TRANSMEMBRANE SEGMENT IDENTIFICATION PROBLEM

FOUR DISTINCT PARTS TO AN ITERATION

- **Four distinct parts to an iteration**
  - **initialization**
  - **termination-testing**
  - **work-performing branch**
  - **update**

PROBLEMS WITH ITERATIONS

- **Problems with iteration**
  - **unsatisfiable termination predicates**
  - **time-consuming nested iterations**
  - **number of steps in any one iteration**
  - **number of iterative steps in an individual**

IMPLEMENTATIONS OF ITERATION

- **Implementations of iteration**
  - **Two-argument DU ("Do Until") operator (GP-1)**
  - **Automatically defined iteration (ADIs) — Restricted iteration**
  - **Automatically defined loops (ADLs)**
  - **Iteration creation operation**

TRANSMEMBRANE SEGMENT IDENTIFICATION PROBLEM

THE 446 RESIDUES OF D3DR_MOUSE

```
MAPLSQISSH INSTCGAENS TGVNRARPHA YYALSYCALI LAIIFGNGLV   50
CAAVLRERAL QTTTNYLVVS LAVADLLVAT LVMPWVVYLE VTGGVWNFSR  100
ICCDVFVTLD VMMCTASILN LCAISIDRYT AVVMPVHYQH GTGQSSCRRV  150
ALMITAVWVL AFAVSCPLLF GFNTTGDPSI CSISNPDFVI YSSVVSFY    200
FGVTVLVYAR IYMVLRQRRR KRILTRQNSQ CISIRPGFPQ QSSCLRLHPI  250
RQFSIRARFL SDATGQMEHI EDKPYPQKCQ DPLLSHLQPL SPGQTHGELK  300
RYYSICQDTA LRHPNFEGGG GMSQVERTRN SLSPTMAPKL SLEVRKLSNG  350
RLSTSLKLGP LQPRGVPLRE KKATQMVVIV LGAFIVCWLP FFLTHVLNTH  400
CQACHVSPEL YRATTWLGYV NSALNPVIYT TFNIEFRKAF LKILSC      446
```

KYTE-DOOLITTLE HYDROPHOBICITY VALUES FOR THE 20 AMINO ACID RESIDUES

Category	Kyte-Doolittle value	One-letter code for amino acid	Amino acid	Three-letter code
Hydrophobic	+4.5	I	Isoleucine	Ile
Hydrophobic	+4.2	V	Valine	Val
Hydrophobic	+3.8	L	Leucine	Leu
Hydrophobic	+2.8	F	Phenylalanine	Phe
Hydrophobic	+2.5	C	Cysteine	Cys
Hydrophobic	+1.9	M	Methionine	Met
Hydrophobic	+1.8	A	Alanine	Ala
Neutral	–0.4	G	Glycine	Gly
Neutral	–0.7	T	Threonine	Thr
Neutral	–0.8	S	Serine	Ser
Neutral	–0.9	W	Tryptophan	Trp
Neutral	–1.3	Y	Tyrosine	Tyr
Neutral	–1.6	P	Proline	Pro
Hydrophilic	–3.2	H	Histidine	His
Hydrophilic	–3.5	Q	Glutamine	Gln
Hydrophilic	–3.5	N	Asparagine	Asn
Hydrophilic	–3.5	E	Glutamic Acid	Glu
Hydrophilic	–3.5	D	Aspartic Acid	Asp
Hydrophilic	–3.9	K	Lysine	Lys
Hydrophilic	–4.0	R	Arginine	Arg

SOME OF THE 246 IN-SAMPLE FITNESS CASES

Protein	Length	Number of TM domains	Length of chosen TM domain	Location of the chosen TM domain	Length of chosen non-TM segment	Chosen non-tTM area
3BH1_MOUSE	372	2	19	287–305	19	330–348
3BH3_MOUSE	372	2	19	287–305	19	330–348
5HT3_MOUSE	487	4	20	465–484	20	385–404
5HTE_MOUSE	366	7	25	24–48	25	235–259
A2AB_MOUSE	455	7	24	411–434	24	277–300
A4_MOUSE	770	1	24	700–723	24	736–759
ACE_MOUSE	1312	1	17	1265–1281	17	625–641

4 OUTCOMES FOR THE TRANSMEMBRANE SEGMENT IDENTIFICATION PROBLEM

$$\boxed{N_{fc} = N_{tp} + N_{tn} + N_{fp} + N_{fn}}$$

CORRELATION

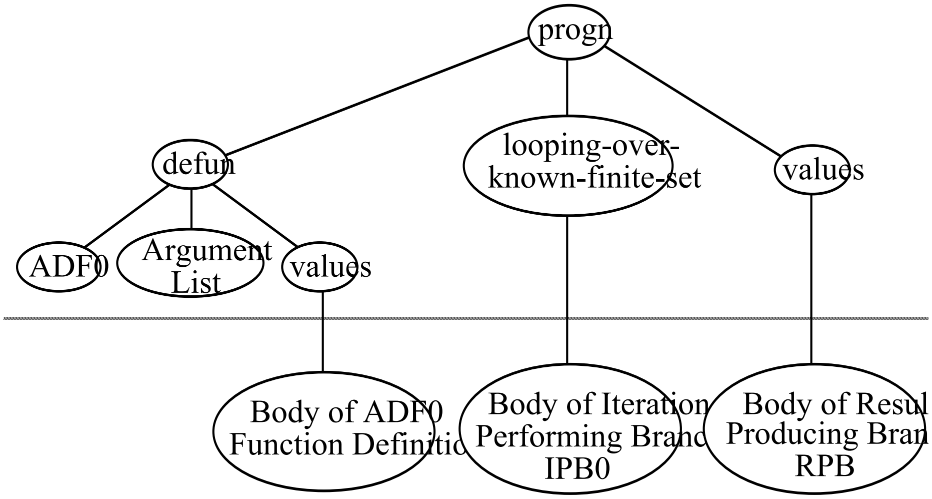
$$\boxed{C = \frac{\sum_j \left(S_j - \overline{S}\right)\left(P_j - \overline{P}\right)}{\sqrt{\sum_j \left(S_j - \overline{S}\right)^2 \sum_j \left(P_j - \overline{P}\right)^2}}}$$

$$\boxed{C = \frac{N_{tp} N_{tn} - N_{fn} N_{fp}}{\sqrt{\left(N_{tn} + N_{fn}\right)\left(N_{tn} + N_{fp}\right)\left(N_{tp} + N_{fn}\right)\left(N_{tp} + N_{fp}\right)}}}$$

STANDARDIZED FITNESS

$$\boxed{\frac{1-C}{2}.}$$

OVERALL PROGRAM FOR THE TRANSMEMBRANE SEGMENT IDENTIFICATION PROBLEM CONSISTING OF AN AUTOMATICALLY DEFINED FUNCTION, ADF0, AN ITERATION-PERFORMING BRANCH, IPB0, AND A RESULT-PRODUCING BRANCH, RPB


progn
defun
looping-over-known-finite-set
values
ADF0
Argument List
values
Body of ADF0 Function Definitio
Body of Iteration Performing Branc IPB0
Body of Resul Producing Bran RPB

RESTRICTED ITERATION

```
1 (loop initially (progn (setf M0 0.0)
                         (setf.M1 0.0)
                         (setf M2 0.0)
                         (setf M3 0.0))
2       for residue-index from 0
         below (length protein-segment)
3       for residue =
         (aref protein-segment
               residue-index)
4       do (eval IPB0)
5       finally (return
                (wrapper (eval RPB))))
```

