In [ ]:
```
# using Pkg
# Pkg.add("LinearAlgebra")
# Pkg.add("GLPK")
# Pkg.add("Convex")
# Pkg.add("JuMP")
# Pkg.add("GLPKMathProgInterface")
# Pkg.add("Cbc")
# Pkg.add("Clp")
# Pkg.add("CPLEX")
# Pkg.add("Gurobi")
```

In [ ]:
```
using LinearAlgebra, GLPK, Convex
```

Given the chemical:

$$C_{11}H_{26}O_6N_2P_2S_2Cl_2$$

Accurate masses of atoms:

$C : 12.000000 \quad H : 1.0078250 \quad O : 15.994915 \quad N : 14.003074 \quad P : 30.973762 \quad S : 31.972071 \quad Cl : 34.968853$

Introduce variables:

$$x = \begin{pmatrix} x_C \\ x_H \\ x_O \\ x_N \\ x_P \\ x_S \\ x_{Cl} \end{pmatrix} \qquad A = \begin{pmatrix} 12.000000 \\ 1.0078250 \\ 15.994915 \\ 14.003074 \\ 30.973762 \\ 31.972071 \\ 34.968853 \end{pmatrix} \qquad f = \begin{pmatrix} 11 \\ 26 \\ 6 \\ 2 \\ 2 \\ 2 \\ 2 \end{pmatrix}$$

Given a fragment with mass c, we want to determin a linear combination of atoms such that m/z is most close to value c

The question is equivalent to solving optimization problem:

$$\min_{x \in \mathbb{N}^7} |a^T x - c|$$

Instead, we solve:

$$\begin{array}{ll} \min_{x \in \mathbb{R}^7} & |a^T x - c| \\ \text{s.t.} & x \geq 0 \\ & x \leq f \end{array}$$

Which is equivalent to the problem with auxillary variable:

$$\begin{array}{ll} \min_{t \in \mathbb{R}} & t \\ \text{s.t.} & t \geq ax - c \\ & t \geq -(ax - c) \\ & x \geq 0 \\ & x \leq f \end{array}$$

With one more step of transformation:

$$\begin{array}{ll} \min_{t \in \mathbb{R}} & t \\ \text{s.t.} & t - ax \geq -c \\ & t + ax \geq c \\ & x \geq 0 \\ & -x \geq -f \end{array}$$

With matix expression, we have:

$$\hat{A} = \begin{pmatrix} -A^T & 1 \\ A^T & 1 \\ I_{7\times7} & 0_{7\times1} \\ -I_{7\times7} & 0_{7\times1} \end{pmatrix} \quad \hat{x} = \begin{pmatrix} x \\ t \end{pmatrix} \quad b = \begin{pmatrix} -c \\ c \\ 0_{7\times1} \\ -f \end{pmatrix}$$

The optimization problem is will be equivalent to:

$$\begin{aligned} \min \quad & \begin{pmatrix} 0 & 1 \end{pmatrix} \hat{x} \\ \text{s.t.} \quad & \hat{A}\hat{x} - b \geq 0 \end{aligned}$$

Now generate variables A, f, test for $C_8H_{11}O_4N_2P_1S_2Cl_1$, $c = 328.9592$

```
In [ ]: A = [12 1.007825 15.994915 14.003074 30.973762 31.972071 34.968853]'
        f = [11 26 6 2 2 2 2]
        c = 328.9592;
```

```
In [ ]: x = Variable(7+1)
        Im = diagm(0=>fill(1, 7))
        A_h = [-A' 1;
                A' 1;
                Im zeros(7,1);
               -Im zeros(7,1)]
        b = [-c;
              c;
              zeros(7,1);
              -f'];
```

```
In [ ]: problem = minimize([zeros(1,7) 1]*x, [A_h * x >= b])
        solve!(problem, GLPK.Optimizer)
        x.value
```

Therefore we cannot simply try to solve the problem by soving $x \in \mathbb{R}^7$, now we need to solve the mixed integer programming problem

For the previous problem, we need to add a new constrain that $x \in \mathbb{Z}^7$

$$\begin{aligned}
\min_{t\in\mathbb{R}} \quad & t \\
\text{s.t.} \quad & t - ax \geq -c \\
& t + ax \geq c \\
& x \geq 0 \\
& -x \geq -f \\
& x \in \mathbb{Z}
\end{aligned}$$

