Computer Corner

Calculation of the isoelectric points of polypeptides from the amino acid composition

B. Skoog and A. Wichman
Stockholm, Sweden

The isoelectric point (p*I*) of a polypeptide is of practical importance in many separation procedures, both analytical and preparative. The p*I* is defined as the pH where the net charge of the peptide is zero. Contributing to the net charge are terminal amino acids and charged amino acid side-chains within the sequence. Therefore, by using the primary structure, a theoretical p*I* value can be calculated. In this Corner a Pascal program is presented for the calculation of the p*I* value of a known carbohydrate-free polypeptide.

The fundamental Henderson–Hasselbach relationship:

$$pH = pK_a' + \log (\text{basic species/acidic species})$$

describes the ratio of conjugate acid and base in a solution. Using this equation an ionization ratio at a specific pH can be calculated from the pK_a value.

The contribution, C_{pos}, for a positively charged amino acid residue (arg, his or lys) or the N-terminal amino acid at a certain pH (pH_{test}) is thus calculated by

$$C_{pos} = 1/(1 + \alpha)$$

where $\alpha = 10^{(pH_{test} - pK_a')}$

In the case of a negatively charged amino acid (asp, cys, glu or tyr) or the C-terminal amino acid the contribution, C_{neg}, is calculated by

$$C_{neg} = \alpha/(1 + \alpha)$$

The pK_a' values from amino acid side chains or for the amino/carboxyl end can be found in literature[1]. A net charge for every pH_{test} can thus be calculated from the number of positive and negative amino acids in the polypeptide. Using Newton–Raphson's iterative procedure it is possible to calculate the pH where the net charge is zero (p*I*). The iteration starts with two pH_{test} values (the secant method) and is repeated, while the successive approximations of p*I* converge with increasing accuracy.

The number of charged amino acids, number of chains and their terminal amino acids are input values from keyboard. Starting with the two extreme pH_{test} values 1 and 14 a p*I* usually is generated within 20 iterations (accuracy 0.0005).

TABLE I. Comparison of p*I* values generated by the Pascal program and literature values.

Protein	p*I*	
	Calculated	Literature
Human growth hormone (hGH)	5.3	5.2[2,3],5.5[4]
Asp 152 desamido hGH	5.1	5.1[3]
Glu 137, asp 152 desamido hGH	5.0	4.9[3]
20K hGH	5.6	5.4[2],5.9[5]
24K hGH (1–134 + 141–191) (plasmin digested hGH)	5.1	< p*I* for hGH[6]
Desialylated human antithrombin III*	5.3-6.0	5.5-5.8[7]
Thrombin, bovine	6.7	5.6-7.1[8] **

* p*I* interval calculated after assuming different degree of desamidation within the antithrombin III molecule.
** Values for human thrombin.

Derived p*I* values are in many cases comparable to reported values (see Table I). Discrepancies will arise due to, for instance, carbohydrate moieties or modifications of amino acid residues.

The program is written in Apple LisaPascal but will work directly on any computer with an UCSD-Pascal compiler and with only minor changes on any Pascal environment. Program code length is 3.6 kB. With only minor modifications the program can also be used to generate theoretical titration curves.

0165-9936/86/$02.00.

References

1 R. M. C. Dawson, D. C. Elliott, W. H. Elliott and K. M. Jones, *Data for Biochemical Research,* Oxford University Press, Oxford, 1974.

2 M. W. Stolar, G. Baumann, M. L. Vance and M. O. Thorner, *J. Clin. Endocrinol. Metab.*, 59 (1984) 235–239.

3 A. Crambach, R. A. Yadley, M. Ben-David and D. Rodbard, *Endocrinology (Baltimore),* 93 (1973) 848–857.

4 C. H. Li, in A. Pecile and E. E. Müller (Editors), *Growth Hormone, Proceedings of the First International Symposium, Italy, Sept. 11–13, 1967,* Exerpta Medica.