TABLEAU WITH ADFS

Objective:	**Find a program to classify whether or not a segment of a protein sequence is a transmembrane domain.**
Architecture of the overall program with ADFs:	**One result-producing branch, one iteration-performing branch, and three zero-argument function-defining branches, with no ADF hierarchically referring to any other ADF.**
Parameters:	**Branch typing for the three ADFs.**
Terminal set for the `IPB`:	**`LEN`, `M0`, `M1`, `M2`, `M3`, and the random constants $\Re_{\text{bigger-reals}}$.**
Function set for the `IPB`:	**`ADF0`, `ADF1`, `ADF2`, `SETM0`, `SETM1`, `SETM2`, `SETM3`, `IFLTE`, `+`, `-`, `*`, and %.**

Terminal set for the result-producing branch:	**LEN, M0, M1, M2, M3, and the random constants $\Re_{\text{bigger-reals}}$.**
Function set for the result-producing branch:	**IFLTE, +, -, *, and %.**
Terminal set for the function-defining branches ADF0, ADF1, and ADF2:	**Twenty zero-argument functions (A?), (C?), ..., (Y?).**
Function set for the function-defining branches ADF0, ADF1, and ADF2:	**Numerically valued two-argument logical disjunction function ORN.**

GENERATION 0 OF RUN 1 OF SUBSET-CREATING VERSION OF TRANSMEMBRANE SEGMENT IDENTIFICATION PROBLEM WITH ADFS

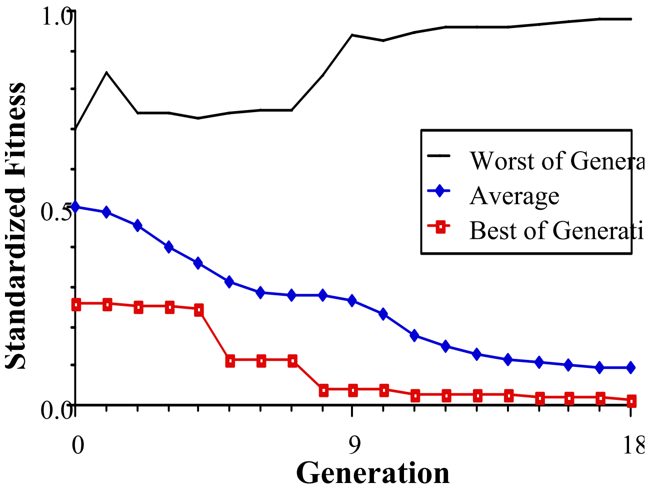
- **in-sample correlation of 0.48**
- **a standardized fitness of 0.26**
- **99 true positives**
- **83 true negatives**
- **40 false positives**
- **24 false negatives**
- **out-of-sample correlation of 0.43**

BEST OF GENERATION 0 OF RUN 1 OF SUBSET-CREATING VERSION OF TRANSMEMBRANE SEGMENT IDENTIFICATION PROBLEM WITH ADFS

```
(progn

(defun ADF0 ()

(values (ORN (ORN (ORN (I?) (M?)) (ORN (V?) (C?)))
(ORN (ORN (W?) (L?)) (ORN (Y?) (A?))))))

(defun ADF1 ()

(values (ORN (ORN (ORN (L?) (L?)) (ORN (R?) (K?)))
(ORN (ORN (I?) (V?)) (ORN (R?) (Q?))))))

(defun ADF2 ()

(values (ORN (ORN (ORN (R?) (S?)) (ORN (F?) (Q?)))
(ORN (ORN (P?) (F?)) (ORN (Y?) (C?))))))

(progn (looping-over-residues
      (SETM0 (SETM3 (SETM0 (ADF0)))))

(values (IFLTE (+ (- M3 M0) (+ M1 M3)) (% (IFLTE M0
M3 6.212 M1) (IFLTE M0 M2 M1 L)) (* (% M1 M2) (* M3
0.419)) (+ (% L M2) (- M0 M2))))))))
```

- **`IPB` only sets `M0` and `M3`. (`M1`, `M2` = 0).**
- **`IPB` only invokes `ADF0`**
- **`IPB` only sets `M0` and `M3` to `ADF0`**
- **`ADF0` returns 1 if residue is A, I, M, L, V (hydrophobic) or C, W, Y (neutral)**

FITNESS CURVES OF RUN 1 OF SUBSET-CREATING VERSION OF TRANSMEMBRANE SEGMENT IDENTIFICATION PROBLEM WITH ADFS



BEST OF GENERATION 5 OF RUN 1

- **in-sample correlation of 0.764**
- **out-of-sample correlation of 0.784**

```
(progn

(defun ADF0 ()

(values (ORN (ORN (I?) (A?)) (ORN (ORN (L?) (G?))
(N?)))))

(defun ADF1 ()

(values (ORN (ORN (ORN (ORN (G?) (D?)) (ORN (E?)
(V?))) (ORN (ORN (R?) (E?)) (ORN (T?) (P?)))) (ORN
(N?) (S?)))))

(defun ADF2 ()

(values (ORN (ORN (ORN (L?) (R?)) (ORN (V?) (P?)))
(ORN (G?) (L?)))))

(progn (looping-over-residues
       (SETM1 (- (+ M1 (ADF0)) (ADF1))))

(values (* (% (+ (% -9.997 M3) M1) 6.602) (+ 6.738 (%
(- M3 L) (+ M3 M2))))))))
```

- **`IPB` only sets `M1`**
- **`IPB` contains running sum of differences**
- **`IPB` only invokes `ADF0` and `ADF1`**
- **`ADF0` returns 1 for A, I, L (hydrophobic) or G (neutral), but N cancels (hydrophilic)**
- **`ADF1` returns 1 for D, E, R or G, P, T (neutral), but N cancels (hydrophilic)**

BEST OF GENERATION 8 OF RUN 1

- **in-sample correlation of 0.92**
- **out-of-sample correlation of 0.89**

```
(progn

(defun ADF0 ()

(values (ORN (ORN (ORN (I?) (M?)) (ORN (V?) (C?)))
(ORN (ORN (L?) (G?)) (N?)))))

(defun ADF1 ()

(values (ORN (ORN (ORN (ORN (G?) (D?)) (ORN (E?)
(V?))) (ORN (ORN (R?) (E?)) (ORN (T?) (P?)))) (ORN
(N?) (S?)))))

(defun ADF2 ()

(values (ORN (ORN (ORN (L?) (R?)) (ORN (V?) (P?)))
(ORN (G?) (L?)))))

(progn (looping-over-residues
       (SETM1 (- (+ M1 (ADF0)) (ADF1))))

(values (* (+ M1 M3) (+ 6.738 (% (- M3 L) (+ M3
M2)))))))
```

- **`IPB` only sets `M1`**
- **`IPB` contains running sum of differences**
- **`IPB` only invokes `ADF0` and `ADF1`**
- **`ADF0` returns 1 for I, L, M (hydrophobic) or C (neutral), but N, G, V cancels**
- **`ADF1` returns 1 for D, E, R (hydrophilic) or P, S, T (neutral) , but N, G, V cancels**

