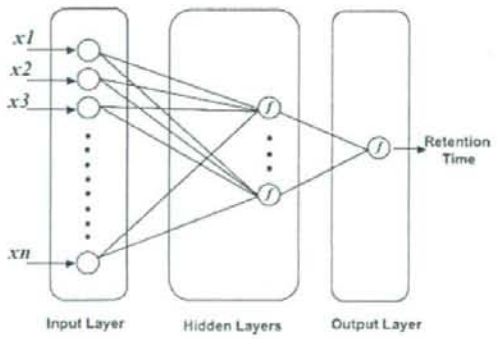

Figure 1. Schematic representation of the artificial neural network architecture. The circles represent input vectors. The small black circles show continuance.

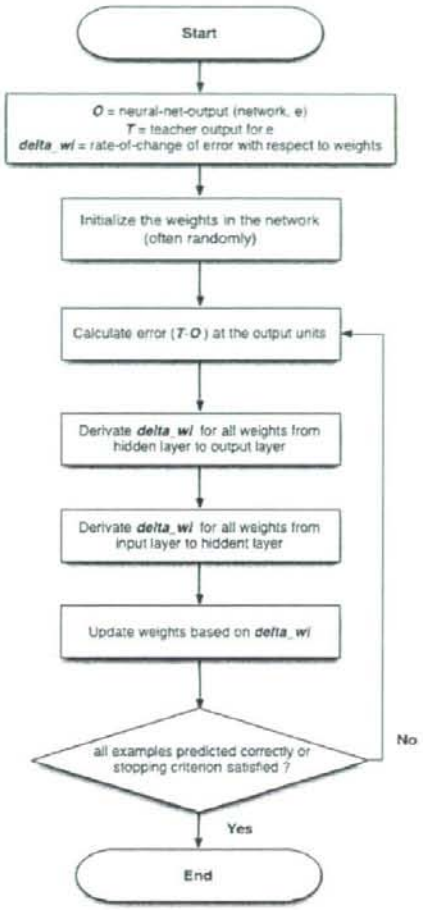

Figure 2. Algorithmic representation of back propagation (back propagation of errors).

hidden layer could approximate any function. Without hidden layers, a neural network with logistic transfer function is identical to logistic regression widely used in statistical modeling. In essence, the application of these equations to nodes in the hidden and output layers allows these ANN to perform multivariate nonlinear regression using a logistic function. Due to the parallel processing of nodes within each layer, these ANN can learn multivariate nonlinear functions.

3.2 Support vector machine

The support vector machine developed by Vapnik *et al.* [57–60] as a novel type of machine-learning method is gaining popularity due to its many attractive features and promising

empirical performance. Compared to traditional ANN, SVM features the following prominent advantages: (i) a strong theoretical background provides SVM with a high generalization capability and can avoid local minima, *i.e.* it has the ability to accurately predict for new data, (ii) SVM always reaches a solution that can be quickly obtained by a standard linear optimization algorithm (quadratic programming), (iii) SVM does not need to determine network topology in advance, rather, it can be automatically obtained at the end of the training process, and (iv) SVM builds a result based on a sparse subset of training samples, thereby reducing the workload. Originally, SVM was developed to solve pattern recognition problems; it is now used for microarray gene expression classification [61], protein folding recognition [62], protein structural class prediction [63], identification of protein cleavage sites, and other pharmaceutical data analyses [61, 64]. Vapnik [58] extended this algorithm for solving regression problems by choosing a suitable cost function (ε-insensitive loss function) that facilitates the acquisition of a sparse set of support vectors (support vector regressions, or SVR). Although SVR has been used in the prediction of chromatographic behavior such as log *k* of peptides [26] and of protein retentions in anion-exchange- [65] and hydrophobic interaction chromatography [66], proteome-wide applications of SVR have just begun. The basic concept of SVM has been described and illustrated in clearly understandable terms by Noble [67].