5 U. J. Lewis, J. T. Dunn, L. F. Bonewald, B. K. Scovey and W. P. Vanderlaan, *J. Biol. Chem.*, 253 (1978) 2679–2687.

6 R. A. Yadley and A. Crambach, *Endocrinology (Baltimore),* 93 (1973) 858–865.

7 B. Skoog and A. Wichman, *Studies of Antithrombin III Charge Heterogeneity, Poster presented at the XXXIIIrd Colloquium, Proteides of the Biological Fluids, Brussels, 1985* .

8 D. Malamud and J. W. Drysdale, *Anal. Biochem.*, 86 (1978) 620–647.

B. Skoog and A. Wichman are at KabiVitrum AB, Research Department, S-112 87 Stockholm, Sweden.

```
{ ================================================================== }
{ Isoelectric point determination of polypeptides                    }
{ ================================================================== }

PROGRAM Iso_point;

CONST   pKa3_arg=12.5; pKa3_asp=3.9; pKa3_cys=8.3;
        pKa3_glu=4.3;  pKa3_his=6.0; pKa3_lys=10.5;
        pKa3_tyr=10.1;
        Max_chains=5;
        Max_iterations=100;
        pH1=1;      pH2=14;        Accuracy=0.00005;

        All = 'alaargasnaspcysglngluglyhisileleulysmetpheproserthrtrptyrva

TYPE    Str3    =       String[3];
        Str80   =       String[80];
        Aminoacid   =   ARRAY [1..Max_chains] OF REAL;
        Turn        =   ARRAY [1..Max_iterations] OF REAL;
        N_values    =   ARRAY [1..20] OF REAL;
        C_values    =   ARRAY [1..20] OF REAL;


VAR     Name        :   Str80;
        Test        :   Str3;

        Num_arg, Num_asp, Num_cys,
        Num_glu, Num_his, Num_lys,
        Num_tyr, Num_chains     : INTEGER;

        N_term,
        C_term  : Aminoacid;

        Test_pH,
        Charge  : Turn;

        TermN,TermC         :   REAL;
        Number, N           :   INTEGER;
        NtermpK             :   N_values;
        CtermpK             :   C_values;
        Sum,Temp            :   REAL;



FUNCTION        Neg_charge (pK_value    : REAL;
                            pH          : Turn)       : REAL;

BEGIN
        Neg_charge:=(exp(2.3026*((pH[N])-(pK_value))))/
                    (1+(exp(2.3026*((pH[N])-(pK_value)))));
END;    (Neg_charge)


FUNCTION        Pos_charge (pK_value    : REAL;
                            pH          : Turn)       : REAL;
BEGIN
        Pos_charge:=(1/(exp(2.3026*(((pH[N])-(pK_value))))+1));

END;    (Pos_charge)


PROCEDURE Findend;

BEGIN
          NtermpK[1]:=9.9;NtermpK[2]:=9.0;NtermpK[3]:=8.8;NtermpK[4]:=9.8;
          NtermpK[5]:=10.8;NtermpK[6]:=9.1;NtermpK[7]:=9.7;NtermpK[8]:=9.8;
          NtermpK[9]:=9.2;NtermpK[10]:=9.8;NtermpK[11]:=9.7;NtermpK[12]:=9.0;
          NtermpK[13]:=9.3;NtermpK[14]:=9.2;NtermpK[15]:=10.6;NtermpK[16]:=9.2;
          NtermpK[17]:=9.1;NtermpK[18]:=9.4;NtermpK[19]:=9.1;NtermpK[20]:=9.7;
          CtermpK[1]:=2.4;CtermpK[2]:=2.2;CtermpK[3]:=2.1;CtermpK[4]:=2.1;
          CtermpK[5]:=1.7;CtermpK[6]:=2.2;CtermpK[7]:=2.2;CtermpK[8]:=2.4;
          CtermpK[9]:=1.8;CtermpK[10]:=2.3;CtermpK[11]:=2.3;CtermpK[12]:=2.2;
          CtermpK[13]:=2.1;CtermpK[14]:=2.2;CtermpK[15]:=2.0;CtermpK[16]:=2.2;
          CtermpK[17]:=2.1;CtermpK[18]:=2.4;CtermpK[19]:=2.2;CtermpK[20]:=2.3;