BEST OF GENERATION 11 OF RUN 1 OF SUBSET-CREATING VERSION OF TRANSMEMBRANE SEGMENT IDENTIFICATION PROBLEM WITH ADFS

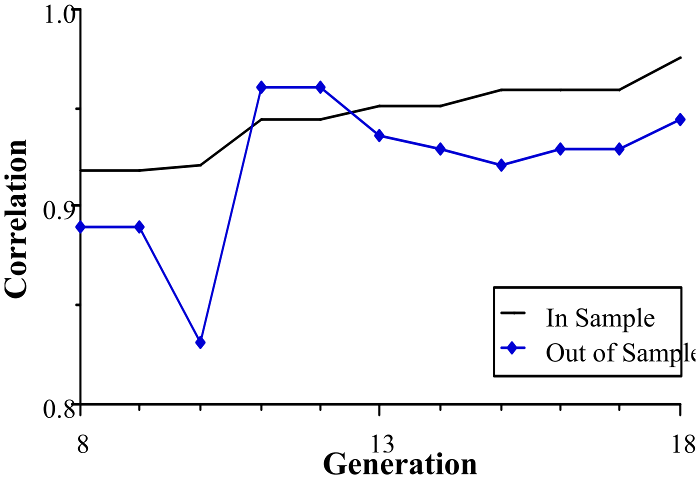
- **in-sample correlation of 0.94**
- **standardized fitness of 0.03**
- **out-of-sample correlation of 0.96**
- **122 true positives**
- **123 true negatives**
- **2 false positives**
- **3 false negatives**
- **out-of-sample error rate 2.0%**

BEST OF ENERATION 11 OF RUN 1 — CONTINUED

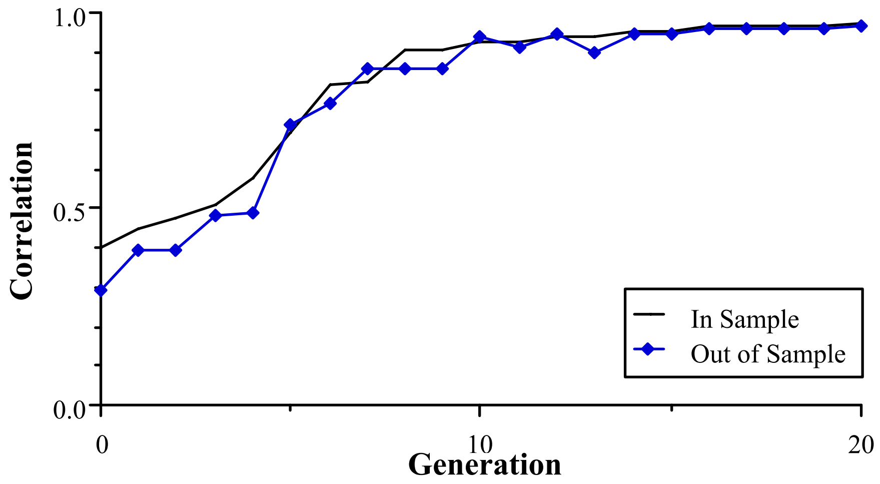
```
(progn

(defun ADF0 ()

(values (ORN (ORN (ORN (I?) (M?)) (ORN (V?) (C?)))
(ORN (ORN (L?) (G?)) (N?)))))

(defun ADF1 ()

(values (ORN (ORN (ORN (ORN (G?) (D?)) (ORN (E?)
(V?))) (ORN (ORN (R?) (E?)) (ORN (ORN (ORN (ORN (G?)
(D?)) (ORN (E?) (V?))) (ORN (ORN (R?) (K?)) (ORN (T?)
(P?)))) (ORN (N?) (S?))))) (ORN (N?) (S?)))))

(defun ADF2 ()

(values (ORN (ORN (ORN (L?) (Y?)) (ORN (V?) (P?)))
(ORN (G?) (L?)))))

(progn (looping-over-residues
       (SETM1 (- (+ M1 (ADF0)) (ADF1))))

(values (* (+ M1 M3) (+ 6.738 (% (- M3 L) (+ M3
M2)))))))
```

- **IPB only sets M1**
- **IPB contains running sum of differences**
- **IPB only invokes ADF0 and ADF1**
- **ADF0 returns 1 for I, L, M (hydrophobic) or C (neutral), but N, G, V cancels**
- **ADF1 returns 1 for D, E, K, R (hydrophilic) or P, S, T (neutral) , but N, G, V cancels**

COMPARISON OF VALUES OF IN-SAMPLE AND OUT-OF-SAMPLE CORRELATION FOR RUN 1



COMPARISON OF VALUES OF IN-SAMPLE AND OUT-OF-SAMPLE CORRELATION FOR RUN 3



STATISTICS FOR THE BEST-OF-ALL PROGRAM FROM RUN 3 FOR THE SUBSET-CREATING VERSION OF THE TRANSMEMBRANE PROBLEM

Out-of-sample statistics	**Best-of-run from generation 20 of run 3**
In-sample correlation *C*	**0.976**
Out-of-sample correlation *C*	**0.968**
Number of fitness cases N_{fc}	**250**
Number of true positives N_{tp}	**123**
Number of true negatives N_{tn}	**123**
Number of false positives N_{fp}	**2**
Number of false negatives N_{fn}	**2**
Standardized fitness	**0.16**
Hits	**98**
Accuracy Q_3	**98.4%**
Error rate	**1.6%**
Percentage of agreement c_a	**98.4%**
Percentage of overprediction c_{na}	**98.4%**

BEST OF GENERATION 20 OF RUN 3

```
(progn

(defun ADF0 ()

(values (ORN (ORN (ORN (I?) (H?)) (ORN (P?) (G?)))
(ORN (ORN (ORN (Y?) (N?)) (ORN (T?) (Q?))) (ORN (A?)
(H?))))))

(defun ADF1 ()

(values (ORN (ORN (ORN (A?) (I?)) (ORN (L?) (W?)))
(ORN (ORN (T?) (L?)) (ORN (T?) (W?))))))

(defun ADF2 ()

(values (ORN (ORN (ORN (ORN (ORN (D?) (E?)) (ORN (ORN
(ORN (D?) (E?)) (ORN (ORN (T?) (W?)) (ORN (Q?)
(D?))))) (ORN (K?) (P?)))) (ORN (K?) (P?))) (ORN (T?)
(W?))) (ORN (ORN (E?) (A?)) (ORN (N?) (R?))))))

(progn (loop-over-residues
       (SETM0 (+ (- (ADF1) (ADF2)) (SETM3 M0))))

(values (% (% M3 M0) (% (% (% (- L -0.53) (* M0 M0))
(+ (% (% M3 M0) (% (+ M0 M3) (% M1 M2))) M2)) (% M3
M0))))))
```

- **`IPB` only sets `M0` and `M3`**
- **`IPB` contains running sum of differences**
- **`IPB` only invokes `ADF1` and `ADF2`**
- **`ADF1` returns 1 for I, L (hydrophobic), but A, T, W (neutral) cancels**
- **`ADF2` returns 1 for D, E, K, N, Q, R (hydrophilic) or P (neutral) , but A, T, W (neutral) cancels**

VALUES OF OUT-OF-SAMPLE CORRELATION FOR SIX SUCCESSFUL RUNS (OUT OF 22) OF THE SUBSET-CREATING VERSION OF THE TRANSMEMBRANE PROBLEM WITH ADFS

Generation	Out-of-sample correlation
12	0.944
16	0.945
7	0.945
13	0.952
11	0.960
20	0.968

VALUES OF OUT-OF-SAMPLE CORRELATION FOR 11 RUNS OF THE TRANSMEMBRANE PROBLEM WITHOUT ADFs

Generation	Out-of-sample correlation
10	0.7124
6	0.7143
6	0.7143
12	0.8044
7	0.8044
13	0.8044
8	0.8044
3	0.8044
16	0.8054
14	0.8250
20	0.9448

TRANSMEMBRANE SEGMENT IDENTIFICATION PROBELM WITH RESTRICTED ITERATION CREATION OPERATION

PREPARATORY STEPS

INITIAL FUNCTIONS AND TERMINALS

$\mathcal{T}_{\text{initial}}$ = {$\Re$, **M0**, **M1**, **M2**, **M3**, **M4**, **M5**, **LEN**, **(A?)**, **(C?)**, **…** , **(Y?)**}