With standarization to standard form:

$$\begin{aligned}
\min_{t\in\mathbb{R}} \quad & t \\
\text{s.t.} \quad & t - ax - s_1 = -c \\
& t + ax - s_2 = c \\
& x + s = f \\
& x, s \geq 0 \\
& s_1, s_2, t \geq 0 \\
& x, s \in \mathbb{Z} \\
& s_1, s_2, t \in \mathbb{R}
\end{aligned}$$

Introduce variables y, z, with matrix expression:

$$\hat{A} = \begin{pmatrix} -A^T & 0_{1\times 7} & -1 & 0 & 1 \\ A^T & 0_{1\times 7} & 0 & -1 & 1 \\ I_7 & I_7 & 0_{7\times 1} & 0_{7\times 1} & 0_{7\times 1} \end{pmatrix} \quad \hat{x} = \begin{pmatrix} y \\ z \end{pmatrix} = \begin{pmatrix} x \\ s \\ s_1 \\ s_2 \\ t \end{pmatrix} \quad b = \begin{pmatrix} -c \\ c \\ f \end{pmatrix}$$

Introducint two variables $d_1 = 0_{14\times 1}, d_2 = \begin{pmatrix} 0 & 0 & 1 \end{pmatrix}$, the question is equivalent to standard form:

$$\begin{aligned} \min \quad & d_1 y + d_2 z \\ \text{s.t.} \quad & \hat{A}\hat{x} = b \\ & y \geq 0, z \geq 0 \\ & y \in \mathbb{Z}, z \in \mathbb{R} \end{aligned}$$

In [ ]:
```
A_h = [-A' zeros(1,7) -1 0 1;
        A' zeros(1,7) 0 -1 0;
        Im Im zeros(7,1) zeros(7,1) zeros(7,1)]
b = [-c; c; f']
d = [zeros(1,16) 1];
```

In [ ]:
```
using JuMP
using GLPKMathProgInterface
```

In [ ]:
```
model = Model()
set_optimizer(model, GLPK.Optimizer)
y = @variable(model, [1:14], base_name="y", Int);
z = @variable(model, [1:3], base_name="z");
@objective(model, Min, z[3]);
```

In [ ]:
```
for i in 1:14
    @constraint(model, y[i]>=0)
end
for i in 1:3
    @constraint(model, z[i]>=0)
end
```

In [ ]:
```
@constraint(model, A_h[:, 1:14]*y + A_h[:, 15:17]*z.==b);
```

In [ ]:
```
model;
```

In [ ]:
```
optimize!(model);
```

In [ ]:
```
optimal_solution = value.(y)
```

As we can see, the result will be

$$C_9 H_{13} O_1 N_2 P_2 S_1 Cl_2$$

there has already matched with $m/z = 329$, and now we need to add additional constrains that restrict the ratio of each atom.

The additional mass restrictions is:

$$0.2 \leq H/C \leq 3.1$$
$$0 \leq O/C \leq 1.2$$
$$0 \leq N/C \leq 1.3$$
$$0 \leq P/C \leq 0.3$$
$$0 \leq S/C \leq 0.8$$
$$0 \leq Cl/C \leq 0.8$$
$$0 \leq hetero/C \leq 2.27$$

Now our problem is:

$$\begin{array}{ll} \min_{t\in\mathbb{R}} & t \\ \text{s.t.} & t - ax - s_1 = -c \\ & t + ax - s_2 = c \\ & x + s = f \\ & x, s \geq 0 \\ & s_1, s_2, t \geq 0 \\ & x, s \in \mathbb{Z} \\ & s_1, s_2, t \in \mathbb{R} \end{array}$$