          Number:=Pos(Test,All);
          Number:=(Number+2) DIV 3;
          TermN:=NtermpK[Number];
          TermC:=CtermpK[Number];
END; (Findend)


PROCEDURE Check(VAR  Instr : Str3);

VAR     Count   :       INTEGER;

BEGIN
          FOR count:= 1 TO 3 DO IF Instr[count] IN ['A'..'Z'] THEN
            Instr[count] := CHR(ORD(Instr[count])+32);
END;    (Check)


PROCEDURE Input;

VAR     n       :       1..Max_chains;

BEGIN
        Write('Name of protein/peptide: ');
        Readln(Name);
        Writeln;
        Writeln('Please, enter the number of charged amino acids');
        Write('arg: ');     Readln(Num_arg);
        Write('asp: ');     Readln(Num_asp);
        Write('cys: ');     Readln(Num_cys);
        Write('glu: ');     Readln(Num_glu);
        Write('his: ');     Readln(Num_his);
        Write('lys: ');     Readln(Num_lys);
        Write('tyr: ');     Readln(Num_tyr);
        Writeln;
        Write('Number of polypeptide chains: ');
        Readln(Num_chains);
        FOR n:=1 TO Num_chains DO
        BEGIN
            Write('Chain',n:2,' N-terminal amino acid: ');
            Readln(Test);
            Check(Test);
            Findend;
            N_term[n]:=TermN;
            Write('Chain',n:2,' C-terminal amino acid: ');
            Readln(Test);
            Check(Test);
            Findend;
            C_term[n]:=TermC;
        END;
END;    (Input)


PROCEDURE       Find_charge;

VAR     i       :       1..Max_chains;

BEGIN
        Sum:=0;
        Sum:=Sum-(Num_asp)*(Neg_charge(pKa3_asp,Test_pH));
        Sum:=Sum-(Num_glu)*(Neg_charge(pKa3_glu,Test_pH));
        Sum:=Sum-(Num_cys)*(Neg_charge(pKa3_cys,Test_pH));
        Sum:=Sum-(Num_tyr)*(Neg_charge(pKa3_tyr,Test_pH));
        Sum:=Sum+(Num_his)*(Pos_charge(pKa3_his,Test_pH));
        Sum:=Sum+(Num_lys)*(Pos_charge(pKa3_lys,Test_pH));
        Sum:=Sum+(Num_arg)*(Pos_charge(pKa3_arg,Test_pH));
        FOR i:=1 to Num_chains DO
        BEGIN
                Temp:=(Pos_charge(N_term[i],Test_pH));
                Sum:=Sum+Temp;
                Temp:=(Neg_charge(C_term[i],Test_pH));
                Sum:=Sum-Temp;
                Temp:=Sum;
        END;
        Charge[N]:=Temp;
END;    (Find_charge)


PROCEDURE       Iterate;

VAR     Factor, Diff    :       REAL;

BEGIN
        N:=1;   Test_pH[N]:=pH1;        Find_charge;
        N:=2;   Test_pH[N]:=pH2;        Find_charge;
        REPEAT
            N:=N+1;
            BEGIN
               Factor:=((Test_pH[N-1])-(Test_pH[N-2]))/((Charge[N-1])-(Charge[N-2]));
               Test_pH[N]:=(Test_pH[N-1])-(Charge[N-1]*Factor);
               Find_charge;
               Temp:=(Charge[N])-(Charge[N-1]);
               Diff:=ABS(Temp);
            END;
        UNTIL (Diff)<=(Accuracy);
        Writeln;
        Writeln('Isoelectric point calculated to pH= ',Test_pH[N]:5:3);
        Writeln;
        Writeln('Accuracy= ',Accuracy:7:5,' pH units');
        Writeln('Number of iterations= ',N);
END;    (Iterate)


BEGIN   (main)
        Input;
        Iterate;
END.
```