$\mathcal{F}_{\text{initial}}$ = {**+**, **-**, *****, **%**, **IFGTZ**, **ORN**, **SETM0**, **SETM1**, **SETM2**, **SETM3**, **SETM4**, **SETM5**}

POTENTIAL FUNCTIONS AND TERMINALS

$\mathcal{T}$**potential** = {**IPB0**, **IPB1**, **IPB2**, **ARG0**, **ARG1**, **ARG2**, **ARG3**}

The set of potential additional functions, $\mathcal{F}$potential, for this problem consists of

$\mathcal{F}$**potential** = {**ADF0**, **ADF1**, **ADF2**, **ADF3**}

PARAMETERS

- **Population size *M* = 64,000**
- **The percentage of operations on each generation after generation 6:**
- **85% crossovers**
- **10% reproductions**
- **0% mutations**
- **1% restricted iteration creations**
- **1% branch duplications**
- **1% argument duplications**
- **0.5% branch deletions**
- **0.5% argument deletions**
- **1% branch creations**
- **0% argument creations**

PARAMETERS - CONTINUED

- **The percentage of operations on each generation after generation 6:**
- **70% crossovers**
- **10% reproductions**
- **0% mutations**
- **6% restricted iteration creations**
- **2% branch duplications**
- **2% argument duplications**
- **2% branch deletions**
- **2% argument deletions**
- **6% branch creations**
- **0% argument creations**

THE MYOPIC PERFORMANCE OF THE BEST OF GENERATION 0 (CORRELATION OF 0.3108)

```
(setm2 (* (setm5 (setm0 (orn LEN M0))) (*
(* (setm4 LEN) (setm4 (M?))) (% (setm1
(W?)) (setm4 (V?)))))))
```

- **No iteration**
- **Classification of the entire protein segment is myopically done on the basis of just the last residue from the protein segment**

A MYOPIC ITERATION-PERFORMING BRANCH FROM GENERATION 1 (CORRELATION OF 0.4702)

- **No iteration**
- **Better correlation of 0.4702, but classification of the entire protein segment is myopically done on the basis of just the last residue from the protein segment**

AN ITERATION-PERFORMING BRANCH THAT GLOBALLY INTEGRATES INFORMATION

- **Uses memory cell `M3`**
- **Iteration-performing branch, `IPB0`, is**
  `(% (setm3 (orn (K?) M3)) (E?))`
- **`IPB0` considers `K`, `E` (hydrophilic)**
- **Result-producing branch, `RPB`, is**
  `(orn (IPB0) (L?))`
- **`L` (hydrophobic) is unhelpful**

GENERATION 2 — AN ITERATION-PERFORMING BRANCH THAT COMPUTES A CONVENTIONAL RUNNING SUM

- **Result-producing branch of first pace-setting program from generation 2 (correlation of 0.7224) is**

```
(IPB0)
```

- **Iteration-performing branch, `IPB0`, is**

```
(setm3 (+ (* (H?) (E?)) (+ (V?)
M3)))
```

- **+1 in contributed by each hydrophobic `V` residue (+4.2 on the Kyte-Dolittle scale), +1 is contributed by each residue that is neither `E` (–3.5 on the Kyte-Dolittle scale) nor `H` (–3.2 on the Kyte-Dolittle scale), and -1 is contributed by either an `E` or a `H`**

GENERATION 6 — EMERGENCE OF AUTOMATICALLY DEFINED FUNCTIONS

- **Pace-setting program from generation 6 contains both a one-argument automatically defined function as well as an iteration-performing branch**

GENERATION 8 — EMERGENCE OF MULTIPLE ITERATION-PERFORMING BRANCHES

- **First pace-setting program from generation 8 has multiple iteration-performing branches. One of these iteration-performing branches globally integrates information over the entire protein segment.**

GENERATION 11 — EMERGENCE OF COOPERATIVITY AMONG ITERATION-PERFORMING BRANCHES

- **First iteration-performing branch, `IPB0`, of second pace-setting program from generation 11 is**

```
(setm3 (+ (* (H?) (E?)) (+ (orn
(setm2 M0) (set2 (W?))) M3)))
```

- **`IPB0`, computes a running sum, `M3`. An increment of +1 is contributed by `W` (tryptophan); +1 is contributed by each residue that is neither `E` nor `H`; and -1 is contributed by either an `E` or a `H` (histidine).**
- **Second iteration-performing branch, `IPB1`, makes an additional contribution to `M3` based on `H`, `E`, and `V` (valine) as follows:**

```
(setm3 (+ (* (H?) (E?)) (+ (V?)
M3)))
```

EMERGENCE OF COOPERATIVITY AMONG ITERATION-PERFORMING BRANCHES

- **Result-producing branch is simply `(IPB1)`. The value of result-producing branch is the running sum to which +1 is contributed by each V; +1 is contributed by each W; +2 is contributed by each residue that is neither E nor H; and -2 is contributed by either an E or a H.**

GENERATION 24 — EMERGENCE OF HIERARCHY AMONG AUTOMATICALLY DEFINED FUNCTIONS

- **A pace-setting program from generation 24 has two automatically defined functions (a one-argument `ADF1` and a zero-argument `ADF3`) such that `ADF3` refers to `ADF1` (and also to `IPB1`).**

GENERATION 26 — EMERGENCE OF MULTIPLE AUTOMATICALLY DEFINED FUNCTIONS AND MULTIPLE ITERATION-PERFORMING BRANCHES

- **The pace-setting program from generation 26 has three one-argument automatically defined functions as well as two iteration-performing branches.**

BEST-OF-RUN PROGRAM FROM GENERATION 42

- **Best-of-generation program of generation 42 scores 122 true positives, 122 true negatives, 1 false positive, and 1 false negative and has an in-sample correlation of 0.9938. It has an out-of-sample error rate of 1.6%.**

- **This program has 2 one-argument automatically defined functions (`ADF0` and `ADF1`) and 2 iteration-performing branches (`IPB0` and `IPB1`) that cooperatively integrate global information about the protein segment.**

BEST-OF-RUN PROGRAM FROM GENERATION 42 – CONTINUED

- **The result-producing branch is**

```
(IPB1)
```

- **`ADF0` is**

```
(adf1 (+ (setm0 (E?))(setm4
(Q?))))
```

- **Since `ADF1` merely returns its one argument, `ADF0` returns 0 if the current residue is `E` or `Q` (glutamine) and otherwise returns –2 (as well as side-effecting the settable variables `M0` and `M4`).**

BEST-OF-RUN PROGRAM FROM GENERATION 42 – CONTINUED

- **First iteration-performing branch, `IPB0`:**

```
(setm1 (– (– (setm1 (setm1 (– (setm1 M1)
(setm3 (setm3 (% (– (I?) (R?)) (adf0
(H?))))))))) (setm3 (setm3 (% (– (+ (V?)
M3) (setm2 (+ (– (D?) (+ (V?) (setm3 (+
(orn (Y?) (* (E?) (setm5 (orn (P?)
(D?))))))(+ (setm5 (orn M0 (L?))) M3)))))
(setm3 (R?)))))) (adf0 (% (setm1 (– (–
(setm1 (setm1 (– (setm1 M1) (setm3 (setm3
(% (– (I?) (R?)) (adf0 (H?))))))))) (setm3
(setm3 (% (– (+ (V?) M3) (setm2 (+ (– (*
(setm5 (orn (P?) (R?))) (setm5 (orn (P?)
(D?)))) (L?)) (setm3 (orn (Q?) (%  M5
(V?)))))))) (setm5 (orn  M0 (L?)))))))
(setm3 (setm3 (% (– (F?) (R?))(adf0
(H?)))))))) (E?)))))))) (setm3 (setm3 (% (–
(F?) (R?))(adf0 (H?))))))))
```