New restrictions will be expressed as:

$$\begin{array}{lll} \frac{x_2}{x_1} \geq 0.2 & x_2 \geq 0.2x_1 & x_2 - 0.2x_1 \geq 0 \\ \frac{x_2}{x_1} \leq 3.1 & x_2 \leq 3.1x_1 & x_2 - 3.1x_1 \leq 0 \\ \frac{x_3}{x_1} \leq 1.2 & x_3 \leq 1.2x_1 & x_3 - 1.2x_1 \leq 0 \\ \frac{x_4}{x_1} \leq 1.3 & x_4 \leq 1.3x_1 & x_4 - 1.3x_1 \leq 0 \\ \frac{x_5}{x_1} \leq 0.3 & x_5 \leq 0.3x_1 & x_5 - 0.3x_1 \leq 0 \\ \frac{x_6}{x_1} \leq 0.8 & x_6 \leq 0.8x_1 & x_6 - 0.8x_1 \leq 0 \\ \frac{x_7}{x_1} \leq 0.8 & x_7 \leq 0.8x_1 & x_7 - 0.8x_1 \leq 0 \\ \frac{hetero}{x_1} \leq 2.27 & hetero \leq 2.27x_1 & hetero - 2.27x_1 \leq 0 \end{array}$$

With surplus and slack, we can transform the restirstions to:

$$
\begin{aligned}
&x_2 - 0.2x_1 - s_3 = 0\\
&x_2 - 3.1x_1 + s_4 = 0\\
&x_3 - 1.2x_1 + s_5 = 0\\
&x_4 - 1.3x_1 + s_6 = 0\\
&x_5 - 0.3x_1 + s_7 = 0\\
&x_6 - 0.8x_1 + s_8 = 0\\
&x_7 - 0.8x_1 + s_9 = 0\\
&x_3 + x_4 + x_5 + x_6 + x_7 - 2.27x_1 + s_{10} = 0\\
&s_3 \geq 0\\
&s_4 \geq 0\\
&s_5 \geq 0
\end{aligned}
$$

Now the problem is equivalent to:

$$
\begin{aligned}
\min_{t\in\mathbb{R}} \quad & t\\
\text{s.t.} \quad & t - ax - s_1 = -c\\
& t + ax - s_2 = c\\
& x + s = f\\
& x_2 - 0.2x_1 - s_3 = 0\\
& x_2 - 3.1x_1 + s_4 = 0\\
& x_3 - 1.2x_1 + s_5 = 0\\
& x_4 - 1.3x_1 + s_6 = 0\\
& x_5 - 0.3x_1 + s_7 = 0\\
& x_6 - 0.8x_1 + s_8 = 0\\
& x_7 - 0.8x_1 + s_9 = 0\\
& x_3 + x_4 + x_5 + x_6 + x_7 - 2.27x_1 + s_{10} = 0\\
& x, s \geq 0\\
& s_1, \ldots, s_{10}, t \geq 0\\
& x, s \in \mathbb{Z}\\
& s_1, \ldots, s_{10}, t \in \mathbb{R}
\end{aligned}
$$

With matrix expression:

$$\hat{A} = \begin{pmatrix} -A^T & 0_{1\times 7} & -1 & 0 & \cdots & 0 & 1 \\ A^T & 0_{1\times 7} & 0 & -1 & \cdots & 0 & 1 \\ I_7 & I_7 & 0_{7\times 1} & 0_{7\times 1} & \cdots & \cdots & 0_{7\times 1} \end{pmatrix} \quad \hat{x} = \begin{pmatrix} y \\ z \end{pmatrix} = \begin{pmatrix} x \\ s \\ s_1 \\ \vdots \\ s_{10} \\ t \end{pmatrix} \quad b = \begin{pmatrix} -c \\ c \\ f \\ 0_{8\times 1} \end{pmatrix}$$

$$\tilde{A} = \begin{pmatrix} -0.2 & 1 & 0 & 0 & 0 & 0 & 0 & 0_{1\times 7} & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -3.1 & 1 & 0 & 0 & 0 & 0 & 0 & 0_{1\times 7} & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1.2 & 0 & 1 & 0 & 0 & 0 & 0 & 0_{1\times 7} & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1.3 & 0 & 0 & 1 & 0 & 0 & 0 & 0_{1\times 7} & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ -0.3 & 0 & 0 & 0 & 1 & 0 & 0 & 0_{1\times 7} & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ -0.8 & 0 & 0 & 0 & 0 & 1 & 0 & 0_{1\times 7} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ -0.8 & 0 & 0 & 0 & 0 & 0 & 1 & 0_{1\times 7} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ -2.27 & 0 & 1 & 1 & 1 & 1 & 1 & 0_{1\times 7} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

