% D0, F0, lb0, ub0 need to be initialized first. D0 is the spectral feature, with each column representing a spectrum, and the number of columns represents the number of G4. For the observation values of the selected secondary structural elements, each row represents five secondary structural elements of a sequence, and the number of rows represents the number of G4. LB0 and UB0 represent the upper and lower limits of each secondary structural element, with each row representing the upper and lower limits of five secondary structural elements in a sequence. The number of rows represents the number of G4.. When not considering the upper and lower limits of constraints, lb defaults to all 1 and ub defaults to all 0

NMRSDfit = zeros(1, size(D0, 2));

numColsD0 = size(D0, 2);

numColsfit = size(D0, 1);

x2024=zeros(numColsD0,5);

fit2024=zeros(numColsfit,5);

Aeq = [1,1,1,1,1];

beq=1;

for N = 1:size(D0, 2)

D3 = D0(:, N);

F3 = F0(N, :);

lb3 = lb0(N, :);

ub3 = ub0(N, :);

D = D0(:, 1:N-1);

D = [D, D0(:, N+1:end)];

F = F0(1:N-1, :);

F = [F; F0(N+1:end, :)];

lb1 = lb0(1:N-1, :);

lb1 = [lb1; lb0(N+1:end, :)];

ub1 = ub0(1:N-1, :);

ub1 = [ub1; ub0(N+1:end, :)];

abc333\_accum = zeros(size(D0, 1), 5); % Assuming there are 5 columns that need to be fitted

for L=1:5;

D1 = D;

F1 = F;

y = abs(F(:, L));

maxE = -inf;

maxMatrix = [];

R = NaN; xabcd = [];

previousMaxE = -inf;

numColsD1 = size(D1, 2);

columnToDelete = 0;

RMSDbianhua = zeros(min(80, numColsD1), 1);

while numColsD1 > 1

for Z = 0:numColsD1-1

D2 = D1(:, 1:Z);

D2 = [D2, D1(:, Z+1:end)];

F2 = F1(1:Z, :);

F2 = [F2; F1(Z+1:end, :)];

[rowsF2, colsF2] = size(F2);

if rowsF2 == colsF2

warning('F2 is singular or close to singular. Consider regularization or other techniques.');

continue; % 跳过当前迭代

end

C1 = linsolve(F2, D2');

C1 = C1';

x = [];

for i = 1:( numColsD0-1)

lb = lb1(i, :);

ub = ub1(i, :);

xi = lsqlin(C1, D(:, i), [], [], Aeq, beq, lb, ub);

x = [x xi];

end

E = x';

E = abs(E(:, L));

spd = std(F(:, L));

Rmsd = sqrt(mean(((E-y).^2)));

E\_min = spd / Rmsd;

if E\_min > maxE

maxE = E\_min;

R1 = Rmsd;

Xooo = x;

Cooo = C1;

columnToDelete = Z + 1;

end

end

if columnToDelete > 0

RMSDbianhua(Z+1, 1) = R1;

R = Rmsd;

xabcd = x;

D1(:, columnToDelete) = [];

F1(columnToDelete, :) = [];

numColsD1 = size(D1, 2);

previousMaxE = maxE; %

else

break;

end

end

xn = lsqlin(Cooo, D3, [], [], Aeq, beq, lb3, ub3);

xn = xn';

lb4=lb3;

lb4(~(L == (1:length(lb4))))=0;

lb4(1,L)=xn(1,L);

ub4=ub3;

ub4(~(L == (1:length(ub4))))=1;

ub4(1,L)=xn(1,L);

xm = lsqlin(Cooo, D3, [], [], Aeq, beq, lb4, ub4);

xm = xm';

x2024(N,L) = xm(1,L);

xm\_expanded = repmat(xm, size(Cooo, 1), 1);

CCC= Cooo .\* xm\_expanded;

abc333\_accum(:, L) = sum(CCC,2);

end

fit2024 = mean(abc333\_accum,2);

|  |
| --- |
| RMSDstr = zeros(numColsD0, 1); % rmsd\_values is RMSDstr    % 遍历每一行  for i = 1: numColsD0  % 计算第i行的差异  diff = x2024(i, :)- F0(i, :);    % 计算RMSD  rmsd\_values(i) = sqrt(mean(diff.^2));  end |
| maxValue = max(D3(:, 1));  minValue = min(D3(:, 1));  disvalue= maxValue- minValue;  diff = D3-fit2024;  squared\_diff = diff .^ 2;  mean\_squared\_diff = mean(squared\_diff)/ disvalue;  NMRSD= sqrt(mean\_squared\_diff);  NMRSDfit(:, N) = NMRSD; % NMRSDfit is NMRSDfit |

end