- **Second iteration-performing branch, `IPB1`:**

```
(setm1 (– (setm1 M1) (setm3 (setm3 (% (–
(I?) (adf1 (* (setm0 (setm1 (orn (orn
(P?) (R?)) (– (setm1 M1) (setm3 (setm3
(ifgtz (setm4 (– (Y?) (R?))) (setm1 (Y?))
IPB0)))))))) (setm0 (* (setm0 (orn (K?)
M0)) (setm1 (orn (setm4 (setm1 (setm4
(P?)))) (Q?))))))))) (adf0 (H?)))))))
```

BEST-OF-RUN PROGRAM FROM GENERATION 42 – CONTINUED

- **Both possible avenues of communication and cooperation are employed by this program.**
  - **First, two of the six settable variables (`M0` and `M1`) are set in `IPB0` and referenced by `IPB1` (as highlighted by bold-faced type in `IPB1`).**
  - **Second, `IPB1` contains a reference to the value returned by `IPB0` (also highlighted by bold-faced type in `IPB1`).**

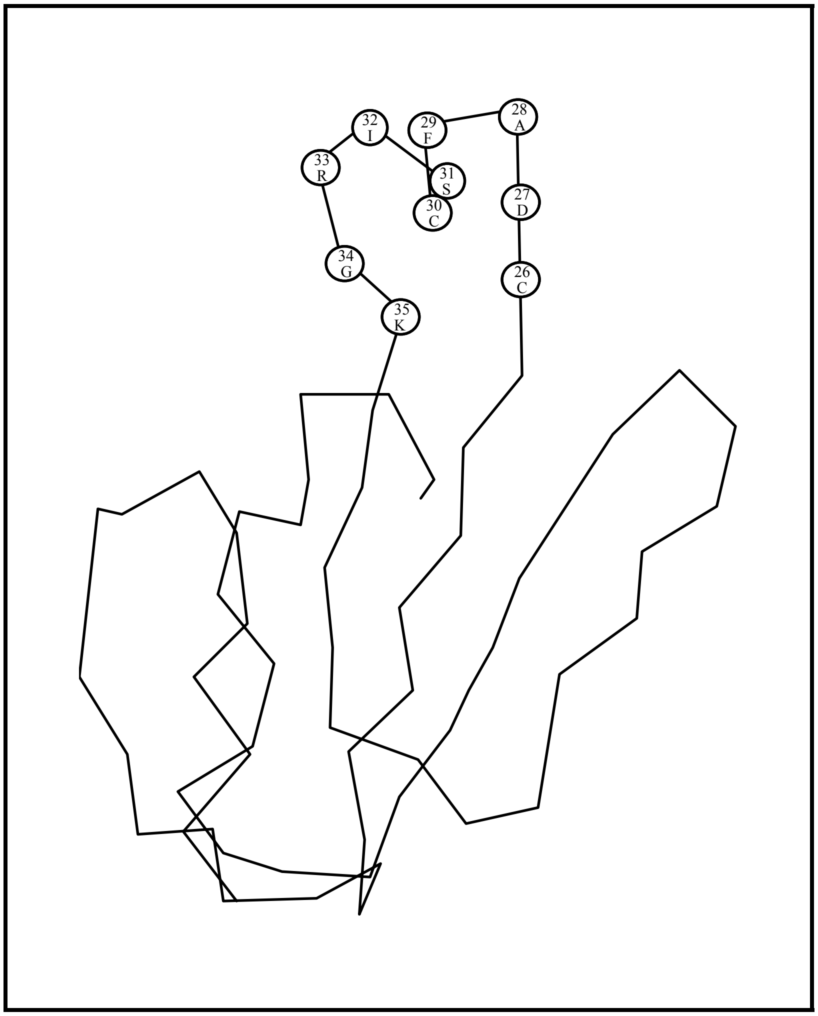
COMPARISON OF 8 METHODS FOR SOLVING TRANSMEMBRANE SEGMENT IDENTIFICATION PROBLEM

Method	Error
von Heijne 1992	2.8%
Engelman, Steitz, and Goldman 1986	2.7%
Kyte and Doolittle 1982	2.5%
Weiss, Cohen, and Indurkhya 1993	2.5%
GP + Set-creating ADFs	1.6%
GP + Arithmetic-performing ADFs	1.6%
GP + ADFs + six architecture-altering operations	1.6%
GP + ADFs + six architecture-altering operations + restricted iteration creation operation	1.6%

PRIMARY SEQUENCE OF COBRA NEUROTOXIN VENOM 1CTX

```
IRCFITPDIT SKDCPNGHVC YTKTWCDAFC SIRGKRVDLG CAATC 50

GVDIQCCSTD NCNPFPTRKR P                          71
```

1CTX WITH OMEGA LOOP AT RESIDUES 26–35



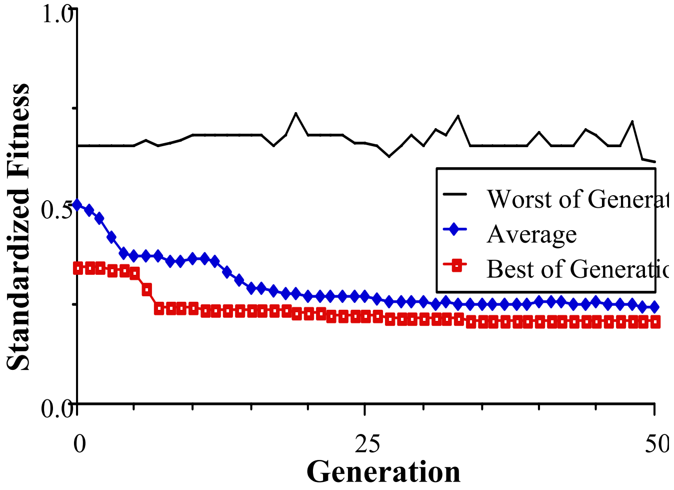
SOME OF THE IN-SAMPLE FITNESS CASES FOR THE OMEGA-LOOP PROBLEM

PDB protein code	Chain	Length of protein	Number of omega loops	Locations of omega loops (positive fitness cases)
351C		82	2	16-26, 51-62
1ABP		306	6	93-99, 142-148, 203-208, 236-248, 289-294, 299-304
2ACT		218	8	8-13, 58-64, 89-103, 139-144, 141-156, 182-192, 198-205, 203-209
1BP2		123	3	23-30, 25-39, 56-66
2BP2		130	3	30-37, 32-46, 68-75
2C2C		112	4	18-33, 30-43, 41-56, 74-89
3CNA		237	8	13-21, 97-104, 116-123, 147-155, 160-165, 199-209, 222-235, 229-237
3CPA		307	7	128-141, 142-156, 156-166, 205-213, 231-237, 244-250, 272-285

SOME OF THE OUT-OF-SAMPLE FITNESS CASES FOR THE OMEGA-LOOP PROBLEM

PDB protein code	Chain	Length of protein	Number of omega loops	Locations of omega loops (positive fitness cases)
351C		82	2	16-26, 51-62
155C		135	3	22-29, 48-55, 84-96
1ABP		306	6	93-99, 142-148, 203-208, 236-248, 289-294, 299-304
2ACT		218	8	8-13, 58-64, 89-103, 139-144, 141-156, 182-192, 198-205, 203-209
8ADH		374	5	14-21, 100-112, 115-122, 122-128, 282-287
3ADK		195	1	134-143
3APP		323	4	42-56, 130-137, 141-151, 184-192
1AZU		127	6	8-14, 34-45, 66-71, 72-82, 83-91, 111-117