The question is equivalent to the standard form:

$$\begin{aligned} \min \quad & d_1 y + d_2 z \\ \text{s.t.} \quad & \begin{pmatrix} \hat{A} \\ \tilde{A} \end{pmatrix} \hat{x} = b \\ & y \geq 0, z \geq 0 \\ & y \in \mathbb{Z}, z \in \mathbb{R} \end{aligned}$$

```
In [12]: using LinearAlgebra
         using JuMP
         using GLPKMathProgInterface
```

In [49]:
```
f = [11 26 6 2 2 2 2] # parameter
c = 328.95864 # parameter

a = [12 1.007825 15.994915 14.003074 30.973762 31.972071 34.968853]'
I = diagm(0=>fill(1, 7))
z17 = zeros(1,7)
z71 = zeros(7,1)
z81 = zeros(8,1)

A   = [-a'                          z17  -1   0    0    0    0    0    0    0    0    0    1;
        a'                          z17  0    -1   0    0    0    0    0    0    0    0    1;
        I                           I    z71  z71  z71  z71  z71  z71  z71  z71  z71  z71  z71;
        -0.2  1    0    0    0    0    0    z17  0    0    -1   0    0    0    0    0    0    0    0;
        -3.1  1    0    0    0    0    0    z17  0    0    0    1    0    0    0    0    0    0    0;
        -1.2  0    1    0    0    0    0    z17  0    0    0    0    1    0    0    0    0    0    0;
        -1.3  0    0    1    0    0    0    z17  0    0    0    0    0    1    0    0    0    0    0;
        -0.3  0    0    0    1    0    0    z17  0    0    0    0    0    0    1    0    0    0    0;
        -0.8  0    0    0    0    1    0    z17  0    0    0    0    0    0    0    1    0    0    0;
        -0.8  0    0    0    0    0    1    z17  0    0    0    0    0    0    0    0    1    0    0;
        -2.27 0    1    1    1    1    1    z17  0    0    0    0    0    0    0    0    0    1    0;]
b = [-c; c; f'; z81]
d = [zeros(1,24) 1];
```

In [50]:
```
model = Model()
set_optimizer(model, GLPK.Optimizer)
y = @variable(model, [1:14], base_name="y", Int);
z = @variable(model, [1:11], base_name="z");
@objective(model, Min, z[11]);
```

In [51]:
```
for i in 1:14
    @constraint(model, y[i]>=0)
end
for i in 1:11
    @constraint(model, z[i]>=0)
end
@constraint(model, A_h[:, 1:14]*y + A_h[:, 15:25]*z.==b);
```

In [52]:
```
optimize!(model)
```

```
In [53]: optimal_solution = value.(y)
```

```
Out[53]: 14-element Array{Float64,1}:
           8.0
          11.0
           4.0
           2.0
           1.0
           2.0
           1.0
           3.0
          15.0
           2.0
           0.0
           1.0
           0.0
           1.0
```