3.3 Genetic algorithm

The GA [68] is an algorithm based on evolutionary computation and survival of the fittest; it is often applied to optimization problems such as optimizing the free variables in a hypothesis function [69]. Solutions to problems are coded as genes (= abstract representations of variables to be optimized) in each individual (= candidate solutions). Traditionally, genes are represented in binary form as strings of 0 and 1; other encodings (*e.g.* real number) are also possible. At first, a number of individuals are initialized with randomly generated genes to form a "population". The fitness of each individual in the population is evaluated by a user-specified fitness function; multiple individuals are stochastically selected from the current population based on their fitness and modified (recombined by crossover operation and randomly mutated) to form a new population. This iterative process, called a "generation", incrementally refines the best solution, the fittest individuals. The GA performs a global search that avoids local minima; many parameters can be estimated simultaneously [70]. In terms of practical applications, the GA was used to select molecular descriptors of retention times in RP-HPLC [23], to optimize numeric parameters of normalization functions [8], and to optimize scoring functions for protein identification [23]. The basic concept and theory as well as influences of parameters empirically tuned by users of GA are described by Leardi [71].

4 Assessment of the performance of predictive models

In using machine learning, a fundamental question is how best to assess the performance of predictors [77]. One way is to obtain a test set of further observations from the same population and to compare the observed values in this set with their predictions from the model using a single criterion measure such as the sum of squared errors. When no test set is available, we need to base assessments on training-set data only. The simplest way is resubstitution, *i.e.* comparing predictions for individual data in the training set with their counterparts. However, this will give optimistic assessments, because predictors perforce find predictive values closest to the values in the training set. Such close matching will not occur for independently gathered data. A favored alternative is cross-validation [72, 73]. Here the training data are divided into *g* equal-sized groups and *g* separate operations are conducted. Each group is omitted in turn from the data, the model is fitted to the remaining $(g - 1)$ groups, and the predictions are obtained for the omitted group. This yields n (= the number of individual data in the whole training set) predictions, none of which used the corresponding training data as part of the modeling stage. Therefore, the performance assessments formed from these predictions should not be optimistically biased. As the number of individuals in each omitted group is $k = n/g$, this method of assessment is termed *leave-k-out*. Leave-*k*-out cross-validation was popularized in the bioinformatics and chemoinformatics areas, where it is often termed "*g*-fold" cross-validation. Theoretical and computational investigations have been conducted into the influence of *k* on the results. Shao [74] established that consistency improves as *k* increases and Altman and Leger [75] reached a similar conclusion with respect to asymptotic optimality. A complication arises because there are $n!/(g!(k!)^g)$ ways of dividing the training set into *g* groups each with size *k*, and different partitions may yield different performance assessments. One solution is to average criterion measures over different partitions to arrive at an overall assessment.

Machine learning-based predictive models often depend on parameters that can only be optimized (estimated) through data-based inspection. For example, SVM reformulates the model in terms of a user-specified parameter ε that controls how closely the function will fit the training data but requires a prior determination of ε before fitting the model [76]. ANN has more varied empirically tuned parameters such as the learning rate, momentum, and number of hidden nodes; these parameters must be selected in anticipation of learning. The GA is more complicated; various parameters such as the number of generations and populations, and the mutation- and crossover rate should be determined empirically. Such parameter selection can be performed using cross-validation. For example, the number of hidden nodes to include in the ANN model can be chosen as the number that yields the lowest predictive error when successively fitting 1, 2, 3, 4 nodes, and so on. This process is called tuning

[77]. Although assessment of the performance of tuned models on test data is the best approach, what is to be done in the absence of a test set? The sum of squared errors for the chosen model is clearly an optimistically biased assessment because the model has been chosen to give the lowest errors on the training data. For unbiased assessment, we need a second layer of cross-validation: leave out each group of individuals in turn, use cross-validation on the remaining individuals to both tune and fit the model, and then make predictions for the omitted individuals using the fitted model. This process is called *two-deep* as opposed to the *one-deep* cross-validation described earlier. The necessity for two-deep cross-validation has been stressed by Ganeshanandam and Krzanowski [78] whenever predictive models are constructed by optimizing cross-validation error rates, and by Krzanowski [79] whenever the selection of variables is based on cross-validated error rates. Despite such warnings, there is often still a reluctance to use two-deep validations. Many papers on the application of machine learning in proteomic studies continued to perform one-deep assessment of error of a predictive model (Table 3); a few appropriately used two-deep cross-validations [21, 25]. Representative results from two-deep assessments of ANN are shown in Fig. 3.

The aim of cross-validation is to mimic the prediction for *future* individuals from the population. This will be achieved if the training data fully represent the sample space and each omitted individual can lie anywhere in this space. Large samples and small dimensionality generally satisfy these requirements. With small samples and high dimensionality, the training data are likely to fall in a very small fraction of the sample space (the "curse of dimensionality": [80]), and any omitted group from the training set will only come from this restricted area. Cross-validation may therefore fall far short of replicating the conditions of a test set, consequently, as dimensionality increases, the method may become less reliable. Therefore, as stated above, there is a need to explore different combinations of descriptors and to discriminate more optimal subsets of descriptors; this can be done by using feature selection methods [45, 46, 49].

5 Application to peptide and protein identifications

There are several studies on the prediction of the LC retention time of tryptic peptides for protein identification. In 2002, Palmblad *et al.* [81] first showed that retention time prediction could be combined with PMF to improve protein identification in proteomic experiments; however, their peptide retention time prediction error was high, presumably

Table 3. Performance of machine learning methods for predicting peptide retentions as reported in the literature

Machine learning	Peptide descriptors	Peptide types	Number of peptides in dataset	Validation method	Reported prediction accuracy		Ref.
					Correlation coefficient (R) or R-squared	Error rate	
ANN/ SMLR	Amino acid composition	LysC-digested peptides	834	Ten-fold two-deep CV	0.928 (R-squared)	<4%	[25]
ANN	Amino acid composition	Tryptic peptides	7080	One-deep CV	-	<3%	[8]
ANN	Peptide sequence, length, hydrophobic moment	Tryptic peptides	345914	One-deep CV	0.967 (R)	<3%	[24]
SVM	Average complementary information content, relative number of single bonds, relative number of S and N, average information content (order 0/2) and number of rings	Enzymatic digestion (trypsin and lysyl endopeptidase) of purified proteins	75	Independent evaluation	0.9801 (R)	0.1523 (in log of retention factor)	[26]
SVM	Number of histidines, histidine pairs and arginines/isoelectric point of peptides sequence	Synthesized peptides	Several hundreds	Bootstrap	0.85 (R-squared)	-	[108]

All data and results shown were collected from the original papers. The reported prediction performances must be interpreted cautiously because they are dependent on factors such as the datasets used and the choice of parameters.

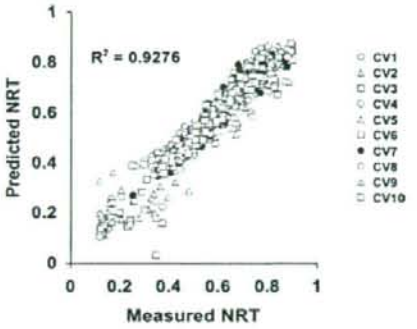

Figure 3. Scattergram of the correlation between experimentally measured and predicted normalized LC retention times (NRT) for all 834 peptides through ten-fold two-deep cross-validations (CV) from [25]; permission was obtained from the authors. NRT was predicted using ANN. Parameters for ANN (training ratio, momentum, and random numbers for initial ANN weights) were tuned through ten-fold CVs using 90% of 834 peptides and performance of the tuned model was evaluated using the remaining 10%.