FITNESS CURVES FOR ONE RUN OF THE SUBSET-CREATING VERSION OF THE OMEGA-LOOP PROBLEM



BEST OF GENERATION 11 OF RUN 1 FOR THE SUBSET-CREATING OMEGA-LOOP PROBLEM

- **in-sample correlation of 0.52**
- **out-of-sample correlation of 0.57**

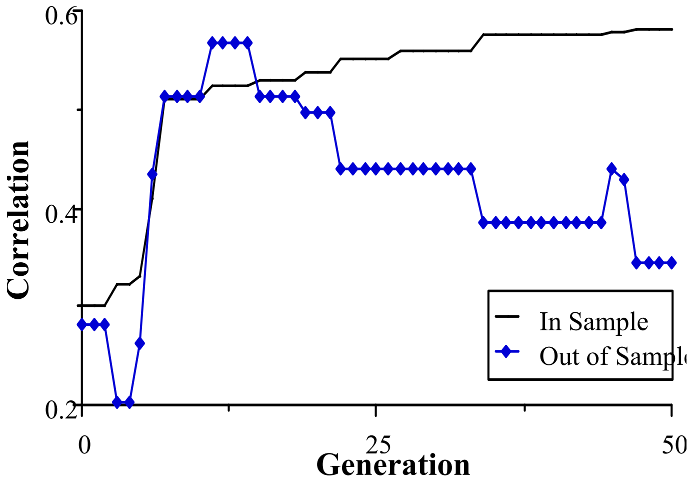
```
(progn

(defun ADF0 ()

(values (ORN (ORN (ORN (A?) (I?)) (ORN
(V?) (Q?))) (ORN (M?) (I?)))))

(defun ADF1 ()

(values (ORN (H?) (T?))))

(defun ADF2 ()

(values (ORN (ORN (N?) (W?)) (ORN (I?)
(W?)))))

(looping-over-residues (SETM0 (- (SETM1
M0)(ADF0))))

(values (+ (IFLTE M0 (+ LEN -5.805) (*
(IFLTE M0 LEN LEN M0) (IFLTE LEN 5.006
2.078 M3)) (IFLTE M1 LEN M3 M3)) (+
(IFLTE M2 M3 M2 M2) (+ (- (* (+ (IFLTE (+
5.17 M1) LEN M3 LEN) M1) (% -4.02 M2)) (-
5.654 LEN)) M1))))))
```

SIMPLIFIED VERSION OF BEST OF GENERATION 11 OF RUN 1

```
(progn
(defun ADF0 ()
(values (ORN* (A?) (I?) (V?) (Q?) (M?))))
(looping-over-residues (SETM0 (- (SETM1
M0) (ADF0))))
(values (+ (IFLTE M0 (+ LEN -5.805) (*
LEN (IFLTE LEN 5.006 2.078 0)) 0) (IFLTE
(+ 5.17 M1) LEN 0 LEN) -5.654 LEN * 2
M1)))))
```

- **`IPB` only sets `M0` and `M1`**
- **`IPB` contains running sum of differences**
- **`IPB` only invokes `ADF0`**
- **`ADF1` returns 1 for A, I, L, M (hydrophobic) and Q (hydrophilic)**
- **Q (hydrophilic) is used to balance `ADF1`**

COMPARISON OF VALUES OF IN-SAMPLE AND OUT-OF-SAMPLE CORRELATION FOR ONE RUN OF THE SUBSET-CREATING VERSION OF THE OMEGA-LOOP PROBLEM



BEST OF GENERATION 14 OF RUN 1 FOR THE ARITHMETIC-PERFORMING OMEGA-LOOP PROBLEM WITH ADFS

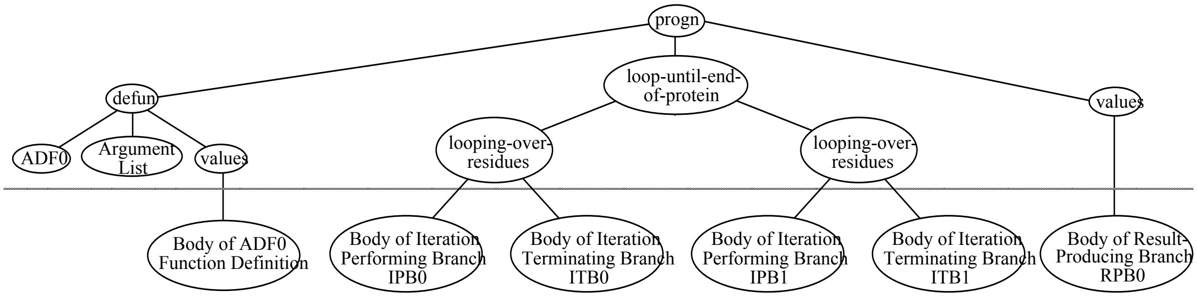
- **in-sample correlation of 0.453**
- **out-of-sample correlation of 0.449**

```
(progn  (defun ADF0 ()
             (values (IFGTZ (H?) 4.172
1.591)))
         (defun ADF1 ()
             (values (IFGTZ (ORN (ORN (R?)
(K?)) (ORN (N?) (K?))) (- (- -8.842
5.865) (% (% -6.399 3.942) (% -5.531
8.623))) (IFGTZ (ORN (E?) (F?)) (- (* -
4.17 4.843) 6.434) (% 0.798004 4.244)))))
```

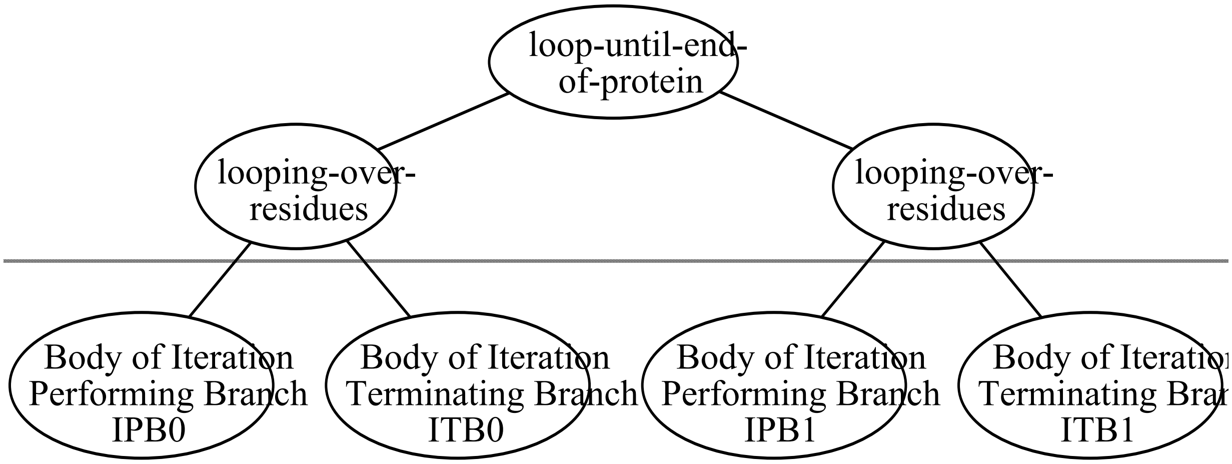
BEST OF GEN 14 OF RUN 1 – CONT

```
(defun ADF2 ()

(values (IFGTZ (ORN (ORN (M?) (I?)) (V?))
(+ (% (+ 1.726 0.0620003) (- 8.783
1.476)) (% -8.943 7.316)) (% (% -8.943
7.316) (- -7.393 2.183)))))

(progn  (looping-over-residues
    (% (SETM1 (+ M1 (ADF2)))
       (+ (* M2 M0) (* -2.037 LEN))))

(values (IFLTE (- (IFLTE (- (IFLTE 6.061
M2 M0 M1) (* M1 1.677)) (% (% M0 M0)
(IFLTE M3 M3 -7.51 0.0160007)) (% (- M0
M1) (+ M1 -5.334)) (* (IFLTE LEN 6.771
4.685 -2.358) (+ M3 LEN))) (* M1 1.677))
(% (% M0 M0) (* 9.91 M3)) (% (- M0 M1) (+
M1 -5.334)) (* (IFLTE LEN 6.771 4.685 -
2.358) (+ M3 LEN))))))
```

HYPOTHETICAL SIX-BRANCH OVERALL PROGRAM FOR LOOKAHEAD VERSION OF THE TRANSMEMBRANE PROBLEM


progn
defun
loop-until-end-of-protein
values
ADF0
Argument List
values
looping-over-residues
looping-over-residues
Body of ADF0 Function Definition
Body of Iteration Performing Branch IPB0
Body of Iteration Terminating Branch ITB0
Body of Iteration Performing Branch IPB1
Body of Iteration Terminating Branch ITB1
Body of Result-Producing Branch RPB0

FOUR-BRANCH OVERALL PROGRAM ACTUALLY USED IN THE LOOKAHEAD VERSION OF THE TRANSMEMBRANE PROBLEM


loop-until-end-of-protein
looping-over-residues
looping-over-residues
Body of Iteration Performing Branch IPB0
Body of Iteration Terminating Branch ITB0
Body of Iteration Performing Branch IPB1
Body of Iteratio Terminating Brar ITB1

LOOP-UNTIL-END-OF-PROTEIN BEHAVIOR AND THE LOOPING-OVER-RESIDUES BEHAVIOR FOR THE LOOKAHEAD VERSION OF THE TRANSMEMBRANE PROBLEM

```
(loop with residue-index = 0
      until (>= residue-index (length protein-sequence)
      do (loop initially (progn (setf M0 0.0) (setf M1 0.0)
                                (setf M2 0.0) (setf M3 0.0))
               for res from residue-index
                      below (length protein-sequence)
               for residue = (aref protein-sequence res)
               do (eval IPB0)
               until (> (eval ITB0) 0.0)
               finally (progn (mark-as-non-transmembrane
                                residue-index res)
                              (setf residue-index res)))
```

LOOP-UNTIL-END-OF-PROTEIN BEHAVIOR – CONTINUED

```
   (loop initially (progn (setf J0 0.0) (setf J1 0.0)
                          (setf J2 0.0) (setf J3 0.0))
         for res from residue-index
             below (length protein-sequence)
         for residue = (aref protein-sequence res)
         do (eval IPB1)
         until (> (eval IPT1) 0.0)
         finally (progn (mark-as-transmembrane
                          residue-index res)
                        (setf residue-index res)))
 finally (return (wrapper (compute-correlation))))
```

THE LOOKAHEAD VERSION OF THE TRANSMEMBRANE PROBLEM

$\mathcal{T}_{ipb0}$ = {**`(PHOBIC)`**, **`(PHILIC)`**, **`(NEUTRAL)`**, **`(CHARGED)`**, **`(VERY-PHOBIC)`**, **`M0`**, **`M1`**, **`M2`**, **`M3`**, $\Re_{\text{bigger-reals}}$}

$\mathcal{T}_{ipb0}$ = {**`(PHOBIC)`**, **`(PHILIC)`**, **`(NEUTRAL)`**, **`(CHARGED)`**, **`(VERY-PHOBIC)`**, **`M0`**, **`M1`**, **`M2`**, **`M3`**, $\Re_{\text{bigger-reals}}$}

THE LOOKAHEAD VERSION OF THE TRANSMEMBRANE PROBLEM

$\mathcal{F}_{ipb0}$ = {**SETM0**, **SETM1**, **SETM2**, **SETM3**, **LOOK**, **IFLTE**, **+**, **-**, *****, **%**, **ORN**}

$\mathcal{T}_{itb0}$ = {**(PHOBIC)**, **(PHILIC)**, **(NEUTRAL)**, **(CHARGED)**, **(VERY-PHOBIC)**, **M0**, **M1**, **M2**, **M3**, $\Re_{\text{bigger-reals}}$}

THE LOOKAHEAD VERSION OF THE TRANSMEMBRANE PROBLEM

$\mathcal{T}_{itb1}$ = {**`(PHOBIC)`**, **`(PHILIC)`**, **`(NEUTRAL)`**, **`(CHARGED)`**, **`(VERY-PHOBIC)`**, **`J0`**, **`J1`**, **`J2`**, **`J3`**, $\Re_{\textbf{bigger-reals}}$}

$\mathcal{F}_{itb0}$ = {**`LOOK`**, **`IFLTE`**, **`+`**, **`-`**, **`*`**, **%**, **`ORN`**}

$\mathcal{F}_{itb1} = \mathcal{F}_{itb0}$.

TABLEAU FOR THE LOOKAHEAD TRANSMEMBRANE PROBLEM

Objective:	**Find a program to classify each individual residue of a protein sequence as to whether it lies in a transmembrane domain or a non-transmembrane area.**
Architecture of the overall program:	**Two iteration-performing branches (`IPB0` and `IPB1`) and two iteration-terminating branches (`ITB0` and `ITB1`).**
Parameters:	**Branch typing.**
Terminal set for the iteration-performing branch `IPB0`:	**`(PHOBIC)`, `(PHILIC)`, `(NEUTRAL)`, `(CHARGED)`, `(VERY-PHOBIC)`, `M0`, `M1`, `M2`, `M3`, and the random constants $\Re_{\text{bigger-reals}}$.**

Terminal set for the iteration-performing branch IPB1:	(PHOBIC), (PHILIC), (NEUTRAL), (CHARGED), (VERY-PHOBIC), J0, J1, J2, J3, and the random constants $\Re_{\text{bigger-reals}}$.
Function set for the iteration-performing branch IPB0:	SETM0, SETM1, SETM2, SETM3, LOOK, IFLTE, +, -, *, %, and ORN.
Function set for the iteration-performing branch IPB1:	SETJ0, SETJ1, SETJ2, SETJ3, LOOK, IFLTE, +, -, *,%, and ORN.
Terminal set for the iteration-terminating branch ITB0:	(PHOBIC), (PHILIC), (NEUTRAL), (CHARGED), (VERY-PHOBIC), M0, M1, M2, M3, and the random constants $\Re_{\text{bigger-reals}}$.

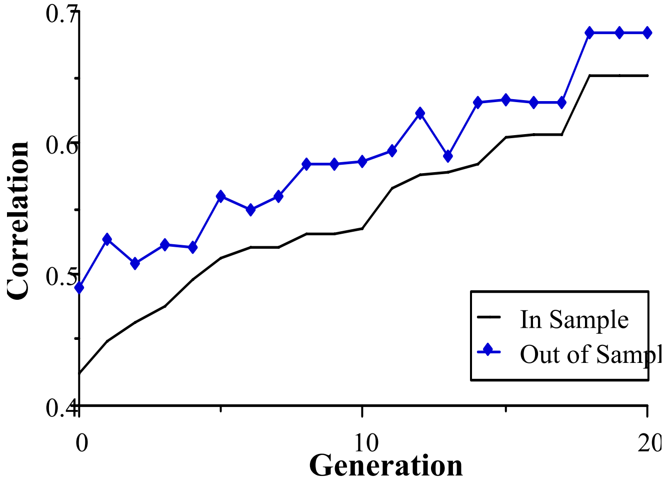
Terminal set for the iteration-terminating branch ITB1:	**(PHOBIC), (PHILIC), (NEUTRAL), (CHARGED), (VERY-PHOBIC), J0, J1, J2, J3, and the random constants $\Re_{\text{bigger-reals}}$.**
Function set for the iteration-terminating branch ITB0:	**LOOK, IFLTE, +, -, *, %, and ORN.**
Function set for the iteration-terminating branch ITB1:	**Same as ITB0.**
Fitness cases:	**Set of 22,981 in-sample residues from 47 mouse transmembrane proteins and 17,158 out-of-sample residues from 38 mouse transmembrane proteins.**
Raw fitness:	**Correlation *C* (ranging from –1.0 to +1.0).**
Standardized fitness:	**Standardized fitness is $\boxed{\frac{1-C}{2}.}$**

Hits:	**100 times the difference of 1.0 minus standardized fitness for the *out-of-sample* set.**
Wrapper:	**Labels each individual residue (fitness case) as being in a transmembrane domain or non-transmembrane area.**
Parameters:	***M* = 4,000. *G* = 21.**
Success predicate:	**A program scores an out-of-sample correlation of 1.00.**

THE 446 RESIDUES OF D3DR_MOUSE

```
MAPLSQISSH INSTCGAENS TGVNRARPHA YYALSYCALI LA  50
CAAVLRERAL QTTTNYLVVS LAVADLLVAT LVMPWVVYLE V  100
ICCDVFVTLD VMMCTASILN LCAISIDRYT AVVMPVHYQH G  150
ALMITAVWVL AFAVSCPLLF GFNTTGDPSI CSISNPDFV  Y  200
FGVTVLVYAR IYMVLRQRRR KRILTRQNSQ CISIRPGFPQ Q  250
RQFSIRARFL SDATGQMEHI EDKPYPQKCQ DPLLSHLQPL S  300
RYYSICQDTA LRHPNFEGGG GMSQVERTRN SLSPTMAPKL S  350
RLSTSLKLGP LQPRGVPLRE KKATQMVVIV LGAFIVCWLP F  400
CQACHVSPEL YRATTWLGYV NSALNPVIYT TFNIEFRKAF L  446
```

COMPARISON OF VALUES OF IN-SAMPLE AND OUT-OF-SAMPLE CORRELATION FOR THE RUN 1 OF THE LOOKAHEAD VERSION



BEST OF GENERATION 0 OF RUN 1 OF THE LOOKAHEAD VERSION OF THE TRANSMEMBRANE PROBLEM

- **in-sample correlation of 0.42**
- **out-of-sample correlation of 0.48**

```
(loop-until-end-of-protein
  (looping-over-residues
    (SETM1 (SETM0 M0))
    (+ (LOOK (VERY-PHOBIC)) (- (VERY-
PHOBIC) M1))
  (looping-over-residues
    (SETJ1 (SETJ0 (CHARGED)))
    (% (* (PHOBIC) (NEUTRAL)) (% (PHOBIC)
J2)))
```

BEST OF GENERATION 6 OF RUN 1 OF THE LOOKAHEAD VERSION OF THE TRANSMEMBRANE PROBLEM

- **in-sample correlation of 0.48**
- **out-of-sample correlation of 0.63**

```
(loop-until-end-of-protein
  (looping-over-residues
    (% (ORN (% (VERY-PHOBIC) (PHILIC))
(SETM1 (PHILIC)))
       (LOOK (% (NEUTRAL) (PHOBIC))))
(LOOK (IFLTE (ORN M1 (PHILIC)) (IFLTE M1
(PHOBIC) M2 (PHOBIC)) (LOOK (* (LOOK
(IFLTE (PHOBIC) M1 M3 (NEUTRAL))) (*
(IFLTE (PHOBIC) (NEUTRAL) M3 (PHILIC))
(ORN (PHOBIC) 2.632)))) (% M0
(CHARGED))))
```

BEST OF GENERATION 6 OF RUN 1 OF THE LOOKAHEAD VERSION OF THE TRANSMEMBRANE PROBLEM

```
(looping-over-residues
  (+ (SETJ2 (PHOBIC)) (+ J2 J0))
  (LOOK (CHARGED)))
```

BEST OF GENERATION 19 OF RUN 1 OF THE LOOKAHEAD VERSION OF THE TRANSMEMBRANE PROBLEM

- **in-sample correlation of 0.68**
- **4,121 true positives**
- **16,162 true negatives**
- **1,509 false positives**
- **1,189 false negatives over the 22,981 in-sample fitness cases**
- **out-of-sample correlation of 0.6988**
3,549 true positives
- **11,593 true negatives'**
- **1,023 false positives**
- **993 false negatives over the 17,158 out-of-sample fitness cases**
- **out-of-sample error rate of 11.7%**

BEST OF GENERATION 19 OF RUN 1 OF THE LOOKAHEAD VERSION OF THE TRANSMEMBRANE PROBLEM

```
(loop-until-end-of-protein
 (looping-over-residues
    (% (ORN (% (VERY-PHOBIC) (PHILIC)) (% (ORN (% (VERY-
    PHOBIC) (PHILIC)) (SETM1 (PHILIC))) (LOOK (% 6.636
    M3)))) (LOOK (% (LOOK (% (% (ORN (% (VERY-PHOBIC)
    (PHILIC)) (SETM1 (PHILIC))) (LOOK (% 6.636 M3)))
    (PHOBIC))) (LOOK (% 6.636 M3)))))

    (LOOK (IFLTE (ORN M1 (PHILIC)) (IFLTE M1 (PHOBIC) M2
    (PHOBIC)) (LOOK (IFLTE (ORN M1 (PHILIC)) (* (% M0
    (CHARGED)) -5.229) (LOOK (IFLTE (ORN M1 (PHILIC))
    (IFLTE (ORN M1 (PHILIC)) (* (LOOK (IFLTE (PHOBIC) M1
    M3 (NEUTRAL))) (* (% M0 (CHARGED)) -5.229)) (LOOK (*
    (LOOK (IFLTE (PHOBIC) M1 M3 (NEUTRAL))) (* (IFLTE
    (PHOBIC) (NEUTRAL) M3 (PHILIC)) (* (NEUTRAL) -
    5.229)))) (* (IFLTE (PHOBIC) (NEUTRAL) M3 (PHILIC))
    (* (NEUTRAL) -5.229))) (LOOK (LOOK (IFLTE (ORN M1
    (PHILIC)) (IFLTE (PHOBIC) M1 M3 (NEUTRAL)) (LOOK (*
    (LOOK (IFLTE (PHOBIC) M1 M3 (NEUTRAL))) (* (IFLTE
    (PHOBIC) (NEUTRAL) M3 (PHILIC)) (ORN (PHOBIC)
    2.632)))) (% M0 (CHARGED))))) (% M0 (CHARGED)))) (%
    M3 (CHARGED)))) (% M0 (CHARGED))))

      (looping-over-residues
        (ORN (SETJ2 (CHARGED))
             (+ (ORN (NEUTRAL) J1) (* J0 (CHARGED))))
        (LOOK (CHARGED)))
```

BEST VALUES OF OUT-OF-SAMPLE CORRELATION FOR FIVE RUNS OF THE LOOKAHEAD VERSION OF THE TRANSMEMBRANE PROBLEM

Run	Generation	Out-of-sample correlation	Error
1	19	0.6988	11.7%
2	20	0.6844	12.3%
3	20	0.6638	13.7%
4	17	0.6556	13.2%
5	20	0.6541	13.5%