With an extremely accurate m/z value, we can converge to a correct answer.
In function, we have:

```
In [57]: using LinearAlgebra
         using JuMP
         using GLPKMathProgInterface
         function pred(c, f)
             a = [12 1.007825 15.994915 14.003074 30.973762 31.972071 34.968853]'
             I = diagm(0=>fill(1, 7))
             z17 = zeros(1,7)
             z71 = zeros(7,1)
             z81 = zeros(8,1)

             A   = [-a'                            z17  -1  0   0   0   0   0   0   0   0   0   1;
                     a'                            z17  0   -1  0   0   0   0   0   0   0   0   1;
                     I                             I    z71 z71 z71 z71 z71 z71 z71 z71 z71 z71 z71;
                     -0.2  1   0   0   0   0   0   z17  0   0   -1  0   0   0   0   0   0   0   0;
                     -3.1  1   0   0   0   0   0   z17  0   0   0   1   0   0   0   0   0   0   0;
                     -1.2  0   1   0   0   0   0   z17  0   0   0   0   1   0   0   0   0   0   0;
                     -1.3  0   0   1   0   0   0   z17  0   0   0   0   0   1   0   0   0   0   0;
                     -0.3  0   0   0   1   0   0   z17  0   0   0   0   0   0   1   0   0   0   0;
                     -0.8  0   0   0   0   1   0   z17  0   0   0   0   0   0   0   1   0   0   0;
                     -0.8  0   0   0   0   0   1   z17  0   0   0   0   0   0   0   0   1   0   0;
                     -2.27 0   1   1   1   1   1   z17  0   0   0   0   0   0   0   0   0   1   0;]
             b = [-c; c; f'; z81]
             model = Model()
             set_optimizer(model, GLPK.Optimizer)
             y = @variable(model, [1:14], base_name="y", Int);
             z = @variable(model, [1:11], base_name="z");
             @objective(model, Min, z[11]);
             for i in 1:14
                 @constraint(model, y[i]>=0)
             end
             for i in 1:11
                 @constraint(model, z[i]>=0)
             end
             @constraint(model, A_h[:, 1:14]*y + A_h[:, 15:25]*z.==b);
             optimize!(model)
             return optimal_solution = value.(y)[1:7]
         end

Out[57]: pred (generic function with 1 method)
```

```
In [81]: f = [16 22 3 1 0 0 0]
         a = [233.1047 205.0859 175.0753 149.0233 135.0440 126.1277 121.0293 84.0810]
         p = [13 15 3 1 0 0 0;
              12 13 3 0 0 0 0;
              11 11 2 0 0 0 0;
              8  5  3 0 0 0 0;
              8  7  2 0 0 0 0;
              8  16 0 1 0 0 0;
              7  5  2 0 0 0 0;
              5  10 0 1 0 0 0;]
         for i in 1:8
             c = a[i]
             t = p[i,:]
             r = pred(a[i]+0.000549, f)
             r = convert(Array{Int64,1}, r)
             if r == t
                 println("C",t[1],"H",t[2],"O",t[3],"N",t[4],"P",t[5],"S",t[6],"Cl",t[7]," is correctly predicted")
             end
             if r != t
                 println("C",t[1],"H",t[2],"O",t[3],"N",t[4],"P",t[5],"S",t[6],"Cl",t[7]," is mispreceted as",
                         "C",r[1],"H",r[2],"O",r[3],"N",r[4],"P",r[5],"S",r[6],"Cl",r[7])
             end
         end
```

```
C13H15O3N1P0S0Cl0 is correctly predicted
C12H13O3N0P0S0Cl0 is correctly predicted
C11H11O2N0P0S0Cl0 is correctly predicted
C8H5O3N0P0S0Cl0 is correctly predicted
C8H7O2N0P0S0Cl0 is correctly predicted
C8H16O0N1P0S0Cl0 is correctly predicted
C7H5O2N0P0S0Cl0 is correctly predicted
C5H10O0N1P0S0Cl0 is correctly predicted
```

```
In [87]: f = [15 23 5 6 0 1 0]
         a = [298.0942 264.0908 250.0918 163.0411 145.0302 136.0614 121.0505 119.0375]
         p = [11 16 3 5 0 1 0;
              11 14 1 5 0 1 0;
              10 12 3 5 0 0 0;
              6  11 3 0 0 1 0;
              6  9  2 0 0 1 0;
              5  6  0 5 0 0 0;
              5  5  0 4 0 0 0;
              7  5  1 1 0 0 0;]
         for i in 1:8
             c = a[i]
             t = p[i,:]
             r = pred(a[i]+0.000549, f)
             r = convert(Array{Int64,1}, r)
             if r == t
                 println("C",t[1],"H",t[2],"O",t[3],"N",t[4],"P",t[5],"S",t[6],"Cl",t[7]," is correctly predicted")
             end
             if r != t
                 println("C",t[1],"H",t[2],"O",t[3],"N",t[4],"P",t[5],"S",t[6],"Cl",t[7]," is mispreected as ",
                         "C",r[1],"H",r[2],"O",r[3],"N",r[4],"P",r[5],"S",r[6],"Cl",r[7])
             end
         end
```

```
C11H16O3N5P0S1Cl0 is mispreected as C14H12O3N5P0S0Cl0
C11H14O1N5P0S1Cl0 is correctly predicted
C10H12O3N5P0S0Cl0 is correctly predicted
C6H11O3N0P0S1Cl0 is mispreected as C4H9O2N3P0S1Cl0
C6H9O2N0P0S1Cl0 is mispreected as C4H7O1N3P0S1Cl0
C5H6O0N5P0S0Cl0 is correctly predicted
C5H5O0N4P0S0Cl0 is correctly predicted
C7H5O1N1P0S0Cl0 is correctly predicted
```

```
In [89]: f = [16 24 5 1 0 0 0]
         a = [276.1600 251.1154 248.1650 207.1023 175.0759 151.0395 142.1229 137.0602 100.0759 88.0761 70.0654]
         p = [16 22 3 1 0 0 0;
              13 17 4 1 0 0 0;
              15 22 2 1 0 0 0;
              12 15 3 0 0 0 0;
              11 11 2 0 0 0 0;
              8  7  3 0 0 0 0;
              8  16 1 1 0 0 0;
              8  9  2 0 0 0 0;
              5  10 1 1 0 0 0;
              4  10 1 1 0 0 0;
              4  8  0 1 0 0 0;]
         for i in 1:11
             c = a[i]
             t = p[i,:]
             r = pred(a[i]+0.000549, f)
             r = convert(Array{Int64,1}, r)
             if r == t
                 println("C",t[1],"H",t[2],"O",t[3],"N",t[4],"P",t[5],"S",t[6],"Cl",t[7]," is correctly predicted")
             end
             if r != t
                 println("C",t[1],"H",t[2],"O",t[3],"N",t[4],"P",t[5],"S",t[6],"Cl",t[7]," is mispredcted as ",
                         "C",r[1],"H",r[2],"O",r[3],"N",r[4],"P",r[5],"S",r[6],"Cl",r[7])
             end
         end
```

```
C16H22O3N1P0S0Cl0 is correctly predicted
C13H17O4N1P0S0Cl0 is correctly predicted
C15H22O2N1P0S0Cl0 is correctly predicted
C12H15O3N0P0S0Cl0 is correctly predicted
C11H11O2N0P0S0Cl0 is correctly predicted
C8H7O3N0P0S0Cl0 is correctly predicted
C8H16O1N1P0S0Cl0 is correctly predicted
C8H9O2N0P0S0Cl0 is correctly predicted
C5H10O1N1P0S0Cl0 is correctly predicted
C4H10O1N1P0S0Cl0 is correctly predicted
C4H8O0N1P0S0Cl0 is correctly predicted
```