because of limitations of the retention coefficient approach they used. Smith and co-workers [82–85] reported an accurate mass and time (AMT) tag proteomics approach that applies accurate mass measurements in conjunction with observed peptide-retention time information to identify peptides more confidently. Le Bihan *et al.* [86] used peptide-retention time prediction parameters to build a model for predicting peptides that are likely to be observable by LC-MS/MS; their model was employed for the targeted MS identification of low-abundance proteins in complex protein samples. Kawakami *et al.* [87] developed a program that validates peptide assignments based on the correlation between the measured and predicted LC retention time of each peptide. Norbeck *et al.* [54] demonstrated how accurate mass- and normalized elution time (NET) information improved peptide identifications in the study of proteomes of high complexity. Such improvements can significantly extend the protein coverage of highly confident peptide identifications. When peptide-retention time prediction was combined with peptide/protein identification programs such as SEQUEST and MASCOT in various applications, the number of false-positive identifications could be decreased [88–90].

In using peptide retention information for identification, comparing multiple LC-MS/MS runs or matching observed and predicted retention times is challenging because small changes in the split ratio, column lengths, column packings, void volumes, etc. unavoidably lead to some retention-time variability. In addition, noncontiguous retention data obtained with different LC-MS systems or by different laboratories must be aligned to confirm and utilize established proteome data. A widely used approach to the chromatographic-alignment problem fits a piece-wise linear function to maximize the correlation between the samples. Methods of this kind are often characterized as correlation optimized warping (COW) [91] and several derivative methods were investigated [92]. In principle, this approach can be extended to aligning multi-dimensional data. However, the handling of proteomic data is extremely difficult because the data are typically characterized by a very large input dimension (*i.e.* tryptic peptides). Thus, more sophisticated alignment algorithms are needed to extract higher quality information from large-scale LC-MS-based experiments. A number of approaches has been developed and used in high-throughput proteomic applications that rely on combined results obtained from different experimental platforms. For example, in the AMT approach [53, 54, 83, 93, 94], results from different LC-MS or MS/MS data sets are combined by finding the transformation functions of mass and retention times that are required to remove variability in mass and retention-time measurements between analyses. An alternative approach by Radulovic *et al.* [95] developed a software suite that bins peaks from MS scans by m/z bins and uses signal-processing algorithms to discover peaks in the chromatographic data, which contain pixels for identified peaks. Pamphlets from different experiments are aligned by using a 2-D smoothing spline function in the m/z and time dimensions to correct for m/z and time drift. Listgarten *et al.* [96] described a method to concurrently align multiple datasets by using a continuous profile model (CPM), a generative model in which each observed time series is a non-uniformly sub-sampled version of a single latent trace, to which local scale transformations are applied. Another proposed pipeline utilizes dynamic programming (DP) [97, 98]. Prakash *et al.* [97] performed DP-based alignment using a score that assumes the similarity of intensity profiles of mass spectra in different LC-MS analyses. Multiple analyses are combined in a progressive strategy of aligning and merging datasets based on similarity.

Machine learning is also applied to develop an "intelligent" system for comparing a large number of LC/MS experiments. Petritis *et al.* [8] introduced GA to optimize the normalization function for peptide retention times. The GA was applied to >50 000 (9121 distinct) peptides identified from 687 LC/MS/MS analyses to establish a common timeline so that the same peptides' variances of NET (normalized between 0 and 1) across the different separations were minimized. The GA optimized two variables of the linear normalization function for each separation to reduce the variance function of specific peptides, *i.e.* the regressed retention times for each separation. While this generated excellent results, this normalization approach became time-prohibitive as the number of peptides used increased significantly due to the many generations (iterations) required to align all analyses [24]. To remove this limitation, Strittmatter *et al.* [82] regressed observed retention times of confidently identified peptides to predicted NET of the sequences using a quadratic function for each LC-MS run. The obtained quadratic equations were used to convert observed retention times to