In [93]:

```
f = [10 15 13 5 3 0 0]
a = [426.0217 408.0105 272.9530 238.8950 176.9390 158.9246 134.0464 96.9683 78.9587]
p = [10 14 10 5 2 0 0;
     10 12 9  5 2 0 0;
     5  7  9  0 2 0 0;
     0  2  9  0 3 0 0;
     0  3  7  0 2 0 0;
     0  1  6  0 2 0 0;
     5  4  0  5 0 0 0;
     0  2  4  0 1 0 0;
     0  0  3  0 1 0 0;]
for i in 1:9
    c = a[i]
    t = p[i,:]
    r = pred(a[i]-0.000549, f)
    r = convert(Array{Int64,1}, r)
    if r == t
        println("C",t[1],"H",t[2],"O",t[3],"N",t[4],"P",t[5],"S",t[6],"Cl",t[7]," is correctly predicted")
    end
    if r != t
        println("C",t[1],"H",t[2],"O",t[3],"N",t[4],"P",t[5],"S",t[6],"Cl",t[7]," is mispredected as ",
                "C",r[1],"H",r[2],"O",r[3],"N",r[4],"P",r[5],"S",r[6],"Cl",r[7])
    end
end
```

```
C10H14O10N5P2S0Cl0 is correctly predicted
C10H12O9N5P2S0Cl0 is correctly predicted
C5H7O9N0P2S0Cl0 is mispredected as C7H2O8N2P1S0Cl0
C0H2O9N0P3S0Cl0 is mispredected as C8H3O4N1P2S0Cl0
C0H3O7N0P2S0Cl0 is mispredected as C8H3O1N0P2S0Cl0
C0H1O6N0P2S0Cl0 is mispredected as C4H2O4N1P1S0Cl0
C5H4O0N5P0S0Cl0 is mispredected as C4H8O4N1P0S0Cl0
C0H2O4N0P1S0Cl0 is mispredected as C4H2O1N0P1S0Cl0
C0H0O3N0P1S0Cl0 is mispredected as C4H1O1N1P0S0Cl0
```

In [ ]: